Long-term dense motion estimation and view synthesis quality assessment with application to joint stereo and motion processing

Pierre-Henri Conze

Thèse

THÈSE INSA Rennes
sous le sceau de l'Université Européenne de Bretagne
pour obtenir le grade de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Traitement du signal et des images

présentée par
Pierre-Henri Conze
ÉCOLE DOCTORALE : MATISSE
LABORATOIRE : IETR

Long-term dense motion estimation and view synthesis quality assessment with application to joint stereo and motion processing

Jury de thèse :

Fabrice Heitz
Professeur à Télécom Physique Strasbourg, ICube / *Examinateur*

Soutenance le 16-04-2014 en présence de :

Charles Kervrann
Directeur de recherche à l'INRIA de Rennes / *Président*

Marco Cagnazzo
Maître de conférence HDR à Télécom ParisTech, LTCI / *Rapporteur*

Vincent Charvillat
Professeur à l'ENSEEIHT, IRIT / *Rapporteur*

Jean-Marc Odobez
Maître d'enseignement et de recherche EPFL, Idiap / *Examinateur*

Patrick Bouthemy
Directeur de recherche à l'INRIA de Rennes / *Examinateur*

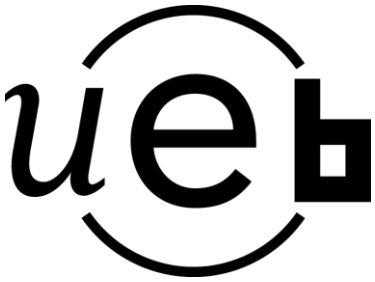
Rémi Mégret
Maître de conférence à l'ENSEIRB-MATMECA, IMS / *Examinateur*

Luce Morin
Professeur à l'INSA de Rennes, IETR / *Directrice de thèse*

Philippe Robert
Ingénieur de recherche à Technicolor / *Co-encadrant de thèse*

Long-term dense motion estimation and view synthesis quality assessment with application to joint stereo and motion processing

Pierre-Henri Conze



En partenariat avec

Contents

Remerciements

J'ai eu la chance d'être entouré et soutenu durant ces trois années de thèse par de nombreuses personnes à qui je tiens à exprimer ma sincère reconnaissance.

Je souhaite remercier en premier lieu les membres du jury pour avoir accepté de prendre part à l'évaluation de mes travaux : M. Cagnazzo et V. Charvillat, mes rapporteurs, ainsi que C. Kervrann, J.-M. Odobez, P. Bouthemy, R. Mégret et F. Heitz.

Je tiens ensuite à remercier Philippe Robert, mon directeur de thèse à Technicolor, pour son encadrement remarquable, sa très grande disponibilité, nos nombreuses discussions techniques enrichissantes ainsi que pour l'ensemble des corrections du manuscrit. Je remercie également ma directrice de thèse à l'INSA, Luce Morin, pour sa présence constante durant ces trois ans, son écoute, ses précieux conseils lors de la rédaction du manuscrit et ses relectures pertinentes. La qualité de leur encadrement m'a permis de travailler dans d'excellentes conditions.

J'adresse bien sur à Tomás Crivelli des remerciements tout particuliers pour tout le travail conjointement mené en estimation de mouvement. Ses conseils, sa disponibilité et ses relectures m'ont été d'une très grande aide. Je n'oublie pas non plus tous les collègues avec qui j'ai eu la chance de travailler. Je pense tout d'abord à Matthieu Fradet avec qui ce fut un plaisir de travailler au quotidien mais également à Cédric Thébault pour les discussions autour de la synthèse de vues ainsi qu'à Thierry Viellard pour ses réponses à mes questions d'implémentation. Je remercie également Patrick Pérez et Lionel Oisel pour les discussions techniques et pour avoir répondu à mes sollicitations lors de mes choix d'orientation. Merci également à Emilie Bosc pour son aide lors de mes travaux sur l'estimation de qualité des vues synthétisées. Enfin, au sein de Technicolor, j'ai eu la chance de côtoyer un grand nombre de collègues. Je pense en particulier à Hasan Sheikh Faridul, Alasdair Newson, Jurgen Stauder, Corine Porée, Patrick Morvan, Emmanuel Jolly, Catherine Serre, Arno Schubert, Cristina Bordei, Claire-Hélène Demarthy, Neus Sabater, Pierre Hellier, Hasan Guermoud, Jonathan Kervec, Harouna Kabré et tant d'autres.

Je souhaite également remercier tous mes amis qui ont été présents tout au long de ces trois années : Antoine et Julie pour tous les moments géniaux partagés aux quatre coins de la France, Florian pour ses encouragements et son aide en LaTeX, Alex et Marie pour leur présence si indispensable à Strasbourg sans oublier Alexandra, Emeline, Timothée, Samy, Damien, Anaïs, Yannick ainsi que les amis de l'ENSEIRB. Merci également aux Fuzzy Garden et notamment à Laurent, FX et Dom pour tous ces moments de décompression partagés en repets, en studio ou au Ty Anna.

Que n'aurait pas été cette thèse sans tous les compositeurs et interprètes géniaux qui ont rythmé mes longues heures de rédaction ? Merci de vous plaindre à Schoenberg pour les fautes de frappes, à Reich pour les répétitions, à Chopin et Rachmaninov pour les trop longues phrases. Reste, je l'espère, la rigueur mathématique des compositeurs baroques à commencer par Bach ou Buxtehude !

Tout ce travail n'aurait pas été possible sans le soutien indéfectible et constant de mes parents. Je leur adresse d'énormes remerciements pour m'avoir toujours encouragé, de la première parution du Sait Tout à la soutenance de thèse. Un grand merci également à ma famille et belle famille et tout particulièrement à mes frères ainsi qu'à Brigitte, Jean-Yves et Camille.

Enfin, je n'ai pas de mots assez forts pour exprimer ma reconnaissance envers Claire, dont la présence et les encouragements ont été indispensables pour mener à bien ce projet comme tant d'autres. Que notre arrivée à Strasbourg soit l'occasion de vivre ensemble de nouvelles belles expériences !

Chapter 1

Résumé en français

1.1 Introduction générale

Les nouvelles technologies de la vidéo numérique tendent vers la production, la transmission et la diffusion de contenus de très haute qualité, qu'ils soient monoscopiques ou stéréoscopiques. Ces technologies ont énormément évolué au cours de ces dernières années afin de faire vivre à l'observateur l'expérience la plus réaliste possible. Pour diverses raisons artistiques ou techniques liées à l'acquisition ou à la transmission de contenu, il est parfois nécessaire de combiner la vidéo acquise à des informations de synthèse tout en veillant à maintenir un rendu photo-réaliste accru.

Pour faciliter la tâche des opérateurs de production et post-production de contenu, le traitement combiné de contenu capturé et de synthèse exige de disposer de fonctionnalités automatiques sophistiquées telles que des algorithmes de suivi long-terme de déformations d'objets, de mise en correspondance d'images capturées à partir de différents points de vues ou de génération de vues synthétisées. Avec l'idée sous-jacente selon laquelle de telles fonctionnalités automatiques doivent répondre à des exigences fortes en terme de qualité, les travaux de recherche présentés ici ont plus précisément porté sur l'évaluation de la qualité de la synthèse de vues et sur l'élaboration de stratégies d'estimation de mouvement dense et long-terme.

La création d'images synthétisées, via des algorithmes de rendu *DIBR* (*Depth-Image-Based Rendering*) combinant à la fois estimation de disparité et synthèse de vues [1][1], est nécessaire dans le cadre de la diffusion de contenus stéréoscopiques par des téléviseurs 3D sans lunettes, dits auto-stéréoscopiques. En raison de difficultés à capturer et à transmettre un nombre important de vues issues de points de vue différents, les téléviseurs 3D auto-stéréoscopiques requièrent la création de vues synthétisées *a-posteriori* à partir d'un nombre réduit de vues réellement capturées (on parle alors de conversion stéréo/multi-vues). Cependant, les algorithmes *DIBR* peuvent parfois être à l'origine d'artéfacts au sein des vues synthétisées, ce qui peut endommager significativement la perception 3D et le confort de l'observateur. C'est dans ce contexte que nous avons étudié quelles sont les différentes sources possibles de distortions et comment évaluer de manière automatique la qualité de la synthèse de vue.

Dans le contexte de l'estimation de mouvement, nos contributions portent sur l'élaboration de stratégies denses et long-termes permettant par exemple de propager de manière automatique divers types d'information tels que des textures ou logo synthétiques insérés manuellement ou des labels de segmentation. Contrairement aux estimateurs de mouvement denses classiques

[1] les références numériques du Chapitre 1 sont détaillées dans la liste des publications, page 303

qui se limitent généralement à un matching d'images consécutives, les stratégies proposées ont pour but de produire une description dense du mouvement au sein de longues séquences vidéos soumises à des situations complexes (large mouvement, occultations temporaires, déformations non-rigides, textures périodiques...).

Les problématiques d'estimation de disparité et de synthèse de vues (Section 1.2) ainsi que d'estimation de mouvement dense long-terme (Section 1.3) ont d'abord été étudiées séparément. Les contributions respectives ont ensuite été considérées conjointement dans le contexte de la coopération stéréo-mouvement. Dans ce cadre, nous avons notamment proposé une méthode de correction de disparité et des artéfacts de synthèse de vues correspondants (Section 1.4).

1.2 Estimation de qualité de la synthèse de vues

1.2.1 Introduction

L'obtention d'images synthétisées de bonne qualité est essentielle pour les écrans 3D auto-stéréoscopiques. Les vues synthétisées générées par l'intermédiaire d'algorithmes de rendu de type *DIBR* font cependant parfois l'objet de défauts dus à une mauvaise estimation de disparité et/ou interpolation. Après étude approfondie des différentes sources de distorsions possibles, nous avons identifié diverses situations à risques : zones sans textures, transparence, objets fins, bords d'objets, variations de luminosité ou différences de couleur entre vues gauche et droite, objets périodiques...

L'état de l'art réalisé en évaluation objective de la qualité des images dans les cas monoscopique et stéréoscopique a révélé que les métriques de qualité traditionnelles ne sont pas adaptées à la détection des artéfacts présents dans les vues synthétisées [BKP+11][2]. De plus, la problématique de l'évaluation explicite de la qualité de la synthèse de vue n'a pas été étudiée en profondeur. Bien que les métriques les plus récentes intègrent des caractéristiques connues du système visuel humain, comme c'est le cas pour la métrique *SSIM* (*Structural SIMilarity*) [WBSS04] ou les extensions dont elle a fait l'objet, elles ont pour la plupart été conçues pour mesurer la qualité visuelle d'une image compressée par rapport à une image originale. De nouvelles méthodes dédiées à l'évaluation de la qualité de la synthèse de vue sont donc nécessaires.

1.2.2 Contribution à l'évaluation de qualité de la synthèse de vue

C'est dans ce contexte que nous avons développé une méthode d'évaluation objective *full-reference* de la qualité des images, dédiée à la synthèse de vue : la métrique *VSQA* [2] (*View Synthesis Quality Assessment*). Notre approche fait l'hypothèse que tous les pixels n'ont pas la même importance en terme de perception visuelle. C'est pourquoi *VSQA* inclut une pondération des valeurs obtenues avec une métrique existante (*SSIM* par exemple) par trois cartes de pondération calculées via trois cartes de visibilité (Fig. 1.1). Ces cartes de visibilité sont créées via trois approches semi-globales dont le but est de quantifier respectivement la présence de textures, la diversité en termes d'orientations de gradient (la vision humaine étant sensible aux caractéristiques d'orientations locales [WJMG10]) et l'existence de différences de luminance.

La carte de pondération basée texture permet de mettre en avant les artéfacts situés dans des zones peu texturées et donc davantage visibles que ceux localisés dans des zones texturées

[2] les références alphabétiques de la Section 1.2 sont détaillées dans la bibliographie de la Partie I, page 95

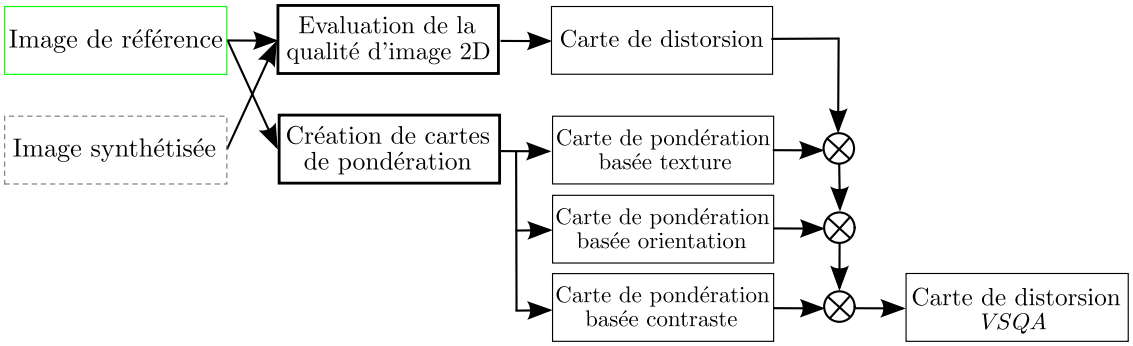


Figure 1.1: Notre système d'évaluation de qualité de la synthèse de vues [2].

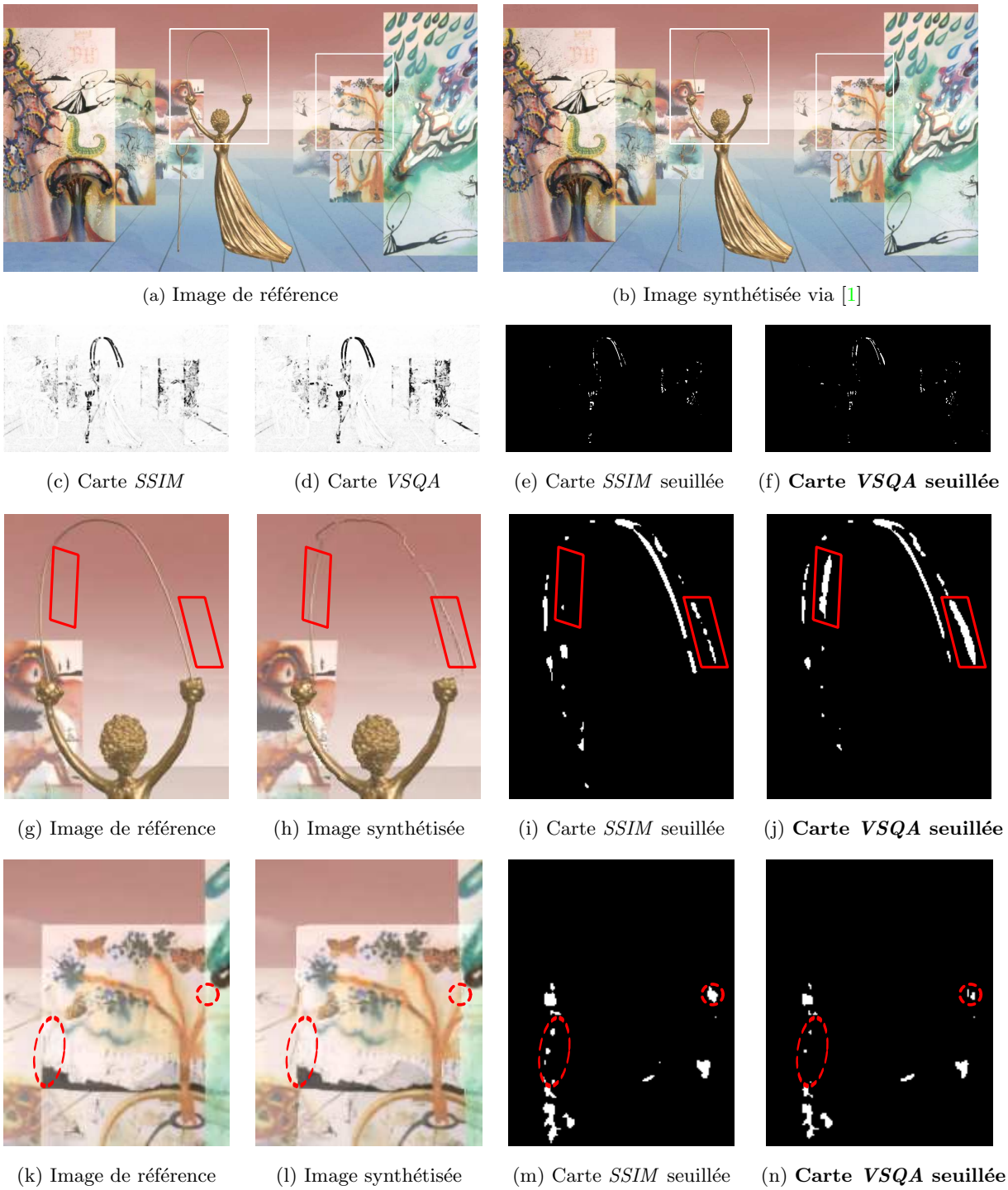
où s'opère un phénomène de masquage. Dans le même esprit, les cartes de pondération basées orientations de gradient et contraste donnent plus de poids aux artéfacts situés respectivement dans des zones de faible diversité en termes d'orientations de gradient et de fort contraste et atténuent les artéfacts se trouvant respectivement dans des zones de forte diversité d'orientations de gradient et de faible contraste.

1.2.3 Résultats obtenus avec *VSQA*

La procédure *VSQA* appliquée aux cartes de distorsions *SSIM* (*VSQA* basée *SSIM*) a été évaluée expérimentalement et comparée aux méthodes de l'état de l'art. Par le biais d'une évaluation qualitative, nous avons prouvé que *VSQA* basée *SSIM* permet d'établir une meilleure hiérarchie des artéfacts en terme d'estimation de visibilité perçue par rapport à *SSIM*. Ainsi, Fig. 1.2 montre les cartes de distorsions obtenues par *SSIM* (c) et *VSQA* basée *SSIM* (d) (artéfacts indiqués en couleurs sombres) pour la paire image de référence (a) / image synthétisée (b). Les deux cartes de distorsions ont ensuite été seuillées en prenant en compte les 2300 pixels les plus erronés selon chaque métrique (pixels erronés indiqués en blanc). En se focalisant respectivement sur les artéfacts présents au niveau de l'arc en or (g,h) et des panneaux semi-transparents (k,l), on remarque que *VSQA* basée *SSIM* :

- met en avant les artéfacts dont la visibilité est accrue par la présence d'un fort contraste et d'une faible diversité d'orientations de gradient au sein de zones peu texturées (arc en or, Fig. 1.2 (g-j)),
- atténue les artéfacts dont la visibilité est diminuée par une forte diversité d'orientations de gradient au sein de zones texturées (panneaux semi-transparent, Fig. 1.2 (k-n)).

D'un point de vue quantitatif, les mesures subjectives fournies par la base de donnée *IRCCyN/IVC DIBR* [BPLC+11b, BPLC+11a] et obtenues par 43 observateurs sur 84 séquences synthétisées ont permis de comparer *VSQA* basée *SSIM* avec de nombreuses métriques existantes. Les comparaisons ont été effectués par le biais de calculs de corrélation avec les scores objectifs de chaque métrique et les mesures subjectives. Les résultats explicités Tab. 1.1 montrent que la métrique proposée améliore de manière significative les résultats obtenus avec les métriques de l'état de l'art [2]. *VSQA* basée *SSIM* obtient un coefficient de corrélation avec les mesures subjectives de $61,42\%$ ce qui correspond à un gain de $17,8\%$ par rapport à *SSIM* ($43,63\%$). De plus, *VSQA* basée *SSIM* est plus performante que la meilleure des métriques de l'état de l'art, à savoir *MS-SSIM* ($55,99\%$), avec un gain de plus de 5%.



(a) Image de référence (b) Image synthétisée via [1]

(c) Carte *SSIM* (d) Carte *VSQA* (e) Carte *SSIM* seuillée (f) **Carte *VSQA* seuillée**

(g) Image de référence (h) Image synthétisée (i) Carte *SSIM* seuillée (j) **Carte *VSQA* seuillée**

(k) Image de référence (l) Image synthétisée (m) Carte *SSIM* seuillée (n) **Carte *VSQA* seuillée**

Figure 1.2: Comparaison entre *SSIM* et *VSQA* basée *SSIM* (séquence *Dali-A*) appliquées aux images de référence (*a*,*g*,*k*) et synthétisée (*b*,*h*,*l*). Les cartes de distorsion *SSIM* et *VSQA* basée *SSIM* (resp. (*c*) et (*d*)) sont seuillées (resp. (*e*,*i*,*m*) et (*f*,*j*,*n*) en prenant en compte les 2300 pixels les plus erronés selon chaque métrique (pixels erronés indiqués en blanc). Les formes rouge en trait plein indiquent les artéfacts mis en avant par *VSQA* tandis que les formes rouges en pointillé correspondent aux artéfacts atténués par *VSQA*.

Métrique	*PLCC DMOS*
PSNR	45, 65
SSIM	43, 63
SSIM* basée *VSQA	**61,42**
MS-SSIM	55, 99
VSNR	35, 89
VIF	32, 03
VIFP	25, 55
UQI	39, 27
IFC	27, 90
NQM	53, 34
WSNR	44, 12
PSNR HVSM	40, 57
PSNR HVS	39, 25

Table 1.1: Coefficients de corrélation linéaire de *Person* (*PLCC*) entre mesures subjectives et scores de qualité objectifs en pourcentage (base de donnée *IRCCyN/IVC DIBR* [BPLC+11a]).

1.2.4 Conclusion

Pour conclure, *VSQA* permet une détection pertinente des artéfacts dus à la synthèse de vue mais pourrait néanmoins être améliorée en appliquant la procédure proposée à d'autres métriques existantes telles que *MS-SSIM*. De plus, une perspective possible consiste à étendre *VSQA* à l'évaluation de la qualité de vidéos synthétisées dans la mesure où les variations temporelles d'artéfacts spatiaux peuvent avoir un impact important sur leur perception.

1.3 Estimation de mouvement dense et long-terme

1.3.1 Introduction

L'estimation de mouvement est l'une des problématiques prédominantes en vision par ordinateur. Basés sur la contrainte de conservation de l'intensité lumineuse, les estimateurs de mouvement dense de l'état de l'art [ZPB07, SRB10, SBK10, BM11][3] se limitent quasi-exclusivement à des paires d'images consécutives. Cependant, certaines applications telles que la segmentation vidéo, des techniques d'analyse ou d'édition vidéo requièrent une estimation dense et long-terme du mouvement le long des séquences d'images. Cela demande à ce que des méthodes puissent être capables d'établir des correspondances dense entre images distantes.

Bien que robustes entre images consécutives, les méthodes d'estimation de mouvement de l'état de l'art fonctionnent généralement moins bien entre images non-consécutives. Une mise en correspondance directe entre images non-consécutives peut s'avérer incorrecte, notamment lorsque le contenu et les déplacements sont complexes.

Une alternative consiste à accumuler des vecteurs de flot optique estimés entre images consécutives. Cependant, les trajectoires denses et long-termes résultantes divergent rapidement du fait de l'accumulation des erreurs et leur fiabilité dépasse rarement les 30 images. De

[3] les références alphabétiques de la Section 1.3 sont détaillées dans la bibliographie de la Partie II, page 276

plus, la concaténation de vecteurs de flot optique entre images consécutives ne permet pas de traiter le cas spécifique des occultations temporaires.

C'est dans ce contexte que nous présentons différentes contributions à l'estimation de mouvement dense et long-terme. Les stratégies proposées manipulent en entrée des vecteurs de flot optique estimés avec des pas variables (*multi-steps*), c'est à dire calculés entre images consécutives ou davantage éloignées. Ces vecteurs de flot optique *multi-steps* sont pré-calculés par un estimateur de flot optique issu de l'état de l'art.

1.3.2 Approche séquentielle *multi-steps*

Nous proposons un estimateur de mouvement dense long-terme, nommé *multi-step* flow fusion (*MSF*) [4], dont le but est de séquentiellement :

1. construire un ensemble de vecteurs de mouvement long-termes candidats en combinant vecteurs de flot optique *multi-steps* et vecteurs de mouvement long-termes optimaux estimés pour les images précédentes (Fig. 1.4),

2. sélectionner parmi les vecteurs de mouvement long-termes candidats générés le vecteur optimal par le biais d'une méthode d'optimisation globale.

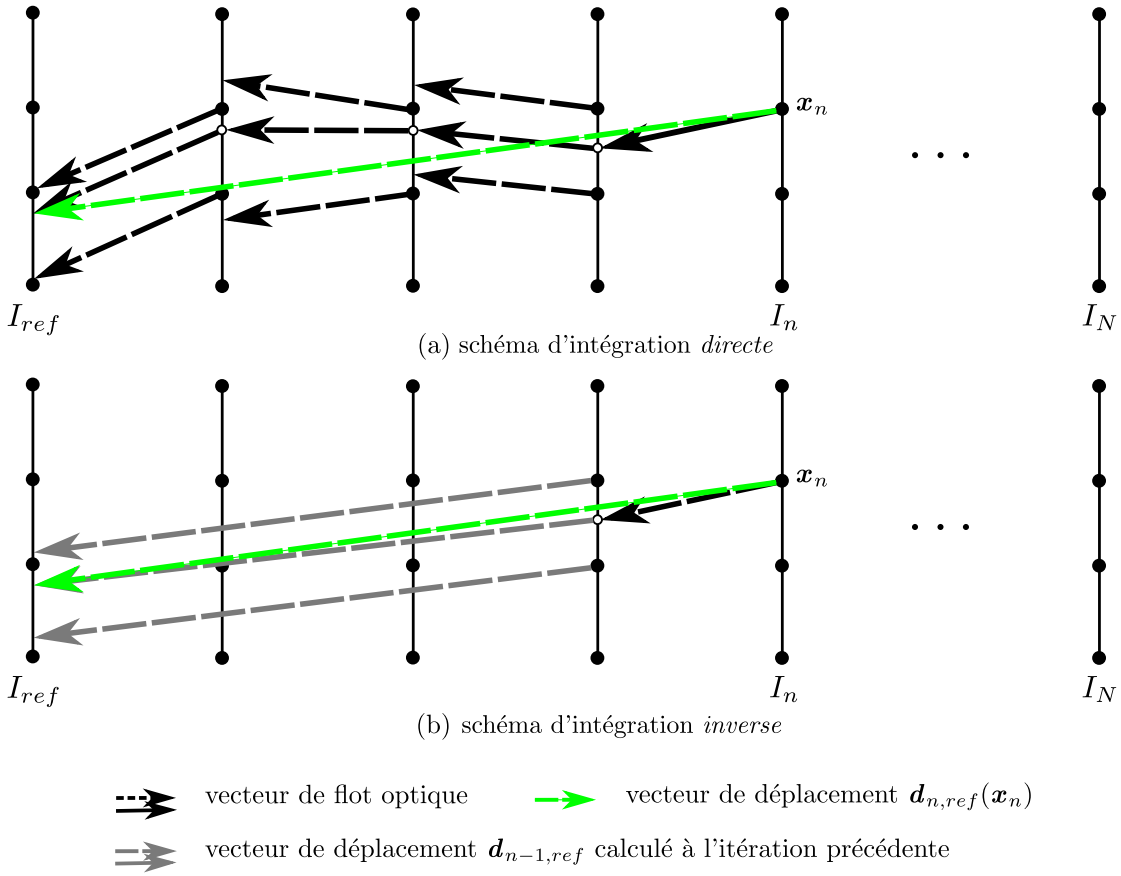
Considérons une séquence de $N+1$ images RGB $\{I_n\}$ avec $n \in [\![0, \dots, N]\!]$ où I_0 est une image de référence. Basons nos explications sur le calcul du vecteur de déplacement $\boldsymbol{d}_{n,0}(\boldsymbol{x}_n)$ qui relie le pixel $\boldsymbol{x}_n$ de I_n à une position sub-pixelique dans l'image de référence I_0. Pour un ensemble donné de Q_n *steps* à l'instant n, $S_n = \{s_1, s_2, ..., s_{Q_n}\}$, nous supposons avoir à disposition un ensemble correspondant de champs de flot optique *forward* ($s_k > 0$) ou *backward* ($s_k < 0$) pré-calculés $\{\boldsymbol{u}_{n,n+s_1}, \boldsymbol{u}_{n,n+s_2}, \dots, \boldsymbol{u}_{n,n+s_{Q_n}}\}$. Pour chaque *step* $s_k \in S_n$, nous pouvons calculer les vecteurs long-termes candidats $\boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n)$ comme suit :

$$\boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n) = \boldsymbol{u}_{n,n+s_k}(\boldsymbol{x}_n) + \tilde{\boldsymbol{d}}_{n+s_k,0}(\boldsymbol{x}_n + \boldsymbol{u}_{n,n+s_k}(\boldsymbol{x}_n)) \tag{1.1}$$

Le processus séquentiel qui mène au calcul des $\boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n)$ s'appuie sur une intégration *inverse* et non *directe* [3], comme l'illustre Fig. 1.3 dans le cas où nous disposons uniquement de *steps* de 1. Au lieu d'accumuler séquentiellement les vecteurs de flot optique en partant de I_n en direction de I_0 (intégration *directe*, Fig. 1.3 (a)), nous proposons un processus itératif consistant à concaténer à chaque itération un vecteur de mouvement long-terme précédemment estimé à un vecteur de flot optique (intégration *inverse*, Fig. 1.3 (b)). L'intégration *inverse* parcourt la séquence dans la direction inverse à cette de l'intégration *directe* dans la mesure où l'on traite d'abord la paire $\{I_1, I_0\}$ puis $\{I_2, I_0\}$ et ainsi de suite jusqu'à $\{I_n, I_0\}$.

Lorsque pour chaque pixel $\boldsymbol{x}_n$, l'ensemble des candidats $\boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n)$ a été estimé en considérant l'ensemble des *steps* $s_k \in S_n$ (Eq. 1.1), nous utilisons une méthode d'optimisation globale pour constituer le champ de mouvement final. L'énergie proposée met en jeu un coût de matching associé à chacun des vecteurs candidats ainsi qu'une régularisation spatiale pondérée par la similarité de couleur entre pixels voisins. La minimisation de cette énergie est réalisée via l'algorithme *fusion moves* [LRR08, LRRB10].

Une fois la séquence traitée dans son ensemble, nous proposons de filtrer itérativement les champs de mouvement long-termes obtenus grâce à un nouveau filtrage multilatéral spatio-temporel [4]. Celui-ci permet tout d'abord de filtrer le long des trajectoires les vecteurs de



(a) schéma d'intégration *directe*

(b) schéma d'intégration *inverse*

Figure 1.3: Comparaison entre deux stratégies différentes d'intégration pour l'estimation du vecteur de déplacement long-terme $\boldsymbol{d}_{n,ref}(\boldsymbol{x}_n)$: (a) l'intégration *directe* et (b) l'intégration proposée, dite *inverse* (technique utilisée au sein de notre estimateur *MSF*).

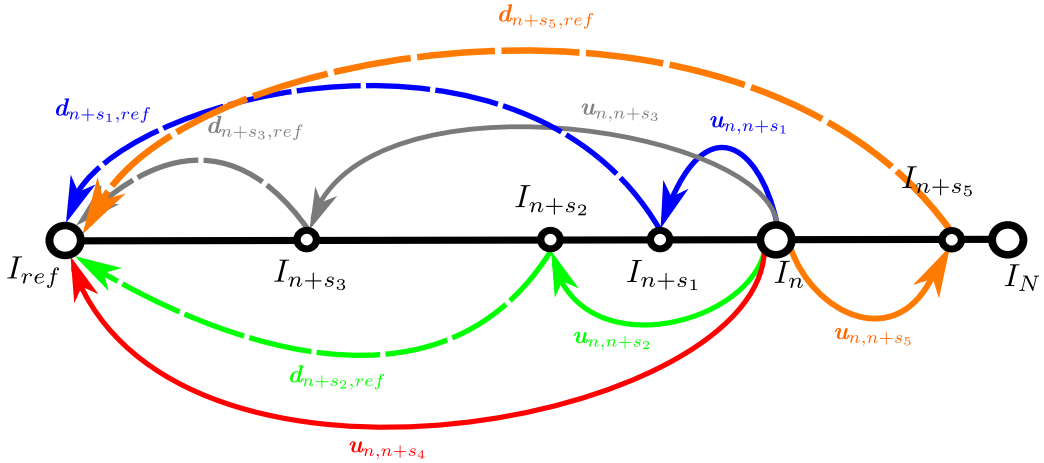


Figure 1.4: Génération de vecteurs de déplacement long-termes candidats via différents chemins combinant vecteurs de flot optique *multi-steps forward* ou *backward* et vecteurs de déplacement long-termes précédemment estimés.

(a) I_0 (b) I_{20} (c) I_{40}

(d) I_{60} (e) I_{80} (f) I_{100}

Figure 1.5: Images sources de la séquence *AmeliaRetro* (propriété de *Dolby*).

mouvement partant des pixels de l'image de référence (*from-the-ref*) avec les vecteurs du voisinage spatio-temporel. La pondération des vecteurs voisins met en jeu la similarité couleur, la distance, le coût de matching ainsi que la similarité entre trajectoires. De plus, nous filtrons conjointement les vecteurs de mouvement long-termes issus de (*from-the-ref*) et allant vers (*to-the-reference*) l'image de référence par filtrage multilatéral. La pondération des vecteurs impliqués est identique à celle présentée précédemment excepté pour la similarité entre trajectoires qui est remplacée par une simple similarité de vecteurs.

Résultats obtenus avec *MSF*

Les champs de déplacement dense long-termes calculés via *MSF*(2*D-DE*) (c'est à dire *MSF* utilisant des champs de flot optique estimés par une version 2D de l'estimateur de disparité décrit dans [1] : 2*D-DE*) ont fait l'objet de nombreuses comparaisons aux méthodes de l'état l'art dont :

- *LDOF acc*, 2*D-DE acc* ou encore *TV-L*1 *acc* : intégration *directe* de vecteurs de flot optique de *step* 1 issus resp. des estimateurs *LDOF* [BM11], 2*D-DE* [1] et *TV-L*1 [ZPB07],
- 2*D-DE inverse* : intégration *inverse* de vecteurs de flot optique 2*D-DE* de *step* 1.

Pour comparer notre méthode à celles de l'état de l'art, la qualité des champs de déplacement peut être évaluée via recalage d'images (reconstruction de I_{ref} à partir de I_n ou inversement) puis calcul de *PSNR* entre images originales et recalées. Des exemples de recalage sont présentés Fig. 1.6 pour *LDOF acc*, 2*D-DE acc* et *MSF*(2*D-DE*) tandis que des scores *PSNR* obtenus par *LDOF acc*, *TV-L*1 *acc*, 2*D-DE inverse* et *MSF*(2*D-DE*) sont explicités Fig. 1.7. Ces expériences révèlent une bien meilleure reconstruction avec *MSF*(2*D-DE*) comparés aux méthodes standards d'intégration de flot optique.

Les exemples de propagation de texture, logo ou labels de segmentation présentés Fig. 1.8 (séq. *AmeliaRetro*), Fig. 1.9 (séq. *Newspaper*) et Fig. 1.10 (séq. *AmeliaRetro*) prouvent que des correspondances denses long-termes peuvent être utiles dans le contexte de l'édition vidéo pour propager de nombreux types différents d'information. Les résultats de propagation de modifications couleur Fig. 1.8 montrent que *MSF*(2*D-DE*) permet d'obtenir des résultats

(a) (b) (c) (d)

Figure 1.6: Reconstruction de la robe de I_0 (a) à partir de I_{100} par recalage via les champs de déplacement long-termes obtenus avec : (b) *LDOF acc*, (c) *2D-DE acc* et (d) *MSF*(*2D-DE*). *Steps* utilisés pour *MSF* : 1, 2, 5, 10, 20, 30, 40, 50 et 100.

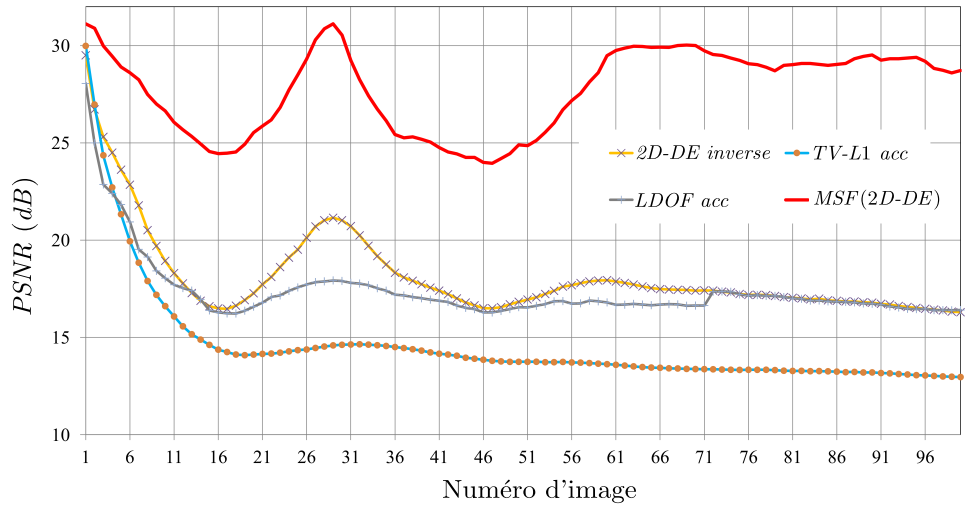


Figure 1.7: Recalage et évaluation *PSNR* d'un bloc de la robe (séquence *AmeliaRetro*) en utilisant les champs de déplacement long-termes obtenus via : *TV-L1 acc*, *LDOF acc*, *2D-DE inverse* et *MSF*(*2D-DE*). *Steps* utilisés pour *MSF* : 1, 2, 5, 10, 20, 30, 40, 50 et 100.

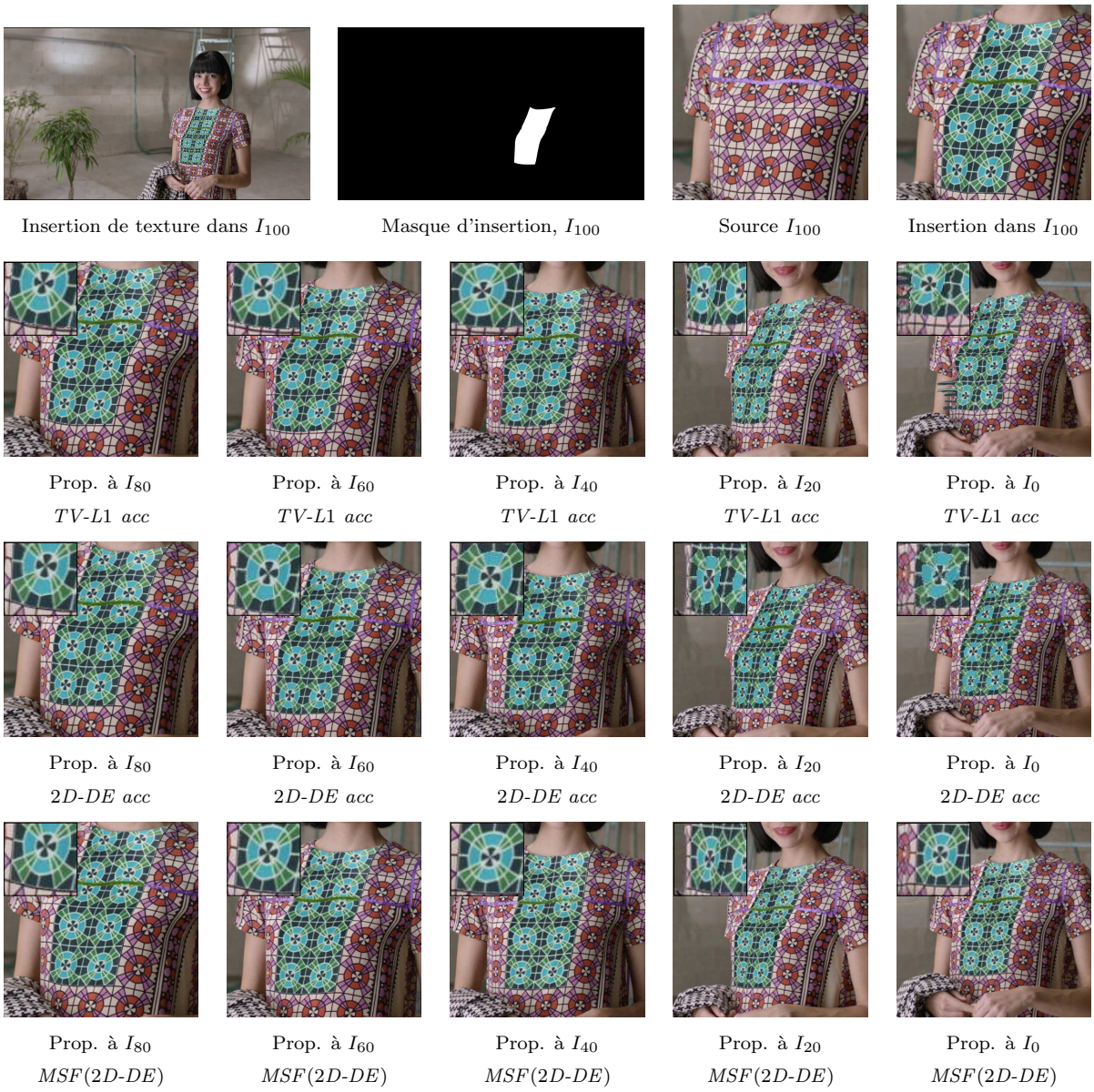


Figure 1.8: Modifications de couleur dans I_{100} et propagation jusqu'à I_0 (séquence *AmeliaRetro*). Nous comparons : *TV-L1 acc*, *2D-DE acc* et *MSF*(2D-DE). *Steps* utilisés pour *MSF* : 1, 2, 5, 10, 20, 30, 40, 50 et 100.

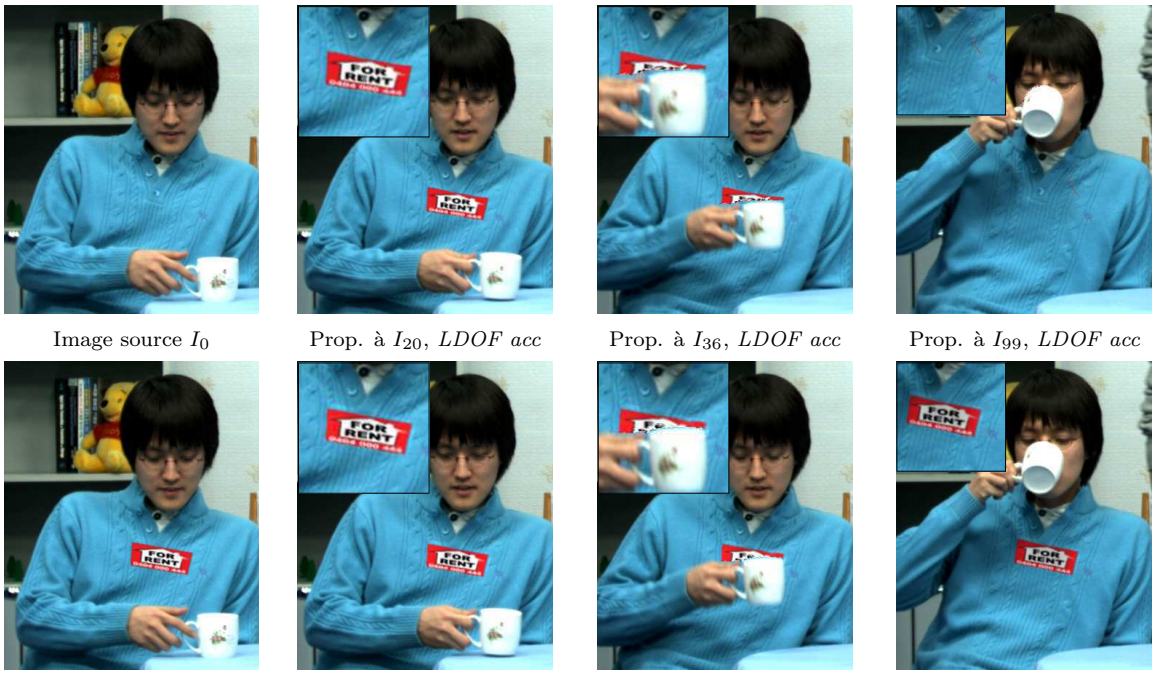
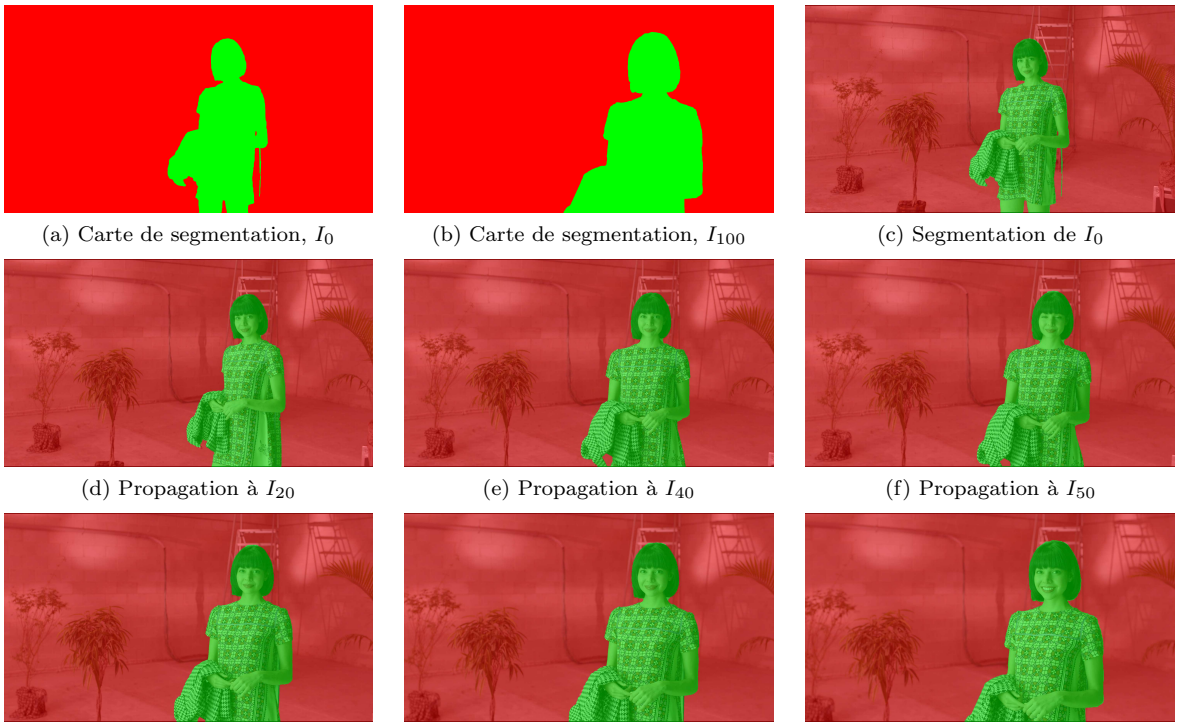


Image source I_0 | Prop. à I_{20}, *LDOF acc* | Prop. à I_{36}, *LDOF acc* | Prop. à I_{99}, *LDOF acc*

Insertion de logo dans I_0 | Prop. à I_{20}, *MSF*(2*D-DE*) | Prop. à I_{36}, *MSF*(2*D-DE*) | Prop. à I_{99}, *MSF*(2*D-DE*)

Figure 1.9: Insertion de logo dans I_0 et propagation via *LDOF acc* et *MSF*(2*D-DE*) jusqu'à I_{99} (séquence *Newspaper*). *Steps* utilisés pour *MSF* : 1, 2, 5, 10, 20, 30, 40, 50 et 100.



(a) Carte de segmentation, I_0 | (b) Carte de segmentation, I_{100} | (c) Segmentation de I_0

(d) Propagation à I_{20} | (e) Propagation à I_{40} | (f) Propagation à I_{50}

(g) Propagation à I_{60} | (e) Propagation à I_{80} | (h) Segmentation de I_{100}

Figure 1.10: Propagation de labels de segmentation, séquence *AmeliaRetro*. L'opérateur fournit des cartes de segmentation pour I_0 (*a*,*c*) et I_{100} (b,h) et *MSF* propage ces labels à l'ensemble de la séquence. *Steps* utilisés : 1, 2, 5, 10, 20, 30, 40, 50 et 100.

beaucoup plus réalistes qu'avec *TV-L1 acc* ou *2D-DE acc*. En plus de ce gain en précision, notre stratégie *MSF*(*2D-DE*) a l'avantage de pouvoir sauter les occultations temporaires grâce à l'utilisation de vecteurs de flot optique *multi-steps* en entrée. Ainsi, l'exemple de propagation de logo Fig. 1.9 permet de voir que celui-ci est propagé après l'occultation temporaire avec *MSF* contrairement à *LDOF acc* qui stoppe la propagation dès lors que l'occultation a lieu. Enfin, Fig. 1.10 montre qu'une propagation de labels de segmentation peut également être réalisée sans défauts visibles grâce à *MSF*.

Pour conclure, les expériences réalisées ont montré qu'en plus d'être robuste aux occultations temporaires, *MSF* permet d'obtenir des champs de déplacement estimés à plus long terme et de meilleure qualité comparé à l'état de l'art.

1.3.3 Intégration combinatoire *multi-step* et sélection statistique

Pour des séquences complexes, on observe parfois la dérive de certaines trajectoires avec *MSF* due à une propagation séquentielle des erreurs. De plus, les critères considérés pour l'estimation du flot optique (dont ceux induits par le coût de matching) ne sont pas adaptés à la sélection de vecteurs de mouvement entre images distantes. C'est pourquoi nous suggérons de suivre une approche de construction de vecteurs long-termes non-séquentielle suivie d'une sélection basée sur un critère statistique exploitant la distribution spatiale d'un large ensemble de candidats ainsi que leur qualité intrinsèque.

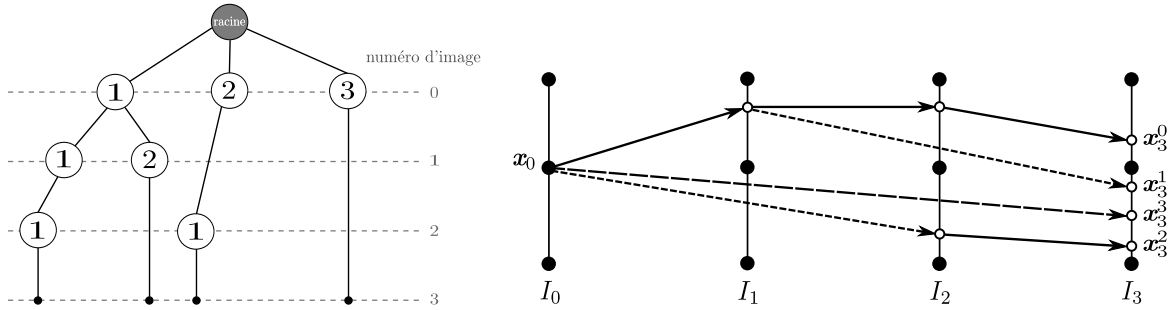
Nous proposons ainsi un nouvel estimateur de mouvement dense long-terme basé sur : 1) une méthode d'intégration combinatoire consistant à construire un grand ensemble de champs de mouvement candidats issus de multiples concaténations *multi-steps* et 2) une sélection du meilleur champ de mouvement parmi les candidats générés en combinant traitement statistique et optimisation globale. Cet algorithme a d'abord été développé entre une paire d'image distantes $\{I_a, I_b\}$ [5, 6] puis le principe a ensuite été étendu à l'ensemble de la séquence pour traiter chacune des paires $\{I_{ref}, I_n\}$ [7]. Il en résulte l'estimateur de mouvement dense long-terme *StatFlow*.

Intégration combinatoire et sélection statistique pour $\{I_a, I_b\}$ [5, 6]

Considérons à nouveau une séquence de $N+1$ images RGB $\{I_n\}$ avec $n \in [\![0, \dots, N]\!]$. Soient I_a et I_b deux images distantes ($0 \leq a < b \leq N$) entre lesquelles nous souhaitons estimer le mouvement. Soit $S_n = \{s_1, s_2, \dots, s_{Q_n}\} \subset \{1, \dots, N-n\}$ l'ensemble des Q_n *steps* disponibles à l'instant n. Cela signifie que les flots optiques élémentaires $\{\boldsymbol{v}_{n,n+s_1}, \boldsymbol{v}_{n,n+s_2}, \dots, \boldsymbol{v}_{n,n+s_{Q_n}}\}$ ont été pré-calculés à partir de l'image I_n.

La méthode de construction de vecteurs de mouvement candidats proposée consiste à générer tout d'abord toutes les séquences de *steps* disponibles entre I_a et I_b (Fig. 1.11 (a)). Chacune d'entre elles définit, après concaténation des flots optiques correspondants, un chemin de mouvement reliant chaque pixel $\boldsymbol{x}_a$ dans I_a à une position sub-pixélique dans I_b (Fig. 1.11 (b)).

Définissons $\boldsymbol{\Gamma}_{a,b}$ comme étant l'ensemble des K séquences de *steps* $\boldsymbol{\gamma}_i$ possibles entre I_a et I_b : $\boldsymbol{\Gamma}_{a,b} = \{\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_{K-1}\}$. $\boldsymbol{\Gamma}_{a,b}$ est calculé en construisant un arbre (Fig. 1.11 (a)) pour lequel chaque nœud correspond à un champ de mouvement défini pour une image donnée et pour une valeur de *step* donnée (valeur du nœud). L'arbre est créé récursivement à partir du nœud racine en générant autant de nœuds fils que de *steps* disponibles à l'instant courant. La génération d'une branche prend fin lorsque I_b est atteint ou dépassé. Les séquences de *steps* sont obtenues en parcourant l'arbre du nœud racine aux feuilles.



(a) (b)

Figure 1.11: (a) Génération de $\mathbf{\Gamma}_{a,b}$, l'ensemble des séquences de *steps* possibles entre I_a et I_b (b) Génération des chemins de mouvement *multi-steps* reliant chaque pixel $\boldsymbol{x}_a$ de I_a à un ensemble de positions candidates dans I_b.

Une fois toutes les séquences de *steps* $\boldsymbol{\gamma}_i \in \mathbf{\Gamma}_{a,b}$ obtenues, la génération des chemins de mouvement est ensuite effectuée par intégration *directe* (Fig. 1.11 (b)). En partant de chaque pixel $\boldsymbol{x}_a$ de I_a et pour chaque séquence de *steps* $\boldsymbol{\gamma}_i \in \mathbf{\Gamma}_{a,b}$ $\forall i \in [\![0, \ldots, K-1]\!]$, l'intégration consiste à accumuler les flots optiques élémentaires dont les *steps* correspondent à ceux constituant la séquence de *steps* courante. En parcourant tous les *steps* $s_j^i \in \boldsymbol{\gamma}_i$, nous obtenons $\boldsymbol{x}_b^i$, la position correspondante à $\boldsymbol{x}_a$ dans I_b via γ_i. L'ensemble des séquences de *steps* permettent d'obtenir l'ensemble des positions candidates dans I_b : $T_{a,b}(\boldsymbol{x}_a) = \{\boldsymbol{x}_b^i\}$ avec $i \in [\![0, \ldots, K_{\boldsymbol{x}_a} - 1]\!]$ où $K_{\boldsymbol{x}_a}$ est le cardinal de $T_{a,b}(\boldsymbol{x}_a)$. En pratique, les séquences de *steps* ne peuvent pas être toutes prises en compte en raison de problèmes mémoires et calculatoires. C'est pourquoi la procédure décrite ci-dessus est restreinte à un sous-ensemble des chemins de mouvement choisis de manière aléatoire guidée.

En ce qui concerne la sélection de vecteurs de mouvement optimaux, le but est de sélectionner pour chaque pixel x_a de I_a la position candidate optimale $\boldsymbol{x}_b^*$ parmi l'ensemble des positions candidates dans I_b, $T_{a,b}(\boldsymbol{x}_a) = \{\boldsymbol{x}_b^i\}$ où $i \in [\![0, \ldots, K_{\boldsymbol{x}_a} - 1]\!]$, obtenu lors de l'étape de construction des vecteurs de mouvement candidats. Avec l'hypothèse d'un modèle *Gaussien* décrivant la distribution spatiale de $T_{a,b}(\boldsymbol{x}_a)$, cette étape de sélection se résume à obtenir la valeur centrale de la distribution via un estimateur du maximum de vraisemblance :

$$\boldsymbol{x}_b^* = arg \min_{\boldsymbol{x}_b^i} med_{j \neq i} \left\| \boldsymbol{x}_b^j - \boldsymbol{x}_b^i \right\|_2^2 \tag{1.2}$$

Chaque candidat $\boldsymbol{x}_b^i$ de la distribution se voit assigner une valeur de qualité entière basée sur sa valeur d'incohérence *forward-backward*. Un mécanisme de vote permet de favoriser les candidats situés dans le voisinage de candidats de bonne qualité et ainsi de renforcer la cohérence *forward-backward*. Le traitement statistique étant appliqué indépendamment pour chaque pixel, nous y combinons une méthode d'optimisation globale incluant une régularisation spatiale.

En pratique, pour chaque pixel $\boldsymbol{x}_a \in I_a$, nous appliquons le traitement statistique à tout l'ensemble $T_{a,b}(\boldsymbol{x}_a)$. Le critère de minimisation de médiane définie dans l'Eq. (1.2) permet de sélectionner les N_{opt} meilleurs candidats qui constituent N_{opt} champs de mouvement. Ces champs sont ensuite fusionnés par paires par une méthode d'optimisation globale et cela jusqu'à la sélection du champ de mouvement final.

Extension à la vidéo (*StatFlow*) [7]

L'intégration combinatoire *multi-steps* et la sélection statistique ont été étendues à l'échelle de la séquence. L'estimateur de mouvement dense long-terme résultant, *StatFlow*, est composé de deux étapes principales :

1. versions étendues de l'intégration combinatoire *multi-steps* et du traitement statistique appliquées indépendamment pour chaque paire $\{I_{ref}, I_n\}$ afin de générer des correspondences initiales de mouvement dense (Fig. 1.12),

2. raffinement itératif impliquant des contraintes de cohérence temporelle en vue d'obtenir un matching dense finale.

La première étape consiste à construire des chemins de mouvement multiples pour chaque paire $\{I_{ref}, I_n\}$ en veillant à n'utiliser que des vecteurs de flot optique intrinsèquement consistents. La sélection aléatoire d'un sous-ensemble de ces chemins de mouvement effectuée pour chacune des paires $\{I_{ref}, I_n\}$ permet de limiter la corrélation entre candidats obtenus pour des images voisines. Cette indépendance statistique des candidats évite la propagation d'erreurs le long des trajectoires comme cela peut être le cas avec *MSF*. En pratique, pour chaque pixel x_{ref} (resp. x_n) de I_{ref} (resp. I_n), nous choisissons parmi tous les candidats résultants des chemins de mouvement sélectionnés K candidats dans I_n (resp. I_{ref}) par traitement statistique et optimisation globale. Enfin, parmi ces K candidats, nous identifions le candidat optimal x_n^* (resp. x_{ref}^*) en appliquant à nouveau la méthode d'optimisation globale.

Sans perdre la caractéristique précédemment évoquée concernant l'indépendance statistique des correspondances initiales temporellement, la deuxième étape a pour but d'obtenir un matching dense final en s'appuyant sur la corrélation temporelle des champs de mouvement long-termes. Pour cela, pour chaque pixel x_{ref} (resp. x_n) de I_{ref} (resp. I_n), nous mettons en jeu le candidat optimal x_n^* (resp. x_{ref}^*) dans I_n (resp. I_{ref}) en le confrontant de manière itérative aux candidats suivants :

- les K candidats de I_n (resp. I_{ref}) calculés lors la première étape,

- le candidat obtenu par inversion du champ de mouvement optimale entre I_n (resp. I_{ref}) et I_{ref} (resp. I_n),

- les candidats venant des images voisines (propagation des candidats optimaux des images voisines via les vecteurs de flot optique *multi-steps*).

Tous ces candidats sont fusionnés par une méthode d'optimisation globale dont le terme de données inclut des contraintes fortes de corrélation temporel. Pour ne pas tomber dans le travers des méthodes séquentielles, les paires $\{I_{ref}, I_n\}$ sont traitées dans un ordre aléatoire et non séquentiellement.

Résultats obtenus avec *StatFlow*

StatFlow a été évaluée par de nombreux tests quantitatifs et qualitatifs. Parmi les expériences réalisées, nous citons ici les comparaisons effectuées entre trajectoires obtenues avec *StatFlow* et trajectoires vérité-terrain par le biais des bases de données *Flag* [GRA11a] et *Hopkins* [TV07].

Sur la base des trajectoires denses vérité-terrain de la séquence *Flag* [GRA11a] (Fig. 1.13), nous observons Tab. 1.2 que l'erreur *RMS* (*Root Mean Square*) globale obtenue avec *StatFlow*(*LDOF*) (c'est à dire *StatFlow* estimé avec des vecteurs de flot optique *multi-steps LDOF*

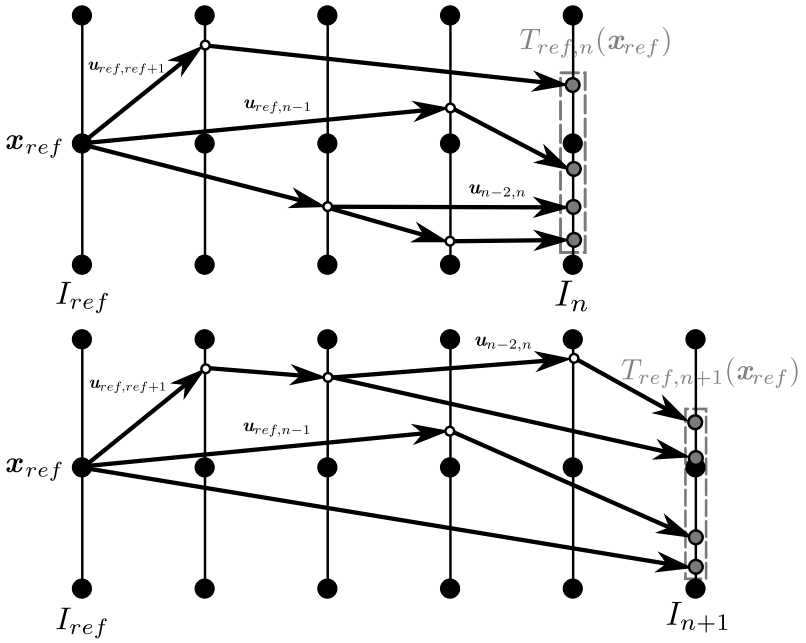


Figure 1.12: Estimation de mouvement dense long-terme via *StatFlow*. Intégration combinatoire et traitement statistique appliqués indépendamment pour chaque paire $\{I_{ref}, I_n\}$ afin de limiter la corrélation entre les candidats sélectionnés pour des images voisines puis raffinement itératif.

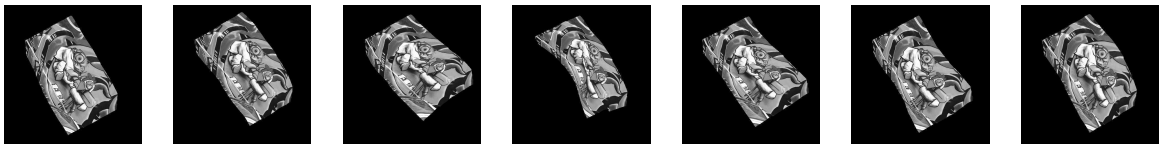
[BM11]) est inférieure ou égale aux erreurs obtenues par les méthodes de l'état de l'art dont *LDOF direct* (*LDOF* directement appliqué aux paires $\{I_{ref}, I_n\}$), *LDOF acc* et *MFSF-PCA* [GRA13], approche variationnelle selon laquelle la séquence de déplacement s'exprime comme une combinaison linéaire d'une base de trajectoire de rang faible. Les erreurs *RMS* obtenues pour chaque paire $\{I_{ref}, I_n\}$ et explicitées Fig. 1.14 montrent que parmi les méthodes se basant sur des vecteurs de flot optique *LDOF*, *StatFlow* est la méthode la plus fiable pour l'obtention de correspondences denses et long-termes de qualité.

Des comparaisons avec des trajectoires vérité-terrain éparses ont également été menées sur certaines séquences de la base de données *Hopkins* [TV07] dont *Hopkins-head* et *Hopkins-truck2*. Sur la base des erreurs de position médianes (*MedE*) illustrées Fig. 1.15, nous constatons que *StatFlow*(*LDOF*) est plus robuste que *MSF*(*LDOF*) ainsi que *LDOF inverse*.

Enfin, Fig. 1.16 compare *LDOF acc*, *MSF*(*2D-DE*), *StatFlow*(*2D-DE*) sur la base de leur faculté à propager des textures au sein d'une séquence comprenant des textures périodiques soumises à un grand mouvement et à de fortes variations de luminosité. Contrairement à *StatFlow* qui permet de réaliser une propagation de très bonne qualité, nous pouvons constater que *LDOF acc* et *MSF* sont sujets à des erreurs de matching importantes.

1.3.4 Stratégies *multi-steps* basées images de référence multiples

Après l'étude approfondie de nouvelles stratégies d'estimation de mouvement dense long-terme basées sur une unique image de référence (Sections 1.3.2 et 1.3.3), nous avons également étudié, implémenté puis testé des traitements avec des images de références multiples. Ces traitements ont pour but de raffiner les trajectoires pour que celles-ci soient robustes aux variations d'illumination importantes ainsi qu'à de fortes occultations. Pour cela, nous combinons des



(a) I_1 (b) I_{10} (c) I_{20} (d) I_{30} (e) I_{40} (f) I_{50} (g) I_{60}

Figure 1.13: Images sources de la séquence *Flag* [GRA11a].

Method	Erreur *RMS* globale (pixels)
***StatFlow* (*LDOF*)**	**0,69**
MSF (*LDOF*)	1,41
LDOF direct [BM11]	1,74
LDOF acc [BM11]	4
MFSF-PCA [GRA13]	**0,69**
MFSF-DCT [GRA13]	0,80
MFSF-PCA [GRA11b]	0,98
MFSF-DCT [GRA11b]	1,06
[PB12] *direct*	1,24
ITV-L1 direct [WPZ+09]	1,43

Table 1.2: Erreurs *RMS* globales obtenues pour différentes méthodes grâce aux trajectoires vérité-terrain fournis par la base de données *Flag* [GRA11a] (séquence *Flag*, Fig. 1.13). *Steps* utilisés pour *MSF* et *StatFlow* : 1, 2, 3, 4, 5, 8, 10, 15, 20, 25, 30, 40 et 50.

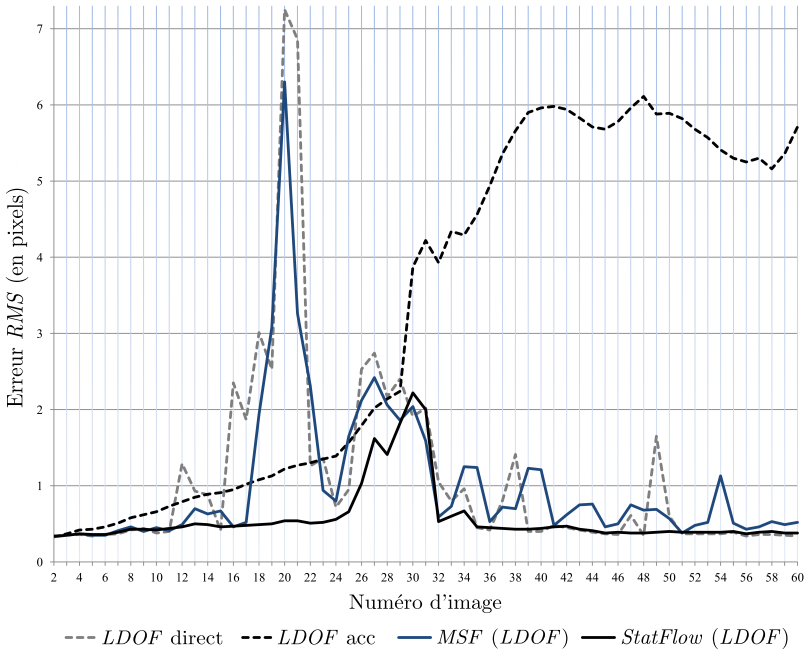
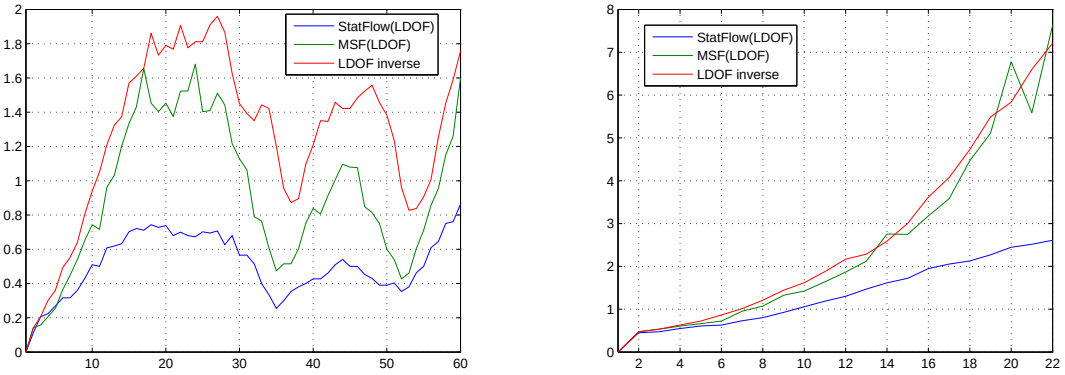


Figure 1.14: Erreurs *RMS* pour chaque paire $\{I_{ref}, I_n\}$ de la séquence *Flag* [GRA11a] avec les méthodes suivantes : *LDOF direct*, *LDOF acc*, *MSF*(*LDOF*) et *StatFlow*(*LDOF*). *Steps* utilisés pour *MSF* et *StatFlow* : 1, 2, 3, 4, 5, 8, 10, 15, 20, 25, 30, 40 et 50.



(a) *Hopkins-head* (b) *Hopkins-truck2*

Figure 1.15: Erreurs *MedE* pour deux séquences (*Hopkins-head* et *Hopkins-truck2*) de la base de données *Hopkins* [TV07] entre les trajectoires vérité-terrain et les celles obtenues avec : *LDOF inverse*, *MSF*(*LDOF*) et *StatFlow*(*LDOF*). *Steps* utilisés pour *MSF* et *StatFlow* : 1, 2, 3, 4, 5, 10, 20 et 40.

tracklets estimées en *forward* ou en *backward* à partir d'images de référence positionnées dans la vidéo sur des critères de qualité de trajectoires. Les algorithmes développés permettent une estimation robuste du mouvement dense à très long-terme.

1.3.5 Conclusion et perspectives

Le travail mené en estimation de mouvement dense et long-terme a conduit à plusieurs contributions, toutes basées sur une manipulation robuste de vecteurs de flot optique *multi-steps*. Pour améliorer ces stratégies, nous préconisons un certain nombre de perspectives qui mériteraient davantage d'attention telles que la sélection automatique d'images de référence et de *steps* candidats, l'introduction explicite de mesures de gain afin de prendre en compte de manière plus robuste encore les fortes variations d'illumination, la prise en compte de nouveaux types d'information de mouvement en entrée de nos algorithmes (trajectoires éparses, vecteurs de flot optique provenant de différents estimateurs, champs paramétriques et non-paramétriques...), une meilleure gestion des occultations ou encore la prise en compte d'interactions semi-automatiques.

1.4 Application à la coopération stéreo-mouvement

1.4.1 De nouvelles perspectives en coopération stéréo-mouvement

De nouvelles perspectives s'offrent à nous lorsque l'on considère simultanément estimation de disparité et estimation de mouvement dense long-terme pour des séquences stéréo/multi-vues. Il en résulte notamment les traitements suivants, dédiés à la coopération stéréo-mouvement :

- estimation ou correction de disparité en utilisant des champs de déplacement long-terme pré-calculés. Ceux-ci peuvent :
  - fournir des contraintes additionnelles (telles que des contraintes de similarité de mouvement/trajectoires) ainsi que de nouveaux candidats pour l'estimation de disparité,

(a) Image source I_0 (d) Propagation à I_{20} *LDOF acc* (e) Propagation à I_{25} *LDOF acc* (f) Propagation à I_{40} *LDOF acc*

(b) Insertion de logo dans I_0 (g) Propagation à I_{20} *MSF* (2*D-DE*) (h) Propagation à I_{25} *MSF* (2*D-DE*) (i) Propagation à I_{40} *MSF* (2*D-DE*)

(c) Image source I_{40} (j) Propagation à I_{20} *StatFlow* (2*D-DE*) (k) Propagation à I_{25} *StatFlow* (2*D-DE*) (l) Propagation à I_{40} *StatFlow* (2*D-DE*)

Figure 1.16: Insertion de texture dans I_0 et propagation jusqu'à I_{40} (séquence *Walking-Couple*). Nous comparons d-f) *LDOF acc*, g-i) *MSF*(2*D-DE*), j-l) *StatFlow*(2*D-DE*). *Steps* utilisés pour *MSF* et *StatFlow* : 1, 2, 3, 4, 5, 8, 10, 15, 20 et 30.

  - propager automatiquement des vecteurs de disparité correctement estimés via un processus automatique ou manuellement corrigés par un opérateur afin de corriger des vecteurs de disparité erronés,

- estimation de mouvement dense et long-terme en utilisant des champs de disparité calculés préalablement pouvant fournir des contraintes additionnelles (similarité de disparité par exemple) ainsi que de nouveaux candidats.

- estimation conjointe disparité/mouvement dense long-terme en alternant estimation de disparité à partir du déplacement dense long-terme et estimation du déplacement dense long-terme à partir de la disparité.

Contrairement aux méthodes de l'art [MPPC09, DMPP10, Gon06, LH10][4] qui se limitent généralement à un traitement séquentiel de quadruplets d'images consécutives, l'introduction du mouvement dense long-terme permet d'étendre chacune des perspectives mentionnées ci-dessus en des traitements long-termes robustes. On peut ainsi envisager des applications telles que la correction d'artéfacts de synthèse de vues par propagation d'information de disparité correcte aux zones incorrectement estimées via des champs de déplacement long-termes, la conversion 2D-3D ou l'édition vidéo automatique ou semi-automatique visant à propager des modifications effectuées dans une seule image à l'ensemble de la séquence binoculaire ou multi-vues.

1.4.2 Une nouvelle chaine de traitement dédiée à la correction de disparité

Dans ce contexte, nous avons approfondi les aspects correction de disparité et d'artéfacts de synthèse de vues correspondants à l'aide des champs de déplacement denses long-termes. La chaine de traitement proposée combine à la fois estimation de disparité, synthèse de vues et évaluation de la qualité de la synthèse de vues via *VSQA* (Section 1.2) ainsi que l'estimation de mouvement dense long-terme (Section 1.3).

La correction de disparité débute par une classification des vecteurs de disparité estimés pour chaque paire d'images gauche/droite appartenant à la séquence binoculaire en deux catégories : vecteurs de disparité correctement estimés et vecteurs de disparité erronés. Cette classification automatique s'opère par synthèse de vue et détection des artéfacts de synthèse de vue via *VSQA*. Les vecteurs de disparité ayant généré les artéfacts de synthèse de vue sont classés comme étant erronés. Les autres vecteurs de disparité sont considérés comme correctement estimés.

Ensuite, une estimation de la qualité globale basée sur *VSQA* permet de sélectionner deux paires d'images de référence. Ces paires d'images de référence sont choisies comme étant les paires faisant l'objet du nombre d'artéfacts de synthèse le plus faible à l'échelle de la séquence. Enfin, les vecteurs de disparité des paires de référence ayant été correctement estimés sont propagés à l'ensemble de la séquence à partir des deux références grâce aux champs de déplacement long-termes pré-calculés. Cette propagation d'information fiable permet de corriger les zones erronées en terme d'estimation de disparité. La correction de disparité a un effet bénéfique sur les artéfacts de synthèse de vue qui se retrouvent atténués voir supprimés une fois une nouvelle synthèse effectuée.

[4] les références alphabétiques de la Section 1.4 sont détaillées dans la bibliographie de la Partie III, page 296

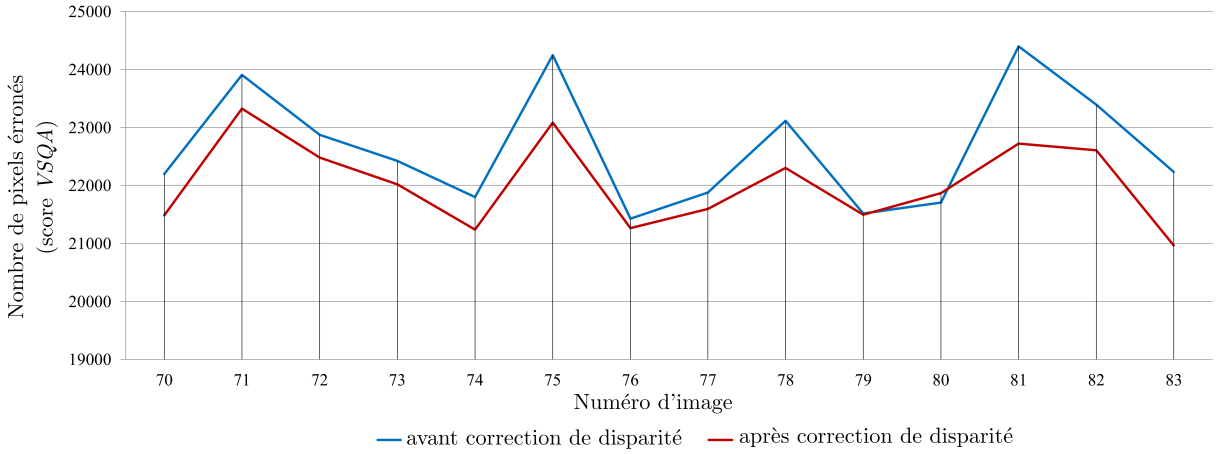
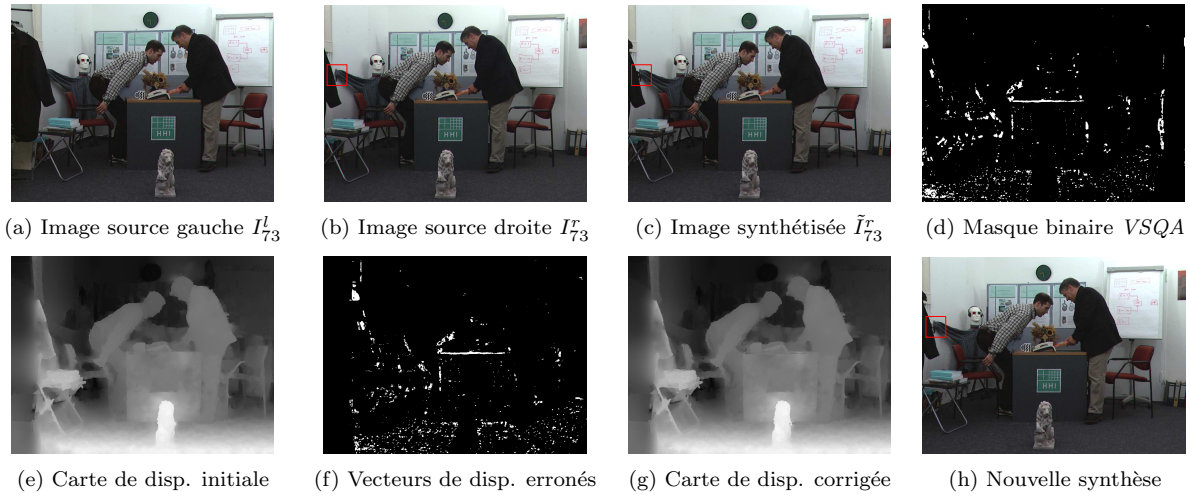


Figure 1.17: Comparaison du nombre de pixels erronés (score *VSQA*) avant et après le traitement de correction de disparité proposé (séquence *Book-Arrival*). Les deux paires d'images de référence sont les suivantes : $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$.



(a) Image source gauche I^l_{73} (b) Image source droite I^r_{73} (c) Image synthétisée $\tilde{I}^r_{73}$ (d) Masque binaire *VSQA*

(e) Carte de disp. initiale (f) Vecteurs de disp. erronés (g) Carte de disp. corrigée (h) Nouvelle synthèse

(i) Zoom sur I^r_{73} (j) Zoom sur $\tilde{I}^r_{73}$ (k) Zoom sur $\tilde{I}^r_{73}$

Avant correction de disparité **Après correction de disparité**

Figure 1.18: Evaluation qualitative via synthèse de vues du traitement de correction de disparité.

1.4.3 Premiers résultats

Notre traitement de correction de disparité a fait l'objet de premiers tests sur la séquence *Book-Arrival* par le biais de l'étude de la qualité des vues synthétisées produites avant et après correction. Comme l'atteste Fig. 1.17, le nombre de pixels considérés comme appartenant à un artéfact de synthèse de vues le long de la séquence diminue une fois la correction de disparité effectuée, avec un gain maximal de $6,88\%$ pour la paire $\{I^l_{81}, I^r_{81}\}$. Visuellement, on remarque Fig. 1.18 que la correction des cartes de disparité permet l'atténuation de certains artéfacts de synthèse de vues comme c'est le cas en (i,j,k).

1.4.4 Conclusion

Pour permettre une correction des cartes de disparité plus efficace encore, il serait intéressant dans le cadre d'un traitement semi-automatique de propager des corrections des vecteurs de disparité manuellement effectuées par un opérateur. Du point de vue automatique, le traitement proposé pourrait être amélioré par l'ajout d'un raffinement *a-posteriori* par filtrage spatio-temporel multilatéral ou par le calcul et la prise en compte de modèles d'évolution temporelle de la disparité qui pourraient servir de contraintes fortes pour la tâche de correction.

1.5 Conclusion générale

Le traitement combiné de contenu capturé et de synthèse requiert des fonctionnalités automatiques robustes afin d'offrir aux observateurs des contenus monoscopiques ou stéréoscopiques dont le rendu est réaliste. Parmi ces fonctionnalités, nos travaux de recherches ont porté notamment sur l'évaluation de la qualité de la synthèse de vue, problématique peu explorée dans la littérature et cependant primordiale en ce qui concerne la diffusion de contenus auto-stéréosopiques de qualité, ainsi que sur l'élaboration de stratégies d'estimation de mouvement dense et long-terme fiables permettant, entre autres, de propager de manière automatique des informations synthétiques dans les séquences vidéo. Les contributions proposées dans chacun de ces deux domaines ont ensuite été considérées conjointement dans le contexte de la coopération stéréo-mouvement. Les larges perspectives qu'ouvre notre étude mériteraient à l'avenir des travaux complémentaires, notamment concernant l'estimation de qualité des vidéos synthétisées, l'amélioration des stratégies automatiques ou semi-automatiques d'estimation de mouvement dense à très long-terme pour des séquences complexes ou encore l'estimation conjointe disparité/mouvement dense long-terme.

Chapter 2

Introduction

2.1 Context

In the last few years, film and consumer electronics industries have known huge technological improvements to capture, transmit and display high-quality monoscopic and stereoscopic video content. These improvements have been made in order to provide to the viewer the most realistic viewing experience. Due to artistic intentions or due to physical limitations to efficiently capture and transmit video contents, it is sometimes necessary to combine simultaneously captured and synthetic data while taking care to maintain a photo-realistic rendering. To efficiently process captured and synthetic content simultaneously, production and post-production operators need to be assisted by sophisticated automatic tools.

Among these functionalities, one predominant issue is the establishment of robust correspondences between images. This is a very challenging task, especially for complex scenes featuring non-rigid deformations, large motion, poorly textured areas, zooming, illumination changes or transparency. Matching algorithms can be involved either for stereo or multi-view setups to match views captured from slightly different viewpoints via disparity estimation or for image sequences to link temporally images acquired at different moments in time through motion estimation.

In both spatial and temporal contexts, establishing image correspondences induces numerous specific applications. First, in the context of stereoscopic imaging, we investigate in particular the field of view synthesis involved together with disparity estimation within *Depth-Image-Based Rendering* (*DIBR*) algorithms. Following such algorithms, the idea is to generate synthetic views of 3D scenes starting from available captured views. Second, regarding temporal aspects, we focus on long-term dense motion estimation whose goal is to construct dense fields of correspondences over extended time periods. These dense motion correspondences can, for instance, automatically propagate synthetic data across long video sequences. Disparity estimation and view synthesis issues as well as long-term dense motion estimation define the general context of our research activities.

More precisely, disparity estimation and view synthesis aspects are studied from the perspective of high-quality image-based rendering since *DIBR* algorithms may introduce distortions which can strongly impair the viewer comfort. Toward the goal of enhancing the viewing experience offered by current 3D technologies, our study is mainly dedicated to the analysis and the detection of artifacts inherent to the *DIBR* process.

In the context of motion estimation, strategies for long-term dense matching applied to monoscopic video sequences are investigated. Numerous video processing tasks such as semi-

automatic video editing require a temporally smooth and accurate dense description of how the video content varies in time. The challenge consists in proposing dense motion estimation techniques whose spectrum is not limited to the matching of consecutive frames only.

Both view synthesis quality assessment and long-term dense motion estimation issues are first of all investigated separately in our study. To make the link between these two topics, contributions of both fields are then applied to joint disparity and motion processing.

This work has been led in both academic and industrial environments. It has been especially involved within research projects whose main goal is to conceive and experiment algorithms in order to provide post-production services in the global entertainment industry.

2.2 Motivations

An overview of the motivations in the fields of view synthesis quality assessment, long-term dense motion estimation and joint disparity and motion processing is provided in what follows.

View synthesis quality assessment

The 3D processing chain, from acquisition to display, has known significant progress since the invention of the first 3D viewing device in 1838. As proved by the advent of 3D video products to the mass consumer market, 3D technologies are now enough mature to reach a relatively satisfying quality level in terms of viewing experience. Among current 3D technologies, 3D autostereoscopic displays especially give a sensation of immersion far beyond what is offered by traditional media thank to the generation of new virtual viewpoints through disparity estimation and view interpolation involved together within *DIBR* algorithms. Indeed, due to physical limitations to efficiently capture and transmit a significant number of views, *DIBR* allows to provide additional virtual views starting from a smaller number of acquired views.

Despite recent advances, *DIBR* algorithms do not always provide artifact-free and realistic-looking synthesized views and induce new types of artifacts whose impact can be harmful for the observer. Both disparity estimation and view synthesis may encounter some issues in a variety of situations and the issues related to the quality assessment of synthesized views has not been widely investigated. Based on these findings, the following questions arise: What are the sources of distortions of view synthesis? Which features can strongly impact their visibility and how to mathematically formalize these features? How to built an image quality assessment method able to efficiently detect such distortions?

Long-term dense motion estimation

Analyzing the temporal dynamic of objects is one of the major tasks of both human and artificial visual systems and motion estimation has consequently become one of the predominant topics in computer vision. Since early *optical flow* formulations in the beginning of the 80's, dense motion estimation has mainly dealt with matching consecutive frames. However, numerous applications such as video segmentation, analysis techniques or video editing have recently motivated both dense and long-term requirements.

Establishing dense and long-term correspondences through dense trajectory computation translates in computing motion between distant frames and therefore in handling simultaneously small and large displacements. Moreover, classical *optical flow* assumptions which may fail between consecutive frames are even less valid between non-consecutive frames, especially for

complex scenes or for significant video content changes in time. Another challenge deals with the occlusion detection task since occluded areas become wider when considering distant frames.

Recent methods have contributed to the purpose of long-term dense motion estimation through *optical flow* concatenation, long-term temporal smoothness regularization or multi-frames *optical flow* formulations. However, the resulting dense long-term trajectories are not able to face with the previously described challenges and their reliability does not exceed more than about thirty frames. Moreover, relying only on motion fields computed between consecutive frames does not allow to recover trajectories after temporary occlusions.

Considering *optical flow* vectors as the natural tool to build long-term dense correspondences and assuming that their use can be extended to distant frames, we claim that alternative strategies can be built in order to limit the motion drift while dealing with temporary occlusions. Our contributions toward this goal are especially motivated by applications that require to propagate across the sequence dense information such as color, disparity, depth, position or any other type of visual information.

Application to joint stereo and motion processing

Motion and disparity information are rarely involved together for solving computer vision tasks. When simultaneously considered, joint stereo and motion processing is generally restricted to quadruplets of images processed sequentially across the binocular sequence. To go further, we propose to rely on our contributions in both domains to imagine new joint stereo and long-term motion processing. Among the different potential applications, we suggest to involve long-term dense motion fields as well as information related to view synthesis quality assessment in order to perform a new disparity correction framework.

2.3 Thesis outline

This thesis dedicated to view synthesis quality assessment and long-term dense motion estimation is divided in three main parts.

Part I investigates the previously mentioned view synthesis quality assessment issues through an introduction to stereoscopic imaging (Chapter 3) and an illustrated description of the possible sources of distortions combined with a study of the main features involved within artifact masking mechanisms (Chapter 4). In addition, we address the state-of-the-art of objective monoscopic and stereoscopic image quality assessment (Chapter 5) before proposing in Chapter 6 a new metric whose goal is to efficiently detect view synthesis artifacts: the *View Synthesis Quality Assessment* (*VSQA*) metric.

Part II focuses on long-term dense motion estimation by first of all describing how *optical flow* computation methods perform dense motion estimation between two consecutive frames (Chapter 7). Then, Chapter 8 studies how the literature has extended *optical flow* to the purpose of long-term motion estimation and reviews applications for which long-term temporal consistency is key. An introduction to our contributions is provided in Chapter 9, especially through the presentation of the concept of *multi-step* elementary *optical flow* estimation which extends the conventional use of existing *optical flow* estimators to distant frames. Based on this concept, we propose in Chapter 10 and Chapter 11 new long-term dense motion estimation approaches:

- *Multi-step* flow via *graph-cuts* (*MS-GC*) and *multi-step* flow fusion (*MSF*) which consist in both accumulating *multi-step* elementary *optical flow* vectors through inverse integration and merging the resulting candidate long-term displacement fields,

- *Statistical multi-step flow* (*StatFlow*), based on a combinatorial integration of *multi-step* elementary *optical flow* vectors followed by a statistical-based long-term displacement vector selection.

We suggest in Chapter 12 to exploit the concept of multi-reference frames estimation to the purpose of very long-term dense motion estimation. In particular, we study how long-term dense motion estimators such as the ones proposed in Chapter 10 and Chapter 11 can be involved within a multi-reference frames framework toward longer accurate dense long-term correspondences. Chapter 13 concludes this Part II and evoke the aspects which must deserve more attention for further research.

Part III explains how the contributions of the two first parts can be combined to consider joint stereo and long-term motion processing. In particular, we propose in Chapter 14 to study a disparity correction framework which includes disparity estimation, view synthesis, view synthesis quality assessment (Part I) as well as long-term dense motion estimation (Part II).

Part I

View synthesis and quality assessment

Chapter 3

Introduction to stereoscopic imaging

Since the invention of the first 3D viewing device in the first half of the XIXth century, stereoscopic imaging has experienced an uneven development. Previous attempts to make 3D videos gain more awareness have been stopped due to immature technologies which were responsible for creating a strong viewer discomfort. The rebirth of stereoscopic imaging is quite recent and the whole 3D processing chain, from acquisition to display, has known significant progress in recent years. Current technologies lead to a much higher level of quality than in the past, as proved by the advent of 3D video products to the mass consumer market.

3D technologies provide a depth perception of the observed scene and therefore increase the sensation of immersion far beyond what is offered by traditional media. The 3D experience is potentially more powerful through multi-view systems such as autostereoscopic displays in the context of *3D Television* (*3D-TV*). Due to physical limitations to efficiently acquire and transmit a significant number of views, the need arises to generate additional realistic-looking virtual images, as if they were truly acquired from different perspective viewpoints.

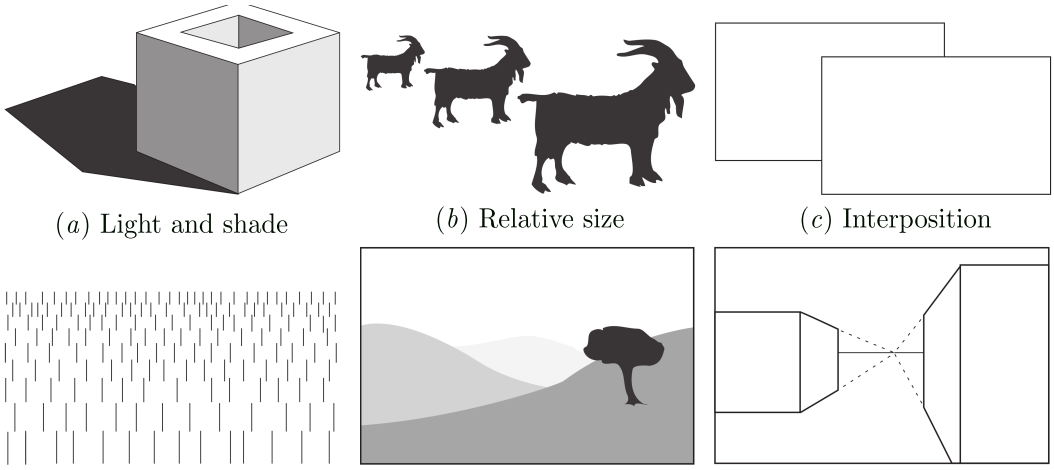
In this context, view synthesis appears to be one of the crucial stages of 3D processing chains for *3D-TV*. When involved within *Depth-Image-Based Rendering* (*DIBR*) algorithms, view synthesis requires as inputs stereo correspondences between the original views. This depth information can be obtained using a disparity estimation algorithm.

Before focusing on view synthesis artifacts and studying the view synthesis quality assessment task, the aims of this Part I, it is necessary to provide the fundamental concepts of stereoscopic imaging and to give an overview of view synthesis principles.

Toward these goals, this chapter is organized as follows. Section 3.1 studies the stereoscopic imaging fundamentals through the description of the mechanisms of stereoscopic perception. History of stereoscopic imaging, current 3D displays as well as 3D content generation techniques are also discussed. Then, Section 3.2 introduces the principles of disparity estimation and focuses on the inherent issues of disparity estimation. Finally, the overview on view synthesis takes place in Section 3.3. Section 3.4 concludes this chapter.

3.1 History and principles of stereoscopic vision

This section describes the basics of stereoscopic vision and focuses more precisely on the mechanisms of stereoscopic perception (Section 3.1.1), the evolution of the underlying technology starting from the invention of the first 3D viewing device in 1838 (Section 3.1.2) as well as 3D displays (Section 3.1.3) and 3D content generation (Section 3.1.4).



(*a*) Light and shade (*b*) Relative size (*c*) Interposition

(*d*) Textural gradient (*e*) Aerial perspective (*f*) Perspective

Figure 3.1: Illustration of six monoscopic depth cues described in [Lip97]. The seventh monoscopic depth cue, motion parallax, is not illustrated here.

3.1.1 Stereoscopic perception

Stereoscopic imaging refers to the process which offers to the viewer both left and right views obtained with different perspective viewpoints. By fusing these two slightly different views, the human brain is able to synthesize an image of the 3D scene and therefore to perceive objects at different depths.

The denomination $3D$ *imaging* is confusing and we should rather use the term stereoscopic imaging because in the field of computer graphics $3D$ refers to the image rendering process through 3D scene models.

The ability to perceive the world in three dimensions, commonly called depth perception, is based on a variety of depth cues. These depth cues can be split into monoscopic depth cues and binocular depth cues [Mat09]. The basic monoscopic depth cues are illustrated in Fig. 3.1 and briefly described below:

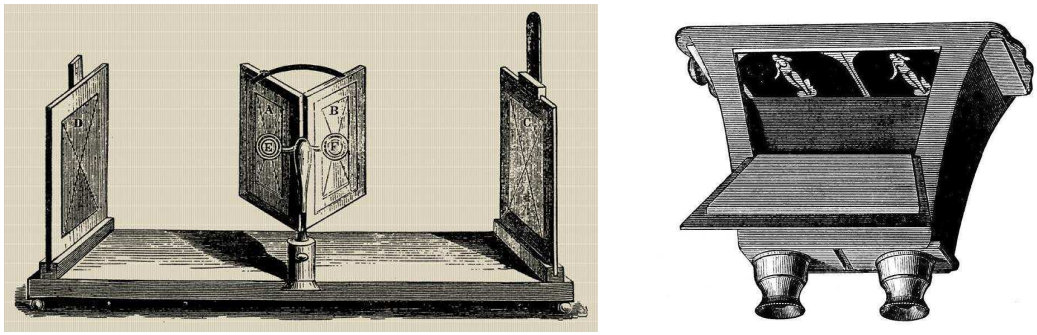
- light and shade: shadows help in providing depth information to the observer. If we assume for instance that the light comes from a position near the observer himself, objects in shadow areas must be farther from the light than objects that are not in shadow (Fig. 3.1 (*a*)).

- relative side: objects appear larger when they are closer, and smaller when they are far from the observer (Fig. 3.1 (*b*)),

- interposition: objects occluding each other suggest their depth ordering (Fig. 3.1 (*c*)),

- textural gradient: a textured material provides depth information (Fig. 3.1 (*d*)) because the texture is more apparent as the object is closer to the observer,

- aerial perspective: suggests the diminution of visibility for distant objects (Fig. 3.1 (*e*)),



(*a*) *Charles Wheatstone* mirror *stereoscope* (*b*) *David Brewster stereoscope*

Figure 3.2: First 3D viewing devices: the *Charles Wheatstone* mirror *stereoscope* (1838) and the *David Brewster stereoscope* (1849).

- perspective: deals with perspective projection laws and in particular with the appearance of parallel 3D lines or planes as vanishing 2D points and lines in retinal images (Fig. 3.1 (*f*)),
- motion parallax: 2D motion of closer objects is faster compared to distant objects for camera motion in front of static scene and/or motion in front of rigidly moving scene,

Stereoscopy, the visual perception process that reconstructs the 3D shape as well as the depth of each object of the scene, brings two additional physiological depth cues:

- vergence: the vergence angle depends on the distance to the observer. For object which are close from the observer, the eyes rotate in order to converge. On the contrary, object at infinite distance require the divergence of the eyes in order to become parallel,
- binocular disparity: the positional difference of a scene point between the two retinal images (parallax). The brain fuses these left and right retinal images and then extracts relative depth information from retinal disparity.

Stereoscopic depth cues are mainly involved for short distances contrary to greater distances for which monoscopic depth cues have a more stronger impact on 3D perception.

The mental interpretation of all these depth cues to perceive the world in three dimensions is an ability which is learned during childhood. It is not until the age of five months that the eyes are able to work together and therefore to perceive in depth [Ass13].

3.1.2 Brief history of stereoscopic imaging

While the binocular vision process has been discovered during the Ancient Greece, the history of 3D technology really starts with the invention of the first 3D viewing device, created by *Charles Wheatstone* in 1838 [Whe38]. This first 3D viewing device, called *stereoscope*, provides a simultaneous observation of two slightly different views via a system of mirrors (Fig. 3.2 (*a*)). The *stereoscope* has been then improved by *David Brewster* who made the first portable 3D viewing device in 1849 (Fig. 3.2 (*b*)).

(*a*) Left and right images (*b*) Anaglyph image

Figure 3.3: Stereo image (*b*) anaglyphed for red and cyan filters with associated left and right original views (*a*).

In the same period appear the first anaglyph images with associated color-coded anaglyph glasses. To achieve stereoscopy, each eye's image is encoded using filters of chromatically opposite colors, as shown in Fig. 3.3 with red and cyan filters. Through color-coded anaglyph glasses, each of the two images reaches one eye which reveals an integrated stereoscopic image. The first method to produce anaglyph images has been developed in 1852 by *Wilhelm Rollmann*.

On september the 30^{th}, 1922, the first 3D feature film was displayed at the *Ambassador Hotel Theater* of *Los Angeles*: *The Power of Love* [Zon07]. In 1928, *John Logie Baird* demonstrates the first stereoscopic 3D television. In the field of stereoscopic cinema, *Devernay* and *Beardsley* report in [DB10] the two first waves of commercial movies: in the early 50's when stereoscopic cinema was a method to get back the audience lost due to the development of the television and then in the 80's with the advent of large format stereoscopic movies. However, these two first attempts have not known great success due to the unavailability of relevant technologies.

The recent rebirth of stereoscopic imaging has been allowed by the significant improvements made in 3D graphics, display technology and 3D content generation.

3.1.3 3D imaging displays

Stereoscopic images are widely used nowadays to give the illusion of depth through 3D display technology. When displayed to an observer, the left and right images must be mutually consistent in order to allow a correct 3D reconstruction by the human brain. This implies the previously described geometric and photometric constraints (Section 3.1.1) between the images displayed to both left and right eyes [DB10].

Existing $3D$ displays can be classified into two categories depending on the use of glasses or not: stereoscopic and autostereoscopic displays.

Stereoscopic displays

Stereoscopic displays are characterized by the necessity for the viewer to wear glasses which allow the left and right images to be seen by the corresponding eye. Stereoscopic displays are classified into time-parallel or time-sequential displays [MIS04]. For time-parallel displays, left and right views are simultaneously sent on the screen and the distinction between left and right views is done for the viewer through:

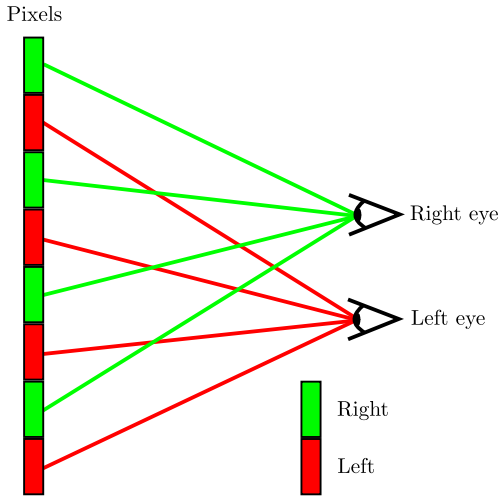


Figure 3.4: A two-view spatially multiplexed autostereoscopic display.

- location multiplexing: redirection of the left and right views through separate channels (in the spirit of the *stereoscope* devices of *Charles Wheatstone* and *David Brewster* displayed in Fig. 3.2),
- color-multiplexing using anaglyph images as well as color-filter glasses,
- polarization multiplexing: image separation using polarized light (monitors or projectors covered with linear or circular polarizing filters) and maintained using polarized glasses.

For time-sequential displays, left and right images are displayed in rapid alternation. Such method requires synchronized active glasses which open alternately for the appropriate eye while closing the other eye's view [MIS04].

Autostereoscopic displays

Autostereoscopic displays present left and right images with the required disparity without requiring any glasses. We can distinguish two-view binocular systems (Fig. 3.4) from multi-view systems. For two-view displays, only a single stereo pair is displayed, whereas in multiview displays, multiple stereo pairs are produced to provide 3D images to multiple users [UCES11].

Autostereoscopic displays are best suited for $3D\text{-}TV$ since the need of glasses is not convenient at home. In practice, such displays can be classified into three categories:

- direction-multiplexed displays: send the left (resp. right) view directly to the corresponding left (resp. right) eye,
- volumetric displays: reproduce the scene within a limited volume in space, as opposed to the planar image of traditional screens,
- holographic displays: perform an ideal free viewing 3D technique by recording and reproducing the properties of light waves via modulation of coherent light.

The two last display categories could potentially offer the most optimal 3D experience but the technology is still under study. Only direction-multiplexed autostereoscopic displays can be considered as a relatively mature technology.

3.1.4 3D content generation

3D displays require appropriate 3D content, i.e several images (at least two) representing the scene from slightly different viewpoints. These views can be originally acquired using multiple cameras.

However, the high number of views required for multi-view autostereoscopic displays (from 5 to 22) is a real challenge in terms of acquisition and transmission due to the physical limitation of cameras and the bandwidth of communication channels [NNKD+10, KNND+10]. To overcome this issue, one can both:

- limit the number of original views to be stored and transmitted (transmitting numerous views of the same 3D scene is extremely expensive in term of transmission costs especially) and,

- rely on *Depth-Image-Based-Rendering* (*DIBR*) algorithms to generate virtual views through interpolation or extrapolation.

Such content generation for autostereoscopic displays is referred to as stereo-to-multiview conversion.

To synthesize new realistic views at a slightly different view perspective, *DIBR* algorithms require original textured images as well as associated depth information. Therefore, *DIBR* algorithms need both an accurate disparity estimation process and robust view synthesis techniques. These two points are precisely described respectively in Section 3.2 and Section 3.3.

3.2 Principles of disparity estimation

In this section, the description of the general principles of disparity estimation is divided into two parts. First, we introduce the concept of disparity estimation in Section 3.2.1. Second, we provide the inherent issues of disparity estimation as well as the constraints found in the literature to solve the resulting matching ambiguities (Section 3.2.2).

3.2.1 Introduction to disparity estimation

Disparity estimation consists in establishing stereo correspondences. This translates in computing for each pixel of an image the corresponding point in the other image. Each 3D line of sight of these 2 points projects onto a 2D line in the other view (called epipolar lines). More generally, all the points located on a 3D plane containing the two optical points (perspective projection centers) project onto two epipolar lines in the image. All the points of one epipolar line have their correspondence in the other epipolar line.

Epipolar geometry requires camera calibration (identification of the parameters that define the relation between the 3D space and image coordinate systems). Moreover, both images are generally rectified before mutual processing so that the same image lines become corresponding epipolar lines.

The field of disparity estimation is a very active field which has known a great evolution at the instigation of *Scharstein* and *Szeliski* who have proposed through the *Middlebury* benchmark [SS11] a way to compare stereo algorithms based on a set of objective criteria [SS02].

To understand the relationship between depth and disparity in the context of projective geometry, let us focus on a standard rectified stereo setup (Fig. 3.5) where the physical point P of the 3D scene is defined by the following 3D coordinates: (X, Y, Z). Let $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ be the projection of P respectively in the left and right views. The depth Z and the disparity d are linked by a relationship which involves both the baseline b and the focal length f, as shown in Eq. 3.1:

$$d = \boldsymbol{x}_l - \boldsymbol{x}_r = \frac{f.b}{Z} \tag{3.1}$$

According to Eq. 3.1, disparity values are inversely proportional to the depth Z which means that far points will have a low disparity value (the disparity of the horizon equals 0 for instance) contrary to close points which will have a high disparity value.

Depending on the configuration, one distinguishes binocular stereo matching from multi-view stereo matching. We focus here on dense two-frame stereo correspondence algorithms having in mind the fact that stereo matching feeds the development of multi-view matching.

3.2.2 Matching ambiguities and constraints

Disparity estimation is a complex problem as numerous cases of image content introduce ambiguity in disparity:

- Noise: a stereo algorithm must be robust to unavoidable light reflections, image blurring and sensor noise,
- Occlusions: occluded pixels in one view should not be matched with points in the other view,
- Textureless areas: information from textured areas must be propagated into textureless areas through spatial regularization since pixels within textureless areas can hardly be distinguished by an intensity criterion,
- Depth discontinuities: the spatial smoothness constraint must be stopped at depth boundaries to rigorously segment objects within disparity maps,
- Periodic structures: the matching process must avoid to switch from one structure of another neighbouring structure,
- Transparency: in this particular situation, disparity estimators should compute a disparity per layer and rely on joint depth and alpha matte estimation [ZLYP09].

The objective of recent disparity estimators is to satisfactorily solve most of these issues via a robust algorithm. Toward the goal of reaching more accurate stereo correspondences, numerous constraints have been identified when studying state-of-the-art disparity estimators:

- Minimal correspondence cost through color (or luminance) similarity: provides a matching cost that must be robust with respect to noise and possible color mismatches,
- Smoothness constraint: neighboring pixels with similar color are favored to have similar disparity. Therefore, disparity discontinuities are encouraged to be located at color discontinuities,

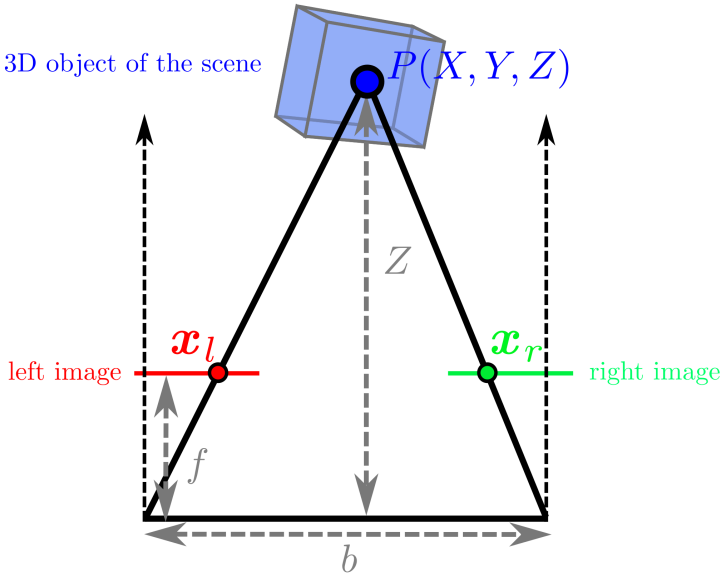


Figure 3.5: Standard stereo setup. The depth Z and the horizontal $1D$ disparity $d = \boldsymbol{x}_l - \boldsymbol{x}_r$ are linked by a relationship which involves both the baseline b and the focal length f, as shown in Eq. 3.1.

- Consistency constraint [EW02]: disparity of a point must be encouraged to have the same module and opposite sign with respect to the disparity of the corresponding point in the other view,
- Visibility constraint [SLKS05]: an occlusion pixel (i.e. a pixel corresponding to a 3D point occluded in the other view) must have no match on the other image and a non-occlusion pixel must have at least one match,
- Ordering constraint [EW02]: two points with a given order along a scanline must have the same order in the other view,
- Uniqueness constraint [ZK00]: enforces a one-to-one mapping between pixels in two images.

The constraints are generally expressed as energy terms and embedded in a global energy for which an iterative global optimization algorithm is used to approximate the minimum. *Graph Cut* [BVZ01] (*GC*) and *Belief Propagation* (*BP*) [SZS03, SLKS05, YWY+09] are the most popular global optimization techniques for such energy minimization. In such global frameworks, the global energy is made of a data term which involves the matching error implied by the extracted disparity maps and a smoothness term which encodes the prior assumption that the world surfaces are piecewise smooth [YWY+09].

On the other hand, disparity estimation can be performed through local window-based algorithms whose goal is to match neighboring pixel values within a window between the left and right views. The choice of the window size is crucial to achieve a smooth and detailed disparity map [ZK00] and this issue has led to algorithms using adaptative window sizes, as done in [KO94]. Added to local matching establishment, bilateral or trilateral filtering has been shown to be an interesting alternative to global optimization techniques, in particular for the stereoscopic *HD* video applications [Bou08, MZK10, RTDC12].

To see an example of implementation, *Appendix A* describes very precisely the disparity estimator proposed in [RTDC12] since it is used in many experiments of this thesis.

3.3 Principles of view synthesis

View synthesis refers to the process of generating new viewpoints of a scene, relying on available color and disparity information. The resulting virtual synthesized views are generated at viewpoints which differ from those captured by the cameras.

In terms of applications, view synthesis is used for $3D\text{-}TV$ which requires to generate new viewpoints from transmitted texture and depth video sequences through *DIBR* algorithms in the context of stereo-to-multiview conversion, as described in Section 3.1.4. The problem of synthesizing new viewpoints is also motivated by the concept of *virtual reality* which consists in giving the possibility to an observer to actively explore an environment [Sch99]. Free navigation inside the scene through *Free-Viewpoint Videos* (FVV) also requires the creation of virtual views in the context of *Free-Viewpoint Television* (FTV).

While in traditional stereo reconstruction systems the desired output is a 3D description of the observed scene, the desired output in the application of view synthesis are realistic-looking images of the scene as it would appear from novel viewpoints and with minimal visual artifacts [Sch96]. Thus, instead of explicitly building a 3D model of the scene and rendering the images through projection onto the recovered 3D surface (which induces point triangulation), view synthesis can be seen as a warping process where depth information is used to warp the existing images.

In this context and as shown in Fig. 3.6, view synthesis requires as input the left and right texture images as well as disparity and occlusion information initially computed between the left and right views, as described in Section 3.2. Let us assume that both left and right texture images have been rectified. From these inputs, the heart of view synthesis is usually divided into two steps which consist in:

1. disparity map projection: interpolating a disparity map at the position of the new synthesized view through projection of the left and/or right disparity map(s) onto the synthesized viewpoint (Section 3.3.1),

2. warping: synthesizing the new view by interpolating color information which are brought back from the left and right views using the projected disparity map computed in step 1 (Section 3.3.2).

Let us study the general concepts of the view synthesis process and more precisely detail these two steps by following the description of the view synthesis method presented in [RTDC12]. This view synthesis method has shown to provide satisfying results in a wide set of stereoscopic data [RTDC12] and is used to generate the synthesized views displayed in this thesis. Section 3.3.3 provides an example of view synthesis with the associated disparity map between the rectified left and right views.

3.3.1 Disparity map projection

Disparity map projection aims at generating a disparity map at the new viewpoint by projecting the left estimated disparity map. Note that sometimes, both left and right disparity maps are

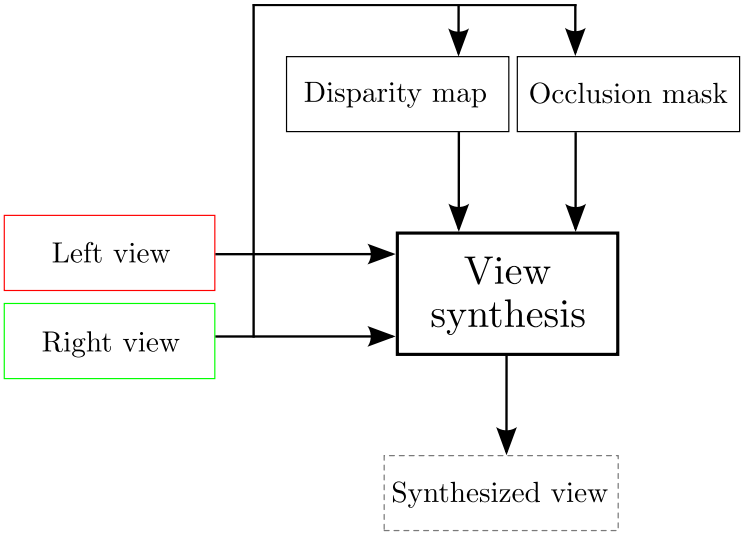


Figure 3.6: View synthesis system: to synthesize a virtual image, view synthesis requires as inputs the left and right texture images as well as disparity and occlusion information.

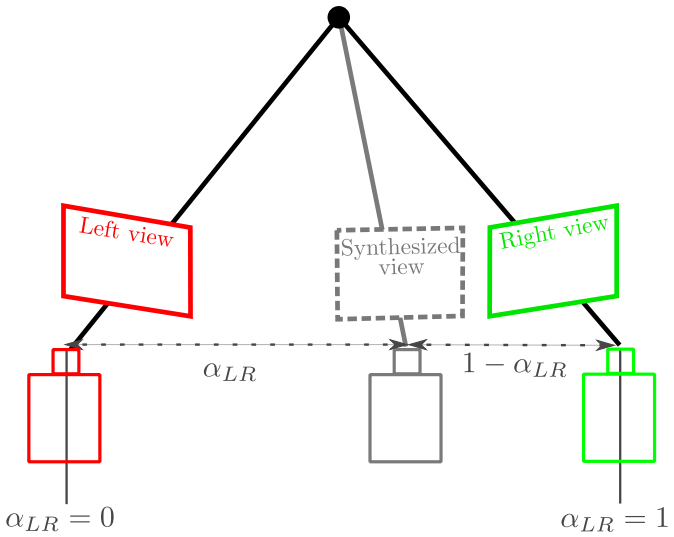


Figure 3.7: Generation of a new viewpoint of the scene relying on color (from left and right views) and disparity information. α_{LR} denotes the ratio between the spatial distance of the virtual viewpoint from the left view and the baseline defined between the left and right views.

used for this task. The position of the view to be synthesized can be defined by a factor α_{LR} (with $0 < \alpha_{LR} \leq 1$) which corresponds to the ratio between the spatial distance of the virtual viewpoint from the left view and the baseline defined between the left and right views (Fig. 3.7). Thus, according Eq. 3.2, this factor α_{LR} is applied to the original left disparity values in order to obtain the projected disparity map.

$$\boldsymbol{x}_i = \boldsymbol{x}_l - d^i(\boldsymbol{x}_l) = \boldsymbol{x}_l - \alpha_{LR}.d^{l/r}(\boldsymbol{x}_l) \tag{3.2}$$

where $\boldsymbol{x}_i$ is the corresponding point in the synthesized image of $\boldsymbol{x}_l$ which belongs to the left image. $d^{l/r}(\boldsymbol{x}_l)$ corresponds to the left/right disparity vector of $\boldsymbol{x}_l$ and $d^i(\boldsymbol{x}_l)$ is the disparity vector with respect to the new synthesized viewpoint.

The left disparity map is scanned from left to right (so that information concerning occluding pixels overwrites information concerning occluded pixels) and each disparity value is shifted at the position defined by this scaled disparity. This position is rounded to the nearest pixel location in the projected disparity map.

Because of disocclusion, some pixels in the view to be synthesized get no disparity value projected to them. These pixels correspond to objects visible in the synthesized view but occluded in the left view. In order to fill these disoccluded areas, during the projection each disparity value is assigned to all pixels on the right side of the previously assigned pixel up to the pixel the current disparity value points at. In other words, this technique aims at filling the occluded areas with background disparity. The same problem can also occur when two neighboring disparity values close to each other (in terms of disparity values) are projected to two non neighboring pixels in the interpolated view. In such situation, the same solution is applied.

3.3.2 Warping

Once the projected disparity map has been created at the virtual viewpoint position, the rendering procedure can start. This translates in computing the color of each pixel of the synthesized view through disparity compensation from either both left and right views or from one of the original images in case of occlusion.

Efficiently managing occlusions is crucial for view synthesis. Indeed, because of occlusions, some objects are visible in the synthesized view but occluded in the left view or in the right view. Fig. 3.8 illustrates such situation and displays, in yellow and blue, pixels of the synthesized view respectively occluded in the left and in the right view. To render the corresponding areas, only one view will be used. During the disparity estimation stage (between left and right views) which precedes view synthesis, occlusion areas in the left and right views have been identified. This occlusion information is projected at the position of the virtual viewpoint in order to identify the regions in the synthesized view which are occluded in the left view or in the right view.

At this stage, we know for each pixel of the virtual view which view (left, right or both views) can be used. To obtain the color for each point of the synthesized view, one takes the color values pointed by the left and/or right disparity vector(s) into the left and/or right view through unidirectional or bidirectional interpolation (Fig. 3.8). For pixels visible in both left and right views (grey and orange pixels in Fig. 3.8), the obtained left and right color samples are combined using the previously described factor α_{LR}. This weighting by α_{LR} allows to rely more on the view which is closer to the virtual viewpoint.

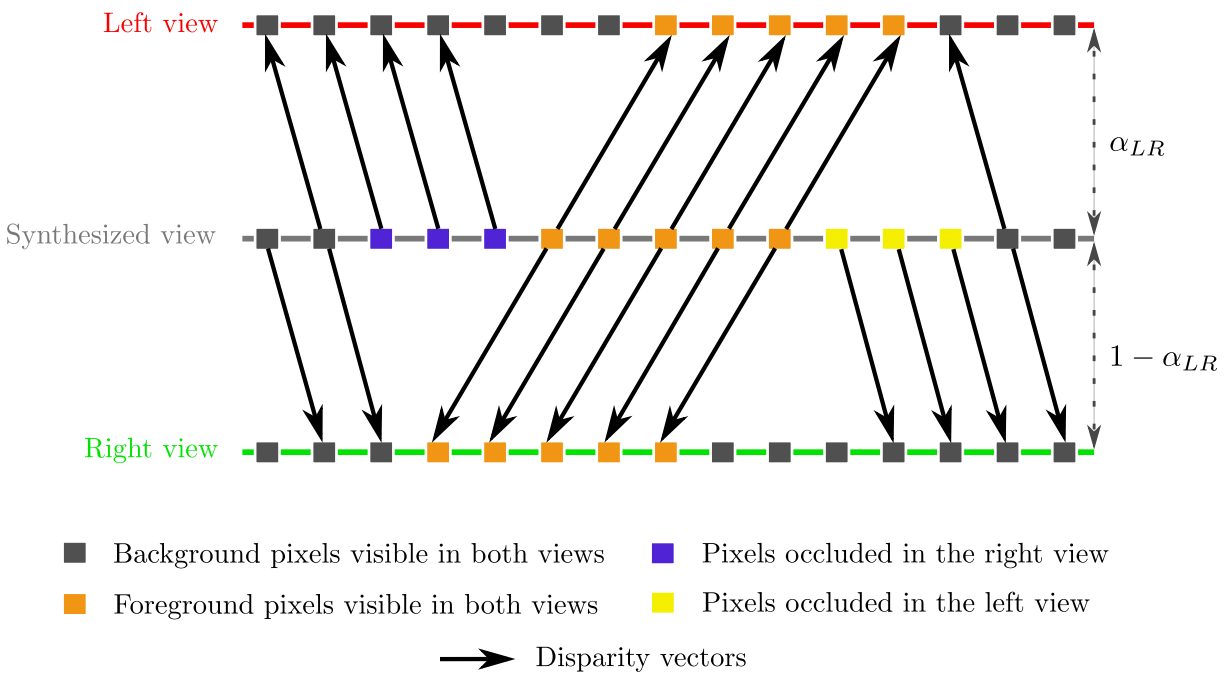


Figure 3.8: Synthesized view rendering. The color value of each point of the synthesized view is obtained using the color values pointed by the left and/or right disparity vector(s) into the left and/or right view(s). To render pixels visible in both views, the obtained left and right color samples are combined using the factor α_{LR}. For areas which are visible in only one view, the interpolation is unidirectional.

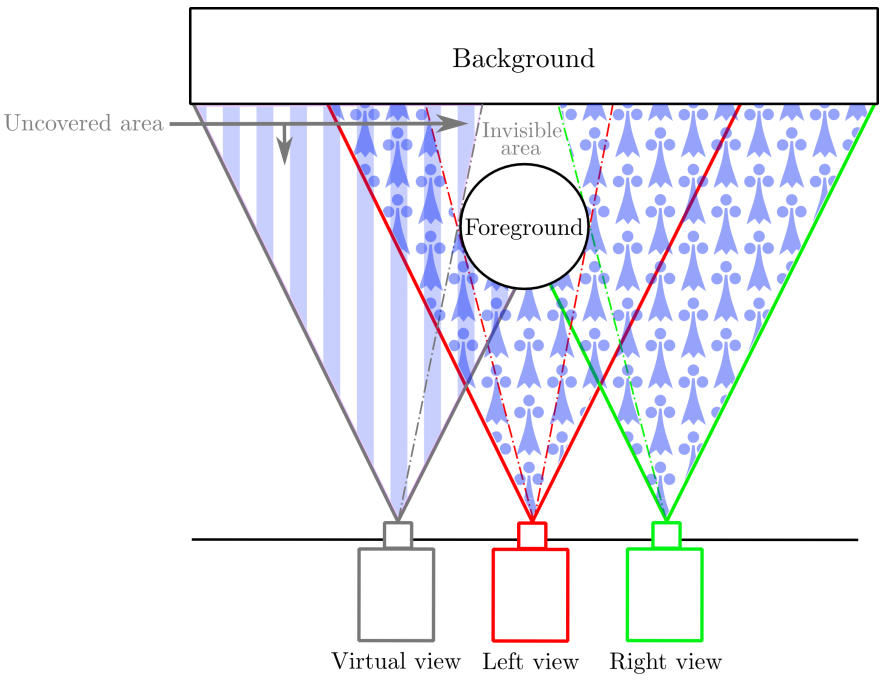


Figure 3.9: Extrapolation of new viewpoints spatially located beyond the viewing range of the original cameras.

Up to now, we saw that view synthesis allows the computation of additional viewpoints between the original left and right views. This interpolation concept can be extended to extrapolate views beyond the viewing range of the original cameras (i.e. α_{LR} outside the interval $]0,1]$). How to render image information which is occluded in both left and right view and which becomes visible in the synthesized view is a critical issue, especially when extrapolating new viewpoints.

Such regions, illustrated in Fig. 3.9 in the context of viewpoint extrapolation, require inpainting and smoothing methods along depth discontinuities [Feh04, MSD+08, NNKD+10, KNND+10] to be accurately filled. Recent works focus on the improvement of view synthesis using temporal information [NNKD+10, KNND+10] in order to obtain temporally consistent synthesized sequences. In particular, how to accurately handle disocclusions to reach spatially and temporally consistent results is still an open issue.

3.3.3 Illustration

To illustrate the description of view synthesis, Fig. 3.10 shows both view interpolation ($\alpha_{LR} = 0.5$) and extrapolation ($\alpha_{LR} = 2$) examples using the disparity estimator and the view interpolation/extrapolation process of [RTDC12]. Fig. 3.10 also displays the left disparity map, i.e. the disparity map which indicates for each pixel of the left view the corresponding location in the right view. The reference views are taken from the *Dali-A* binocular sequence, provided by courtesy of $3DTV\ Solutions^{TM}$.

Despite recent significant progress, view synthesis still requires improvement and existing methods do not always give perfect realistic-looking synthesized views. In particular, view synthesis induces specific distortions which mainly deal with geometric artifacts, as proved by the view synthesis examples in Fig. 3.10 by paying attention on the synthesized thin objects of the sculpture or on the synthesized semi-transparent panels.

The overall description of the different types of view synthesis artifacts as well as their automatic detection will be discussed in the following of this Part I.

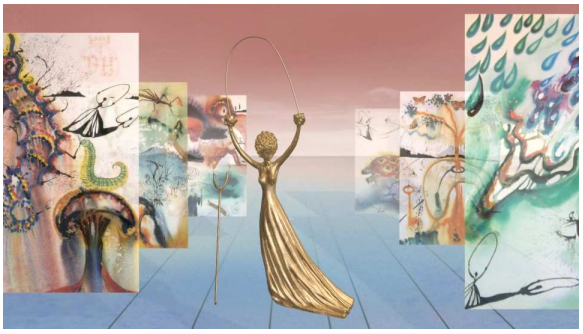
3.4 Conclusion

The history and the general principles of stereoscopic imaging have been provided to allow a good understanding of the topic on which this Part I is dedicated: view synthesis quality assessment. This chapter was the opportunity to present the context on which the view synthesis process takes place. We have seen in particular that view synthesis is involved for $3D\text{-}TV$ that offers a depth impression for home entertainment through stereo-to-multiview conversion including *Depth-Image-Based Rendering* (*DIBR*) algorithms.

Image-based view synthesis generates new viewpoints of a scene relying on textures from original views as well as disparity information. Disparity information is obtained by establishing stereo correspondences between existing views and the accuracy of the resulting estimation is of course crucial for view synthesis.

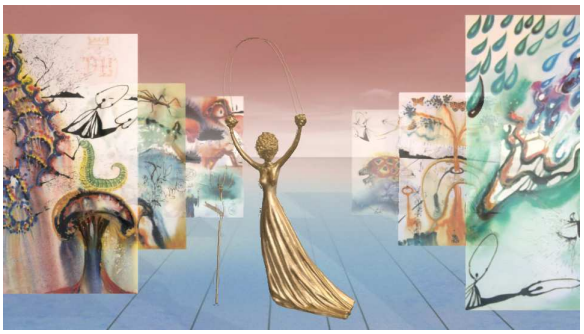
Unfortunately, both disparity estimation and view interpolation/extrapolation processes induce specific distortions on synthesized views. Before studying the way to detect such distortions, we need to identify all the possible different types of view synthesis artifacts. This study is provided in the next chapter, Chapter 4.

(a) Left view



(b) Right view

(c) Disparity map



(d) Interpolated view with $\alpha_{LR} = 0.5$

(e) Extrapolated view with $\alpha_{LR} = 2$

Figure 3.10: Example of view synthesis performed using the disparity estimator and the view interpolation/extrapolation process of [RTDC12]. The reference views are taken from the *Dali-A* binocular sequence, provided by courtesy of $3DTV\ Solutions^{TM}$. Black holes in (e) indicate occlusions.

Chapter 4

View synthesis artifacts: sources and perception

We have seen in Chapter 3 that applications such as *DIBR* mainly rely on the generation of virtual viewpoints based on acquired views as well as disparity information. However, view synthesis may cause structural modifications to synthesized objects and therefore seriously impair the quality of the synthesized views.

In order to know how to assess the quality of synthesized views, we first of all need to identify the main causes of view synthesis artifacts. Toward this goal, we propose in this chapter to present the different situations for which view synthesis may encounter some issues. This study, presented in Section 4.1, is illustrated by various view synthesis examples involving both disparity estimation and view interpolation algorithms presented in [RTDC12]. Moreover, the issue of artifact perception is discussed in Section 4.2. In particular, toward the goal of creating our own view synthesis detection method, we focus on features which can greatly impact the artifact visibility. Finally, Section 4.3 concludes this chapter.

4.1 Sources of distortion

View synthesis involves both disparity estimation and virtual view interpolation/extrapolation. The artifacts occurring in the synthesized views can be due to any error in either two mentioned modules. Unfortunately, they may seriously impair the quality of the synthesized views and therefore impact the viewer comfort.

In practice, view synthesis may encounter some issues in a variety of situations. By studying the commonly observed view synthesis artifacts, we tried to identify the different possible sources of distortion. These sources of distortion are presented and illustrated in what follows.

The view synthesis examples displayed in this Section 4.1 have been obtained using the disparity estimator and the view synthesis process of [RTDC12]. Virtual views have been generated with $\alpha_{LR} = 0.5$ which means that the virtual viewpoint is located exactly in the middle of the left and right viewpoints (Eq. 3.2, Chapter 3).

Some identified sources of distortion are listed and briefly described as follows:

- Color difference between views: when both left and right views involve strong color differences, the disparity matching process may be disturbed. Consequently, this translates in bad stereo correspondences which induce view synthesis issues. Fig. 4.1 illustrates such

situation: the spatial variations of the sky color are not distributed in the same manner in both views due to quantization. Consequently, some parts of the sky in the interpolated image are distorted (see especially below the logo). An incorrect rendering of the homogeneous regions of the interpolated image of Fig. 4.2 is due to the same reason.

- Transparency: As described in Section 3.2, disparity estimators which do not follow a multi-layer approach are not able to compute accurate stereo correspondences when transparency occurs. Disparity maps switch from one layer to another without an efficient multi-layer spatial regularization. This leads to a bad synthesis of semi-transparent objects, as shown for water bubbles in Fig. 4.2, semi-transparent letters in Fig. 4.3, semi-transparent panels in Fig. 4.4 or the gray smoke of Fig. 4.7. Semi-transparent objects generally appear structurally distorted after interpolation/extrapolation and may lose their compactness.

- Thin objects: The size of certain objects is sometimes too small to allow their accurate disparity estimation, especially when the involved disparity estimator implies a hierarchical coarse-to-fine strategy. Due to such depth inaccuracy, thin object are generally degraded in the synthesized views, as illustrated in Fig. 4.4 (golden arc), Fig. 4.5 (foreground flowers) and Fig. 4.6 (thin blue, red and green wires). For this reason, the logos in Fig. 4.1, Fig. 4.2, Fig. 4.5 and Fig. 4.7 also appear very distorted after view synthesis. We notice in particular three types of artifacts for thin object:

  - object shifting (abnormal spatial displacement),
  - object duplication (see the duplication of the golden arc in Fig. 4.4 (c)),
  - structural distortions given that some parts of the object may disappear.

- Periodic objects: As revealed in Section 3.2, disparity estimation may fail to accurately match periodic patterns. The matching ambiguity induces some issues for synthesizing periodic objects, as displayed in Fig. 4.6.

- Variation of illumination: Strong variations of illumination between the left and right views, as shown in Fig. 4.8, are difficult to handle for view synthesis without relying on light scattering models.

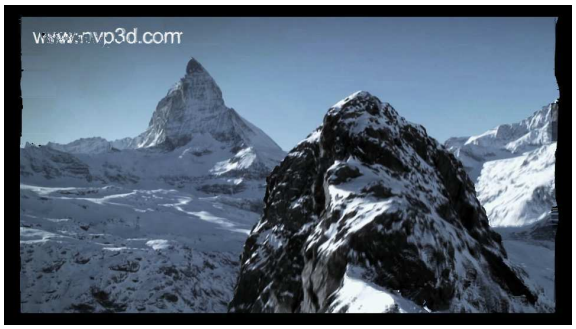
- Depth discontinuities: Areas near object boundaries suffer from echoes in the synthesized views, as illustrated with the boundaries of the foreground mountain in Fig. 4.1, sculpture boundaries in Fig. 4.2 or synthesized black borders (Fig. 4.1, Fig. 4.2, Fig. 4.5 and Fig. 4.7). Also called ghosting artifacts, the halos which are created around objects are due to both:

  - a bad disparity estimation around object boundaries, especially in case of blurred borders (difficult in this case to detect the object border),
  - the transition between unidirectional and bidirectional interpolations (see Fig. 3.8 with corresponding description in Section 3.3, Chapter 3) which may be abrupt during view synthesis.

  Areas located near depth discontinuities also suffer from crumbling. Such distortion appears as floating bits or holes and produces eroded synthesized objets (see in particular around the foreground flowers in Fig. 4.5).

(a) Left view

(b) Right view



(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

Figure 4.1: View synthesis in presence of color differences between left and right views (see in particular the artifacts located in the sky or on depth discontinuities).

(a) Left view

(b) Right view

(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

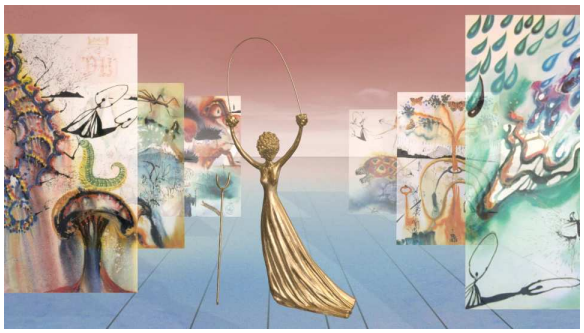
Figure 4.2: View synthesis in presence of color differences and transparency (see in particular the artifacts located on the water bubbles).

(a) Left view

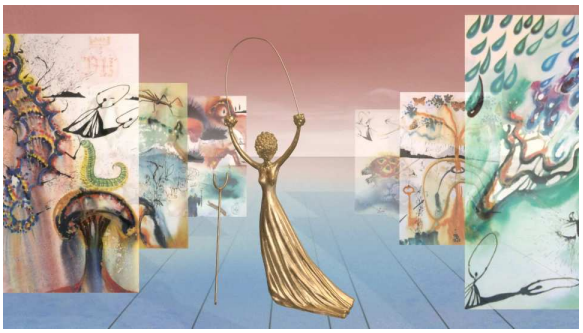
(b) Right view

(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

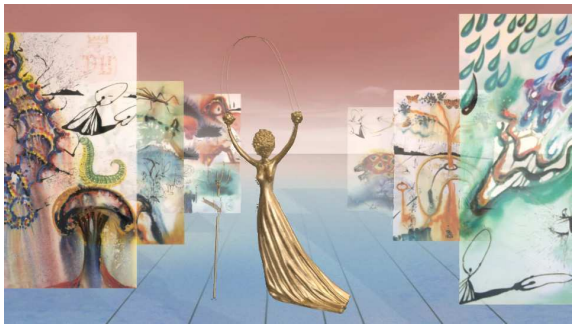
Figure 4.3: View synthesis in presence of transparency (see in particular the artifacts located on the semi-transparent letters).



(a) Left view



(b) Right view



(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

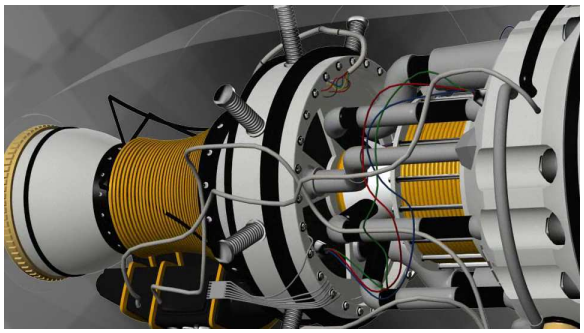
Figure 4.4: View synthesis in presence of thin objects and transparency (see in particular the artifacts of the thin golden arc).

(a) Left view

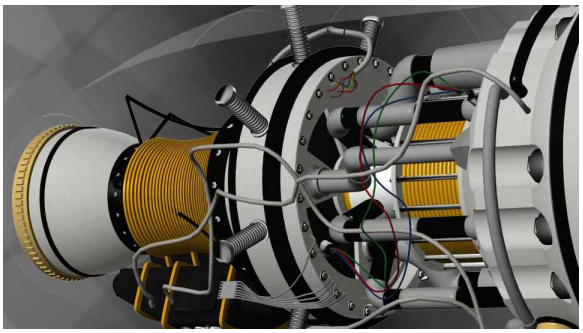
(b) Right view

(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

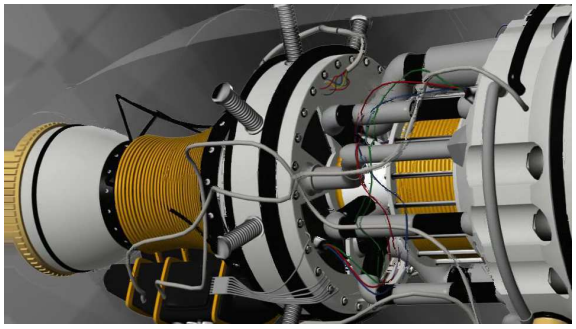
Figure 4.5: View synthesis in presence of thin objects (see in particular the artifacts of the thin foreground flowers).



(a) Left view



(b) Right view



(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

Figure 4.6: View synthesis in presence of thin objects and periodic patterns (see respectively the artifacts on the thin wires and on the left periodic patterns).

(a) Left view

(b) Right view

(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

Figure 4.7: View synthesis in presence of grey smoke (see in particular the seriously impaired left foreground part).

(a) Left view

(b) Right view

(c) Interpolated view with $\alpha_{LR} = 0.5$ using [RTDC12]

Figure 4.8: View synthesis in presence of variations of illumination (see in particular the distorted light source).

This non-exhaustive list proves that sources of distortion are numerous. They can sometimes be combined as in Fig. 4.6 where periodic patterns and thin objects make view synthesis become a difficult task. Such combinations may make synthesized images look very un-natural.

The baseline between left and right views is a crucial point since the larger the baseline is, the harder view synthesis is. In addition, compared to interpolation, the extrapolation process which allows to generate virtual views beyond the viewing range of the original cameras is even more complicated. In this context, the major issue deals with how to accurately fill disoccluded regions to obtain realistic-looking views [NNKD+10, KNND+10].

4.2 Artifact perception

The issue of assessing and quantizing the visual perception of artifacts, and in particular view synthesis artifacts, is an open problem which feeds on research works dedicated to the understanding of the complex mechanisms of the *Human Visual System* (*HVS*). The impact on the viewer comfort depends on many cues and varies from an observer to another.

The spatial environment (i.e. the local neighborhood) of a given view synthesis artifact can greatly impact its perception by an observer. In particular, we focus on three features which appear to be involved in such process:

- the complexity of textures,

- the diversity in terms of gradient orientations,

- the presence of high contrast.

The complexity in terms of textures can modify the artifact visibility [CYX07]. If the distorted entity (object or region) is located in an area characterized by strong textures, its visibility may be strongly reduced due to visual masking. On the contrary, it may be increased within an un-textured area.

In this direction, since view synthesis causes geometric distortions, the diversity in terms of gradient orientations of the spatial environment can also seriously increase or decrease the artifact visibility [WJMG10]. Indeed, view synthesis may distort straight objects into un-structured and curved objects. Therefore, this impacts the object integrity in terms of gradient orientations. Thus, an artifact located within an area of low-diversity in terms of gradient orientations may be highlighted whereas an artifact located within an area with various gradient orientations can be attenuated.

Finally, contrast can have also an important effect in terms of artifact visibility [WBSS04]. Badly synthesizing an object having a color similar to its neighborhood is less annoying than wrongly performing the synthesis of an object involving a strong contrast with its neighborhood.

In this thesis, we only focus on spatial distortions. Nevertheless, the temporal aspects are also crucial and would deserve further investigation since spatial distortions over time can be significantly modified by their temporal changes [NLMLCB09]. In addition, when seen in stereoscopic viewing conditions, the perception of view synthesis artifacts can be different compared to monoscopic viewing conditions. The distortions may seriously impact the stereoscopic perception of the 3D scene and therefore make the scene looks un-realistic.

4.3 Conclusion

Due to wrongly performed disparity estimation or view interpolation/extrapolation, view synthesis artifacts may occur in various situations: color differences between left and right views, transparency, thin objects, periodic patterns, variations of illumination, depth discontinuities...

The visibility of view synthesis artifacts depends on the viewing conditions (monoscopic or stereoscopic) and on the observer himself. Moreover, various features of the artifact neighborhood can have a great impact on the viewer comfort in terms of masking. In particular, features such as complexity of textures, diversity in terms of gradient orientations or presence of high contrast can highlight or attenuate the visibility of the view synthesis artifacts. The question that arises now deals with the automatic detection of these view synthesis artifacts: how to accurately detect the distortions which are due to view synthesis?

Before proposing in Chapter 6 our own method based on this study of the possible sources of distortion and on the previous discussion relative to the artifact perception issue, we aim in Chapter 5 at raising two main points. First, our goal is to study the state-of-the-art in image quality assessment in order to question the reliability of the existing image quality assessment metrics toward an efficient assessment of synthesized views. Second, we aim at extracting from the literature the first attempts dedicated to view synthesis quality assessment.

Chapter 5

State-of-the-art of objective monoscopic and stereoscopic image quality assessment

Objective quality assessment of monoscopic or stereoscopic content is a challenging research task. The goal is to automatically evaluate the perceived image quality and to be able to do it online. The emergence of objective image quality assessment metrics occurred in order to substitute subjective assessment which is time consuming and costly.

The scope of image quality assessment is very wide. It can be used for monitoring quality control systems, for the selection of the best method from multiple image processing systems or for the optimization of algorithms and parameter settings [WLB04]. In practice, objective image quality metrics can be classified into three categories according to the availability of the reference image. If the reference image is completely known, image quality metric is defined as *Full-Reference* (*FR*). If only a set of extracted features is available, a *Reduced-Reference* (*RR*) assessment is performed. At last, if the reference image is not available at all, quality assessment methods are defined as *No-Reference* (*NR*) or *blind* methods.

The most widely used tool for *FR* image quality assessment is the *Mean Square Error* (*MSE*) which is usually involved through *Peak Signal-to-Noise Ratio* (*PSNR*) measurements. *MSE* and *PSNR* are widely used because they are easy in terms of computation and have clear physical meanings [WBSS04]. Unfortunately, *MSE* and *PSNR* do not take into account the way the *HVS* perceives monoscopic or stereoscopic images. This is why great efforts have been spent in order to develop methods for image quality evaluation that take advantage of known characteristics of the *HVS*.

Some papers have more recently investigated the field of stereoscopic content quality assessment. The efforts devoted to the quality assessment of stereoscopic images have been motivated by the widespread of 3D technology (entertainment, medical applications...) described in Chapter 3. Despite these advances, the specific field of view synthesis quality assessment has not been widely investigated and how to detect the different types of view synthesis artifacts mentioned in Chapter 4 is still an open issue.

With view synthesis quality assessment in mind, we start this Chapter 5 by giving an overview of existing objective 2D image quality assessment algorithms (Section 5.1). Then, we analyze in Section 5.2 the state-of-the-art dedicated to stereoscopic content quality assessment. Note that the field of video quality assessment is not studied here. We only focus on monoscopic or

stereoscopic still images. The overviews on objective monoscopic and stereoscopic image quality assessment are followed by the description of early attempts to quantify view synthesis quality in Section 5.3. Finally, Section 5.4 concludes this chapter.

5.1 Objective image quality assessment

Among the plethora of objective quality assessment metrics which can be found in the literature, we focus on the most widely used ones in the state-of-the-art overview given below. In particular, we describe *PSNR* (Section 5.1.1), *SSIM* (Section 5.1.2) and some of its variants (Section 5.1.3) before focusing on image quality assessment based on local orientation features (Section 5.1.4).

5.1.1 The most widely used tool: *PSNR*

The *Peak-Signal-to-Noise Ratio* (*PSNR*) measure assesses the fidelity of a distorted image with respect to a reference image. Its computation relies on the *Mean Squared Error* (*MSE*) which computes the average of the squared intensity differences of both reference and distorted image pixels, as follows:

$$MSE = \frac{1}{M \times N} \sum_i \sum_j [I(i,j) - \tilde{I}(i,j)]^2 \tag{5.1}$$

where I and $\tilde{I}$ respectively deal with the reference and distorted frames. M and N correspond to the dimension of I. The *PSNR* measure is usually expressed in terms of the logarithmic decibel scale and can be computed as follows:

$$PSNR = 20.log_{10}(\frac{max(I)}{\sqrt{MSE}}) \tag{5.2}$$

MSE and *PSNR* are widely used because they are simple in terms of computation, have clear physical meanings and are mathematically easy to deal with for optimization purposes [WBSS04]. Unfortunately, *MSE* and *PSNR* do not take into account the way the *HVS* perceives images, contrary to more sophisticated image quality assessment metrics described in the following.

5.1.2 *SSIM*, a structural similarity-based metric

In [WBSS04], *Wang* et al. focus on *FR* image quality assessment and assume that the *HVS* is highly adapted for extracting structural information from a scene. The created measure, called the *SSIM* index, is based on the assessment of the degradation of structural information. [WBSS04] separates the task of similarity measurement into three comparisons: luminance, contrast and structure. x and y refer respectively to the reference and distorted pixels. The three comparison functions yield a general form of the *SSIM* index:

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma} \tag{5.3}$$

The functions $l(x,y)$, $c(x,y)$ and $s(x,y)$ correspond respectively to the luminance, contrast and structure comparison functions and are detailed in Eq. 5.4, Eq. 5.5 and Eq. 5.6. α, β and γ are positive parameters used to adjust the relative importance of the three comparison functions.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{5.4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{5.5}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{5.6}$$

where μ_x and μ_y are the means of x and y, σ_x and σ_y the standard deviations of x and y and σ_{xy} the correlation coefficient between x and y. These parameters are computed within windows respectively centered around x and y with a *Gaussian* filter. C_1, C_2 and C_3 prevent the situation where the denominator is close to zero. Usually, $\alpha = \beta = \gamma = 1$, $C_1 = (0.01 \times L)^2$ and $C_2 = C_3 = (0.03 \times L)^2$ where L is the dynamic range of the pixel values.

From image pairs (made of both reference and distorted images), the *SSIM* measure yields to quality maps in which a similarity value is given pixel-wise. In [WBSS04], *Wang* et al. also propose to compute an overall quality measure, the *Mean-SSIM* (*M-SSIM*), described in Eq. 5.7, where M and N correspond to the dimension of the images:

$$M - SSIM = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} SSIM(i,j) \tag{5.7}$$

5.1.3 *SSIM* extensions

The previously described *SSIM* metric has motivated numerous alternatives (including [WSB04, CYX07, LW09]) whose aim is to apply *SSIM* using:

- multi-scale images [WSB04],
- gradient reference and gradient distorted images [CYX07],
- macro and micro edge images [LW09].

Other methods such as [LB10, WDM10, WL10] rely in the fact that *SSIM* quality values can be weighted according to an error visibility detection map obtained from:

- gradient information [LB10],
- global phase coherence information [WDM10],
- information content [WL10].

In the following, we briefly describe these alternative metrics.

MS-SSIM [WSB04]

The *Multi-Scale SSIM* (*MS-SSIM*) index has been proposed by *Wang* et al. in [WSB04]. The goal of this method is to deal with image details at different resolutions in order to take into account the variation of the *HVS* sensibility with respect to different frequencies. *MS-SSIM* is an extension of *SSIM* to a multi-scale approach and consists in performing the quality assessment

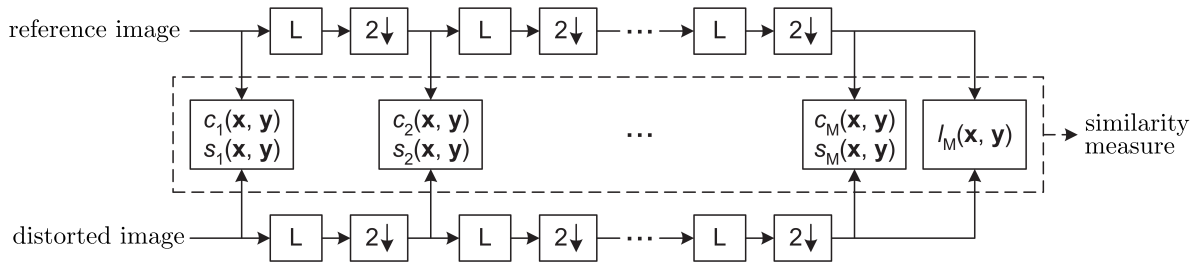


Figure 5.1: *MS-SSIM* measurement system proposed in [WSB04].

of the reference and distorted image patches over multiple scales by iteratively low-pass filtering and down-sampling the signals, as described in Fig. 5.1.

The contrast comparison (Eq. 5.5) and the structure comparison (Eq. 5.6) are performed at each scale. The luminance comparison (Eq. 5.4) is only computed at the highest scale. Finally, the *MS-SSIM* measure is obtained by combining the similarity values computed at different scales:

$$MS-SSIM(x,y) = [l_M(x,y)]^{\alpha_M} . \prod_{j=1}^{M} [c_j(x,y)]^{\beta_j} . [s(x,y)]^{\gamma_j} \tag{5.8}$$

where M corresponds to the number of scales ($M = 5$ in practice). The exponents α_M, β_j and γ_j with $j \in [\![1, \ldots, M]\!]$ aim to adjust the relative importance of the different components and scales.

G-SSIM [CYX07]

According to comparisons performed in [CYX07] between objective and subjective measurements on a dataset dedicated to compression artifacts, the main drawback of *SSIM* is to fail in measuring the quality of images which are blurred due to compression. Moreover, according to *Chen* et al. in [CYX07], the edge and contour information is the most important structural information for the *HVS* to capture the scene. In order to take into account these two findings, a *Gradient-based Structural SIMilarity* (*G-SSIM*) metric has been proposed in [CYX07].

The procedure for the *G-SSIM* quality map computation is similar to the *SSIM* quality map computation procedure except that the inputs are different. With *SSIM*, the reference and distorted images are used as inputs for luminance, contrast and structure comparisons. For *G-SSIM*, the gradient reference and gradient distorted images replace the reference and distorted images as inputs. Two *Sobel* operators are used in order to obtain the gradient maps. In addition, a *Mean-G-SSIM* (*M-G-SSIM*) can be computed in the same manner as Eq. 5.7 to give an overall quality score.

E-SSIM [LW09]

In [LW09], an image quality metric called *Edge Structural SIMilarity* (*E-SSIM*) is introduced. The goal of this method is to divide the structural comparison performed using Eq. 5.6 into the computation of a macro edges similarity function and a micro edges similarity function.

The whole *E-SSIM* scheme is performed as follows. First, a 5×5 average low filtering is performed on the reference and distorted images x and y which gives x^0 and y^0. Then, the gradient images are created from x, y, x^0 and y^0 which gives E_x, E_y, E_x^0 and E_y^0. The third step consists in creating the micro edge information, as described in Eq. 5.9.

$$\begin{cases} E_x^1 = E_x - E_x^0 \\ E_y^1 = E_y - E_y^0 \end{cases} \tag{5.9}$$

The macro edge information is defined as the edge images E_x^0 and E_y^0. The micro and macro edge images are divided into blocks and the following similarity function is computed first between E_x^1 and E_y^1 and then between E_x^0 and E_y^0:

$$e(x,y) = \frac{\sigma_{xy}}{\sigma_x \sigma y} \tag{5.10}$$

where σ_x and σ_y are the standard deviations of x and y and σ_{xy} the correlation coefficient between x and y. The two obtained results $e_0(x,y)$ (between E_x^0 and E_y^0) and $e_1(x,y)$ (between E_x^1 and E_y^1) are then combined:

$$e(x,y) = \alpha.e_0(x,y) + (1-\alpha).e_1(x,y) \tag{5.11}$$

The final step yields a general form of the *E-SSIM* index:

$$E - SSIM(x,y) = [l(x,y)]^\alpha.[c(x,y)]^\beta.[e(x,y)]^\gamma \tag{5.12}$$

In addition, a global quality score, the *Mean-E-SSIM* (*E-G-SSIM*), can be computed in the same manner as Eq. 5.7.

Content-partitioned *SSIM* [LB10]

The method proposed in [LB10] is based on the fact that different image regions have different importance in terms of vision perception and that neither *SSIM* nor *MS-SSIM* take into account explicitly the visual importance of image features. This is why *Li* and *Bovik* suggest in [LB10] to weight the *SSIM* quality values by a perceptual importance function within a theoretic approach using information content-weighted pooling. More precisely, the pixels of the reference and distorted images are parsed into four categories: changed edges, preserved edges, texture and smooth regions.

Let TH_1 and TH_2 be two thresholds computed as in Eq. 5.13 where g_{max} is the maximum gradient magnitude value computed over the reference image.

$$\begin{cases} TH_1 = 0.12.g_{max} \\ TH_1 = 0.06.g_{max} \end{cases} \tag{5.13}$$

Let $P_r(i,j)$ and $P_d(i,j)$ be respectively the gradient of the reference image and the gradient of the distorted image at coordinate (i,j). The coarse segmentation proposed in [LB10] is carried out according to the following rules:

- (i,j) is considered as a preserved edge pixel if $P_r(i,j) > TH_1$ and $P_d(i,j) > TH_1$,
- (i,j) is considered as a changed edge pixel if $P_r(i,j) > TH_1$ and $P_d(i,j) \leq TH_1$ or if $P_d(i,j) > TH_1$ and $P_r(i,j) \leq TH_1$,

- (i, j) is considered as belonging to a smooth region if $P_r(i, j) < TH_2$ and $P_d(i, j) > TH_1$,
- otherwise (i, j) is considered as belonging to a textured area but is not considered as an edge pixel.

Then, the input *SSIM* quality map is weighted according to the coarse segmentation done previously. The idea is to allocate greater weight to the scores at edge regions than on smooth and textured regions. Moreover, smooth regions are more important than textured regions because some distortions can be masked by the presence of textures while they would be visible in untextured areas.

***SSIM* and *Global Phase Coherence* (*GPC*)** [WDM10]

[WL10] proposes an image quality assessment metric based on *SSIM* and *Global Phase Coherence* (*GPC*), where *GPC* is applied on *SSIM* as a weighted factor. The aim is to take into account both the structural information in the spatial domain and the phase characteristics in the frequency domain. Estimated using a *Discrete Fourier Transform* (*DFT*) applied on both reference and distorted images, *GPC* is used as a measure of image sharpness and informs about the quality of image edges in smooth transition regions. It has been demonstrated that *GPC* decreases when the level of blur, aliasing or ringing is increasing.

Information content weighting [WL10]

As in [LB10, WDM10], [WL10] proposes to perform a weighting of local quality measurement values obtained with *SSIM*. It combines information content weighting through mutual information computations, distortion and perception channel modelling with multiscale *SSIM* measures.

5.1.4 Image quality assessment based on local orientation features

Inspired by the idea that *HVS* is sensitive to image local orientation features, the *Histograms of Oriented Gradients* (*HOG*) based metric has been created in [WJMG10]. Not based on *SSIM*, the quality assessment performed in [WJMG10] consists in computing the difference between gradient orientations of image patches which allows a robust distortion detection with a low computational complexity. More precisely, this method aims at describing the distribution of the pixel orientations within blocks in the reference and distorted images and to compare the obtained results block-wise. For each block, two *Histograms of Oriented Gradients* (*HOG*) are built (one for the reference image, one for the distorted image) and then compared with an appropriate distance in order to obtain an error value for the corresponding block in the distortion map.

5.2 Objective stereoscopic content quality assessment

We have seen in Section 5.1 that numerous metrics have been proposed in the literature to assess the quality of 2D images. More recently, quality assessment of stereoscopic content has also known increasing attention. The resulting metrics, dedicated to stereoscopic content quality assessment, do not directly target view synthesis artifacts. Nevertheless, it is useful to discuss state-of-the-art methods in this field in order to understand how the first attempts toward an efficient stereoscopic content assessment have been designed from 2D metrics.

[CLCM07] proposes to design a quality metric tailored to stereoscopic images by applying 2D quality metrics to stereo content. Thus, reference left and right views are compared with distorted left and right views in the context of image compression. *SSIM*, among other metrics, are separately computed on each eye and three fusion methods are investigated:

- average: the scores computed separately from the left and right views are averaged,
- mean-eye: only the main eye of each observer is taken into account,
- visual acuity: the scores obtained for the left and right image are weighted by using the visual acuity of the observers.

These two last approaches have led to no performance improvement with respect to the classic average approach, which appears as the most effective fusion.

In [BLCCC08], the previously described quality metric for the stereoscopic images quality assessment task (average approach of [CLCM07]) has been improved by taking into account the disparity map between stereoscopic image pairs. Compared to [CLCM07], [BLCCC08] aims at incorporating information related to the 3D nature in the whole process by fusing the disparity map with the scores coming from the metrics employed in both left and right views in [CLCM07]. A similar reasoning is followed in [YXPW10] which presents a study on the integration of disparity information into quality assessment.

Both papers [BLCCC08, YXPW10] relate a significant performance enhancement when disparity information is added to original images as inputs of the stereoscopic image quality assessment stage. This proves the two following assumptions:

- 2D image quality metrics cannot be directly adapted in evaluating stereoscopic image quality,
- disparity has a significant impact on stereoscopic image quality assessment.

In [BGE+], *Boev* et al. suggest an alternative method which involves two components within a stereo quality metric based on *SSIM*: a monoscopic quality component and a stereoscopic quality component. The goal is to dissociate monoscopic artifacts (blur, noise, blockiness and other structural changes) from stereoscopic artifacts (changes in the disparity estimation, color distortions, vertical disparity...).

Another interesting approach to assess the quality of stereoscopic content is presented in [JMFK10]: the *Perceptual Quality Metric* (*PQM*). *PQM* renders color and depth images to left and right views using the *DIBR* algorithm presented in [Feh04]. An average of the 2D left and right objective scores is finally carried out in order to obtain the final quality measurement. Distortions between both reference and distorted color views are quantified using luminance differences as well as contrast distortions.

The objective quality assessment method presented in [YHZ+09] evaluates stereo images from the perspective of image quality and stereo sense. Image quality assessment based on *PSNR* applied on both left and right views is combined with *Stereo Sense Assessment* (*SSA*) which involves absolute disparity images.

Finally, the authors of [SYZ09] have created a stereo image quality assessment method called *DSSIM*. They have tried to focus on psychological stereoscopy factors (related to depth cues like

shading, motion, texture...) and physiological stereoscopy factors (stereoscopic sense given by binocular disparities, as in [YHZ+09]). The full system includes a disparity estimation step and a segmentation step using *K-means*. The *SSIM* metric is then computed separately by regions which are finally weighted by physiological cues.

To conclude this state-of-the-art overview, we notice that stereoscopic content quality assessment using 2D objective metrics have been widely used. An extension of this principle is to add information from disparity maps to the whole measurement system since disparity information has an important impact in terms of visual quality. Several interesting further ideas have been found in the literature: distinction between monoscopic and stereoscopic artifacts [BGE+], introduction of *SSA* [YHZ+09] as well as considerations about segmentation [SYZ09].

5.3 Early attempts to quantify view synthesis quality

All the state-of-the-art methods presented in Section 5.1 and Section 5.2 quantify the quality of monoscopic or stereoscopic content. The common point between all these techniques is to tend forward subjective results. However, most of them focus on compression artifacts and, on the other hand, there is a need of techniques able to deal with artifacts due to view synthesis (Chapter 4) since such distortions can seriously impair the viewer comfort.

The reliability of traditional 2D objective quality assessment metrics to assess synthesized views has been studied in [BKP+11]. Although these methods perform quite well when detecting compression artifacts such as blocking effects, blurring artifacts, ringing, edge distortion or false contouring, *Bosc* et al. have showed that they are not appropriate for assessing synthesized views. The assessment of seven *DIBR* algorithms through both objective measurements (coming from many well known metrics such as *PSNR*, *SSIM*, *MS-SSIM*...) and subjective ratings performed in [BKP+11] proves that traditional objective metrics are not suited for assessing virtual synthesized views because the correlation coefficients between subjective and objective scores are relatively low. This conclusion implies the need for developing new methods dedicated to view synthesis quality assessment in order to improve the ability to automatically predict human experience in the context of *3D-TV* for instance.

In what follows, we give a brief overview of first attempts exclusively dedicated to view synthesis quality assessment.

[EWDS+10] proposes an unequal weighting based quality evaluation approach which can be applied on 2D quality metric such as *PSNR* and *SSIM*. The weighting computation step is based on the fact that the *HVS* is more affected by the distortions happening on the front part of the scene. Therefore, [EWDS+10] suggests to weight the quality measurements by weighting coefficients $WC(x, y)$ defined between 0 and 1 and assigned to each individual pixel (x, y) in order to both:

- highlight artifacts in the front part of the scene,
- attenuate artifacts in the back part of the scene.

Thus, the weighting coefficients $WC(x, y)$ are near 0 when the local scene depth is above an upper depth threshold Z_f and near 1 when the local scene depth is under a lower depth threshold Z_n. Between Z_n and Z_f, it varies linearly, inversely proportional to the local scene depth range. Such approach is performed using as inputs the original view with associated depth map as well as the synthesized view.

In the context of *FVV* production, [SKH08] addresses the problem of objectively measuring the view synthesis quality by assessing synthesized views in terms of structural registration error. The structural registration error is quantified by measuring the most apparent geometric errors with respect to the underlying geometry of the scene. The proposed metric is hybrid since it can be used to compare a synthesized view to a ground-truth image (*FR* assessment) or to assess the appearance of the virtual view without ground-truth image (*NR* assessment).

Another interesting *NR* assessment method is the approach developed in [BLL+10] which deals with the detection of ghosting artifacts (i.e. shadow-like artifact). According to the authors, ghosting artifacts are perceived as the most distracting artifacts in image interpolation. In addition, they assume that ghosting can be detected locally and that ghosting artifacts are only visible in areas containing strong object edges.

The view synthesis artifacts detection proposed by *Devernay* and *Peon* in [DRP10] is carried out by creating a confidence map over the whole interpolated image. The aim is to assign a weight to each pixel to indicate how certain we are about its correctness given that view synthesis artifacts are characterized by small high-frequency defects which appear as crumbling artifacts (Section 4.1, Chapter 4).

The general idea of the artifact detection step of [DRP10] is to find similar pixel intensities, gradients and *Laplacian* in the synthesized view compared to the left and right views, but at different locations. Thus, three confidence maps are created based on intensity, gradient and *Laplacian* values. For each map, the following procedure occurs:

1. get the (intensity, gradient or *Laplacian*) value of the pixel on the synthesized view,

2. fetch the value of the corresponding pixels in the left and right views,

3. compute the absolute value of the difference with each value (values from left and right views are blended).

Experiments have shown that *Laplacian* only detects artifact on contour and not in the inner region which can be detected with a better accuracy with the intensity or gradient differences. Thus, the detection procedure proposed in [DRP10] finally aims at detecting artifacts as:

- areas surrounded by high *Laplacian* differences,

- areas inside which the intensity or gradient difference with the original images is high.

Finally, [SAB11] introduces the concept of ideal depth maps which are compared to estimated depth maps through three distortion measures including a spatial inconsistency measure. The three distortion measures are combined to finally constitute a vision-based quality measure for 3D *DIBR*-based content.

Despite these interesting early attempts dedicated to view synthesis artifacts, the field of view synthesis quality assessment does not seem to have been widely investigated. New methods are required in order to increase the correlation between objective measurements and subjective results and to finally perform a more efficient detection of view synthesis artifacts.

5.4 Conclusion

This chapter was the opportunity to present image quality assessment techniques dedicated to monoscopic and stereoscopic content. Many existing image quality metrics are based on *SSIM* [WBSS04], assuming that the *HVS* is highly adapted for extracting structural information and therefore considering that the degradation of structural information greatly impacts the perceived quality.

In terms of stereoscopic content quality assessment, existing methods are very often based on 2D objective image quality assessment metrics. Moreover, the introduction of disparity information within the proposed measurement systems has been widely accepted.

However, we have seen that a majority of the created methods is dedicated to compression. Attempts to quantify view synthesis quality and in particular to focus directly on view synthesis artifacts (through traditional metrics or dedicated methods) are not numerous. Moreover, the correlation between objective measurements and subjective results is still relatively low.

Starting from these findings, we claim that new methods are required in order to more efficiently detect artifacts introduced by view synthesis. Toward this goal, we propose our own method in Chapter 6.

Chapter 6

Objective view synthesis quality assessment: *VSQA* metric

The state-of-the-art dedicated to image quality assessment (Chapter 5) has revealed that early attempts to quantify the view synthesis quality are not numerous and that the few existing methods do not work satisfactorily. Moreover, the multiple existing metrics dedicated to compression artifacts have shown their inefficiency to assess the different types of view synthesis artifacts detailed Chapter 4. Starting from these two findings, we propose to study how to efficiently detect the distortions introduced by view synthesis while relying on any existing metric.

Toward this goal, our motivation is to create a new full-reference objective image quality assessment metric dedicated to view synthesis quality assessment. Moreover, we propose to follow an approach which can be based on any existing image quality assessment metric, including metrics which are usually used to detect compression artifacts. The idea here is to convert a given metric toward an accurate detection of all the possible view synthesis artifacts.

The proposed metric dedicated to the automatic detection of view synthesis artifacts is called the *VSQA* metric. *VSQA* stands for *View Synthesis Quality Assessment*. Section 6.1 describes exactly how this *VSQA* metric is defined and focuses more precisely on how the complexity in terms of textures, the diversity of gradient orientations and the presence of high contrast, previously described in Section 4.2 for their visual masking properties, can be involved to accurately assess synthesized views and therefore to automatically predict the perceived synthesized image quality.

In order to illustrate our study, we have chosen to build our metric as an extension of the *Structural SIMilarity image index* (*SSIM*) metric [WBSS04]. Therefore, Section 6.2 presents experimental results with the resulting *SSIM*-based *VSQA* metric including visual results and comparisons with subjective data obtained through the *IRCCyN/IVC DIBR* images database [BPLC+11a, BPLC+11b].

On the scale of the whole sequence, Section 6.3 studies how the proposed quality assessment approach can be involved to both describe how the quality evolves temporally and to identify the more distorted frames within a synthesized sequence. Finally, Section 6.4 gives conclusions and some cues toward further work.

This study led to one publication published in an international conference: [CRM12]. This publication introduces and experimentally assess the *VSQA* metric. Note that in [RTDC12], published in the same international conference, we used the *VSQA* approach to compare different disparity estimation and view synthesis algorithms.

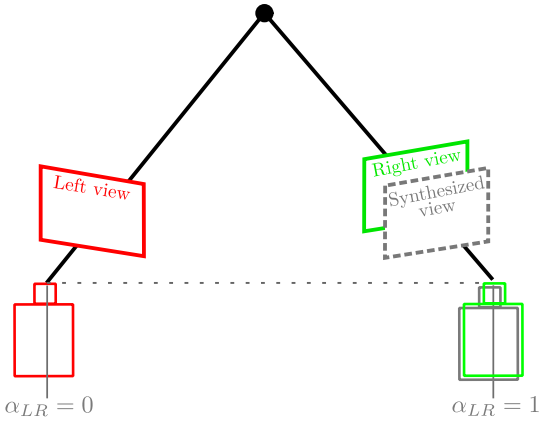


Figure 6.1: The comparison between the right view and the view synthesized at the right position gives enough information to assess the quality of the whole view synthesis.

6.1 The proposed *VSQA* metric

The context of our study deals with view synthesis from binocular input. To create this new metric dedicated to the detection of view synthesis artifacts, we have considered that the comparison between the right (resp. left) view and the view synthesized at the right (resp. left) position gives enough information in order to assess the quality of view synthesis for any virtual viewpoint. These two views, displayed in Fig. 6.1 are referred to as reference and synthesized views. Note that our approach does not assess the quality of areas which are not visible in the left (resp. right) view if the right (resp. left) view is used as reference view.

In the following, we describe the general principle of the *VSQA* quality metric in Section 6.1.1. Then, we focus more precisely on the three main features involved within the metric: the complexity in terms of textures (Section 6.1.2), the diversity of gradient orientations (Section 6.1.3) and the presence of high contrast (Section 6.1.4). Finally, our spatial pooling method is explained in Section 6.1.5.

6.1.1 General principle

We present a new quality metric dedicated to view synthesis: the *View Synthesis Quality Assessment* (*VSQA*) metric. It can be based on any existing quality metric. As described in Fig. 6.2, this *VSQA* metric takes as inputs the right view and the synthesized view (obtained at the position of the right view) and it gives as outputs:

- a distortion map which indicates the exact location of the view synthesis artifacts,
- a global score which informs about the overall quality of the synthesized view.

VSQA is based on the fact that the perception of view synthesis artifacts is strongly linked with the features of the spatial environment and notably with the complexity in terms of textures, the diversity of gradient orientations and the presence of high contrast. Therefore, as described in Fig. 6.3, our approach consists in weighting the distortion values obtained from the chosen existing image quality assessment metric *dist* (*PSNR* or *SSIM* for instance). Weights are computed based on three weighting maps directly created from three visibility maps.

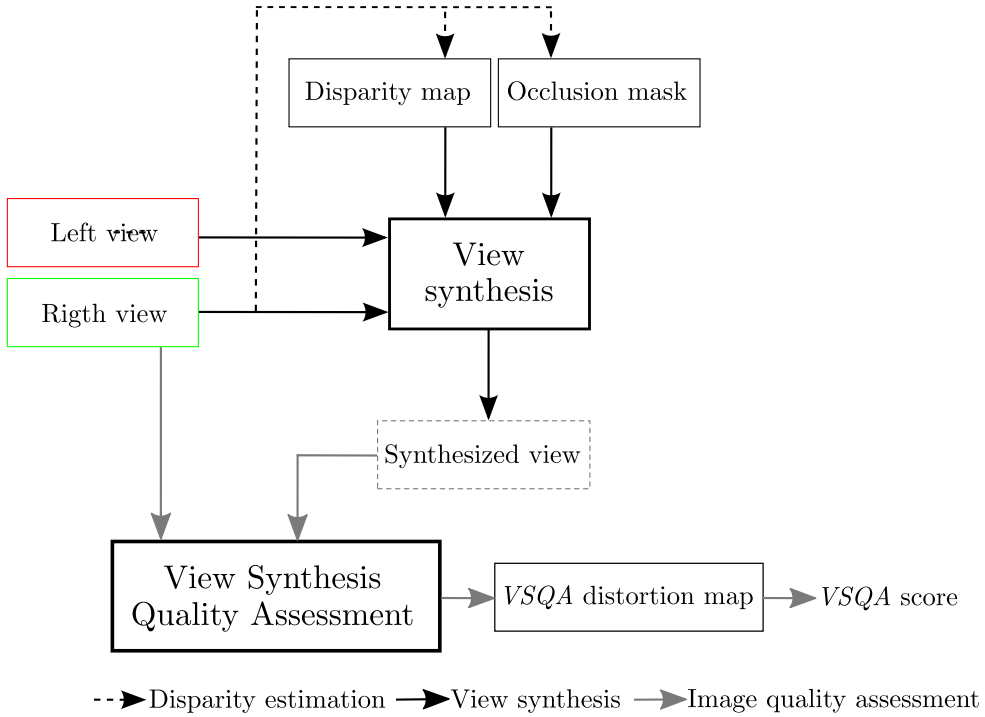


Figure 6.2: View synthesis at the right position and quality assessment by comparing the reference and the synthesized views by the proposed metric, the *View Synthesis Quality Assessment* (*VSQA*) metric.

The three weighting maps characterize the complexity in terms of textures, the diversity of gradient orientations and the presence of high contrast. These maps are referred to as texture-based weighting map, orientation-based weighting map and contrast-based weighting map in the following. The computation of these three weighting maps consists in extracting image features within large windows in order to take into account possible masking effects due to the environment.

The weighting of the distortion values obtained with the chosen existing image quality assessment metric leads to a final distortion map called *dist*-based *VSQA* distortion map (where *dist* corresponds to the chosen metric). The *dist*-based *VSQA* distortion map is computed as follows:

$$VSQA(i,j) = dist(i,j).[W_t(i,j)]^{\delta}.[W_o(i,j)]^{\epsilon}.[W_c(i,j)]^{\xi} \tag{6.1}$$

where $dist(i,j)$ denotes the distortion value given by the chosen existing image quality assessment metric for a given pixel (i,j). W_t, W_o and W_c correspond respectively to the texture, orientation and contrast-based weighting maps. δ, ϵ, ξ are real positive parameters used to adjust the relative importance of the three weighting maps.

The three following sub-sections focus on how to create the three weighting maps and describe the effect of each one on distortion values.

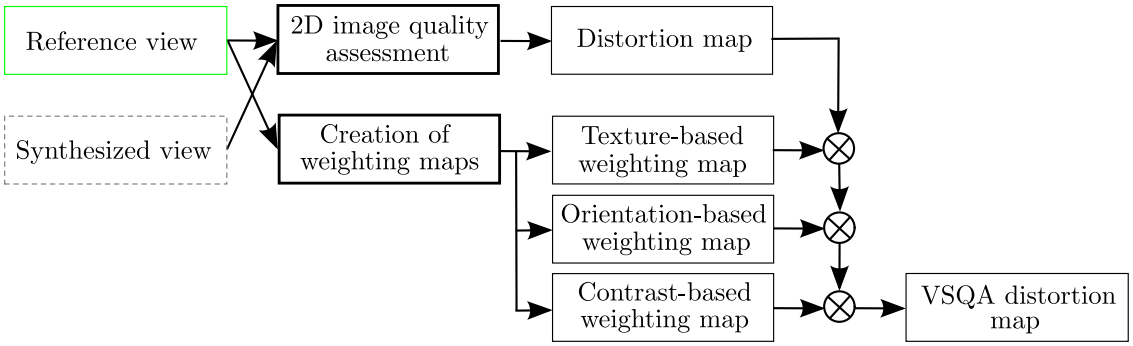


Figure 6.3: Diagram of the proposed quality measurement system with $\delta = \epsilon = \xi = 1$.

6.1.2 Texture-based weighting map

The texture-based weighting map describes the complexity of the neighborhood in terms of textures. Actually, the perception of artifacts surrounded with high gradient pixels is attenuated due to masking effect. Inversely, artifacts within untextured areas are more visible. Thus, when an artifact is located in a low texture complexity area, the weighting aims at increasing the corresponding distortion value (or to decrease it if we consider a similarity metric, i.e. the smaller the distortion value is, the stronger the artifact is). On the contrary, if an artifact is located in a high texture complexity area, the weighting aims to decrease the corresponding distortion value (or to increase it if we consider a similarity metric).

More precisely, the texture-based weighting map is computed as follows. First of all, the first step is the computation of the texture-based visibility map V_t from which the texture-based weighting map W_t will be computed. The method consists in deriving an image gradient from the reference view with a *Sobel* operator. Then, each pixel value of V_t is computed as the mean of the gradient magnitude values over a large surrounding window, as described in Eq. 6.2. Note that a *Gaussian* weighting function normalized to unit sum with standard deviation σ_t is involved in the computation.

$$V_t(i,j) = \frac{1}{N_t^2} \sum_{l=i-\left\lfloor \frac{N_t}{2} \right\rfloor}^{i+\left\lfloor \frac{N_t}{2} \right\rfloor} \sum_{k=j-\left\lfloor \frac{N_t}{2} \right\rfloor}^{j+\left\lfloor \frac{N_t}{2} \right\rfloor} w_{l,k}.grad(l,k) \tag{6.2}$$

where V_t denotes the texture-based visibility map. $grad(.)$ is the gradient magnitude map and $w_{l,k}$ is the *Gaussian* weighting function. $N_t \times N_t$ corresponds to the window size.

Finally, the texture-based weighting map W_t is created by linear rescaling of the texture-based visibility map V_t between 0 and 2 as described in Eq. 6.3. Let $min(V_t)$ and $max(V_t)$ be the minimum and maximum values over the visibility map V_t. To compute W_t, $min(V_t)$ and $max(V_t)$ are mapped from 0 to 2 and intermediate values $V_t(i,j)$ are then mapped to the line defined by these two values.

This rescaling procedure is performed as shown in Fig. 6.4 (a) and Eq. 6.3 if the chosen image quality assessment metric on which *VSQA* relies is a similarity metric. In this case, when the complexity in terms of textures is important (resp. low), i.e. when the artifact visibility is attenuated (resp. highlighted), V_t increases (resp. decreases) which make W_t increasing (resp.

decreasing) too. Consequently, for pixels (i,j) belonging to highly-textured regions, $W_t(i,j)$ is usually greater than 1 which allows, after weighting, to increase the corresponding similarity values $dist(i,j)$. On the contrary, for pixels (i,j) belonging to untextured regions, $W_t(i,j)$ is usually lower than 1 which allows, after weighting, to decrease the corresponding similarity values $dist(i,j)$.

$$\begin{aligned} W_t(i,j) &= \frac{2}{max(V_t) - min(V_t)}.V_t(i,j) - \frac{2.min(V_t)}{max(V_t) - min(V_t)} \\ &= \frac{2.\left(V_t(i,j) - min(V_t)\right)}{max(V_t) - min(V_t)} \end{aligned} \tag{6.3}$$

6.1.3 Orientation-based weighting map

Secondly, the orientation-based weighting map aims at quantifying the diversity of gradient orientations of the neighborhood for each pixel. Only the high textured areas are taken into account (thresholding over the gradient values). The use of such map has been inspired by the fact that the *HVS* is quite sensitive to image local orientation features [WJMG10]. The main idea is to take into account the masking effect due to large diversity of gradient orientations. Indeed, we can easily think that a large diversity of gradient orientations can decrease the artifacts visibility. Inversely, if all neighbors of a considered pixel have the same gradient orientation, an artifact located in that point attracts the gaze. This is especially true given that view synthesis artifacts are geometric artifacts and that objects are subject to structural modifications.

Thus, when an artifact is located in a low gradient orientations diversity area, the weighting aims at increasing the corresponding distortion value (or to decrease it if we consider a similarity metric). On the contrary, in a high gradient orientations diversity area, the weighting aims at decreasing the corresponding value (or to increase it if we consider a similarity metric).

Before computing the orientation-based weighting map W_o, the gradient orientation map θ at pixel level is computed from the reference view with Eq. 6.4.

$$\theta(i,j) = tan^{-1}\left(\frac{f_y(i,j)}{f_x(i,j)}\right) + \frac{\pi}{2} \tag{6.4}$$

where $f_x(i,j)$ and $f_y(i,j)$ correspond respectively to the horizontal and vertical gradients for a given pixel (i,j). Note that all the obtained values are defined modulo π.

Let us consider the orientation-based visibility map V_o from which the orientation-based weighting map W_o will be computed. The idea is to compute each pixel value of the orientation-based visibility map $V_o(i,j)$ as the standard deviation in terms of gradient orientations with respect to a reference gradient orientation value $\theta_q(i,j)$. This reference value corresponds to the orientation $\theta \in [0,\pi]$ which minimizes the standard deviation in terms of gradient orientations over the window centered around (i,j):

$$\theta_q(i,j) = min_\theta \left[\sum_{l=i-\lfloor \frac{N_o}{2} \rfloor}^{i+\lfloor \frac{N_o}{2} \rfloor} \sum_{k=j-\lfloor \frac{N_o}{2} \rfloor}^{j+\lfloor \frac{N_o}{2} \rfloor} min\left[(\theta(l,k)-\theta)^2, (\theta(l,k)+\pi-\theta)^2\right] \right] \tag{6.5}$$

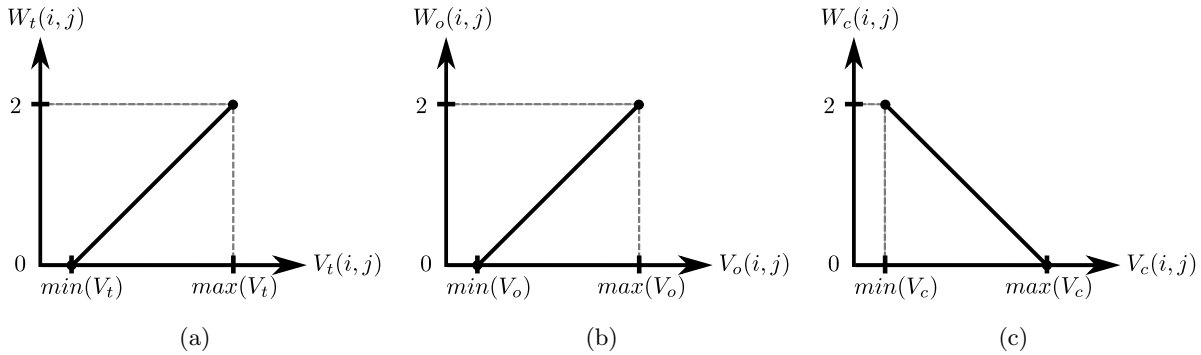


Figure 6.4: Illustration of the rescaling between visibility maps and weighting maps for: a) texture-based (Section 6.1.2), b) orientation-based (Section 6.1.3) and c) contrast-based maps (Section 6.1.4).

where $\theta(l,k)$ is the gradient orientation values for a given position (l,k). $N_o \times N_o$ corresponds to the window size. The complete formula of V_o is given in Eq. 6.6. As previously, a *Gaussian* weighting function normalized to unit sum with standard deviation σ_o is considered.

$$V_o(i,j) = \sum_{l=i-\lfloor \frac{N_o}{2} \rfloor}^{i+\lfloor \frac{N_o}{2} \rfloor} \sum_{k=j-\lfloor \frac{N_o}{2} \rfloor}^{j+\lfloor \frac{N_o}{2} \rfloor} w_{l,k}.min\left[(\theta(l,k) - \theta_q(i,j))^2, (\theta(l,k) + \pi - \theta_q(i,j))^2\right] \tag{6.6}$$

where V_o denotes the orientation-based visibility map and $w_{l,k}$ is the *Gaussian* weighting function.

Finally, the orientation-based weighting map is created by rescaling the orientation-based visibility map between 0 and 2 as described in Fig. 6.4 (b) and Eq. 6.7 for similarity metrics.

$$W_o(i,j) = \frac{2.\left(V_o(i,j) - min(V_o)\right)}{max(V_o) - min(V_o)} \tag{6.7}$$

6.1.4 Contrast-based weighting map

The contrast-based weighting map highlights luminance differences between pixels and their neighborhood. The goal is to give a better importance to artifacts located on pixels whose luminance value differs significantly with the luminance values of the neighborhood. Thus, in this type of area, the weighting aims at increasing the corresponding distortion value (or to decrease it if we consider a similarity metric). On the contrary, if the luminance difference is not significant, the weighting aims at decreasing the corresponding distortion value (or to increase it if we consider a similarity metric).

Let us consider the contrast-based visibility map V_c from which the contrast-based weighting map W_c will be computed. This third visibility map can be created by first computing the luminance image *Lum* from the reference view. Then, as described in Eq. 6.8, we compute each pixel value of the contrast-based visibility map $V_c(i,j)$ as the mean of the absolute differences between the luminance values over a large surrounding window and the luminance value of the current pixel (center of the window). Note that a *Gaussian* weighting function normalized to unit sum is also involved here with standard deviation σ_c.

$$V_c(i,j) = \frac{1}{N_c^2} \sum_{l=i-\lfloor \frac{N_c}{2} \rfloor}^{i+\lfloor \frac{N_c}{2} \rfloor} \sum_{k=j-\lfloor \frac{N_c}{2} \rfloor}^{j+\lfloor \frac{N_c}{2} \rfloor} w_{l,k}.\,|Lum(l,k) - Lum(i,j)| \tag{6.8}$$

where V_c denotes the contrast-based visibility map, Lum the luminance of the reference image, $w_{l,k}$ the *Gaussian* weighting function. $N_c \times N_c$ corresponds to the window size.

Finally, the contrast-based weighting map is created by rescaling the contrast-based visibility map between 0 and 2 as described in Fig. 6.4 (c) and Eq. 6.9 for similarity metrics.

$$\begin{aligned} W_c(i,j) &= \frac{2}{min(V_c) - max(V_c)}.V_c(i,j) - \frac{2.min(V_c)}{min(V_c) - max(V_c)} \\ &= \frac{2.\,(V_c(i,j) - min(V_c))}{min(V_c) - max(V_c)} \end{aligned} \tag{6.9}$$

Contrary to both texture and orientation-based weighting maps (Fig. 6.4 (a,b)), the mapping function between the contrast-based visibility and the contrast-based weighting maps for similarity metrics is a monotonically decreasing function (Fig. 6.4 (c)). Indeed, the artifact visibility is highlighted when the contrast is important whereas it is attenuated when the textures or the diversity of gradient orientations are important.

Once the three weighting maps have been computed respectively following Eq. 6.3, Eq. 6.7 and Eq. 6.9, they are used to weight the distortion map obtained with the chosen existing quality metric (Fig. 6.3). This weighting procedure is not applied to high quality pixels since *VSQA* mainly aims at reorganizing the prioritization of erroneous pixels. At the end, we finally obtain the *VSQA* distortion map which assigns a view synthesis distortion value for each pixel of the synthesized view.

6.1.5 Spatial pooling method: *VSQA* score

In order to obtain a global score for the whole synthesized frame, we suggest to create a thresholded *VSQA* distortion map and to count the number of remaining pixels after thresholding. The obtained result gives what we call the *VSQA* score. Following such procedure, we quantify the global image quality by referring to the most annoying artifacts.

Let min_{VSQA} and max_{VSQA} be respectively the minimum and maximum distortion values within the *VSQA* distortion map. If the used objective image quality assessment metric is a similarity metric, the threshold th can be fixed as described in Eq. 6.10.

$$th = min_{VSQA} + p.\frac{max_{VSQA} - min_{VSQA}}{100} \tag{6.10}$$

where p is a positive parameter (with $p \in [0, 100]$) and min_{VSQA} and max_{VSQA} the minimum and maximum *VSQA* distortion values respectively.

Pixels whose *VSQA* distortion value is under the threshold th are activated in the thresholded *VSQA* distortion map and taken into account in the computation of the *VSQA* score.

6.2 Experimental evaluation of the *VSQA* metric

The goal of this section is to validate the *View Synthesis Quality Assessment* (*VSQA*) metric proposed in Section 6.1. In particular, we aim at assessing the *SSIM*-based *VSQA* metric, i.e. the *VSQA* approach based on distortion maps obtained with the *SSIM* metric [WBSS04].

The experiments have been conducted as follows. First, we use two examples to detail how *VSQA* acts on different image areas and to demonstrate the effectiveness of *VSQA* in evaluating the perceptible synthesized image quality compared to *SSIM* (Section 6.2.1). Second, we carry out an overall performance comparison thanks to subjective data provided by the *IRCCyN/IVC DIBR* database [BPLC+11a, BPLC+11b] (Section 6.2.2).

For these experiments, the proposed quality assessment approach has been performed with the following parameters.

First, the *SSIM* metric has been applied with its classical parameters: $\alpha = \beta = \gamma = 1$ in Eq. 5.3 (Chapter 5). In Eq. 5.4, Eq. 5.5, Eq. 5.6, $C_1 = (0.01 \times L)^2$ and $C_2 = C_3 = (0.03 \times L)^2$ where $L = 255$ is the dynamic range of pixel values. Finally, in these equations, σ_x, σ_y, μ_x, μ_y and μ_{xy} have been computed within 11×11 windows using a centered *Gaussian* weighting function of standard deviation 1.5.

Second, concerning *VSQA*, the three weighting maps (texture-based, orientation-based and contrast-based) have been involved with the same importance: $\delta = \epsilon = \xi = 1$ in Eq. 6.1. The texture-based weighting map and the contrast-based weighting map are computed with 31×31 windows ($N_t = N_c = 31$) with a standard deviation of $\sigma_t = \sigma_c = 17$ whereas the orientation-based weighting map uses 17×17 windows ($N_o = 17$) with a standard deviation of $\sigma_o = 9$. Finally, the parameter p involved for the computation of the *VSQA* score equals to 19 (Eq. 6.10).

6.2.1 Visual results

In this section, we visually compare *SSIM* (described in Chapter 5) and *SSIM*-based *VSQA* (described in Section 6.1). *SSIM* and *SSIM*-based *VSQA* have been applied on reference and synthesized views taken from two sequences: 1) *Dali-A*, Fig. 6.5 (courtesy of $3DTV\ Solutions^{TM}$), 2) *Lovebird1*, Fig. 6.8. In both cases, the synthesized views have been created by the disparity-compensated view synthesis proposed in [RTDC12].

Experiments with the *Dali-A* sequence

The reference (right) view and the synthesized view (obtained at the position of the right view) for the *Dali-A* sequence is shown in Fig. 6.5. One can perceive a sculpture of *Dali* in the middle of the scene as well as six semi-transparent panels on both sides. Fig. 6.5 focuses on two view synthesis artifacts:

- the distortion of the golden arc: Fig. 6.5 (*c,d*),
- the fluctuations of the vertical edge of two of the semi-transparent panels: Fig. 6.5 (*e,f*).

For these two images, Fig. 6.6 shows the *SSIM* distortion map (Fig. 6.6 *(a)*) and the *SSIM*-based *VSQA* distortion map (Fig. 6.6 *(f)*). For these two distortion maps, the darker the pixel, the larger the distortion.

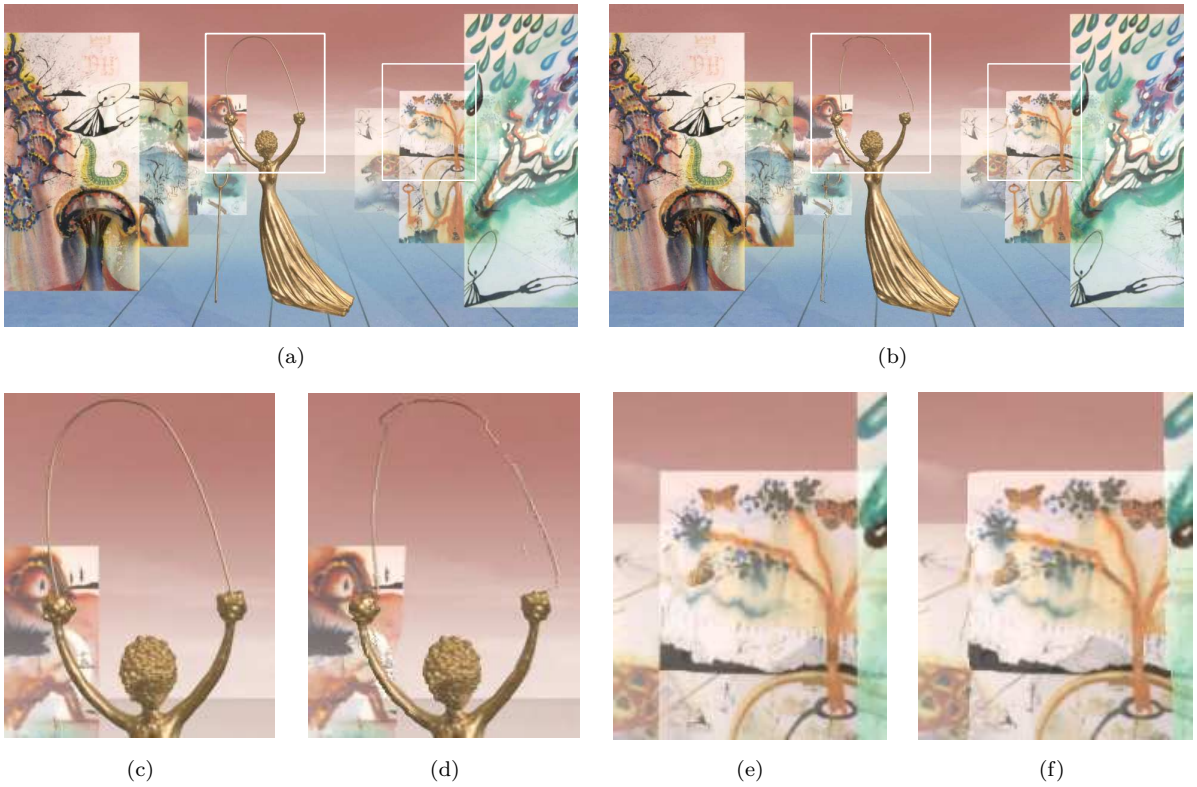


Figure 6.5: Reference (a) and synthesized (b) views with zooms on two distorted areas: (c,e) for the reference view and (d,f) for the synthesized view. The reference view is taken from the *Dali-A* binocular sequence provided by courtesy of $3DTV\ Solutions^{TM}$. The synthesized view has been created through the disparity-compensated view synthesis proposed in [RTDC12].

Fig. 6.6 also gives the texture (Fig. 6.6 *(b)*), orientation (Fig. 6.6 *(c)*) and contrast-based weighting maps (Fig. 6.6 *(e)*). They are consistent respectively with Eq. 6.3 (Section 6.1.2), Eq. 6.7 (Section 6.1.3) and Eq. 6.9 (Section 6.1.4). These weighting maps are the maps used if the selected existing image quality assessment metric is a similarity metric, which is the case here with *SSIM*. In particular, dark areas indicate high-visibility areas for distortions. Thus, for these areas, the weighting decreases *SSIM* values which accentuates the corresponding artifacts. Conversely, bright areas indicate low-visibility areas. In this case, the weighting increases *SSIM* values in order to attenuate the corresponding artifacts.

Moreover, note that only highly-textured areas are taken into account for the weighting performed by the orientation-based map (indeed, no gradient means no gradient orientation). Thus, Fig. 6.6 *(d)* shows a mask, associated to the orientation-based weighting map, which indicates in white the considered highly-textured areas, i.e. pixels which are taken into account during the orientation-based weighting.

As detailed in Section 6.1.4, the *VSQA* procedure is not applied to high quality pixels. For this, we set a threshold of 0.75 on *SSIM* values (for which the maximum similarity value is 1). Therefore, for high quality pixels (i.e. *SSIM* value above 0.75), the *VSQA* distortion values are taken as equal to *SSIM* values. For pixels with a lower quality (i.e. *SSIM* value below 0.75), the *VSQA* procedure is applied as described in Section 6.1.

The artifact displayed in Fig. 6.5 (c,d) (distortion of the thin golden arc) is due to the inaccuracy of disparity estimation for thin objects. After projection, thin objects are completely unstructured. Here, artifacts are all the more noticeable since the background around the golden arc is textureless and different of the arc in terms of luminance values (i.e. strong contrast). The second artifact on which Fig. 6.5 (e,f) focuses (distortion of the vertical edges of the panels) is essentially due to transparency which is not efficiently handled during disparity estimation. Vertical edges are not perfectly straight. However, this second artifact is not as noticeable as the first artifact because it is in an high-textured area containing various orientation features.

Let us now consider thresholded versions of the *SSIM* and the *SSIM*-based *VSQA* distortion maps as shown respectively in Fig. 6.7 (a) and Fig. 6.7 (b). The same number of erroneous pixels has been kept (2300 pixels) in order to allow a comparison between the two approaches. Concerning these two binary maps, white pixels indicate the pixels considered as erroneous after thresholding. As explained in Section 6.1, *VSQA* reorganizes the prioritization of the pixels in terms of quality. Indeed, we notice that the thresholded *SSIM* and *SSIM*-based *VSQA* distortions maps do not highlight the same erroneous pixels.

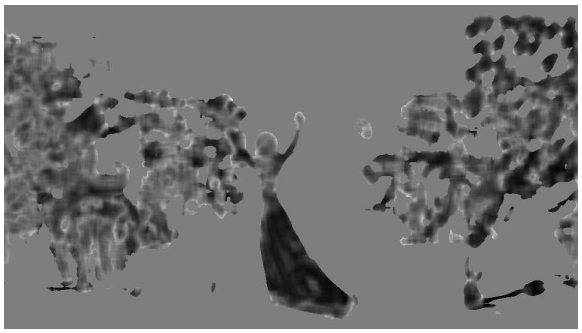
In order to be consistent with the human perception of artifacts, an image quality metric should focus on the first artifact (zooms Fig. 6.5 (c,d) reproduced in Fig. 6.7 (c) and (d)) and give smaller distortion values for the second distorted area (zooms Fig. 6.5 (e,f) reproduced in Fig. 6.7 (g) and (h)). If we study the zooms on the thresholded *SSIM* distortion map Fig. 6.7 (e) and (f) and on the thresholded *SSIM*-based *VSQA* distortion map Fig. 6.7 (i) and (j), it appears that *SSIM* is not consistent with the previous considerations contrary to *VSQA* which:

- highlights visually important distortions,
- attenuates insignificant distortions.

More precisely, Fig. 6.7 shows that some important erroneous areas are detected in the thresholding *SSIM*-based *VSQA* distortion map and not in the *SSIM* one (parts of the golden

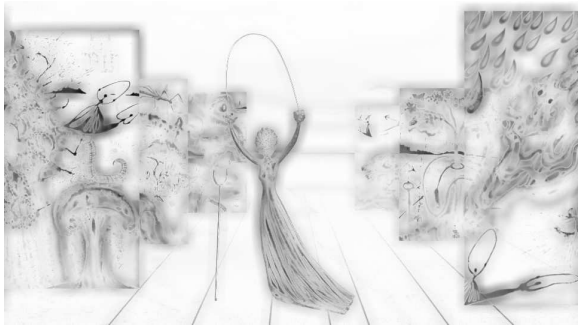
(a) *SSIM* distortion map

(b) Texture-based weighting map W_t



(c) Orientation-based weighting map W_o

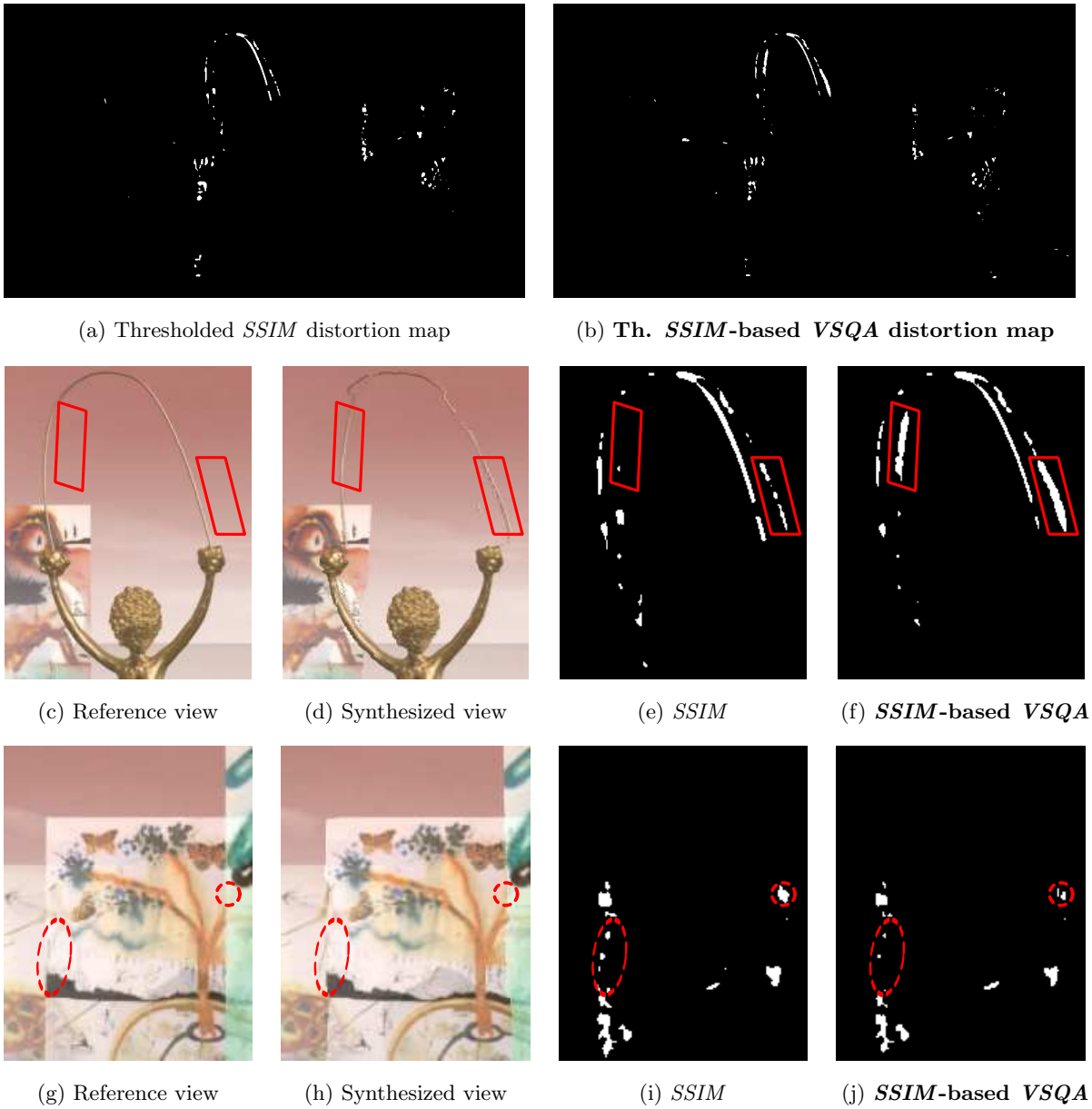
(d) Binary mask associated to W_o



(e) Contrast-based weighting map W_c

(f) ***SSIM*-based *VSQA* distortion map**

Figure 6.6: Experimental evaluation of the *VSQA* metric using the binocular *Dali-A* sequence (Fig. 6.5). For the distortion maps in (a) and (f): the darker the pixel, the larger the distortion. Concerning the weighting maps in (b), (c) and (e): dark and bright pixels indicate respectively high-visibility and low-visibility areas for distortions. White pixels in (d) correspond to pixels taken into account during the orientation-based weighting (black pixels not considered).



(a) Thresholded *SSIM* distortion map

(b) **Th. *SSIM*-based *VSQA* distortion map**

(c) Reference view

(d) Synthesized view

(e) *SSIM*

(f) ***SSIM*-based *VSQA***

(g) Reference view

(h) Synthesized view

(i) *SSIM*

(j) ***SSIM*-based *VSQA***

Figure 6.7: Comparison between *SSIM* and *SSIM-based VSQA* (*Dali-A* sequence). For both distortions maps (*SSIM* in (*a*,*e*,*i*) and *SSIM*-based *VSQA* in (*b*,*f*,*j*)), the 2300 most erroneous pixels have been kept. White pixels indicate the pixels considered as erroneous after thresholding. Results from (*c*) to (*j*) focus on two distorted areas. Red shapes with solid lines focuses on artifacts highlighted by *VSQA*. Conversely, red shapes with dotted lines indicated artifacts attenuated by *VSQA*.

Figure 6.8: Reference (a) and synthesized (b) views with zooms on two distorted areas: (c,e) for the reference view and (d,f) for the synthesized view. The reference view is taken from the *Lovebird*-1 binocular sequence. The synthesized view has been created by the disparity-compensated view synthesis proposed in [RTDC12].

arc for instance). Moreover, compared to *SSIM*, *SSIM*-based *VSQA* highlights more artifacts located within untextured areas where the contrast is high (golden arc). Conversely, it can be observed that masking in terms of texture complexity and gradient orientations diversity has been taken into account during the weighting procedure. Indeed, some artifacts located on the transparent panels have been attenuated by *SSIM*-based *VSQA* compared to *SSIM*.

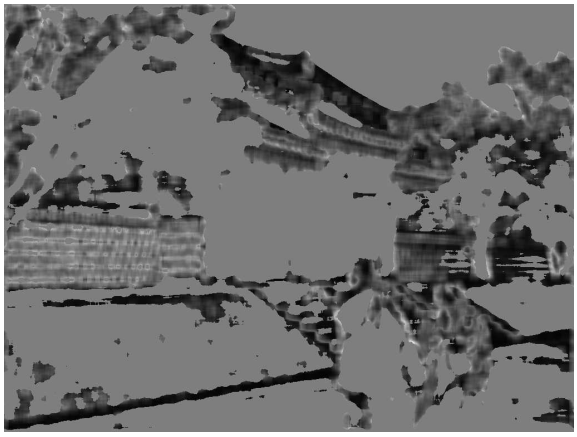
Experiments with the *Lovebird1* sequence

In the same spirit as previously, we now aim at visually comparing *SSIM* and *SSIM*-based *VSQA* using a reference (right) frame coming from the *Lovebird-1* sequence and a synthesized frame obtained at the right position through the disparity-compensated view synthesis of [RTDC12]. These two frames are displayed in Fig. 6.8 and represent a couple walking in front of a tree-lined temple. Fig. 6.8 focuses especially on two regions:

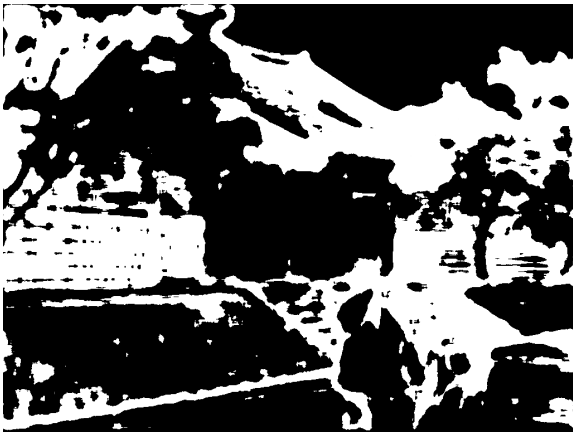
- the area around the couple with quite disturbing artifacts (see especially the artifacts on the column pillar, the temple steps and the wall behind the woman),
- the area centered on the trees at the left side of the temple for which the high complexity in terms of textures and the diversity in terms of orientations tend to attenuate the artifacts visibility.

(a) *SSIM* distortion map

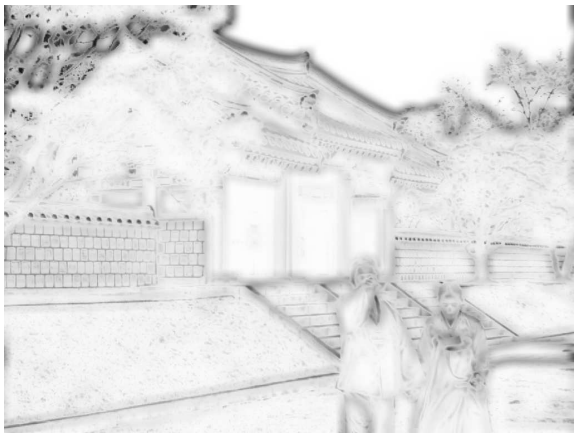
(b) Texture-based weighting map W_t



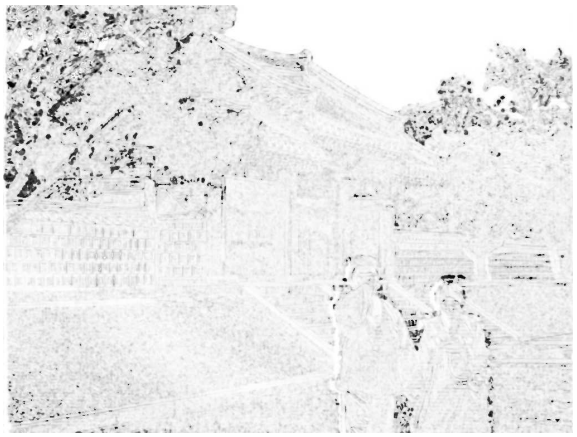
(c) Orientation-based weighting map W_o



(d) Binary mask associated to W_o

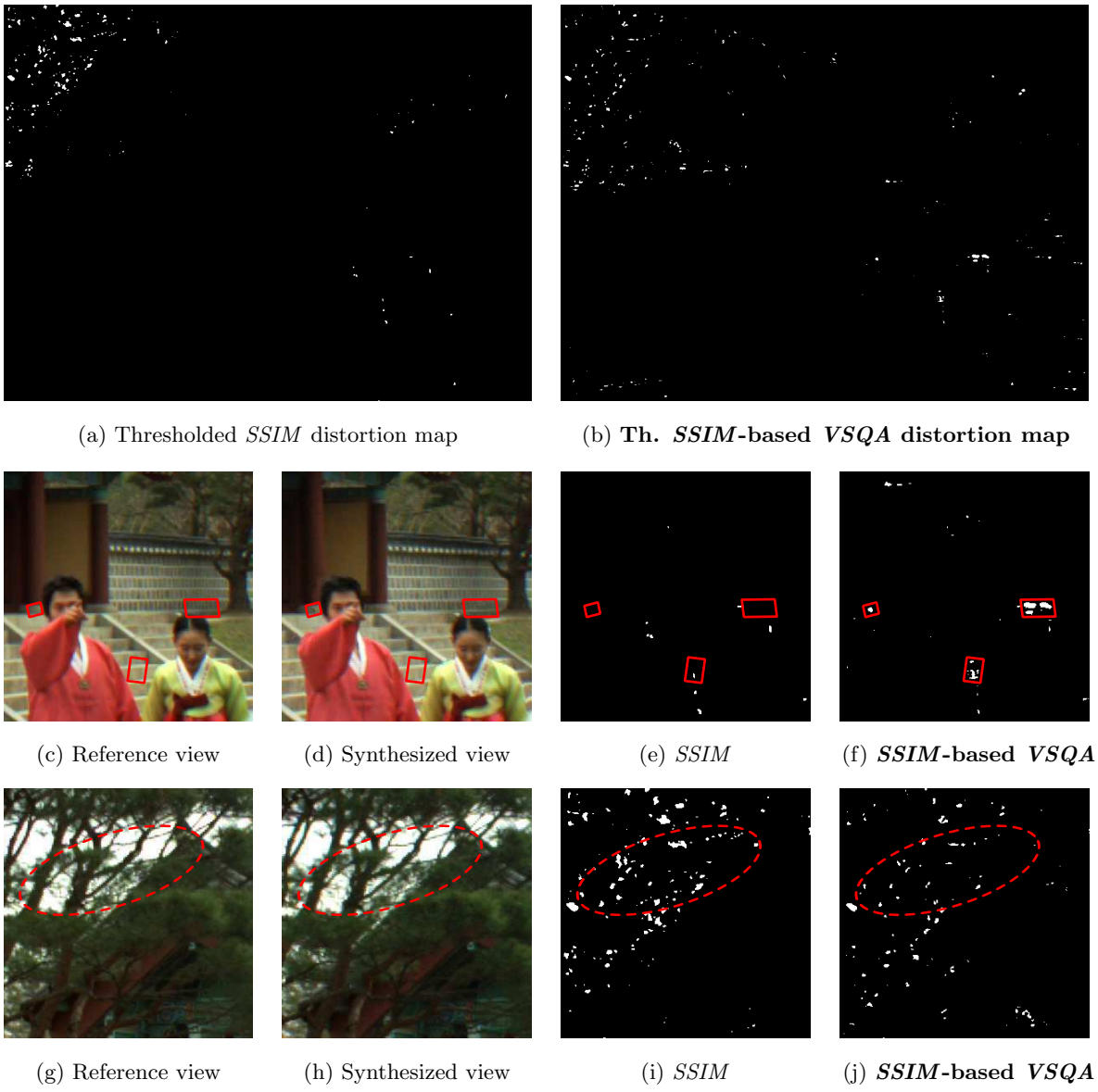


(e) Contrast-based weighting map W_c



(f) ***SSIM*-based *VSQA* distortion map**

Figure 6.9: Experimental evaluation of the *VSQA* metric using the binocular *Lovebird-1* sequence (Fig. 6.8). For the distortion maps in (a) and (f): the darker the pixel, the larger the distortion. Concerning the weighting maps in (b), (c) and (e): dark and bright pixels indicate respectively high-visibility and low-visibility areas for distortions. White pixels in (d) correspond to pixels taken into account during the orientation-based weighting (black pixels not considered).



(a) Thresholded *SSIM* distortion map

(b) **Th. *SSIM*-based *VSQA* distortion map**

(c) Reference view

(d) Synthesized view

(e) *SSIM*

(f) ***SSIM*-based *VSQA***

(g) Reference view

(h) Synthesized view

(i) *SSIM*

(j) ***SSIM*-based *VSQA***

Figure 6.10: Comparison between *SSIM* and *SSIM-based VSQA* (*Lovebird-1* sequence). For both distortions maps (*SSIM* in (*a*,*e*,*i*) and *SSIM*-based *VSQA* in (*b*,*f*,*j*)), the 2300 most erroneous pixels have been kept. White pixels indicate the pixels considered as erroneous after thresholding. Results from (*c*) to (*j*) focus on two distorted areas. Red shapes with solid lines focuses on artifacts highlighted by *VSQA*. Conversely, red shapes with dotted lines indicated artifacts attenuated by *VSQA*.

Similarly to Fig. 6.6, Fig. 6.9 shows the *SSIM* distortion map (Fig. 6.9 *(a)*), the *SSIM*-based *VSQA* distortion map (Fig. 6.9 *(f)*) as well as the texture-based weighting map (Fig. 6.9 *(b)*), the orientation-based weighting map (Fig. 6.9 *(c)*) with associated binary mask (Fig. 6.9 *(d)*) and the contrast-based weighting map (Fig. 6.9 *(e)*). Using the information provided by these three weighting maps, *VSQA* reorganizes the prioritization of the pixels quality initially done by *SSIM*.

As previously, to objectively compare *SSIM* and *SSIM*-based *VSQA*, Fig. 6.10 displays thresholded versions of the *SSIM* (Fig. 6.10 (a)) and the *SSIM*-based *VSQA* (Fig. 6.10 (b)) distortion maps for which the most erroneous pixels (2300 pixels) have been highlighted. Fig. 6.10 also focuses on the two previously described areas and allows to draw the same conclusions as the one obtained with the *Dali-A* sequence. In particular, compared to *SSIM*, *SSIM*-based *VSQA* is able to:

- give more importance to visually important artifacts such as the ones surrounded by red rectangular shapes (column pillar, steps, wall),
- attenuate less disturbing artifacts as indicated by the red ellipsoidal shape.

According to these subjective results, *SSIM*-based *VSQA* shows a better ability to assess the perceptible synthesized image quality compared to *SSIM*. The next section, Section 6.2.2 provides objective comparison results between *SSIM*-based *VSQA* and *SSIM* as well as other state-of-the-art quality metrics.

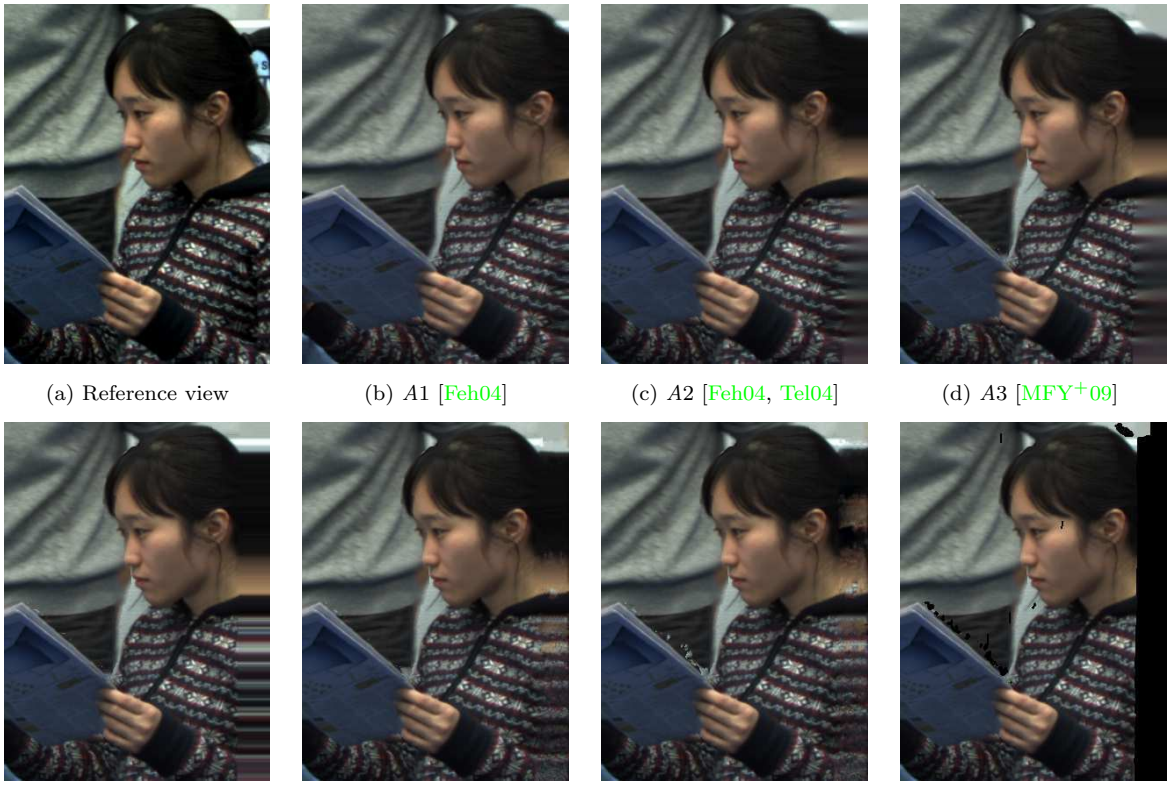
6.2.2 Performance comparison between VSQA and existing quality metrics

In this section, we aim at objectively comparing *SSIM*-based *VSQA* with existing quality metrics including *SSIM*. More precisely, the objective is to compare the correlation between subjective measurements and existing objective quality metrics with the correlation between subjective measurements and the proposed *SSIM*-based *VSQA* metric. These comparisons are done on the *IRCCyN/IVC DIBR* images database [BPLC+11a, BPLC+11b] which provides the subjective quality data described in the following.

The *IRCCyN/IVC DIBR* database is made of three test sequences which are used to generate four different viewpoints (12 sequences to synthesize in total). The test sequences are *BookArrival* (1024×768, 16 cameras with 6.5cm spacing), *Lovebird*1 (1024×768, 12 cameras with 3.5cm spacing) and *Newspaper* (1024×768, 9 cameras with 5cm spacing). The synthesized sequences are obtained by seven *depth-image-based rendering* (*DIBR*) methods in order to reach 84 synthesized sequences in total.

These *DIBR* methods, referenced from $A1$ to $A7$, are illustrated Fig. 6.11 for the *Newspaper* sequence and are briefly described in the following:

- $A1$: *Fehn* [Feh04] applies a 2D *Gaussian* low-pass filter to disparity maps in a manner that no disocclusions occur in the synthesized view. However, this approach fails to extrapolate holes on the left or right border image. The issue is avoided by cropping left or right borders and interpolating in order to reach the original image size.
- $A2$: This second approach is also based on [Feh04] except that borders are not cropped but inpainted by the method proposed by *Telea* [Tel04].



(a) Reference view (b) *A*1 [Feh04] (c) *A*2 [Feh04, Tel04] (d) *A*3 [MFY+09]

(e) *A*4 [MSD+08] (f) *A*5 [NNKD+10] (g) *A*6 [KNND+10] (h) *A*7 (unfilled view)

Figure 6.11: Reference view and *DIBR*-based synthesized views (*Newspaper* sequence).

- *A*3: *Mori* et al. describe in [MFY+09] a 3D view generation system which has been adopted as reference software for *MPEG* standardization experiments in the 3D Video group. This method consists in first projecting depth maps to virtual image plane and then post-filtering these projected depth maps from the assumption that the depth value inside same object changes smoothly.

- *A*4: *Muller* et al. [MSD+08] propose different hole-filling methods as well as a final smoothing filtering along depth discontinuities in order to provide high-quality synthesized views.

- *A*5: In [NNKD+10], each synthesized view disocclusion is compensated using image information from a causal picture neighborhood via a background sprite. Residual holes are filled with an advanced patch-based texture synthesis method.

- *A*6: *Koppel* et al. [KNND+10] extend A5 [NNKD+10] by generating and updating temporally a background sprite.

- *A*7: It corresponds to the unfilled sequences, i.e. with holes.

Key frames taken within the synthesized sequences are evaluated with the following objective metrics:

- *Peak Signal-to-Noise Ratio* (*PSNR*),

- *Universal Quality Index*[1] [WB02] which corresponds to *SSIM* with $C_1 = C_2 = C_3 = 0$ in Eq. 5.4, Eq. 5.5 and Eq. 5.6 (Chapter 5),
- *Single-scale Structural SIMilarity* (*SSIM*[1]) [WBSS04],
- Our *SSIM*-based *VSQA* metric and more precisely the proposed *VSQA* score described in Section 6.1.5. The *VSQA* score has been computed with $p = 19$ (Eq. 6.10),
- *Multi-scale Structural SIMilarity* (*MS-SSIM*[1]) [WSB04],
- *Visual Signal-to-Noise Ratio* (*VSNR*[1]) [CH07] which consists in quantifying the visual fidelity of natural images based on both low-level and mid-level properties of the *HVS*,
- *Weighted Signal-to-Noise Ratio* (*WSNR*[1]) which uses a function adapted to the *HVS*,
- *Visual Information Fidelity* (*VIF*[1]) [SB06] which is an information fidelity metric based on *Shannon* information that is shared between the reference and the distorted images relative to the information contained in the reference image itself,
- *VIFP*[1], a pixel-based version of *VIF*,
- *Information Fidelity Criterion* (*IFC*[1]) [SBDV05] which uses the mutual information between reference and distorted images,
- *Noise Quality Measure* (*NQM*[1]) [DVKG+00] which quantifies the impact on the *HVS* of frequency distortion and noise injection in the tested image.
- *PSNR-HVS* [EAP+06] is based on *PSNR* and *UQI* modified to take into account the *HVS* properties.
- *PSNR-HVSM* [PSE+07] which corresponds to a simple model based on *PSNR* and between-coefficient masking of *DCT* basis functions.

According to *Bosc* et al. [BPLC+11b], the subjective experiments have been done with 43 non-expert observers which have provided five-level absolute categorical ratings (*ACR*-5) for every key frames. More precisely, each one of the *ACR*-5 ratings indicates the opinion of a given observer for a given key frames. The opinion translates in one adjective which corresponds to an integer ranging from 1 to 5 as shown in Tab. 6.1. These opinion scores are then averaged in order to obtain observers *Mean Opinion Scores* (*MOS*). The *MOS* are finally used to obtain the *Difference Mean Opinion Scores* (*DMOS*) where *DMOS* corresponds to the difference between *MOS* computed on reference and synthesized views.

Before performing the comparison, the objective quality scores (*M-SSIM* or *VSQA* score for instance) must be fitted to the subjective measurements using a logistic function according to the *Video Quality Expert Group* (*VQEG*) Phase I *FR-TV* [RLC+00]. Here, the regression is performed with a cubic function as follows:

$$DMOS_p(I) = a_3.dist_s^3(I) + a_2.dist_s^2(I) + a_1.dist_s(I) + a_0 \tag{6.11}$$

where $DMOS_p(I)$ corresponds to the predicted difference mean opinion score for the synthesized view I, $dist_s$ the score obtained with the tested objective metric (*M-SSIM*, *VSQA* score...) and

[1]These metrics have been evaluated thanks to the MeTriX MuX Visual Quality Assessment Package [Gau11]

5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 6.1: *ACR*-5 comparison scale.

Methods	*PLCC DMOS*
PSNR	45.65
SSIM	43.63
SSIM*-based *VSQA	**61.42**
MS-SSIM	55.99
VSNR	35.89
VIF	32.03
VIFP	25.55
UQI	39.27
IFC	27.90
NQM	53.34
WSNR	44.12
PSNR HVSM	40.57
PSNR HVS	39.25

Table 6.2: *Person* linear correlation coefficients (*PLCC*) between *DMOS* and objective quality scores in percentage.

$\{a_0, a_1, a_2, a_3\}$ the parameters of the cubic function to be determined during the regression step (minimization of $DMOS - DMOS_p$).

Once the regression is achieved, the *Person Linear Correlation Coefficient* (*PLCC*) is computed for each fitted objective metrics $DMOS_p$, as described in Eq. 6.12.

$$PLCC = \frac{\sum_{i=1}^{N}(DMOS(I_i) - \overline{DMOS})(DMOS_p(I_i) - \overline{DMOS_p})}{\sqrt{\sum_{i=1}^{N}(DMOS(I_i) - \overline{DMOS})^2 \sum_{i=1}^{N}(DMOS_p(I_i) - \overline{DMOS_p})^2}} \tag{6.12}$$

where $\overline{DMOS}$ and $\overline{DMOS_p}$ denote the average of $DMOS$ and $DMOS_p$ over the N tested key frames I_i.

PLCC measures the consistency between subjective measurements and quality scores for *SSIM*-based *VSQA* and existing metrics. Tab. 6.1 shows the results.

Within the existing metrics and according to the obtained correlation coefficients, MS-SSIM and NQM are the most correlated metrics. These metrics are the only ones which overtake 50% in terms of similarity with human judgment. *SSIM*-based *VSQA* succeeds in improving the correlation coefficients reached by *SSIM* (43.63%) and obtains 61.42%, which means a gain of

around 17.8%. *SSIM*-based *VSQA* achieves the best correlation results. Indeed, Tab. 6.1 shows that *SSIM*-based *VSQA* exceeds the best existing metric, *MS-SSIM*, with a *PLCC* more than 5% higher.

6.3 View synthesis quality assessment along the sequence

For a given still synthesized view, we saw in Section 6.1 that our view synthesis quality assessment approach is able to both: 1) indicate the position of the view synthesis artifacts, 2) give an information about the overall view synthesis quality through the *VSQA* score. On the scale of the whole sequence, this overall quality information can be involved to describe the temporal evolution of the view synthesis quality along the synthesized sequence.

More precisely, we can assign a *VSQA* score to each synthesized frame of the sequence following the procedure described in Section 6.1.5. These *VSQA* scores can then be plotted with respect to time in order to obtain a curve called the *VSQA* curve. This *VSQA* curve allows to identify the more distorted frames along the sequence. In an applicative point of view, this information can be decisive for semi-automatic or automatic view synthesis artifacts removal framework.

View synthesis artifacts removal consists in correcting either the wrongly estimated disparity vectors which have caused the artifacts (which requires a new generation of the synthesized views after disparity refinement) or directly the synthesized views themselves. Instead of correcting all the synthesized views which can be very tedious, we suggest to rely on corrected or already correct disparity vectors coming from a few binocular pairs of the sequence. Then, one requires motion vectors to propagate this accurate information to the frames for which we aim at removing the view synthesis artifacts.

In this context, we can imagine to order all the synthesized frames according to their own *VSQA* score to help the user (or the automatic processing) to know on which frames he must pay more attention during the correction stage. Moreover, depending on the application, a quality threshold can be set with respect to the required quality level. In this case, only the frames with a *VSQA* score under this threshold need to be corrected (in a semi-automatic or automatic way).

This issue about view synthesis artifacts removal, discussed briefly here, will be addressed more thoroughly in Part III (Chapter 14).

6.4 Conclusions and perspectives

To conclude, we have designed a new objective view synthesis quality assessment metric: the *View Synthesis Quality Assessment* (*VSQA*) metric. It aims to handle areas where either disparity estimation or interpolation has caused view synthesis artifacts. The key feature of the proposed method is the use of three visibility maps which characterize the complexity in terms of textures, the diversity of gradient orientations as well as the presence of high contrast. *VSQA* can be based on any existing image quality assessment metric.

Our view synthesis quality assessment approach is able to both indicate the exact position of the view synthesis artifacts and give an information about the overall view synthesis quality through the *VSQA* score.

An overall performance comparison between *SSIM*-based *VSQA* and existing quality metrics has been done. The obtained correlation coefficients between subjective measurements and objective quality scores have shown that *SSIM*-based *VSQA* improves the results obtained by a simple quality measurement reached by *SSIM*. Moreover, experimental tests have proved that *SSIM*-based *VSQA* exceeds all the tested existing quality metrics.

Future work aims at improving the proposed quality measurement system in order to obtain a method more correlated to human perception of artifacts. In this direction, *VSQA* as an extension of other metrics could be tested. According to the results obtained with the *IRCCyN/IVC DIBR* images database, it could be interesting in particular to extend *MS-SSIM* [WSB04] or *NQM* [DVKG+00] toward an accurate view artifacts detection.

In addition, *VSQA* could be improved by incorporating a *No-Reference* (*NR*) assessment stage dedicated to the quality evaluation of disoccluded regions, i.e. regions occluded in the original views which become visible in the synthesized view.

We would also like to introduce temporal consistency in the quality measurements in order to take into account temporal fluctuations of spatial distortions. The extension of the *VSQA* image metric into a video quality assessment metric would require temporal analysis of the spatial artifacts through motion information to be able to track objects or areas along the video sequence. In this context, modelling the mechanisms of the perception of the temporal distortions by the *HVS* is a crucial issue since the perception of spatial distortions over time can be significantly modified by their temporal changes.

Bibliography Part I

[Ass13] American Optometric Association. Infant vision: Birth to 24 months of age. Steps in infant vision development. http://www.aoa.org/patients-and-public/good-vision-throughout-life/childrens-vision/infant-vision-birth-to-24-months-of-age, 2013.

[BGE+] A. Boev, A. Gotchev, K. Egiazarian, A. Aksay, and G.B. Akar. Towards compound stereo-video quality metric: a specific encoder-based framework. In *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 218–222.

[BKP+11] E. Bosc, M. Köppel, R. Pépion, M. Pressigout, L. Morin, P. Ndjiki-Nya, and P. Le Callet. Can 3D synthesized views be reliably assessed through usual subjective and objective evaluation protocols? In *IEEE International Conference on Image Processing*, 2011.

[BLCCC08] A. Benoit, P. Le Callet, P. Campisi, and R. Cousseau. Using disparity for quality assessment of stereoscopic images. In *IEEE International Conference on Image Processing, ICIP 2008, San Diego, United States*, 2008.

[BLL+10] K. Berger, C. Lipski, C. Linz, A. Sellent, and M. Magnor. A ghosting artifact detector for interpolated image quality assessment. In *Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on*, pages 1–6, 2010.

[Bou08] F. Boughorbel. Adaptive filters for depth from stereo and occlusion detection. In *SPIE IS&T Electronic Imaging*, volume 6803, 2008.

[BPLC+11a] E. Bosc, R. Pépion, P. Le Callet, M. Köppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin. *IRCCyN/IVC DIBR* database. http://www.irccyn.ec-nantes.fr/spip.php?article866, 2011.

[BPLC+11b] E. Bosc, R. Pépion, P. Le Callet, M. Köppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin. Towards a new quality metric for 3D synthesized view assessment. *IEEE Journal on Selected Topics in Signal Processing*, 2011.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[CH07] D.M. Chandler and S.S. Hemami. VSNR: A wavelet-based visual signal-to-noise ratio for natural images. *Image Processing, IEEE Transactions on*, 16(9):2284–2298, 2007.

[CLCM07] P. Campisi, P. Le Callet, and E. Marini. Stereoscopic images quality assessment. In *Proceedings of 15th European Signal Processing Conference*, 2007.

[CRM12] P.-H. Conze, P. Robert, and L. Morin. Objective view synthesis quality assessment. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.

[CYX07] G.H. Chen, C.L. Yang, and S.L. Xie. Gradient-based structural similarity for image quality assessment. In *IEEE International Conference on Image Processing*, pages 2929–2932, 2007.

[DB10] F. Devernay and P. Beardsley. Stereoscopic cinema. In *Image and Geometry processing for 3-D cinematography*, pages 11–51. Springer, 2010.

[DRP10] F. Devernay and A. Ramos-Peon. Novel view synthesis for stereoscopic cinema: Detecting and removing artifacts. In *3DVP 2010: ACM Workshop on 3D Video Processing*, 10 2010.

[DVKG+00] N. Damera-Venkata, T.D. Kite, W.S. Geisler, B.L. Evans, and A.C. Bovik. Image quality assessment based on a degradation model. *Image Processing, IEEE Transactions on*, 9(4):636–650, 2000.

[EAP+06] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli. New full-reference quality metrics based on hvs. In *Second International Workshop on Video Processing and Quality Metrics, Scottsdale, USA*, volume 4, 2006.

[EW02] G. Egnal and R.P. Wildes. Detecting binocular half-occlusions: empirical comparisons of five approaches. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1127–1133, 2002.

[EWDS+10] E. Ekmekcioglu, S. T. Worrall, D.V.S.X. De Silva, W.A.C. Fernando, and A. M. Kondoz. Depth based perceptual quality assessment for synthesised camera viewpoints. In *Second International Conference on User Centric Media, UCMedia 2010, Palma de Mallorca, Spain*, 2010.

[Feh04] C. Fehn. Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In *SPIE IS&T Electronic Imaging*, volume 5291, page 93, 2004.

[Gau11] M. Gaubatz. MetriX MuX Visual Quality Assessment Package. http://foulard.ece.cornell.edu/gaubatz/metrix_mux/, 2011.

[JMFK10] P. Joveluro, H. Malekmohamadi, W.A. Fernando, and A.M. Kondoz. Perceptual video quality metric for 3D video quality assessment. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1–4. IEEE, 2010.

[KNND+10] M. Köppel, P. Ndjiki-Nya, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand. Temporally consistent handling of disocclusions with texture synthesis for depth-image-based rendering. In *IEEE International Conference on Image Processing*, pages 1809 –1812, sept. 2010.

[KO94] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(9):920–932, 1994.

[LB10] C. Li and A.C. Bovik. Content-partitioned structural similarity index for image quality assessment. *Signal Processing: Image Communication*, 25(7):517–526, 2010.

[Lip97] L. Lipton. Stereographics, developers handbook. *StereoGraphics Corporation*, 3, 1997.

[LW09] L. Liu and Y. Wang. A mean-edge structural similarity for image quality assessment. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, pages 311–315. IEEE, 2009.

[Mat09] G. Mather. *Foundations of sensation and perception*, volume 2. Psychology Press, 2009.

[MFY+09] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto. View generation with 3D warping using depth information for FTV. *Signal Processing: Image Communication*, 24(1-2):65–72, 2009.

[MIS04] L.M.J. Meesters, W.A. IJsselsteijn, and P.J.H. Seuntiens. A survey of perceptual evaluations and requirements of three-dimensional TV. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):381–391, 2004.

[MSD+08] K. Muller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. View synthesis for advanced 3D video systems. *EURASIP Journal on Image and Video Processing*, pages 1–12, 2008.

[MZK10] M. Mueller, F. Zilly, and P. Kauff. Adaptivecross-trilateral depth map filtering. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2010*, pages 1–4. IEEE, 2010.

[NLMLCB09] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba. Considering temporal variations of spatial visual distortions in video quality assessment. *Selected Topics in Signal Processing, IEEE Journal of*, 3(2):253–265, 2009.

[NNKD+10] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand. Depth image-based rendering with advanced texture synthesis for 3D video. In *IEEE International Conference on Multimedia & Expo*, volume 13, pages 453 –465, June 2010.

[PSE+07] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin. On between-coefficient contrast masking of dct basis functions. In *Third International Workshop on Video Processing and Quality Metrics, Scottsdale, Arizona, USA*, page 4, 2007.

[RLC+00] A.M. Rohaly, J. Libert, P. Corriveau, A. Webster, et al. Final report from the video quality experts group on the validation of objective models of video quality assessment. *ITU-T Standards Contribution COM*, pages 9–80, 2000.

[RTDC12] P. Robert, C. Thébault, V. Drazic, and P.-H. Conze. Disparity-compensated view synthesis for s3D content correction. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.

[SAB11] M. Solh, G. AlRegib, and J.M. Bauza. 3VQM: A vision-based quality measure for DIBR-based 3D videos. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pages 1–6, 2011.

[SB06] H.R. Sheikh and A.C. Bovik. Image information and visual quality. *Image Processing, IEEE Transactions on*, 15(2):430–444, 2006.

[SBDV05] H.R. Sheikh, A.C. Bovik, and G. De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *Image Processing, IEEE Transactions on*, 14(12):2117–2128, 2005.

[Sch96] D. Scharstein. Stereo vision for view synthesis. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 852–858. IEEE, 1996.

[Sch99] D. Scharstein. *View synthesis using stereo vision*. Springer-Verlag, 1999.

[SKH08] J. Starch, J. Kilner, and A. Hilton. Objective quality assessment in free-viewpoint video production. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2008*, pages 225–228. IEEE, 2008.

[SLKS05] J. Sun, Y. Li, S.B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 399–406. IEEE, 2005.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[SS11] D. Scharstein and R. Szeliski. *Middlebury* stereo benchmark. http://vision.middlebury.edu/stereo/eval/, 2011.

[SYZ09] L. Shen, J. Yang, and Z. Zhang. Quality assessment of stereo images with stereo vision. In *2nd International Congress on Image and Signal Processing*, pages 1–4. IEEE, 2009.

[SZS03] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(7):787–800, 2003.

[Tel04] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics, GPU, and Game Tools*, 9(1):23–34, 2004.

[UCES11] H. Urey, K.V. Chellappan, E. Erden, and P. Surman. State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, 99(4):540–555, 2011.

[WB02] Z. Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[WDM10] D. Wang, W. Ding, and Y. Man. A joint image quality assessment method based on global phase coherence and structural similarity. In *3rd International Congress on Image and Signal Processing*, volume 5, pages 2307–2311. IEEE, 2010.

[Whe38] C. Wheatstone. Contributions to the physiology of vision.–part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical transactions of the Royal Society of London*, 128:371–394, 1838.

[WJMG10] Y. Wang, T. Jiang, S. Ma, and W. Gao. Image quality assessment based on local orientation distributions. In *28th Picture Coding Symphosium, PCS2010, December 8-10, 2010, Nagoya, Japan*, 2010.

[WL10] Z. Wang and Q. Li. Information content weighted structural similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 2010.

[WLB04] Z. Wang, L. Lu, and A.C. Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2):121–132, 2004.

[WSB04] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1398–1402. IEEE, 2004.

[YHZ+09] J. Yang, C. Hou, Y. Zhou, Z. Zhang, and J. Guo. Objective quality assessment method of stereo images. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video, 2009*, pages 1–4. IEEE, 2009.

[YWY+09] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):492–504, 2009.

[YXPW10] J. You, L. Xing, A. Perkis, and X. Wang. Perceptual quality assessment for stereoscopic images based on 2d image quality metrics and disparity analysis. In *Proc. Int. Workshop Video Processing and Quality Metrics, Scottsdale, Arizona, USA*, 2010.

[ZK00] C.L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, 2000.

[ZLYP09] J. Zhu, M. Liao, R. Yang, and Z. Pan. Joint depth and alpha matte optimization via fusion of stereo and time-of-flight sensor. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 453–460. IEEE, 2009.

[Zon07] Ray Zone. *Stereoscopic Cinema and the origins of 3-D Film, 1838-1952*. University Press of Kentucky, 2007.

Part II

Long-term dense motion estimation

Chapter 7

Introduction to motion estimation

Image motion estimation aims at studying the displacement between two images, or more generally across a video sequence. The motion estimation process translates in determining motion vectors which represent the temporal dynamic of entities such as pixels, areas or objects between the considered frames. It is one of the major tasks of both human and artificial visual systems and it has become a predominant topic in the field of computer vision since the seminal formulations of *Lucas* and *Kanade* [LK81] and *Horn* and *Schunck* [HS81] in the early 80's.

Motion estimation has been largely studied since these early works, especially because it covers many applications in many advanced tasks such as visual surveillance, visual servoing, action detection, description and recognition, scene segmentation, video editing, structure-from-motion, augmented reality, motion modification, video indexing, annotation or video compression. It is still an active research topic due to its inherent complexity, especially when dealing with real scenes and complex situations. Real scenes may contain objects with non-rigid motion or may deal with occlusion, spatially and temporally non-uniform illumination changes or any other complex scenario. Such situations require a very robust estimator to be processed efficiently. Moreover, some applications require specific requirements such as real-time processing for embedded equipment or high quality rendering for movie postproduction (visual effects or any other video editing tasks...). These recent and new challenges make the motion estimation field still an important area of research.

Motion estimation process aims more precisely at establishing correspondences between images which have been captured at different instants, analogously to disparity estimation which works in the spatial domain between different points of view. Disparity estimation is usually limited to a 1D horizontal problem since the captured data have been rectified which restricts the area of search of correspondences to an horizontal epipolar line. On the contrary, motion estimation deals with a 2D matching issue.

Depending of the context and according to the required precision, different types of motion estimation algorithms can be considered from object-based estimation to sparse, region-based or pixel-wise matching. First, object-based motion estimation involves an object model based on cues such as color, geometry or shape which is used to identify and then match one or several objects across a video sequence [CRM00, PHVG02]. Second, the field of motion estimation shares a long history with sparse estimates and more precisely *feature* points [TK91, ST94]. Establishing *feature* matches consists in focusing on a sparse set of salient points which are extracted and then matched based on their distinctive appearance. Object-based motion estimation and *feature* matching are more frequently involved in the context of long-term tracking than between two frames only. Therefore, these methods will be detailed in Chapter 8 whose

topic is long-term motion estimation. A third way to perform motion estimation is to rely on region-based approaches. The most well known approach deals with block matching algorithms. The idea behind block matching is to divide the frames into *macro-blocks* and then to estimate for each *macro-block* a single motion vector that points to the target frame. Different strategies such as the *three step search* (*TSS*) or the *diamond search* (*DS*) have been established to restrict the search area and therefore to limit the computational complexity [Bar04]. A cost function like the *mean absolute difference* (*MAD*) or the *mean squares error* (*MSE*) is minimized to obtain the best matching for each *macro-block*. Finally, if the full density is required, we rely on pixel-wise (i.e. dense) motion estimation, also called *optical flow* estimation, whose aim is to link each pixel of a first image to positions in a second image.

This chapter focuses on dense motion estimation between two consecutive frames and gives an overview of the existing methods. According to [IA00], the existing dense motion estimators can be categorized into pixel-based methods or *feature*-based methods. Pixel-based methods require the computation of a displacement vector for each pixel whereas *feature*-based methods rely on sparse motion estimation before densifying the flow. In other words, sparse information coming from a *feature* matching algorithm is used to guide the dense motion estimation process. More precisely, one extracts a sparse set of *features*, analyses their correspondences and finally determines dense motion relying on motion models such as affine, homographic or non-parametric models. In practice, pixel-based methods minimize an error computed using image information collected from all the pixels whereas *feature*-based approaches involve distances between a few corresponding *features* [IA00].

In the following, we study the state-of-the-art of dense motion estimation between two consecutive frames from early *optical flow* formulations (Section 7.1) to very recent algorithms (Section 7.2). This study includes both pixel-based and *feature*-based methods. Section 7.3 focuses on the occlusion detection task which allows to identify which pixels in the first image are occluded in the second frame. Finally, we introduce the concept of long-term motion estimation in Section 7.4 with in mind the fact that *optical flow* can be a key tool toward this goal.

7.1 Early formulations

Early formulations in this context have been introduced into the two following papers: [LK81] which address the problem through image registration and [HS81] which deals explicitly with *optical flow* and defines it as the distribution of apparent velocities of movement of brightness patterns in an image. In other words, *optical flow* is the dense velocity field observed from objects, surfaces and edges of an image sequence.

7.1.1 The *brightness constancy constraint*

The starting point for most pixel-based methods including seminal works [LK81, HS81] is the *brightness constancy constraint*. The idea is to assume that the brightness of a particular point remains constant while its location changes. Let $\boldsymbol{x} = (x, y)$ be the coordinates of a given pixel at instant t and let $\boldsymbol{u}(\boldsymbol{x}) = [u(x, y), v(x, y)]$ be the two-dimensional displacement vector for the position $\boldsymbol{x}$ between t and $t + \Delta t$. Finally, let $I(\boldsymbol{x}, t)$ be the image brightness for $\boldsymbol{x}$ at time t and $I(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), t + \Delta t)$ the image brightness at the position $\boldsymbol{x}$ displaced from $\boldsymbol{u}(\boldsymbol{x})$ at time $t + \Delta t$. The *brightness constancy constraint* can be expressed as follows:

$$I(\boldsymbol{x}, t) = I(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}), t + \Delta t) \tag{7.1}$$

$I(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}),t+\Delta t)$ can be linearized locally using a first order *Taylor* expansion:

$$I(\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}),t+\Delta t)=I(\boldsymbol{x},t)+\Delta t.\frac{\partial I(\boldsymbol{x},t)}{\partial t}+\boldsymbol{u}(\boldsymbol{x}).\nabla I(\boldsymbol{x},t)+\vartheta(\boldsymbol{u}^2(\boldsymbol{x}),\Delta t^2) \tag{7.2}$$

In Eq. 7.2, $\nabla I(\boldsymbol{x},t)=(\frac{\partial I(\boldsymbol{x},t)}{\partial x},\frac{\partial I(\boldsymbol{x},t)}{\partial y})^t$ is the spatial gradient at time t. $\vartheta(\boldsymbol{u}^2(\boldsymbol{x}),\Delta t^2)$ can be neglected assuming that the displacement $\boldsymbol{u}(\boldsymbol{x})$ and Δt are small enough. Combining the two last equations gives:

$$\Delta t.\frac{\partial I(\boldsymbol{x},t)}{\partial t}+\boldsymbol{u}(\boldsymbol{x}).\nabla I(\boldsymbol{x},t)=0 \tag{7.3}$$

Finally, Eq. 7.3 can be rewritten as follows assuming that $I_t(\boldsymbol{x},t)=\frac{\partial I(\boldsymbol{x},t)}{\partial t}$ denotes the temporal derivative of $I(\boldsymbol{x},t)$:

$$I_t(\boldsymbol{x},t)+\frac{\boldsymbol{u}(\boldsymbol{x})}{\Delta t}.\nabla I(\boldsymbol{x},t)=0 \tag{7.4}$$

Using this *brightness constancy constraint* results in an ill-posed problem. Eq. 7.4 is not sufficient to obtain the two unknown variables, $u(\boldsymbol{x})$ and $v(\boldsymbol{x})$. This problem, known as the *aperture problem*, can be overcome by adding constraints following local [LK81] or global [HS81] considerations.

7.1.2 A local approach to solve the *optical flow* equation

The idea of *Lucas* and *Kanade* in [LK81] was to assume that the displacement is constant within the neighborhood of the pixel under consideration, $\boldsymbol{x}_0=[x_0,y_0]$. Let us consider the pixels $\boldsymbol{x}_i$ $\forall i\in[\![1,\dots,N^2-1]\!]$ with $\boldsymbol{x}_i\neq\boldsymbol{x}_0$ located in the spatial window $\mathcal{N}(\boldsymbol{x}_0)$ of size $N\times N$ around $\boldsymbol{x}_0$. Thus, $\boldsymbol{u}(\boldsymbol{x}_i)=\boldsymbol{u}(\boldsymbol{x}_0)$ $\forall\boldsymbol{x}_i\in\mathcal{N}(\boldsymbol{x}_0)$. Consequently, $\boldsymbol{u}(\boldsymbol{x}_0)$ satisfies the following system of equations:

$$I_t(\boldsymbol{x}_i,t)+\frac{\boldsymbol{u}(\boldsymbol{x}_0)}{\Delta t}.\nabla I(\boldsymbol{x}_i,t)=0\ \forall i\in[\![1,\dots,N^2-1]\!] \tag{7.5}$$

In this context, a least-squares approach can be considered to obtain $\boldsymbol{u}(\boldsymbol{x}_0)$. It leads to the minimization of the following energy:

$$E(\boldsymbol{u}(\boldsymbol{x}_0))=\sum_{\boldsymbol{x}_i\in\mathcal{N}(\boldsymbol{x}_0)}[I_t(\boldsymbol{x}_i,t)+\frac{\boldsymbol{u}(\boldsymbol{x}_0)}{\Delta t}.\nabla I(\boldsymbol{x}_i,t)]^2 \tag{7.6}$$

In the matrix form, Eq. 7.6 can be rewritten as follows:

$$\begin{pmatrix}\frac{\partial I(\boldsymbol{x}_1,t)}{\partial x} & \frac{\partial I(\boldsymbol{y}_1,t)}{\partial y}\\ \frac{\partial I(\boldsymbol{x}_2,t)}{\partial x} & \frac{\partial I(\boldsymbol{y}_2,t)}{\partial y}\\ \vdots & \vdots\\ \frac{\partial I(\boldsymbol{x}_{N^2-1},t)}{\partial x} & \frac{\partial I(\boldsymbol{y}_{N^2-1},t)}{\partial y}\end{pmatrix}\times\begin{pmatrix}u(x_0,y_0)\\ v(x_0,y_0)\end{pmatrix}=-\begin{pmatrix}\frac{\partial I(\boldsymbol{x}_1,t)}{\partial t}\\ \frac{\partial I(\boldsymbol{x}_2,t)}{\partial t}\\ \vdots\\ \frac{\partial I(\boldsymbol{x}_{N^2-1},t)}{\partial t}\end{pmatrix} \tag{7.7}$$

In a more compact way, Eq. 7.7 becomes:

$$A\boldsymbol{u}^t=b \tag{7.8}$$

with $A = [\nabla I(\boldsymbol{x}_1, t)^t, ..., \nabla I(\boldsymbol{x}_{N^2-1}, t)^t]^t$ and $b = [-I_t(\boldsymbol{x}_1, t), ..., -I_t(\boldsymbol{x}_{N^2-1}, t)]^t$. The solution of this system of equations is given in Eq. 7.9. A^tA, also called the *structure tensor*, must be invertible, i.e. with no zero eigen-values.

$$\boldsymbol{u}^t = (A^tA)^{-1}.A^tb \tag{7.9}$$

Instead of giving the same importance to all the pixels of the spatial window, the *Lucas-Kanade* algorithm uses a weighted least-squares formulation. The idea behind this formulation is to increase the influence of the central pixels of $\mathcal{N}(\boldsymbol{x}_0)$ compared to pixels at its periphery. Assuming that W is a *Gaussian* window for instance, the solution becomes:

$$\boldsymbol{u}^t = (A^tWA)^{-1}.A^tWb \tag{7.10}$$

where W is a $N \times N$ diagonal matrix containing the weights w_i assigned to pixels $\boldsymbol{x}_i$ $\forall i \in [\![1, \ldots, N^2 - 1]\!]$.

A severe drawback of this method concerns especially the *structure tensor* which is not invertible for several situations including homogeneous regions (eigen-values of A^tA close to zero) or image edges (no single solution) for instance.

7.1.3 A global approach to solve the *optical flow* equation

Contrary to [LK81] which deals implicitly with spatial consistency, global approaches have added explicitly spatial smoothness assumptions within the energy to be minimized. In this context, *Horn* and *Schunck* have studied in [HS81] a variational formulation, Eq. 7.11, where a regularization term is used to model how the flow is expected to vary spatially:

$$min_{\boldsymbol{u}} \left(\int_\Omega [I(\boldsymbol{x}) - I(\boldsymbol{x} + \boldsymbol{u})]^2 d\Omega + \lambda \int_\Omega |\nabla u|^2 + |\nabla v|^2 d\Omega \right) \tag{7.11}$$

This formulation over the image grid Ω contains a data term which corresponds to the *brightness constancy constraint* previously described. Moreover, Eq. 7.11 is composed of a spatial regularization term which penalizes high variations of the motion field $\boldsymbol{u}$. In other words, the goal is to limit the difference between the displacement at a point and the average motion over a small neighborhood in order to obtain smooth motion fields. λ gives the relative weight between both terms.

Compared to local methods, global methods such as [HS81, HB93] are able to compute *optical flow* vectors for homogeneous areas. The regularization term spatially propagates the motion estimates on areas where the data term is not discriminant. Although this spatial regularization has a significant advantage, it causes over-smoothing along motion discontinuities. The formulation forces motion vectors to be close to the average of its neighbors which reduces the accuracy across boundaries. In addition, deviations are penalized in a quadratic way which does not allow an efficient *outlier* rejection.

7.2 Significant progress since early formulations

In what follows, we focus on recent papers and last progresses from early formulations. The seminal formulations of [LK81] and [HS81] have been extended in numerous papers. Let us describe the main aspects of state-of-the-art *optical flow* methods.

7.2.1 Robustness to motion outliers

Many different robust functions $\phi(.)$ have been explored in order to substitute the quadratic formulation $[I(\boldsymbol{x}) - I(\boldsymbol{x}+\boldsymbol{u})]^2$ in Eq. 7.11.

First, *Black* and *Anandan* have reformulated the least-squares estimation problem using robust estimation techniques. Robust statistics are used in [BA96] to accurately estimate the *optical flow* vectors across motion boundaries. The goal of robust statistics is to best fit the majority of the data while rejecting the influence of *outliers*. Two examples of robust functions are the *Geman-McClure* function $\phi(x) = \frac{x^2}{\epsilon+x^2}$ and the *Lorentzian* function $\phi(x) = \log(1+\frac{1}{2}(\frac{x}{\epsilon})^2)$ where ϵ is a positive constant.

Furthermore, using robust functions in the regularization term allows to preserve sharp discontinuities whereas a least-squares formulation results in an over-smoothing of the resulting *optical flow* fields.

Many other functions have been tested as in [BBPW04] where *Brox* et al. use $\phi(x^2) = \sqrt{x^2+\epsilon^2}$ (ϵ is a small positive constant), a robust concave function which reduces the influence of the *outliers* contrary to quadratic penalizers.

[BBPW04] introduces also a gradient constancy constraint which is used to complement the *brightness constancy constraint* in the energy function. Analogously to Eq. 7.1, we can write the following equation assuming that the gradient does not vary due to the displacement:

$$\nabla I(\boldsymbol{x}, t) = \nabla I(\boldsymbol{x}+\boldsymbol{u}, t+\Delta t) \tag{7.12}$$

where $\nabla I(\boldsymbol{x}, t) = (\frac{\partial I(x,y,t)}{\partial x}, \frac{\partial I(x,y,t)}{\partial y})^t$ deals with the spatial gradient at instant t. Thus, the following data term has been proposed:

$$E_{data}(\boldsymbol{u}) = \int_\Omega \phi([I(\boldsymbol{x}) - I(\boldsymbol{x}+\boldsymbol{u})]^2 + \gamma[\nabla I(\boldsymbol{x}) - \nabla I(\boldsymbol{x}+\boldsymbol{u})]^2) d\Omega \tag{7.13}$$

where γ weights the relative influence between gradient and brightness constancy constraints. Eq. 7.13 is robust to illumination effect since image gradients are invariant to additive brightness changes.

The regularization, Eq. 7.14, is done in [BBPW04] by penalizing the *Total Variation* (TV) of the flow field and involves a discontinuity-preserving spatio-temporal smoothness constraint. The idea is to rely on a model of a piecewise smooth flow field. The robust concave function $\phi(x^2) = \sqrt{x^2+\epsilon^2}$ is also used in the regularization term so that the energy is convex after linearization. The smoothness constraint is applied in the spatio-temporal domain if the displacements across more than two frames are required. In this case, the spatio-temporal gradient $\nabla_3. = (\frac{\partial .}{\partial x}, \frac{\partial .}{\partial y}, \frac{\partial .}{\partial t})^t$ is involved. For a single pair of frames, the spatial gradient substitutes the spatio-temporal gradient.

$$E_{reg}(\boldsymbol{u}) = \int_\Omega \phi(|\nabla_3 u|^2 + |\nabla_3 v|^2) d\Omega \tag{7.14}$$

Lempitsky et al. explore in [LRR08] a different robust function by removing lower spatial frequencies to make the data term robust to illumination and exposure changes.

Many motion estimators use both the L^1 robust penalty function and a *Total Variation* (TV) optimization method, as done in [ZPB07, WPZ+09, XJM10, XJM12]. It is one of the most common formulations among all the existing ones. These estimators are defined as TV-L^1 estimators. This formulation can preserve discontinuities and offers a certain robustness against illumination changes, occlusions and noise.

The TV-L^1 approach leads to the following functional displayed in Eq. 7.15. Instead of using $\phi(x) = x^2$ and $\psi(\nabla \boldsymbol{u}) = |\boldsymbol{u}|^2$ as robust functions for the data and the regularization term as done in the *Horn-Schunk* model (Eq. 7.11), the TV-L^1 formulation involves $\phi(x) = |x|$ and $\psi(\nabla \boldsymbol{u}) = |\boldsymbol{u}|$.

$$min_{\boldsymbol{u}} \left(\int_{\Omega} |I(\boldsymbol{x}) - I(\boldsymbol{x} + \boldsymbol{u})| d\Omega + \lambda \int_{\Omega} |\nabla u| + |\nabla v| d\Omega \right) \tag{7.15}$$

The computational difficulties due to the fact that the data term and the regularization term are not continuously differentiable can be overcome as follows. A first idea consists in using a differentiable approximation such as $\phi(x^2) = \sqrt{x^2 + \epsilon^2}$ which is used in the data term of [BBPW04]. An alternative deals with a numerical scheme based on a dual formulation of the TV minimization and on a point-wise thresholding step [ZPB07].

7.2.2 Photo consistency assessment

Whatever the robust function $\phi(.)$ used in the data term, let us focus on how the *brightness constancy constraint* is involved in practice between the first image I_n and the second frame I_{n+1}. A common tool for this task is the matching cost [RTDC12], a quality measure which is directly related on the image residual $\phi(I(\boldsymbol{x}) - I(\boldsymbol{x} + \boldsymbol{u}))$. Computing the matching cost $C(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))$ consists in assessing the photo consistency between the starting point $\boldsymbol{x}$ of the motion vector $\boldsymbol{u}(\boldsymbol{x})$, and its ending point in the second frame, $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$. The matching cost can be computed very basically as follows:

$$C(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = \sum_{c \in \{r,g,b\}} \phi(I_n^c(\boldsymbol{x}) - \tilde{I}_{n+1}^c(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))) \tag{7.16}$$

where $\tilde{I}_{n+1}^c$ has been interpolated (through bilinear interpolation for instance) using the R, G, or B component of the nearest pixels located around $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ in I_{n+1}. This classic matching cost can be extended to assess motion vectors through a texture patch similarity by considering the local neighborhood of both the starting and the ending point of $\boldsymbol{u}$. An alternative consists in considering a local warping quality by involving in the computation the motion vectors defined with respect to each pixel $\boldsymbol{y}$ in the neighborhood $N(\boldsymbol{x})$ of $\boldsymbol{x}$ as in Eq. 7.17 where $|N(\boldsymbol{x})|$ is the cardinal of the neighborhood $N(\boldsymbol{x})$.

$$C(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = \frac{1}{|N(\boldsymbol{x})|} \sum_{\boldsymbol{y} \in N(\boldsymbol{x})} \sum_{c \in \{r,g,b\}} \phi(I_n^c(\boldsymbol{y}) - \tilde{I}_{n+1}^c(\boldsymbol{y} + \boldsymbol{u}(\boldsymbol{y}))) \tag{7.17}$$

The correlation can be used instead of these previous *sum of absolute differences* (SAD) formulations. This includes in particular *non-normalized cross-correlation* (CC), *normalized cross-correlation* (NCC) or *zero-mean normalized cross-correlation* ($ZNCC$) [LK10].

The correlation quantifies the photo consistency between a patch P_n of I_n and another patch P_{n+1} of I_{n+1}. In practice, we can assign a color correlation factor $G(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = [G^R, G^G, G^B]^t$ to an *optical flow* vector $\boldsymbol{u}(\boldsymbol{x})$ starting from $\boldsymbol{x}$ such as:

$$I^c_{n+1}(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})) = G^c(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})).I^c_n(\boldsymbol{x}) \tag{7.18}$$

with $c = \{r, g, b\}$. The correlation is computed by comparing the patch around $\boldsymbol{x}$ in I_n and the one aroung the ending point of $\boldsymbol{u}(\boldsymbol{x})$ in I_{n+1}. This comparison can be done as follows:

$$G^c(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = \frac{E[P_n.P_{n+1}]}{E[P_n^2]} = \frac{\sum\limits_{\boldsymbol{y} \in N(\boldsymbol{x})} I^c_n(\boldsymbol{y}).\tilde{I}^c_{n+1}(\boldsymbol{y} + \boldsymbol{u}(\boldsymbol{x}))}{\sum\limits_{\boldsymbol{y} \in N(\boldsymbol{x})} [I^c_n(\boldsymbol{y})]^2} \tag{7.19}$$

where $E[.]$ corresponds to the expectation. The same computation can be done on luminance values (instead on RGB components) which finally gives a scalar correlation factor.

7.2.3 Coarse-to-fine strategies to overcome aliasing

Current practices generally include coarse-to-fine strategies as in [MP02, BBPW04, ZPB07]. Starting from the coarsest scale, multi-resolution strategies aim at warping one of the two images according to a current motion estimation at each scale of the image pyramid (obtained through successive down-samplings). Then, the obtained solution is used as initialization for the next finer level. This procedure is repeated until the initial image resolution is reached.

Coarse-to-fine warping strategies allow to overcome aliasing which causes matching ambiguities since many pixels has the same intensity. Such strategies are especially used for motion estimation with large displacements (Section 7.2.4). Indeed, the first order *Taylor* approximation which linearizes locally $I(\boldsymbol{x} + \boldsymbol{u}, t + \Delta t)$ in Eq. 7.1 is only valid for small displacements. Unfortunately, these multi-resolution strategies could fail to recover fine motion structures.

7.2.4 Accuracy against large displacements

The algorithm proposed in [SPC09] is dedicated to large displacement *optical flow* estimation. It does not require a coarse-to-fine approach and its respective warping strategy. Indeed, the initial finding of this work is that coarse-to-fine warping can fail when the relative motion of small scale structures is larger than their own spatial scale. It is the case for instance for small body parts such as hands since they can move very rapidly.

[SPC09] works on the full scale image and proposes to decouple the data term and the regularization term. The authors decompose the original non-convex functional into a functional which can be minimized by alternating two globally optimal algorithms. This is possible by introducing an auxiliary vector field via a quadratic relaxation scheme [ZPB07].

Brox et al. propose in [BM11] an alternative to perform motion estimation including large displacements which combines descriptor matching, variational model and a coarse-to-fine strategy. The proposed approach starts by considering both brightness and gradient constancy constraints (data term similar to Eq. 7.13) and penalizes the TV of the flow field (regularization term similar to Eq. 7.14) as in [BBPW04]. Point correspondences obtained through descriptor

matching are added into the variational approach and result in two new terms. First, E_{desc} which performs the matching task between descriptors (Eq. 7.20). Second, E_{match} which penalizes the differences of motion estimation with respect to the correspondence vectors $\boldsymbol{u}_{desc}$ established between feature points (Eq. 7.21).

$$E_{desc}(\boldsymbol{u}_{desc}) = \int_{\Omega} \delta(\boldsymbol{x}) |\boldsymbol{f}_2(\boldsymbol{x} + \boldsymbol{u}_{desc}(\boldsymbol{x})) - \boldsymbol{f}_1(\boldsymbol{x})|^2 d\Omega \tag{7.20}$$

where $\boldsymbol{f}_1(.)$ and $\boldsymbol{f}_2(.)$ correspond to the sparse fields of feature vectors in both frames. $\delta(\boldsymbol{x})$ equals to 1 if a descriptor is available at grid point $\boldsymbol{x}$, 0 otherwise. In practice, [BM11] investigates SIFT-like features [Low04], descriptors obtained through histograms of oriented gradients (HOG) [DT05] and descriptors based on a variant of geometric blur (GB) [BM01] (local histograms computed at different integration scales).

$$E_{match}(\boldsymbol{u}) = \int_{\Omega} \delta(\boldsymbol{x}) C(\boldsymbol{x}) \phi(|\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}_{desc}(\boldsymbol{x})|^2) d\Omega \tag{7.21}$$

where $\phi(x^2) = \sqrt{x^2 + \epsilon^2}$ as in [BBPW04]. $C(\boldsymbol{x})$ corresponds to a matching score which assesses the intrinsic quality of each descriptor match. As shown in Eq. 7.22, this matching score deals with d_1 and d_2, respectively the distances between the best and the second best match in terms of sums of squares differences of warped patches.

$$C(\boldsymbol{x}) = \frac{d_2 - d_1}{d_1} \tag{7.22}$$

The minimization of the resulting functional combines descriptor matching (i.e. the minimization of E_{desc}, Eq. 7.20) and a continuation method which deals with the rest of the energy. Both methods are complementary since descriptor matching neglects regularization whereas the continuation method neglects image details. Note that descriptor matching is preponderant at coarse levels due to a weak ratio between the point correspondences and the total number of pixels at these scales. This improves the estimation accuracy for large displacements, contrary to classical coarse-to-fine strategies. The effects of descriptor matching decrease for finer levels. Moreover, the outliers which can be introduced through descriptor matching are generally removed as more and more data from the image is taken into account.

The *optical flow* estimation method presented in [XJM10, XJM12] addresses the same issue: accurately estimate both large and small displacements. Similarly to [BM11], *Xu* et al. rely in [XJM10, XJM12] on a coarse-to-fine refinement. They aim at reducing the reliance of the flow estimates on their initial values propagated from the coarser level.

For this task, classical flow initialization is also extended using robust sparse feature matching. It deals more precisely with *SIFT* features [Low04] (replaced by dense nearest neighbor patch matching in [XJM12]). The resulting flow candidates are used to improve the flow field computed in the immediately coarser level. Contrary to [BM11] which takes into account descriptor matching via minimization of Eq. 7.21, [XJM10, XJM12] identify the best match via a labeling problem solved using the *fusion moves* algorithm [LRR08, LRRB10] (Section 7.2.8).

The data term and the regularization term of the *optical flow* model of [XJM10, XJM12] relies respectively on the L^1 norm to reject outliers and on a TV formulation (see Section 7.2.1). The

proposed robust data function deals with both brightness and gradient constancy constraints as in [BBPW04, BM11] but the formulation slightly differs. According to *Xu* et al., a good model should only incorporate the more informative constraint. Consequently, they binarize the use of the two terms employing a binary weight map which switches between brightness or gradient constraints.

7.2.5 Handling illumination changes via texture decomposition

Wedel et al. propose in [WPZ+09] to perform a structure-texture decomposition to be robust to illumination changes. The goal is to decompose the image into a structural part which describes the large objects and a textural part which contains fine scale-details. Performing motion estimation using the textural part does not suffer from shadow and shading reflection artifacts.

7.2.6 Discontinuity-preserving smoothness through filtering heuristics

A significant improvement since early formulations consists in considering a filtering heuristic in order to perform smoothing while preserving discontinuities.

A bilateral (or multilateral) filtering approach can be considered to prevent smoothing across boundaries while still averaging motion vectors within untextured regions. In this context, the original bilateral filter has been introduced by *Tomasi* and *Manduchi* in [TM98]. It can be seen as an alternative to isotropic [ZPB07] or anisotropic image-driven diffusion processes [WTP+09]. Isotropic processes deal with identical properties in all directions. Anisotropic processes consist in averaging each motion vector from neighboring vectors with a kernel size and a kernel shape which depend on local variations.

The original formulation of bilateral filtering [TM98] includes two *Gaussian* functions respectively defined in the spatial and intensity domain. It has been presented first as a way to smooth gray or color images. Using such an approach in the context of motion estimation (i.e. directly filter motion maps) means replacing the motion vector defined at a given pixel with an average motion vector computed using nearby pixel values.

The approach presented in [XCS+06] combines an occlusion detection approach (see Section 7.3) with an adaptive multilateral filtering which extends the original formulation [TM98] for a motion smoothing task. Both tools reach to spatial coherence inside each piecewise-smooth region and keep accurate flow discontinuities at motion boundaries while dealing with the two main issues of classic regularizers. Classic regularizers do not accurately perform motion estimation for occluded regions which generally appear over-smoothed and for large uniform areas whose motion is difficult to compute due to the lack of texture or gradient. These issues can be overcome using parametric models or motion segmentation (see Section 7.2.7) or by combining occlusion detection and adaptive bilateral filtering as in [XCS+06].

[XCS+06] introduces more precisely three *Gaussian* kernels combined with an occlusion function, as shown Eq. 7.23 in a discrete formulation. These *Gaussian* kernels involve information about spatial vicinity (g_s), similarity in terms of intensity (g_i) and motion similarity (g_m).

$$\boldsymbol{u}^{+}(\boldsymbol{x}) = \frac{1}{k(\boldsymbol{x})} \sum_{\boldsymbol{y} \in N(\boldsymbol{x})} g_s(\boldsymbol{x}-\boldsymbol{y}).g_i(I(\boldsymbol{x})-I(\boldsymbol{y})).g_m(\boldsymbol{u}(\boldsymbol{x})-\boldsymbol{u}(\boldsymbol{y})).\nu(\boldsymbol{y}).\boldsymbol{u}(\boldsymbol{y}) \tag{7.23}$$

where $\boldsymbol{u}^+(\boldsymbol{x})$ is the filtered motion vector, $k(\boldsymbol{x})$ a normalization term described in Eq. 7.24, $N(\boldsymbol{x})$ a spatial window defined around $\boldsymbol{x}$, g_s, g_I and g_m, the three *Gaussian* kernels in the spatial, intensity and motion domains. $\nu(\boldsymbol{y})$ corresponds to a binary information of occlusion and equals to 1 if the pixel $\boldsymbol{y}$ is considered as un-occluded in the other frame, 0 otherwise (see Section 7.3 for further details).

$$k(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in N(\boldsymbol{x})} g_s(\boldsymbol{x} - \boldsymbol{y}).g_i(I(\boldsymbol{x}) - I(\boldsymbol{y})).g_m(\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{y})).\nu(\boldsymbol{y}) \tag{7.24}$$

The size σ_s of the spatial kernel g_s can change adaptively with respect to the occlusion information ν and χ, a varied occlusion region radius, as follows:

$$\sigma_s(\nu) = \begin{cases} \sigma_0 \text{ if } \nu = 1 \\ \sigma_0 + \frac{\chi}{3} \text{ if } \nu = 0 \end{cases} \tag{7.25}$$

where σ_0 is a default value and where χ, the occlusion region radius function, is pre-computed for each pixel. In case of occlusion, the kernel size becomes larger than the radius of the occluded region which allows to obtain a relevant motion value while rejecting the influence of the unreliable occluded region and other dissimilar regions according to the intensity and motion similarities.

Related to multilateral filtering, a median filtering approach can be also considered. Thus, in [WPZ+09], a median filtering heuristic is used for each warping and at each level of the pyramid within the coarse-to-fine processing. The analysis of [SRB10] reveals the practical importance of median filtering to denoise motion fields.

7.2.7 Parameterized flow models to estimate rigid motion

Global parameterized models (also called parametric models) can be used to describe the apparent motion using a small finite number of parameters while benefiting from a low computational cost [OB95, BA96]. Involving parametric motion models within techniques for performing image alignment [MAB08] or directly recovering *optical flow* can be judicious. For large uniform areas for instance, using parametric models in the motion estimation task instead of only relying on the regularization process of classical *optical flow* estimators can lead to better results.

A class of parameterized approaches model the *optical flow* with a low-order polynomial on point coordinates. More precisely, it includes:

- constant motion model to deal with translations,
- affine motion model (6 parameters) in order to take into account translations, rotations, scaling or any linear combinations of these transformations,
- homographic (or projective) motion model (8 parameters) which can describe more generally the motion of a 3D rigid plan projected onto the 2D image plane.

Let us take the example of an affine linear transformation, the model most commonly used. For a given pixel $\boldsymbol{x} = (x, y)$, its displacement vector $\boldsymbol{u}(\boldsymbol{x}) = [u(x, y), v(x, y)]$ can be expressed as follows:

$$\begin{cases} u(x,y) = a_0 + a_1(x - x_c) + a_2(y - y_c) \\ v(x,y) = a_3 + a_4(x - x_c) + a_5(y - y_c) \end{cases} \tag{7.26}$$

where $\boldsymbol{a} = [a_0, a_1, a_2, a_3, a_4, a_5]$ corresponds to the vector of parameters and $\boldsymbol{x}_c = (x_c, y_c)$ the center of the image or the center of the current region. Indeed, parametric models can give a representation of the global velocity of the whole image but can also be computed for a single object or area. In this latter case, choosing a region size is a crucial issue and the dilemma between small and large areas is referred to as the *generalized aperture problem* (GAP) [JB93]. Indeed, large regions may be not well modelled by a single parametric model due to the complexity of the motion or the possible presence of multiple motions which can contaminate the single-motion estimate. On the contrary, small regions may not provide enough information for a robust motion estimation.

Applying parametric motion models over arbitrary areas is in general not a good solution. Some papers have investigated this issue like [BJ96] which segments the image into regions using brightness information before computing such models within each region. An alternative solution consists in dividing the image using a *quadtree* decomposition [Sam90] as in [SS96] where Szeliski and Shum describe the motion field as a collection of smoothly connected patches of varying size. These patches of varying size are obtained through a *quadtree* decomposition procedure which consists more precisely in recursively subdividing rectangular areas of the image into four pieces until a criterion of homogeneity is reached.

With the assumption of a parametric model, the energy is no longer defined and minimized with respect to $\boldsymbol{u}$ but with respect to $\boldsymbol{a}$, the vector of parameters of the model.

One way to robustly estimate the parameters of the chosen parametric motion model is to use the *RANSAC* (*RANdom SAmple Consensus*) algorithm (initially introduced in [FB81]) or one of its variants such as [MS04]. Starting from the whole set of *optical flow* vectors belonging to the area of interest, the idea of *RANSAC* is to iteratively select a random subset of the *optical flow* vectors. This subset is used to compute the parameters of a first parametric model which is then tested with the whole set of *optical flow* vectors. For each pixel, we compute the residual error in terms of displacements with respect to the model. The estimated model is considered as relevant if it is possible to find a sufficient number of points whose residual error is beyond a given quality threshold. Otherwise, another model is computed using another subset of *optical flow* vectors randomly chosen. This procedure is repeated a fixed number of times in order to find which model leads to the best consensus.

Another aspect with global parameterized approaches is the fact that multiple motion models can be considered simultaneously using a layered motion estimation scheme [BA96, JBJ96]. In [JBJ96] for instance, *Ju* et al. take fixed regions of the image and assume that the motion within each region can be represented by a small number of affine models which can be considered as *layers*. The motion of each layer is computed using a robust mixture model formulation and each pixel is assigned to one of these *layers*. [JBJ96] tries to find a good balance between local dense *optical flow* computation and the robustness of global parameterized flow models. Consequently, a spatial smoothness constraint on the affine flow parameters of neighboring patches is added in order to enforce the continuity of the motion. In practice, a particular layer of a given region is regularized by taking into account the motion similarity with every layer of the adjacent neighboring regions.

7.2.8 Fusing candidate flows

Instead of relying only on one single motion estimation, an alternative consists in considering different flow proposals before fusing them following an energy minimization strategy (minimum cuts on graphs) as done in [LRR08]. These different flow proposals can be obtained using different flow computation methods. Even if the considered estimators may fail in some regions, the idea is to pool the strengths of each one.

Furthermore, the same estimator can be used several times by modifying its parameter settings. In practice, *Lempitsky* et al. obtain in [LRR08] a large set of flow proposals using the *Lucas-Kanade* and the *Horn-Schunk* formulations with various parameter settings: strength of the regularization, number of levels in the coarse-to-fine hierarchy... Shifted copies in each direction of the resulting flow fields and constant flow fields are also produced and taken into account as input of a fusion procedure.

The fusion procedure makes a choice among all the candidate motion fields by relying on the *fusion flow* algorithm [LRR08, LRRB10]. The *fusion flow* algorithm consists in merging candidate motion fields pair by pair using *fusion moves*, up to obtain the optimal one.

7.2.9 Video signal reconstruction

If we focus only on the problem of colour video signal representation/reconstruction, we can mention impressive dedicated algorithms which has been recently proposed in the literature: *SIFT-Flow* [LYT+08], *PatchMatch* [BSFG09], *Coherency Sensitive Hashing* [KA11]. These methods do not compute the *optical flow* but aim at establishing dense patch/feature correspondences using tools such as *Nearest-Neighbor Field* (*NNF*) mapping to match two images [BSFG09]. *SIFT-Flow* for instance adopts a computational framework of *optical flow* estimation but in order to match *SIFT* descriptors [Low04].

7.3 Occlusion-aware *optical flow*

The occlusion detection task aims at identifying which pixels in the first frame I_n become occluded in the second frame I_{n+1}. Two reasons can explain an occlusion. First, a given object surface in the scene observed in I_n can go outside the field of view of I_{n+1}. Second, a given object surface observed in I_n can be occluded in I_{n+1} by a foreground surface which moves between the camera and the surface under consideration. Consequently, the pixels corresponding to the object surface in the first frame I_n have no correspondence in the second frame I_{n+1}. In both cases, we say that these pixels are occluded in I_{n+1}.

Generally, the motion of such occluded pixels cannot be satisfactorily estimated via a method that does not explicitly take into account this case. A relevant *optical flow* estimation requires a robust occlusion detection in order to identify exactly which areas of the first image are present in the second image. On the contrary, the detection of occluded areas requires accurate motion vectors to be well estimated. Both motion estimation and occlusion detection processes feed each other which explains why many *optical flow* estimators are occlusion-aware. This means that they estimate both motion vectors and occluded pixels.

In what follows, we give a very brief overview of the existing occlusion detection techniques. Note that we do not consider here the detection of occlusion boundaries which are distinct from occlusion regions and beyond the scope of our study.

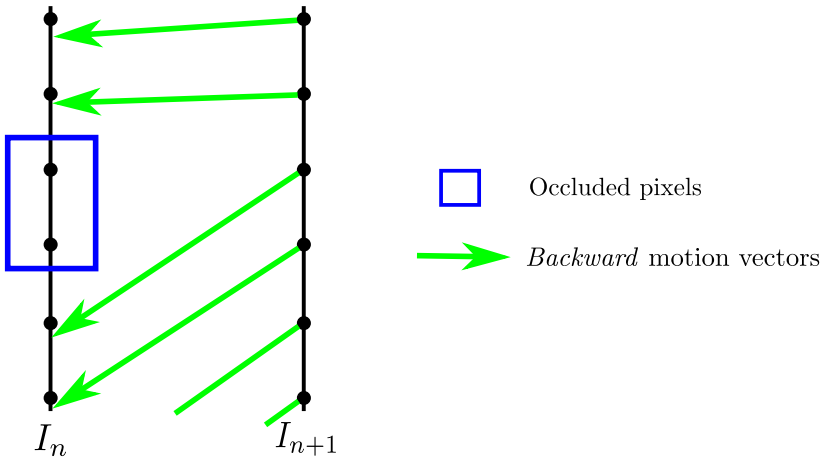


Figure 7.1: Occlusion detection

Some papers combine both motion estimation and occlusion detection by adding explicitly an occlusion penalty term into the formulation [ADPS02, XCS+06, XJM10, XJM12]. Let us focus on [XCS+06] to see how it works. In this paper, *Xiao* et al. propose a simple occlusion criterion based on the square image residual:

$$\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = \begin{cases} 0 \text{ if } [I_n(\boldsymbol{x}) - I_{n+1}(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}))]^2 > \epsilon_{occ} \\ 1 \text{ otherwise} \end{cases} \tag{7.27}$$

where ϵ_{occ} is a threshold to decide the occlusion. $\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = 0$ means that the pixel $\boldsymbol{x}$ of I_n is occluded in I_{n+1}. $\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = 1$ describes the situation where $\boldsymbol{x}$ is visible both in I_n and I_{n+1}. Note that $\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))$ in Eq. 7.27 and $\nu(\boldsymbol{x})$ in Eq. 7.23 and Eq. 7.24 exactly denotes the same feature. A smooth approximation of the *Heavyside* function is then applied to this square image residual in order to obtain a function $\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))$ which becomes continuous:

$$\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) = \frac{1}{2} + \frac{1}{\pi} \tan^{-1}((I_n(\boldsymbol{x}) - I_{n+1}(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})))^2 - \epsilon_{occ}) \tag{7.28}$$

The resulting continuous occlusion information $\nu(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x}))$ (or simply denoted $\nu(\boldsymbol{u})$) is then directly added to the energy function in order to both exclude the occluded pixels from the minimization and add a penalty term which balances between occlusion and visibility. Therefore, the energy function becomes:

$$E(\boldsymbol{u}) = [E_{data}(\boldsymbol{u}) + E_{reg}(\boldsymbol{u})].\nu(\boldsymbol{u}) + [E^{occ}_{data}(\boldsymbol{u}) + E^{occ}_{reg}(\boldsymbol{u})].(1 - \nu(\boldsymbol{u})) \tag{7.29}$$

where E^{occ}_{data} and E^{occ}_{reg} respectively deal with an occlusion energy term and a motion regularization term dedicated to occluded pixels. Moreover, the multilateral filtering described in Eq. 7.23 also incorporates this occlusion information in order to reject the influence of occluded areas.

Humayun et al. describe in [HMAB11] many other ways to perform occlusion detection, without considering how motion estimation and occlusion detection can be combined. Through a *Random Forest* based framework dedicated to feature selection and training, they study the robustness of around 20 different features (with several different parameter settings for each one). More precisely, they investigate the ability of each feature to correctly predict occlusion

regions with respect to ground-truth occlusion data obtained with the dataset from [MABP10]. In the context where motion estimation is performed bi-directionally, i.e. from I_n to I_{n+1} (i.e. in the *forward* direction) and from I_{n+1} to I_n (i.e. in the *backward* direction), reverse flow features such as the *forward-backward* consistency value (as used in [ADPS02]) or the *forward-backward* flow angle consistency value are described as the most robust tools to identify occlusions. These two features follow the idea that the *backward* flow should be the exact opposite of the *forward* flow. Among all the other tested features, features based on the square image residual appear to be also efficient according to the results. Features such as the gradient of the median flow or the variance of motion vector length using multiple *optical flow* estimators have proved to give useful information for the occlusion detection task.

As described in Fig. 7.1, the occlusion detector of [RTDC12] consists in projecting the *backward* motion vectors estimated from I_{n+1} to I_n and then, to identify for each ending point of these vectors the closest pixel in I_n. At the end of this visibility detection step, the pixels in I_n which have not been marked are considered as occluded in I_{n+1}. Adapted from the occlusion constraint (OCC) method described in [EW02], this approach assumes that motion estimation is performed bi-directionally.

7.4 Conclusion

Starting from the same color constancy assumption, state-of-the-art *optical flow* estimators involve many different strategies, numerical solutions and robust functions and may rely on tools such as sparse feature correspondences, parameterized flow models, multilateral or median filtering heuristics to robustify as much as possible the dense motion estimates. Additionally to these resulting motion fields, occlusion information can help to describe more precisely how the scene content behave temporally.

Despite significant progress from early formulations, improvements are still under study since estimating the flow for real scenes is a very difficult task. This is particularly the case when the considered scenes include non-rigid deformations, large motion, zooming, poorly textured areas, transparency, occlusions or illumination changes.

In addition, our study has mainly focused on motion estimation between two consecutive frames up to now. However, there are applications for which long-term dense correspondences are better suited, if not required. In this context, the following chapter, Chapter 8, describes the concept of long-term motion estimation and especially introduces how *optical flow* can be used to reach long-term requirements.

Chapter 8

From *optical flow* to long-term motion estimation

We studied in the previous chapter how recent *optical flow* estimators have focused on both ensuring the robustness under noise and improving the spatial consistency of the flow by introducing respectively more efficient data models than the single classical *brightness constancy constraint* and discontinuity-preserving smoothness constraints.

In practice, most of the best known *optical flow* estimation methods only focus on estimating motion between two consecutive frames. These methods totally ignore that the sequence comprises numerous images which are in fact inter-related. In addition, some applications including scene segmentation methods [BM10, LASL11], analysis techniques [WKSL11] and a number of automatic and semi-automatic video editing tasks such as graphic elements insertion [RAKRF08] or 2D-to-3D video conversion [CLD11] need to reach dense and long-term requirements.

In the context of sparse point tracking, numerous *feature-based tracking* (or more simply *feature tracking*) algorithms have been proposed to compute sparse long-term trajectories across video sequences. The standard tool for this task is the *Kanade-Lucas-Tomasi* (KLT) tracker [LK81, TK91, ST94] which consists in tracking a sparse set of salient points based on their distinctive appearance. Alternative techniques consist in extracting features in the considered frames and in using *SIFT* (*Scale Invariant Featyre Transform*) [Low04], *SURF* (*Speeded Up Robust Features*) [BTVG06] or another descriptor for matching. In practice, sparse motion estimation generally focuses on around a few hundred of points. In the same spirit, object-based motion estimation computes the temporal dynamic of one or several objects along video sequences via a bounding box which is tracked using algorithms such as particle filter for instance [PHVG02].

Although object-based or sparse motion estimation is sometimes sufficient, some applications such as the ones previously described explicitly require a dense and long-term motion estimation. Therefore, we focus on the following problem: how to construct dense fields of point correspondences over extended time periods?

Establishing long-term correspondences finally translates in computing motion between distant frames and therefore in handling simultaneously small and large or very large displacements. *Optical flow* estimation algorithms can be seen as a tool in order to perform the long-term motion estimation task. However, classical *optical flow* assumptions which may fail between consecutive frames are even less valid between non-consecutive frames. This is especially true for complex scenes whose content changes significantly in time. Not only the *brightness constancy assump-*

tion becomes more critical as the distance increases between the input frames. Occluded areas become wider and therefore more difficult to be detected.

When dealing with multiple frames, another key aspect which instantly comes to mind is the temporal consistency of the flow which must ensure a certain temporal smoothing along trajectories. This aspect can provide additional constraints to deal with the aperture problem.

In this context, we study in Section 8.1 how the literature has extended *optical flow* to the purpose of long-term motion estimation. Applications for which long-term temporal consistency is key are then reviewed in Section 8.2.

8.1 Towards long-term motion estimation

In this section, we review the state-of-the-art of long-term motion estimation by first introducing how long-term point trajectories can be obtained by simply concatenating consecutive *optical flow* fields (Section 8.1.1). Then, we describe the extension of the classical *optical flow* formulation from two-frames to multi-frames using trajectorial regularization constraints (Section 8.1.2). In Section 8.1.3, we study sophisticated particle representations to perform semi-dense trajectory estimation. Finally, we describe how subspace constraints can be used to compute long-term trajectories even in the presence of strong deformations (Section 8.1.4).

8.1.1 Straightforward temporal integration

Optical flows estimated between consecutive frames can straightforwardly be concatenated to describe motion trajectories along video sequences. Numerically, this amounts to temporal integration, for which tools such as *Euler* and *Runge-Kutta* integration schemes are available [But08]. Let us describe more precisely how this straightforward integration approach can lead to dense pixel trajectories.

The context of our study is as follows. We consider a sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\dots,N]\!]}$ including I_{ref} considered as a reference frame. In this section, we focus on point tracking starting from the grid point $\boldsymbol{x}_{ref}$ belonging to I_{ref}. Point tracking consists in computing the set of *forward* long-term displacement vectors $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ $\forall n \in [\![0,\dots,N]\!]$ with $I_n \neq I_{ref}$. Each displacement vector $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ connects the grid point $\boldsymbol{x}_{ref} \in I_{ref}$ to a position in I_n (not necessarily a grid-position). The set of displacement vectors $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ defines the trajectory of $\boldsymbol{x}_{ref}$, referred to as $\boldsymbol{T}(\boldsymbol{x}_{ref})$, across the sequence.

Let us study in particular how the trajectory of $\boldsymbol{x}_{ref}$ can be computed along the sequence using a set of *forward optical flow* fields $\boldsymbol{u}_{n,n+1}$ $\forall n \in [\![0,\dots,N-1]\!]$ computed between consecutive frames, commonly called consecutive *optical flow* fields. We assume that this set of consecutive *optical flow* fields has been pre-computed by any of the estimators described in Chapter 7 (Fig. 8.1 (a)).

In terms of notation, we make a difference between $\boldsymbol{u}_{n,n+1} = [u_{n,n+1}, v_{n,n+1}]^t$ which corresponds to an elementary *optical flow* field computed between the consecutive frames I_n and I_{n+1} and $\boldsymbol{d}_{ref,n} = [dx_{ref,n}, dy_{ref,n}]^t$ which denotes a long-term displacement field between I_{ref} and I_n. In addition, we assume in the following that the reference frame corresponds to the first frame of the sequence (i.e. $I_{ref} = I_0$) in order to simplify both explanations and notations.

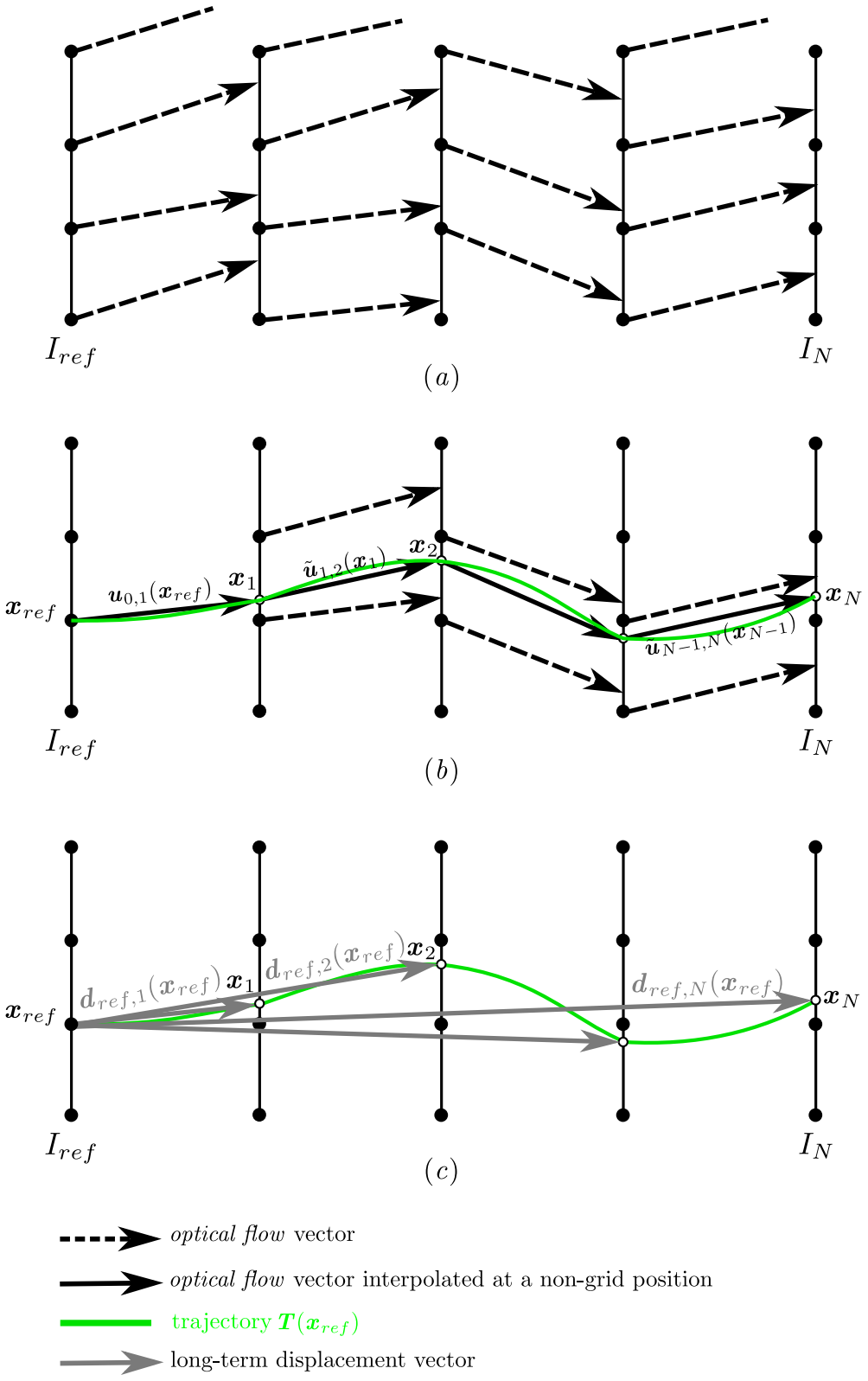


Figure 8.1: Estimation of $\boldsymbol{T}(\boldsymbol{x}_{ref})$, the motion trajectory starting from $\boldsymbol{x}_{ref}$ in the reference frame I_{ref} (b) through integration of *optical flow* fields $\boldsymbol{u}_{n,n+1}$ $\forall n \in [\![0, \ldots, N-1]\!]$ pre-computed between consecutive frames (a). Estimating $\boldsymbol{T}(\boldsymbol{x}_{ref})$ consists in computing the set of *forward* long-term displacement vectors $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ $\forall n \in [\![0, \ldots, N]\!]$ with $I_n \neq I_{ref}$ (c).

Starting from $\boldsymbol{x}_{ref} \in I_{ref}$, $\boldsymbol{T}(\boldsymbol{x}_{ref}) = \{\boldsymbol{x}_0 = \boldsymbol{x}_{ref}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ can be obtained through iterative integration of consecutive *optical flow* fields $\boldsymbol{u}_{n,n+1}$ following Eq. 8.1. In practice, each *optical flow* propagates the current position of the trajectory into the next frame. Therefore, $\boldsymbol{T}(\boldsymbol{x}_{ref})$ is built sequentially from frame to frame, i.e. for $n = 1, 2, \ldots, N$, until I_N is reached (Fig. 8.1 (b)).

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \tilde{\boldsymbol{u}}_{n,n+1}(\boldsymbol{x}_n) \tag{8.1}$$

where the notation $\tilde{.}$ indicates a displacement field probably interpolated at a non-grid position. Indeed, an interpolation is required when $\boldsymbol{x}_n$ does not reach a grid point.

In terms of long-term displacement fields, obtaining the long-term displacement vector $\boldsymbol{d}_{ref,n+1}(\boldsymbol{x}_{ref}) = \boldsymbol{x}_{n+1} - \boldsymbol{x}_{ref}$ (Fig. 8.1 (c)) $\forall n \in [\![1, \ldots, N-1]\!]$ translates in the iterative estimation displayed in Eq. 8.2:

$$\begin{aligned} \boldsymbol{d}_{ref,n+1}(\boldsymbol{x}_{ref}) &= \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}) + (\boldsymbol{x}_{n+1} - \boldsymbol{x}_n) \\ &= \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}) + \tilde{\boldsymbol{u}}_{n,n+1}(\boldsymbol{x}_n) \end{aligned} \tag{8.2}$$

where $\boldsymbol{x}_{n+1}$ is obtained thanks to $\boldsymbol{u}_{n,n+1}$. This iterative procedure can be repeated for each pixel $\boldsymbol{x}_{ref}$ of I_{ref} in order to obtain the set of dense trajectories starting from I_{ref}.

Such a straightforward way to integrate *optical flow* fields has been exploited in several works including [CMP02, BM10, SBK10, WKSL11, WKSL13]. In [CMP02], trajectories are computed from *optical flows* in the context of fluid flow estimation with a 4-th order *Runge-Kutta* integration scheme. In [BM10], *Brox* and *Malik* obtain temporally consistent segmentation of moving objects using long-term trajectories computed through integration of elementary *optical flows*. [SBK10] exploits this concept to compute trajectories starting from *optical flow* estimations made by a fast GPU implementation of LDOF (*Large Displacement Optical Flow*) presented in [BM11] (see Section 9.3.4). Finally, [WKSL11, WKSL13] perform such accumulation scheme to describe videos and especially recognize action using dense trajectories.

This simple scheme can be combined with a more sophisticated global formulation for track estimation, as in [ST06, ST08] (described in Section 8.1.3) which performs a semi-dense trajectory estimation through a particle representation.

Simple concatenation of consecutive *optical flow* fields may lead to large error accumulation that can result in a substantial drift over extended periods of time. Results in the literature are generally reported on fairly short sequences and reliable tracks usually do not exceed thirty frames. [CFC+14] studies in details why this classical integration sooner or later fails and especially deals with sources of motion drift inherent to both estimation and interpolation processes. This includes the motion biais which is due to the fact that motion estimation may be performed at positions which are different from the true ones due to motion drift.

Finally, both estimation and accumulation errors result in dense trajectories which can rapidly diverge and become inconsistent after some time, especially for complex scenes. Another important drawback of this classical integration approach deals with the fact that concatenating motion fields computed between consecutive frames does not allow to recover trajectories after temporary occlusions. Trajectories are stopped once the occlusion occurs.

8.1.2 Multi-frame *optical flow* estimation using trajectorial regularization

To limit the motion drift, the *optical flow* estimation process has been extended from two frames to multi-frames. Multi-frame *optical flow* estimation frameworks generally deal with soft or hard spatio-temporal constraints which enable an overall smoothing. Although far away from the previously described consecutive *optical flow* integration scheme, we study in this section the most related papers in the field of temporally consistent multi-frame *optical flow* estimation. We more particularly focus on how these methods incorporate temporal information in their underlying models.

Historically, early works in this domain have considered straightforward temporal smoothing approaches, like in [MB87] which involves a temporal smoothing following the temporal axis. This method has been extended by [Nag90] with directional smoothness assumptions formulated in the spatio-temporal domain in order to take into account displacements which do not necessarily follow the temporal axis. Then, 3-frames estimation methods, including [WTP+09], have been introduced with robust spatio-temporal formulations symmetric with respect to previous and next frames. [WTP+09] extends the classic spatial anisotropic regularization to the computation of a flow symmetric with respect to the central frame. The proposed anisotropic regularization considers two *brightness constancy constraints*: one between the previous frame I_{n-1} and the current frame I_n and another between the next frame I_{n+1} and the current frame I_n. This symmetric constraint involves a model which implies a linear motion from I_{n-1} to I_{n+1} $\forall n$. In this context, we can also mention [WS01, BBPW04] whose regularization is done with respect to spatial and temporal derivatives treated in the same manner (see Eq. 7.14 in Section 7.2.1, Chapter 7).

The temporal or spatio-temporal smoothing constraints involved in the previously described algorithms fail with large and sudden displacements since the temporal derivatives in the smoothness term are not able to model such motion. Such smoothing constraints suppose a continuity of the flow and are therefore in contradiction with the data terms which can tolerate discontinuities in time. Moreover, the spatial and temporal components are generally coupled which can be an issue with large displacement given the fact that large temporal estimates can inhibit spatial regularization. To overcome these two main issues, *Salgado* and *Sanchez* has designed in [SS07] a new spatio-temporal regularizer which explicitly focuses on large displacements. In [SS07], spatial and temporal regularizers are separated. In addition, the *optical flow* vector $\boldsymbol{u}_{n,n+1}(\boldsymbol{x})$ starting from $\boldsymbol{x}$ of I_n is assumed to be similar to the *optical flow* vector $\boldsymbol{u}_{n+1,n+2}$ starting from $\boldsymbol{x} + \boldsymbol{u}_{n,n+1}(\boldsymbol{x})$. Therefore, instead of taking into account the classic formulation $\phi(|\nabla_3 u|^2 + |\nabla_3 v|^2)$, the temporal regularizer E_{t_reg} of [SS07] extends the *optical flow* constancy constraint to the whole sequence which gives:

$$E_{t\_reg} = \sum_{n=0}^{N-1} \int_{\Omega} \phi_t(||\boldsymbol{u}_{n,n+1}(\boldsymbol{x}) - \boldsymbol{u}_{n+1,n+2}(\boldsymbol{x} + \boldsymbol{u}_{n,n+1}(\boldsymbol{x}))||^2) d\Omega \tag{8.3}$$

where $\phi_t(.)$ is the *Charbonnier* penalizer. Contrary to [WTP+09] which enforces the trajectorial constancy (i.e. the regularization along trajectories) as a hard constraint, this first-order penalization of motion variations along trajectories acts as a soft constraint. Note that it approximates the temporal derivative for very small displacement. Finally, this trajectorial regularization is turned symmetrically to include also *backward optical flows*, as in [WTP+09].

Such trajectorial regularization can oversmooth complex motion. Consequently, [VBVZ11] has proposed to combine the two following assumptions: 1) assumption of a temporally coher-

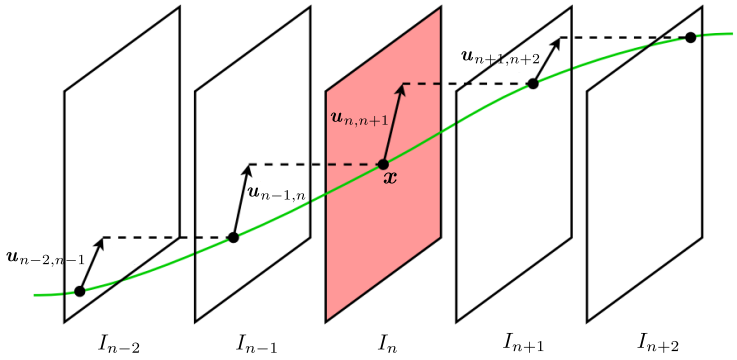


Figure 8.2: Temporally consistent multi-frame *optical flow* estimation proposed in [VBVZ11]: simultaneous computation of $\boldsymbol{u}_{n,n+1}$ starting from $\boldsymbol{x} \in I_n$ as well as $\boldsymbol{u}_{n-2,n-1}$, $\boldsymbol{u}_{n-1,n}$ and $\boldsymbol{u}_{n+1,n+2}$ starting from the positions which belong to the trajectory of $\boldsymbol{x}$, respectively in I_{n-2}, I_{n-1} and I_{n+1}.

ent spatial flow structure which is considered by expanding spatial regularization over multiple frames (five or more), 2) assumption that the *optical flow* must be smooth along motion trajectories. This latter assumption leads to two trajectorial smoothness terms (first- and second-order) with an adaptive regularization degree which can relax one or both terms to avoid oversmoothing. As illustrated in the temporal scenario made of five frames $\{I_{n-2}, \ldots, I_{n+2}\}$ displayed in Fig. 8.2, the multi-frame *optical flow* method of *Volz* et al. in [VBVZ11] consists in minimizing an energy which simultaneously:

- computes and refines the *optical flow* vector $\boldsymbol{u}_{n,n+1}$ starting from the grid point $\boldsymbol{x}$ of I_n, the center frame of the temporal window,

- estimates $\boldsymbol{u}_{n-2,n-1}$, $\boldsymbol{u}_{n-1,n}$ and $\boldsymbol{u}_{n+1,n+2}$ starting from the positions which belong to the trajectory of $\boldsymbol{x}$, respectively in I_{n-2}, I_{n-1} and I_{n+1}.

I_n can be seen as a reference frame since all the *optical flow* vectors to be defined are parameterized with respect to I_n in order to refer to the same coordinate system. The functional involved in [VBVZ11] (E detailed in Eq. 8.4) is made of three terms which account for data constraints (ϵ_{data}), spatial regularization (ϵ_{reg}) and adaptive trajectorial regularization (ϵ_t) which is decoupled from the spatial regularization, as in [SS07].

$$E = \int_{\Omega} (\epsilon_{data} + \epsilon_{reg} + \epsilon_t)\, d\Omega \tag{8.4}$$

The data term ϵ_{data} extends the *brightness constancy constraint* to the purpose of multi-frame *optical flow* estimation (Eq. 8.5). Therefore, the brightness differences between each pair of consecutive positions along the trajectory of $\boldsymbol{x}$ (within the five-frame temporal window) are minimized through the four constraints displayed in Eq. 8.6.

$$\epsilon_{data} = \epsilon_{n-2,n-1} + \epsilon_{n-1,n} + \epsilon_{n,n+1} + \epsilon_{n+1,n+2} \tag{8.5}$$

$$\begin{aligned}
\epsilon_{n-2,n-1} &= \theta.\phi(|I_{n-1}(\boldsymbol{x}-\boldsymbol{u}_{n,n-1}) - I_{n-2}(\boldsymbol{x}-\boldsymbol{u}_{n,n-1}-\boldsymbol{u}_{n-1,n-2})|^2) \\
\epsilon_{n-1,n} &= \phi(|I_n(\boldsymbol{x}) - I_{n-1}(\boldsymbol{x}-\boldsymbol{u}_{n,n-1})|^2) \\
\epsilon_{n,n+1} &= \phi(|I_{n+1}(\boldsymbol{x}+\boldsymbol{u}_{n,n+1}) - I_n(\boldsymbol{x})|^2) \\
\epsilon_{n+1,n+2} &= \theta.\phi(|I_{n+2}(\boldsymbol{x}+\boldsymbol{u}_{n,n+1}+\boldsymbol{u}_{n+1,n+2}) - I_{n+1}(\boldsymbol{x}+\boldsymbol{u}_{n,n+1})|^2)
\end{aligned} \tag{8.6}$$

where $\theta = 0.5$ allows to weight the influence of constraints that have a larger temporal distance to the reference frame I_n. In addition, $\phi(.)$ denotes the robust function introduced by *Brox* et al. in [BBPW04] (see Section 9.3.1). These previous constraints (Eq. 8.6) are also extended to spatial gradient values following the gradient constancy assumption introduced in [BBPW04].

The spatial regularization (ϵ_{reg}) of [VBVZ11] is achieved via an anisotropic smoothness term which relaxes the smoothness assumption in direction of the data constraints to limit possible interferences between both data and spatial smoothness terms.

Finally, the trajectorial regularization (ϵ_t) involves both first- and second-order regularizers, ϵ_t^{1st} Eq. 8.7 (in the same spirit of Eq. 9.3 [SS07]) and ϵ_t^{2nd} Eq. 8.8, in order to encourage temporal smoothness along trajectories. Assuming $\boldsymbol{u}_{i,i+1} = [u_{i,i+1}, v_{i,i+1}]^t$ with $i = n-2, \ldots, n+1$, *Volz* et al. [VBVZ11] propose:

$$\epsilon_t^{1st} = \sum_{i=n-2}^{n} \phi_t([u_{i+1,i+2} - u_{i,i+1}]^2 + [v_{i+1,i+2} - v_{i,i+1}]^2) \tag{8.7}$$

$$\epsilon_t^{2nd} = \sum_{i=n-1}^{n} \phi_t([u_{i+1,i+2} - 2u_{i,i+1} + u_{i-1,i}]^2 + [v_{i+1,i+2} - 2v_{i,i+1} + v_{i-1,i}]^2) \tag{8.8}$$

where $\phi_t(.)$ is the *Charbonnier* penalizer, as in [SS07]. We can notice that ϵ_t^{1st} assumes a piecewise smooth *optical flow* along the trajectory whereas ϵ_t^{2nd} models piecewise linearly smooth *optical flow* in the trajectorial direction.

Following adaptive schemes such as the one in [XJM10, XJM12] which chooses between brightness or gradient constancy constraint (see Section 9.3.4), [VBVZ11] proposes to adapt the degree of regularization in order to:

- strongly enforce the trajectorial regularization if it is known that the flow follows a certain trajectorial motion model,
- avoid oversmoothing if the flow does not fulfil the trajectorial smoothness model.

For this task, a parabola is fitted to each component (x and y) of the flow along the trajectories using an iterative reweighted least squares formulation. The coefficients of the two computed parabolas $\{a_x, b_x, c_x\}$ and $\{a_y, b_y, c_y\}$ infer the required degree of regularization. Assuming that T_a and T_b are two pre-defined thresholds, three situations are finally considered:

- no trajectorial regularization when $max(|a_x|, |a_y|) > T_a$,
- only 2nd-order trajectorial regularization if $max(|a_x|, |a_y|) \leq T_a$ and $max(|b_x|, |b_y|) > T_b$,
- only 1st-order trajectorial regularization if $max(|a_x|, |a_y|) \leq T_a$ and $max(|b_x|, |b_y|) \leq T_b$.

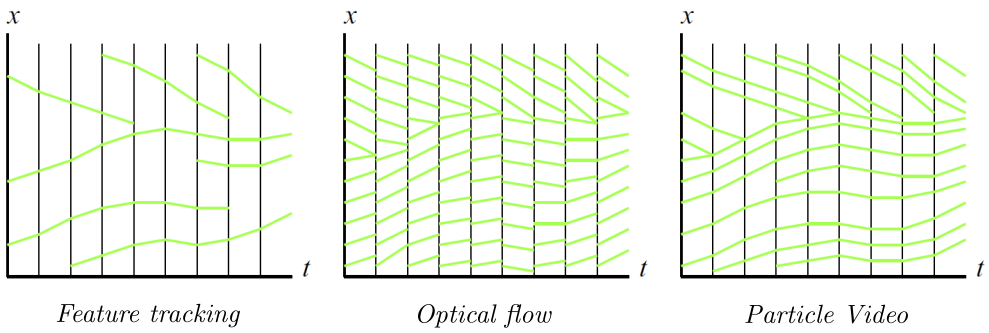


Figure 8.3: The *Particle Video* representation [ST06, ST08] combines both *optical flow* and *feature tracking* approaches to produce semi-dense trajectories across video sequences.

Despite significant progresses in the field of multi-frames *optical flow* formulation based on trajectorial regularization, more sophisticated motion models must be investigated in order to deal with complex motion. Let us describe how a particle representation can represent motion in a non-parametric manner.

8.1.3 Long-range motion estimation through particle trajectories

In [ST06, ST08], *Sand* and *Teller* propose to represent motion using a set of particles that move across the video sequence. The proposed framework is called *Particle Video*. It computes variable-length point trajectories from *optical flow* fields computed between consecutive frames. Using a particle representation allows to forsake rigidity assumptions and motion model considerations which may fail in complex situations.

The primary goal of *Sand* and *Teller* is to combine both *optical flow* and *feature tracking* approaches to produce semi-dense and long-range trajectories (see Fig. 8.3). *Particle Video* relies on *optical flow* estimation to sequentially propagate particles across the sequence and to obtain a semi-dense presentation which is able to model details with fewer particles than pixels. *Feature tracking* approaches are exploited for their long-term tracking aspects.

As shown in Fig. 8.4, *Particle Video* is a sophisticated framework made of five steps: propagation, linking, optimization, pruning and addition. Let us describe how these steps work precisely.

Propagation step. Particles which belong to adjacent frames are propagated to the current frame using *forward* (or *backward*) previously computed *optical flow*. This first step gives an initial location for each particle in a given frame. This initial particle location will be refined during the optimization step. In the *forward* direction for instance, the particle i located in $\boldsymbol{x}_i(t) = (x_i(t), y_i(t))$ at instant t is simply propagated to $\boldsymbol{x}_i(t+1)$ at instant $t+1$ using the *optical flow* vector $\boldsymbol{u}(\boldsymbol{x}_i(t), t) = [u_i(t), v_i(t)]^t$ (as in Section 8.1.1):

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{u}(\boldsymbol{x}_i(t), t) \tag{8.9}$$

The *optical flow* estimator used in [ST06, ST08] is occlusion-aware and embedded in a multi-resolution scheme. The propagation does not occur when the corresponding *optical flow* vector indicates an occlusion. In addition, the *optical flow* estimator is based on the *HAOF* (*High*

Accuracy Optical Flow) method [BBPW04] (see Section 9.3.1) extended using the multilateral filtering proposed in [XCS+06] (see Section 7.2.6).

Linking step. Links between spatially neighboring particles are firstly established through *Delaunay* triangulation. Then, particle links are considered by taking into account both this spatial *Delaunay* connectivity and the temporal link which connects particles via *optical flow* vectors. Such particle link is created only if the corresponding triangulation edge exists for the current frame or one of the adjacent frames. Finally, a weight l_{ij} is assigned for each of these resulting particle links based on the trajectory similarity $D(i,j)$, as described in Eq. 8.10 where $\mathcal{N}(.)$ is a zero-mean *Gaussian* function with standard deviation σ_l. $D(i,j)$, detailed in Eq. 8.11, computes a trajectory similarity between the trajectories of particles i and j through mean squared motion differences estimated within the temporal window T (composed of $|T|$ frames). The link between two particles whose trajectories have a similar behavior receives a large weight. On the contrary, the situation where two particles are separated by an occlusion boundary gives a weight near zero.

$$l_{ij} = \mathcal{N}(\sqrt{D(i,j)}; \sigma_l) \tag{8.10}$$

$$D(i,j) = \frac{1}{|T|} \sum_{t \in T} [u_i(t) - u_j(t)]^2 + [v_i(t) - v_j(t)]^2 \tag{8.11}$$

Optimization step. The initial particle positions obtained during the propagation step are refined via a particle optimization process. The idea is to minimize an energy functional made of both data and regularization terms which are computed simultaneously over the F frames of the sequence and for the P identified particles:

$$E = \sum_{t \in F, i \in P} E(i,t) \text{ with } E(i,t) = \sum_{k \in K_i(t)} \epsilon^k_{data}(i,t) + \alpha \sum_{j \in L_i(t)} \epsilon_{reg}(i,j,t) \tag{8.12}$$

where $L_i(t)$ corresponds to the particles linked to particle i at instant t. The computation of the energy involves five channels: image brightness, green minus red channel, green minus blue channel, x gradient and y gradient. k corresponds to the channel index and $K_i(t)$ is the set of active channels. α makes the balance between both terms.

The data term involves temporal appearance smoothness assuming that the particle appearance must change slowly across the sequence. Let $I^k(x_i(t), y_i(t), t)$ be the particle's appearance for the kth channel. *Sand* and *Teller* propose to compute a temporally filtered version $\hat{I}^k(x_i(t), y_i(t), t)$ in order to finally assess the quality of the current particle i by comparing this filtering version to $I^k(x_i(t), y_i(t), t)$ (with $\phi(.)$ as a robust function):

$$\epsilon^k_{data}(i,t) = \phi([I^k(x_i(t), y_i(t), t) - \hat{I}^k(x_i(t), y_i(t), t)]^2) \tag{8.13}$$

The regularization term measures the relative motion of particles modulated by the link weight l_{ij} computed during the linking step (Eq. 8.10). This term (see Eq. 8.14) aims at encouraging particles that have moved together to continue moving together:

$$\epsilon^k_{reg}(i,j,t) = l_{ij}\phi([u_i(t) - u_j(t)]^2 + [v_i(t) - v_j(t)]^2) \tag{8.14}$$

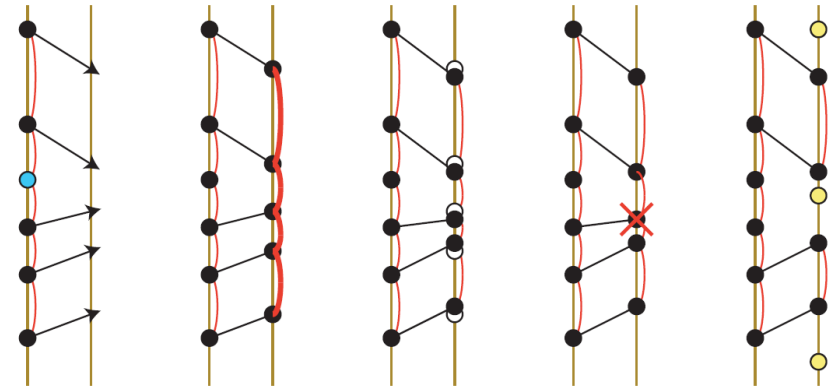


Figure 8.4: The five steps of the *Particle Video* framework [ST06, ST08].

Pruning step. Particles which have a high energy value (with respect to Eq. 8.12) are simply deleted. In practice, large matching error or strong dissimilarities with neighbouring particles may indicate occlusions.

Addition step. This final step aims at adding new particles in gaps between existing particles by considering an adaptative particle density based on the visual complexity. This relies on the fact that motion complexity often implies visual complexity. Indeed, a region with a strong visual complexity must correspond to a large particle density in order to model complex motion.

The visual complexity is quantified as follows. Images are filtered using a *Gaussian* kernel at different scales: $\{\sigma(j) = 1.9^j | 0 \leq j \leq 5\}$. Then, for each pixel, the maximum scale index $k(x, y)$ over which the blurred pixel value does not change substantially is computed. Finally, $\sigma(k(x, y))$ is compared to the distance to the nearest particle. If this distance is greater than $\sigma(k(x, y))$, a new particle is added in the current location. This process is repeated for each pixel.

Particle Video focuses on handling occlusions and more particularly occluding/disoccluding boundaries which are estimated studying the *optical flow* divergence and then refined using a multilateral filtering heuristic similar to [XCS+06]. Although there is a careful reasoning on occlusion and trajectory termination, [ST06, ST08] do not account for temporarily occluded particles. Moreover, using *Particle Video* means renouncing to the full tracking density contrary to what can be required for some applications, as described in Section 8.2.

As noted by *Rubinstein* et al. in [RLF12], particle trajectories from [ST06, ST08] can be lost due to occlusion, mis-tracking or camera motion. In this case, new particles can be generated by the addition stage to restart the tracking. Thus, the displacement of a single physical point can be represented by several particle trajectories. The same scenario occurs in case of temporary occlusion: a first trajectory terminates when the occlusion occurs while a second one is generated once the corresponding point reappears. Starting for this finding, [RLF12] proposes to re-correlate these short-term trajectories (also called *tracklets*) in order to construct long-term trajectories (Fig. 8.5).

For the trajectory linking task, three constraints are taken into account: 1) linked trajectories must encode the same scene feature, 2) each query trajectory (trajectory to merge

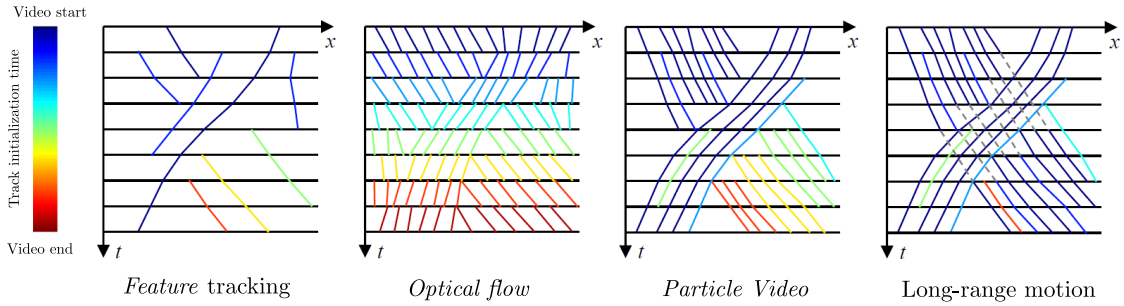


Figure 8.5: [RLF12] extends the *Particle Video* representation to produce long-range semi-dense trajectories across video sequences.

with another) and candidate trajectory must be merged with at most one trajectory, 3) spatio-temporally neighboring trajectories must be associated with neighboring candidate trajectories. This combinatorial assignment problem is defined and optimized globally over the entire sequence through *belief propagation*. In particular, it involves three trajectory compatibility components: appearance similarity (based on similarity between *SIFT* features), motion similarity (computed with respect to the end and start velocities) and a prior on the feature's motion while unobserved which predicts via a constant model the trajectory position while occluded. Moreover, the formulation includes a link regularization which enforces the spatio-temporal smoothing while penalizing links that cross trajectories behind occluders.

8.1.4 Multi-frame *optical flow* estimation using subspace constraints

Let us come back to dense estimation by describing an alternative way to perform long-term dense motion estimation: multi-frame *optical flow* estimation methods using subspace constraints. Starting from the fact that trajectories of points which belong to an object are very correlated (even with strong deformations), this type of methods assumes that the set of all flow fields reside in a low-dimensional subspace. Therefore, the related works deal with a low-rank space which is built to constrain the *optical flow* estimation process. Moreover, these constraints provide additional information which allows to resolve the ambiguity in regions that suffer from the aperture problem. Let us review the most directly related research.

Historically, low-dimensional subspaces have been used for recovering 3D information from known 2D correspondences [TK92]. We can quote here also the non-rigid low-rank shape model introduced in [BHB00] which is used to express the 3D shape of an non-rigid object as a linear combination of a low-rank shape basis. In the context of low-dimensional subspace for multi-frame motion estimation, a straightforward approach consists in projecting the set of inter-frame *optical flow* fields (obtained with any existing *optical flow* estimator) into a low-dimensional subspace. However, as described in Chapter 7, the *optical flow* estimation process is under-constrained and therefore can result in erroneous motion vectors. Depending on the overall quality of the motion estimation process, the computation of the low-dimensional space may be corrupted by these outliers. With such classical approach, *optical flow* vectors are taken into account with the same weight, without considering any intrinsic vector quality to balance their relative importance.

The first major contributions in the context of multi-frame *optical flow* estimation using subspace constraints have been introduced in [Ira99, Ira02]. Their method has been presented

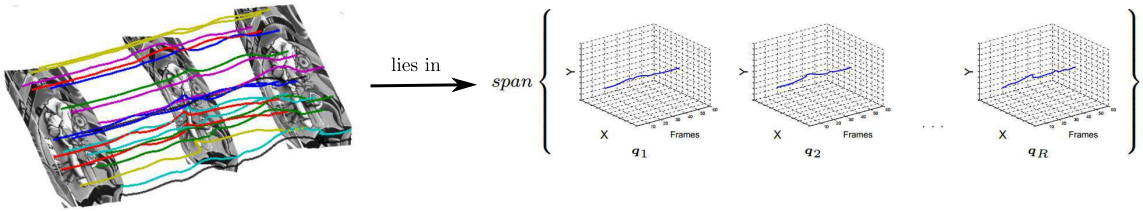


Figure 8.6: Multi-frame *optical flow* estimation using 2D trajectory subspace constraints [GRA11b]. Trajectories are estimated along the sequence assuming that they must lie close to a low-dimensional subspace which implicitly acts as a long-term temporal regularization.

as an alternative to *heuristic* constraints such as spatio-temporal smoothness (described in Section 8.1.2). Multi-frame linear subspace constraints are used to guide the 2D correspondence estimation process itself, without following a two-step approach (inter-frame *optical flow* fields estimation and then projection of the resulting displacements) as for straightforward approaches. More precisely, *Irani* proposes to apply low-dimensionality subspace constraints directly on measurable image quantities (such as brightness) and not on displacements.

In particular, two different brightness-based subspace constraints are invoved simultaneously in [Ira99, Ira02]: a multi-point multi-frame pixel-wise constraint based on the *brightness constancy equation* and a multi-point multi-frame region-based constraint which relies on the *Lucas* and *Kanade* formulation. The approach implicitly translates in a confidence-weighted subspace projection formulation since a confidence measure computed for each pixel or area is involved in the subspace projection process. Indeed, pixels or areas whose matching is reliable will have more influence in the subspace projection task than pixels or areas for which the matching fails.

However, the method of [Ira99, Ira02] is limited to rigid motion assumptions. It has been then extended to the non-rigid case in [TB02] where a non-rigid model-free tracking solution exploiting space-time rank constraints has been proposed. More recently, Garg *et al.* perform dense multi-frame *optical flow* estimation in a variational framework using 2D trajectory subspace constraints [GRA11b, GRA13]. This variational approach with subspace constraints generates trajectories starting from a reference frame in a non-rigid context. Therefore, trajectories are estimated along the sequence assuming that they must lie close to a low-dimensional trajectory subspace which implicitly acts as a long-term temporal regularization. Temporary occlusions are handled through temporal information which allows to predict the position of non-visible points.

In [GRA11b, GRA13], *Garg* et al. focus on the estimation of $\{\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})\}_{n\in[\![0,\dots,N]\!]\neq ref}$, i.e. the set of motion fields starting from $\boldsymbol{x}_{ref}$ belonging to I_{ref} and going to the subsequent frame I_n. This set of motion fields define the discrete-time 2D trajectory $\boldsymbol{T}(\boldsymbol{x}_{ref})$ whose estimation is submitted to the following linear subspace model:

$$\boldsymbol{T}(\boldsymbol{x}_{ref}) = \sum_{i=1}^{R} \boldsymbol{q}_i(n).L_i(\boldsymbol{x}_{ref}) + \epsilon(\boldsymbol{x}_{ref}, n) \tag{8.15}$$

where the trajectory of $\boldsymbol{x}_{ref}$ is approximated as the linear combination of R basis trajectories $\boldsymbol{q}_1(n), ..., \boldsymbol{q}_R(n)$. $\epsilon(\boldsymbol{x}_{ref}, n)$ denotes a modeling error term. $L_i(\boldsymbol{x}_{ref})$ corresponds to coefficients which control the linear combination of the R basis trajectories. The basis is considered orthonormal and can be computed by applying a Principal Component Analysis (PCA) to a small

subset of point tracks. The point tracks involved for the choice of the basis are chosen with respect to their intrinsic quality (as in [Ira02]). Moreover, the *Discrete Cosine Transform* (*DCT*) can also be used instead of *PCA*. Trajectory subspace constraints and dense motion estimation are combined following variational principles leading to the minimization of the following energy:

$$\begin{aligned} E[\boldsymbol{T}(\boldsymbol{x}_{ref}), \boldsymbol{L}(\boldsymbol{x}_{ref})] &= \alpha \int_{\Omega} \sum_{\substack{n=0 \\ n \neq ref}}^{N} |I_n(\boldsymbol{x} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})) - I_{ref}(\boldsymbol{x}_{ref})| d\Omega \\ &+ \beta \int_{\Omega} \sum_{i=1}^{R} |\boldsymbol{T}(\boldsymbol{x}_{ref}) - \sum_{i=1}^{R} \boldsymbol{q}_i(n).L_i(\boldsymbol{x}_{ref})|^2 d\Omega \\ &+ \int_{\Omega} \sum_{i=1}^{R} g(\boldsymbol{x}_{ref}).|\nabla L_i(\boldsymbol{x}_{ref})|_{\epsilon} d\Omega \end{aligned} \tag{8.16}$$

The energy of Eq. 8.16 is minimized with respect to both the point trajectories and their components in the low-dimensional subspace (i.e. $L_i(\boldsymbol{x}_{ref})$). The first term is an extension of the *brightness constancy constraint* in a multi-frame context. The second term penalizes trajectories that do not reside near the low-dimensional subspace. The third term corresponds to the *total variation* and performs a spatial regularization of the trajectory model coefficients in order to avoid spatial oscillations of each coefficient. Finally, the constants α and β balance the relative weight of each term. The third term involves an edge weighting $g(\boldsymbol{x})$ and the *Huber* norm $|\nabla L_i(\boldsymbol{x})|_{\epsilon} = H_{\epsilon}(|\nabla L_i(\boldsymbol{x})|^2$ with:

$$H_{\epsilon}(s^2) = \begin{cases} \frac{s^2}{2\epsilon} & \text{if} \quad s \leq \epsilon \\ s - \frac{\epsilon}{2} & \text{otherwise} \end{cases} \tag{8.17}$$

As described in 8.17, using the *Huber* norm $H_{\epsilon}(.)$ [Hub73] consists in penalizing in a quadratic way small gradient magnitudes while performing a linear penalization for larger gradient magnitudes which maintains the discontinuity preserving properties.

This smart method requires nevertheless strong *a priori* assumptions on scene contents. Moreover, dense tracking of multiple objects is possible only if the reference frame is segmented and no results on several objects are shown in [GRA11b, GRA13]. Finally, only trajectories starting from the reference frame are considered and the computation of motion fields starting from I_n and going to I_{ref} is not under consideration.

8.2 Applications of long-term motion estimation

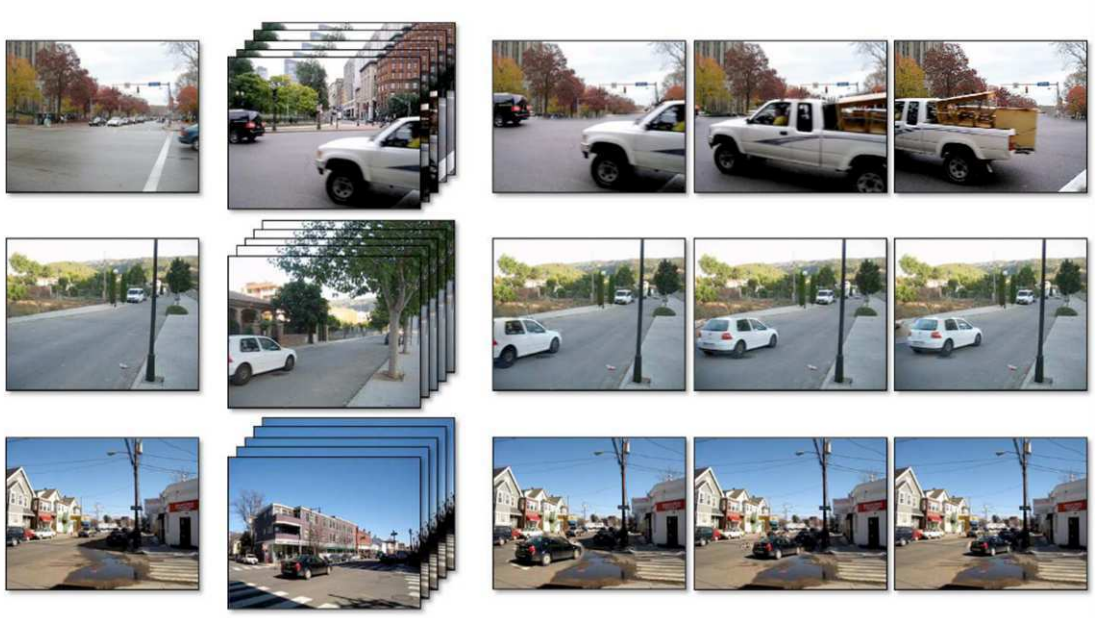
Temporal consistency undeniably plays a key role in many applications related to computer vision and robust multi-frame or long-term motion estimators presented in Section 8.1 can be used to obtain a temporally smooth description of how the content of a video sequence varies in time. In the following, we briefly review possible applications to long-term motion estimation. These applications cover the whole spectrum of long-term motion estimation algorithms from object-based or sparse *feature* tracking to block matching or dense long-term matching. Nevertheless, whatever the algorithm usually dedicated and suitable for each of these applications, note that dense long-term motion estimators can provide long-term estimates in any case.

The first straightforward application deals with tracking in the field of video surveillance or for visual servoing. Visual servoing, also known as vision-based robot control, allows to control the displacements of robots. Tracking applications are well-known due to the large amount of surveillance systems which require automatic object tracking and automatic abnormal event detection in order to replace or almost limit tedious manual video checking. 2D or 3D visual tracking [DMC10, OVCP13] can also be seen as a way to automatically extract tracks (people tracks or face tracks for instance) for annotation tasks and more generally in the field of audio-visual content analysis.

Long-term motion estimation serves as a fundamental brick for a number of more advanced tasks such as structure-from-motion [ZDJ+10] and camera tracking [NLD11] (for mobile robotics, scene reconstruction or augmented reality) or video indexing (video copy detection or video synchronization). It can be also a key tool for video compression schemes since it allows to eliminate the temporal redundancy between frames [CMPP09]. A multi-frame motion estimation based on spatio-temporal tubes has been for instance incorporated into the video coding standard H.264/MPEG-4 Advanced Video Coding (AVC) in order to improve compression performances [BDRB07]. In the context of video quality assessment, temporal evolutions of spatial distortions can be assessed within spatio-temporal tubes following pre-computed long-term trajectories [WLB04, NLMLCB09].

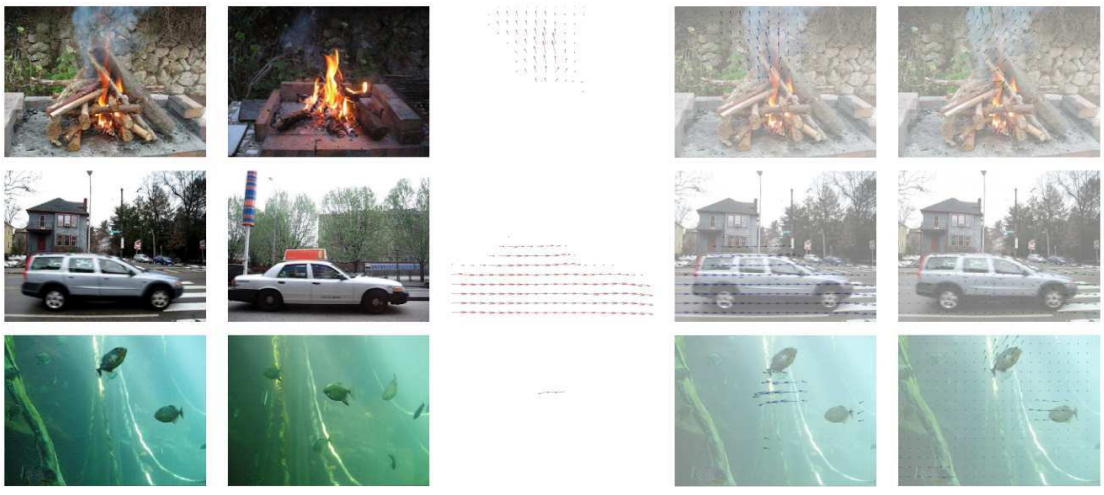
More recently, some papers have proposed to manipulate long-term motion estimates in order to consider applications such as:

- emulating camera motion by following a single object [SEASM09, SN13],
- performing motion synthesis via moving objects transfer from similar video scenes [LYT+08, LYT11] (see Fig. 8.7 where a moving car can be transferred in a street scene initially empty),
- predicting the direction and velocity of objects in still images by comparisons with a video database [LYT+08, LYT11]. Also known as motion hallucination, this application is similar to recognition methods but possible motions instead of labels are assigned to each pixel. Once the correspondence has been established between the still query image and a sequence of the database, the temporally estimated motion of the sequence is used to animate the still image (examples shown in Fig. 8.8),
- determining what motions are plausible given a single static image by comparisons with a video database [LYT+08, LYT11] (a car can move *forward*, *backward*, turn, remain static but cannot move upwards for instance),
- performing multiple object operations by long-term trajectory manipulation [SEASM09, SN11, SN13]: re-ordering, retiming, removal, inversion... Fig. 8.9 shows multiple object operations made on a surveillance video sequence by the object centric user interface proposed in [SN13]. More precisely, the trajectory of the blue car is truncated, copied, shifted in time, and inverted whereas the red car's trajectory is copied and shifted in time, truncated and then extended until the end of the sequence. The example displayed in Fig. 8.10 (also extracted from [SN13]) focuses on object duplication which allow to produce a composite video showing multiple non-overlapping segments in the same image space,
- magnifying motion whose goal is to find (small) motion in the video, amplify them and render the output sequence with the desired magnified motions [LTF+05]. The sequence is



Still query image | Top video match | Synthesized sequence obtained by transferring moving objects

Figure 8.7: Motion synthesis via object transfer. This method, presented in [LYT+08, LYT11], consists in transferring the motion of moving objects from a video to the still query image.



Original image | Database match | Motion of database match | Warped and transferred motion | Ground truth of original

Figure 8.8: Motion hallucination [LYT+08, LYT11]: motion of still images is predicted through transfer and warping from the top database matches.

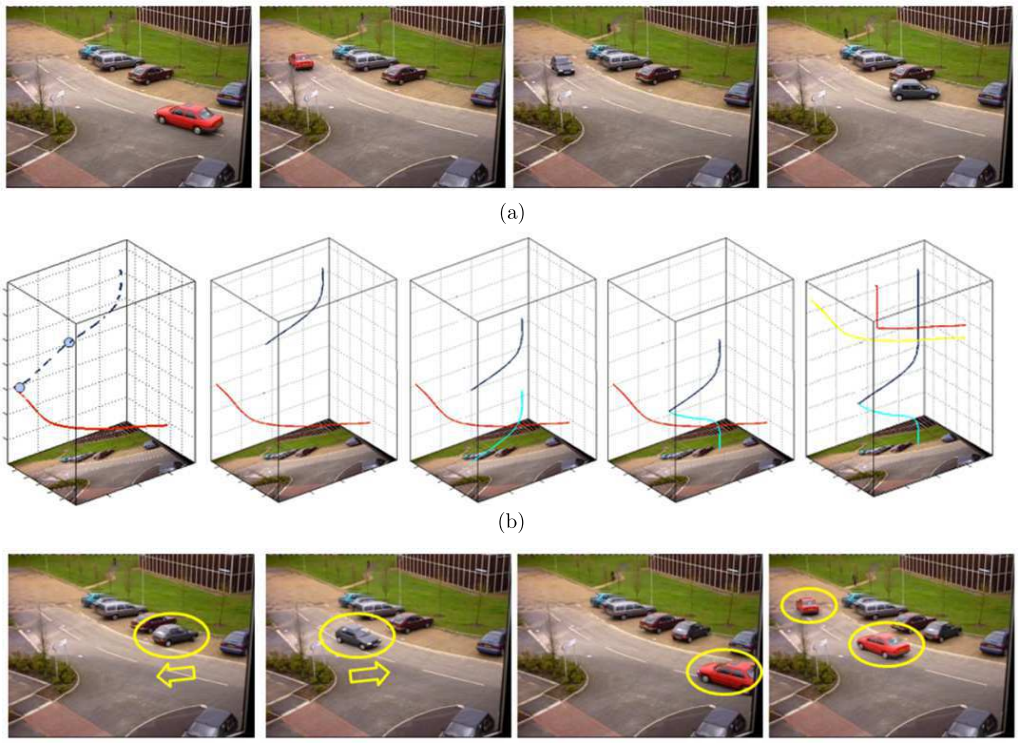


Figure 8.9: Multiple object operations made on a surveillance video sequence [SN13]: (a) frames from the original video, (b) multiple object operations (see details in the text), (c) frames from the modified video.

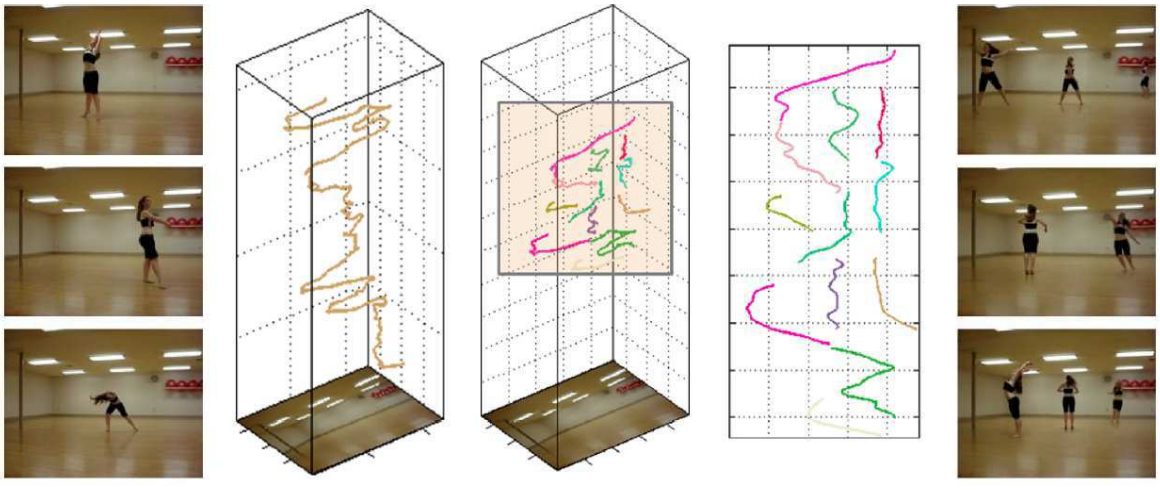


Figure 8.10: Composition of a dance video montage with object duplication [SN13]: (a) frames from the original video, (b) dancer's original trajectory, (c) user's arrangement of the trajectory segments, (d) magnified XY view of the interaction grid, (e) frames from the montage video.

first described using trajectories which are then clustered in order to build layers of related motion and appearance. The user specifies the group of trajectories he wants to magnify. Finally, the rendering generally requires in-painting to fill holes since motion magnification disoccludes some parts of the sequence. In the same spirit, [WRS+12, RWDF13] propose to reveal subtle or invisible changes in the world. Following these works, we can for instance analyze for each pixel of a face image the color variation along its own trajectory before amplifying this variation and finally being able to visualize the human pulse. Another interesting application consists in amplifying baby's breathing motion in a medical context,

- applying cartoon animation filters by animating input motion in such way that the output motion is more alive and animated [WDAC06]. Such filter creates perceptually appealing motion exaggeration like motion magnification algorithms. The filter presented in [WDAC06] adds a smoothed, inverted and time shifted version of the acceleration into the original motion signal.

In most applications, a sparse set of points is sufficient. However, there are cases though where dense sets of trajectories are better suited, if not required. These include recent spatio-temporal scene segmentation [BM10, LASL11] and 2D-to-3D video conversion schemes [CLD11] for which spatial density and long-term temporal consistency are key points.

Action description, detection and recognition, and dynamic scene analysis at large can also rely on dense motion estimates. In [WKSL11], *Wang* et al. propose to perform action recognition based on dense motion trajectories. Descriptors such as HOG [DT05] (Histograms of Oriented Gradients) or HOG-HOF [LMSR08] (Histograms of *Optical Flow* combined with HOG) are computed within space-time volumes along trajectories based on appearance and motion information. Finally, these trajectory descriptors are able to distinguish actions such as walking, biking, jogging or skateboarding.

Finally, a number of automatic and semi-automatic video editing tasks explicitly require dense and long-term motion estimation. For example, the objective can be to insert graphic elements on mosaics [RAKRF08] or directly on one or several reference frames [CCRP12, CCR+12, CCRM14] and then to automatically propagate the graphic elements across the sequence using pre-computed dense motion estimates. Such technology is usually involved in movie postproduction (visual effects, artistic video editing...) or in advertising (logo insertion and propagation) for which a high quality rendering is highly required (more than real-time requirements). In this context of video editing, we can quote [FSBC11, SFAC13] which proposes new numerical schemes to perform temporally consistent video editing based on *Poisson* and *total variation* type formulations. Note that not only color textures or logos can be inserted and propagated. As seen previously, segmentation [BM10, LASL11], depth or disparity information [CLD11] or any type of visual information at large can be propagated across the sequence using dense motion estimation.

When dealing with consumer tools, a compromise must be found between rendering quality and processing complexity. The way the user interacts and manipulates motion is one key issue not only in terms of application usability but also considering that user interactions can improve automatic motion estimation algorithm [KRLM11, RHK+12]. The tool proposed in [KRLM11] assists the user in selecting and correcting mismatched correspondences. In the same spirit, [RHK+12] combines realtime interactive correspondence display, multi-level user guidance and algorithmic subpixel precision to counteract failure cases of automated estimation algorithms.

8.3 Conclusion

Optical flow is a fundamental tool to extend motion estimation to long-term and to finally enable information transfer across the sequence. Up to now, the classic concatenation of consecutive *optical flows* has been combined with more sophisticated global frameworks or replaced by more robust multi-frame formulations using trajectorial regularization or subspace constraints. These improvements have been developed to avoid large error accumulation and therefore to limit the drift of dense trajectories. However, reliable dense trajectories still do not exceed more than thirty frames according to experiments reported in the literature.

Applications to long-term motion estimation need temporally smooth and accurate description of how the content varies in time. Some of these applications such as video editing tasks have very high quality requirements, especially in the field of movie postproduction. However, our conclusion after the analysis of the literature is that significant progress must be carried on to satisfy the application requirements and more generally to reach a robust estimation of dense point correspondences in long video sequences.

Towards this goal, we propose several contributions toward long-term dense motion estimation. They are described in the following chapters. In particular, our analysis starts in Chapter 9 by the introduction of the concept of *multi-step* integration whose underlying idea is to combine *optical flow* fields estimated with various inter-frame distances in order to perform long-term dense motion estimation.

Chapter 9

Introduction to *multi-step* integration strategies

In this chapter, we aim at laying the foundations of the long-term dense motion estimation frameworks described in the following of this part II. For this sake, we focus on concepts which are shared by our different contributions, especially those presented in Chapter 10 and 11.

More precisely, we describe both *from-the-reference* and *to-the-reference* point correspondence estimation schemes in Section 9.1. Then, we introduce the concept of *multi-step* elementary *optical flow* estimation (Section 9.2) which extends the conventional use of existing *optical flow* estimators. Starting from a set of *multi-step* elementary *optical flow* fields, we finally study in Section 9.3 the different possible integration strategies to compute long-term trajectories. This preliminary study gives an overview of the context and initiates the further analysis described in the next chapters.

The concepts introduced here has been proposed and detailed into two papers published in international conferences: [CCRP12, CCR+12].

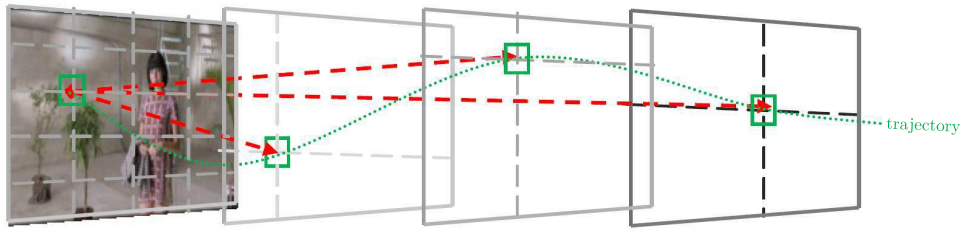
9.1 *From-the-reference* and *to-the-reference* schemes

We propose to study the estimation of long-term dense displacement fields between a reference frame I_{ref} and all the subsequent frames of the sequence $\{I_n\}_{n\in[\![0,\dots,N]\!]\neq ref}$. Depending on the application, two different schemes for point correspondence estimation can be identified: motion estimation between the reference frame and all the images of the sequence and motion estimation to match each image to the reference frame.

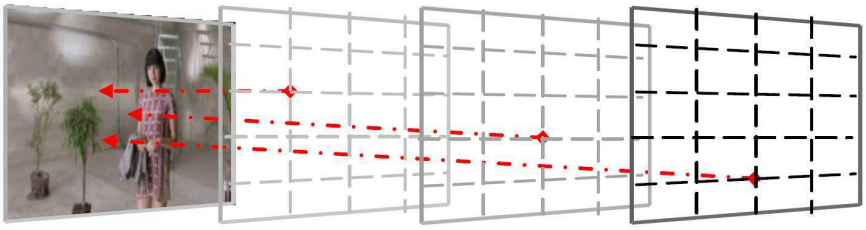
These two schemes are illustrated in Fig. 9.1 and are referred to as *from-the-reference* (Fig. 9.1 (a)) and *to-the-reference* (Fig. 9.1 (b)). In terms of displacement fields, we make the distinction between *from-the-reference* and *to-the-reference* displacement fields.

From-the-reference displacement fields $\boldsymbol{d}_{ref,n}$ connect each grid point $\boldsymbol{x}_{ref} \in I_{ref}$ to a non necessary integer position in I_n and therefore define trajectories $\boldsymbol{T}(\boldsymbol{x}_{ref})$. Such displacement fields are involved in the context of point tracking, trajectory clustering, long-term object segmentation or action recognition since they allow information *pushing* from the reference frame I_{ref} to the whole sequence.

On the contrary, *to-the-reference* displacement fields $\boldsymbol{d}_{n,ref}$ determine for all grid locations of all the frames of the sequence their position in the reference frame. Such representation is more suitable for applications related to propagating information from the reference frame to



(a) *From-the-reference* scheme



(b) *To-the-reference* scheme

Figure 9.1: Point correspondence schemes: (a) *From-the-reference* scheme which corresponds to the problem of determining the trajectory of each initial grid point in the reference frame along the sequence, (b) *To-the-reference* scheme which aims at determining the position in the reference image of each grid point of each image of the sequence.

the rest of the sequence such as automatic and semi-automatic video editing tasks whose aim is to assign information over each frame by *pulling* it from I_{ref}.

Compared to the *forward* (in the direction of time) and *backward* (opposite direction of time) terminology, the *from/to-the-reference* terminology better suits to characterize long-term displacement fields since it includes the two following situations: I_{ref} at the left or at the right side of the sequence.

9.2 *Multi-step* elementary *optical flow* fields

As generally considered in the literature, *optical flow* estimators focus on motion estimation between consecutive frames. To go further, we introduce the concept of *multi-step* elementary *optical flow* estimation whose underlying idea is to extend the use of *optical flow* estimators to distant frames.

In this thesis, we define a "elementary *optical flow* field" defined between two images as a motion field obtained through an *optical flow* estimator directly applied between these two images. The fundament of all the contributions presented in the following of this Part II is to exploit two types of elementary *optical flow* fields:

- elementary *optical flow* fields $\boldsymbol{u}_{n,n+1}$ computed between consecutive frames,
- elementary *optical flow* fields estimated with larger inter-frame distances: $\boldsymbol{u}_{n,n+s_i}$ with $s_i > 1$.

Let s_i correspond to a *step*, i.e. an inter-frame distance between the current frame I_n and I_{n+s_i}. Thus, computing an elementary *optical flow* field of *step* s_i, i.e. $\boldsymbol{u}_{n,n+s_i}$, translates in performing an *optical flow* estimator between the pair of frames $\{I_n, I_{n+s_i}\}$ directly.

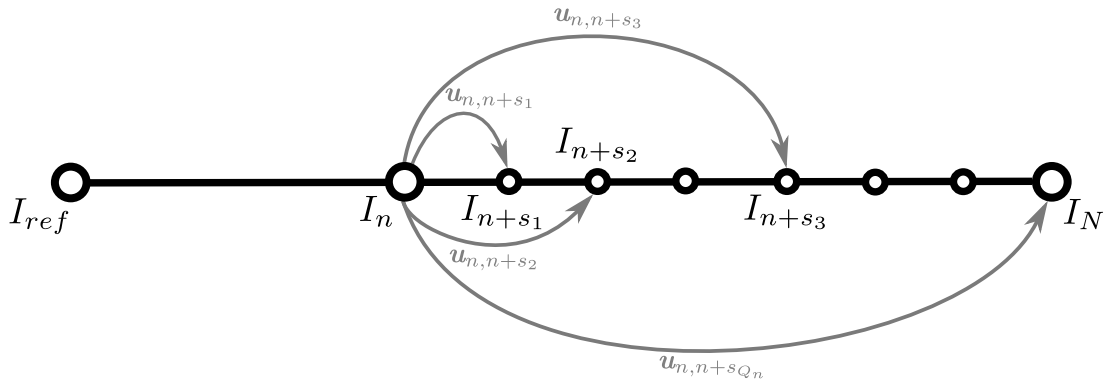


Figure 9.2: *Multi-step* elementary *optical flow* fields from the current frame I_n.

In order to obtain accurate and dense point correspondences in long video shots, we propose to combine elementary *optical flow* fields with various inter-frame distances, i.e. *multi-step* elementary *optical flow* fields. Therefore, "*multi-step* elementary *optical flow* estimation" refers to the combination of elementary *optical flow* fields computed between consecutive and non-consecutive frames.

Such approach follows the following finding: considering the matching between a pair of distant frames, it appears that some regions of the image can be better matched by concatenating *optical flow* fields computed between consecutive frames, while for others a direct long-term point matching is preferred. Contrary to consecutive *optical flow* concatenation, direct matching is more robust in terms of position drift. However, direct matching is very sensitive to ambiguous correspondences which can occur for periodic structures for instance. In this context, we propose to involve *multi-step* elementary *optical flow* fields to combine the benefits of both approaches.

Another reason which explains why we aim at exploiting *multi-step* elementary *optical flow* estimates is that this allows to jump occluding objects when temporary occlusions occur. By this way, we resolve the issue of consecutive *optical flow* concatenation schemes which does not recover trajectories after temporary occlusions as the target entity (point, object, area...) is lost once the occlusion takes place.

9.3 Introduction to *multi-step* integration strategies

As the concept of *multi-step* estimation has been introduced, we address now the following problem: how to perform dense and long-term motion estimation using *multi-step* elementary *optical flow* fields?

Let $S_n = \{s_1, s_2, \ldots, s_{Q_n}\} \subset \{1, \ldots, N-n\}$ be the ordered set of Q_n available *steps* at instant n. As described in Fig. 9.2, this means that the following set of elementary *optical flow* fields starting from I_n is therefore available: $\{\boldsymbol{u}_{n,n+s_1}, \boldsymbol{u}_{n,n+s_2}, \ldots, \boldsymbol{u}_{n,n+s_{Q_n}}\}$. S_n does not necessarily contains all the *steps* $\{1, 2, 3, 4, \ldots, N-n\}$, i.e. $Q_n \leq (N-n)$.

Starting from such sets of elementary *optical flow* fields defined from each frame I_n $\forall n$, we introduce in what follows three different possible strategies to generate trajectories (i.e. *from-the-reference* displacement fields $\boldsymbol{d}_{ref,n}$) along video sequences through integration (i.e. concatenation) of *multi-step* elementary *optical flow* fields.

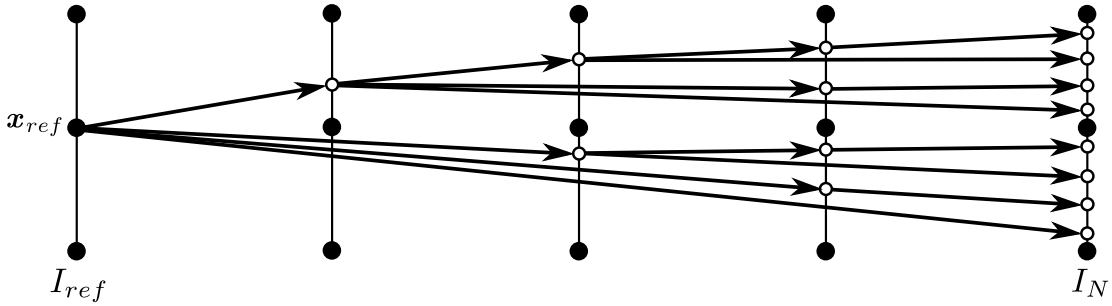


Figure 9.3: Long-term motion estimation through concatenation of *multi-step* elementary *optical flow* fields $\{\boldsymbol{u}_{n,n+1}, \boldsymbol{u}_{n,n+2}, \ldots, \boldsymbol{u}_{n,N}\}$ $\forall n \in [\![0, \ldots, N-1]\!]$.

We consider a sequence made of $N+1$ images $\{I_n\}_{n\in[\![0,\ldots,N]\!]}$ with I_0 as reference frame. We focus on the estimation of $\boldsymbol{T}(\boldsymbol{x}_{ref}) = \{\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})\}_{n\in[\![1,\ldots,N]\!]}$, the trajectory starting from $\boldsymbol{x}_{ref} \in I_{ref}$. Contrary to what generally happens in practice, we assume for the sake of simplicity that the set S_n of *steps* at instant n contains all the possible *steps*: $S_n = \{1, 2, \ldots, N-n\}$.

To begin with, let us consider all the candidate positions along the sequence which can be obtained starting from the grid point $\boldsymbol{x}_{ref}$ through concatenation of the *multi-step optical flow* fields $\{\boldsymbol{u}_{n,n+1}, \boldsymbol{u}_{n,n+2}, \ldots, \boldsymbol{u}_{n,N}\}$ defined $\forall n \in [\![0, \ldots, N-1]\!]$. The concatenation of these *multi-step* fields and the resulting candidate positions are illustrated Fig. 9.3 for $N = 4$. We assume that elementary *optical flow* vectors starting from non-grid positions are interpolated using the motion vectors from the nearest pixels.

Starting from this configuration, we describe in the following the three *multi-step* strategies proposed to the purpose of long-term motion estimation: an exhaustive approach (Section 9.3.1), an approach based on *dynamic programming* (Section 9.3.2) and a sequential approach (Section 9.3.3). For each strategy, we briefly introduce the general concept and we study particularly the complexity of the trajectory construction process. Through this theorical analysis, we aim at giving an overview of the possible approaches before going into detail in the next chapters.

9.3.1 Exhaustive *multi-step* strategy

A first approach consists in exhaustively generating all the possible ways to connect I_{ref} to I_N going through each frame I_n (Fig. 9.4). Such an exhaustive approach requires the generation of all the possible candidate positions along the sequence. This leads to a number of candidate positions $N_{cand}(n)$ per frame I_n which is:

$$N_{cand}(n) = 2^{n-1} \tag{9.1}$$

The goal is then to know which set of candidate positions gives the optimal trajectory (see the red *path* in Fig. 9.4) as the selection of a single trajectory among all the generated ones is required (the selection task is not described for the time being). The number of generated trajectories N_{traj} at each instant n (starting from I_{ref}) can be computed as follows:

$$N_{traj}(n) = \prod_{i=1}^{n} N_{cand}(i) = \prod_{i=1}^{n} 2^{i-1} = 2^{\wedge}(\sum_{i=1}^{n} i - 1) = 2^{\frac{n(n-1)}{2}} \tag{9.2}$$

The last equality of Eq. 9.2 is specified below:

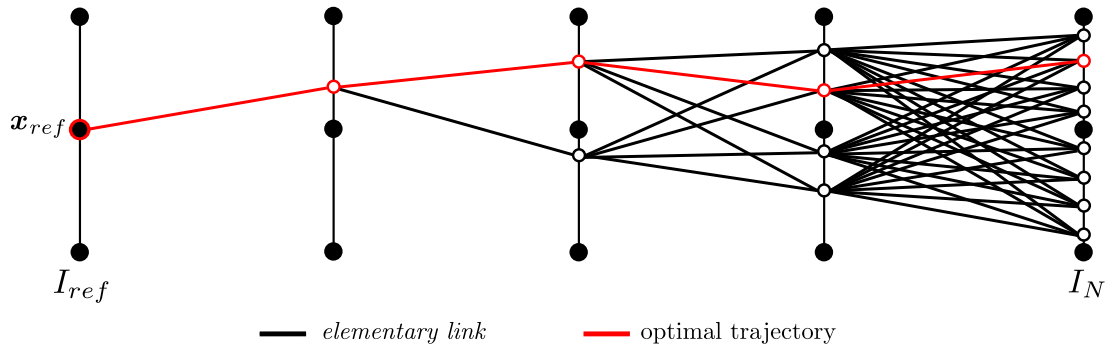


Figure 9.4: Generation of the trajectory starting from $\boldsymbol{x}_{ref} \in I_{ref}$: exhaustive approach. *Elementary links* connect candidate positions of I_{n-1} to candidate positions in I_n $\forall n$.

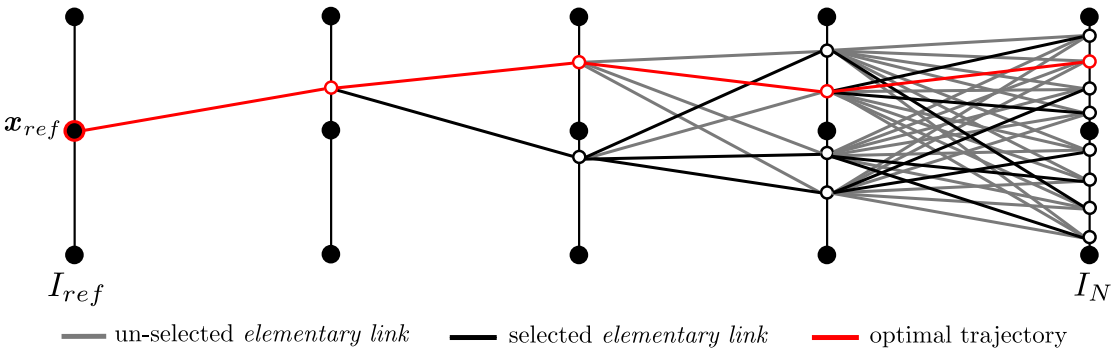


Figure 9.5: Generation of the trajectory starting from $\boldsymbol{x}_{ref} \in I_{ref}$: *dynamic programming* based approach.

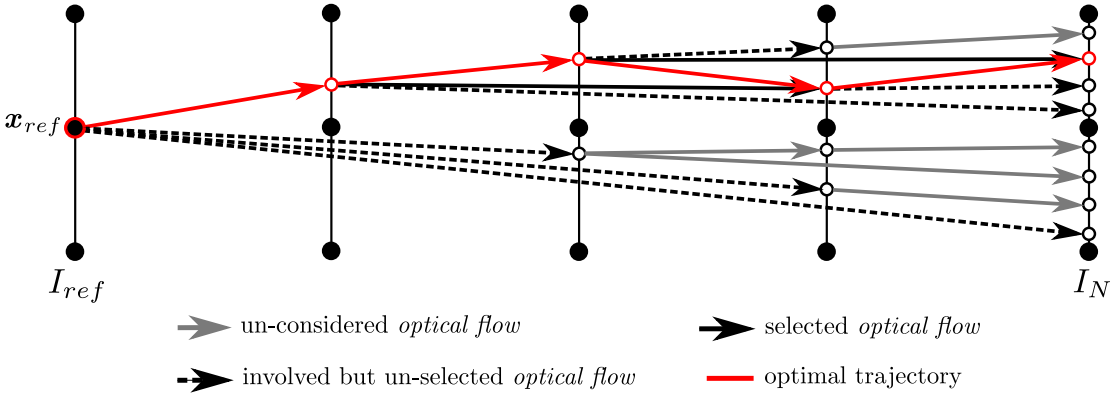


Figure 9.6: Generation of the trajectory starting from $\boldsymbol{x}_{ref} \in I_{ref}$: sequential approach

$$\sum_{i=1}^{n} i - 1 = -n + \sum_{i=1}^{n} i = -n + \frac{n(n+1)}{2} = \frac{n(n-1)}{2} \tag{9.3}$$

It appears that for long video shots, it is not possible to consider all these trajectories due to computational and memory issues. Even with a sequence of only 6 frames (i.e. $N = 5$), such an exhaustive approach requires the construction of $N_{traj}(5) = 1024$ trajectories.

9.3.2 *Multi-step* strategy based on *dynamic programming* (*DP*)

Instead of studying all the possible ways to go from I_{ref} to I_N via each frame I_n, another solution consists in following a *dynamic programming* (*DP*) approach as shown in Fig. 9.5. For this task, a sequential process is performed across the sequence as follows: for each candidate position of the current frame I_n, we choose the best correspondence within the set of candidate positions in I_{n-1} according to a given cost. Generally with *DP*, this selection is based on the cost of the *path* from $\boldsymbol{x}_{ref}$ to the current candidate position. This cost is usually accumulated from frames to frames in order to know which candidate position in I_N is the best ending point of $\boldsymbol{T}(\boldsymbol{x}_{ref})$ [BF06]. Finally, once the whole sequence has been processed this way, the optimal trajectory (see the red *path* in Fig. 9.5) can be directly obtained.

Once arrived at frame I_n, a trajectory between $\boldsymbol{x}_{ref}$ and each candidate position of I_{n-1} has been already established. Consequently, to compute the number of possible trajectories N_{traj} at instant n, we have to consider that each candidate position in I_{n-1} has already received a single optimal trajectory. Each of these trajectories has to be extended by one of the $N_{cand}(n)$ available vectors in order to join the positions of I_n (illustrated by un-selected and selected *elementary links* in Fig. 9.5). Thus, N_{traj} can be computed as follows:

$$N_{traj}(n) = N_{cand}(n).N_{cand}(n-1) = 2^{n-1}.2^{n-2} = 2^{2n-3} \tag{9.4}$$

This type of processing based on *DP* has been used to the purpose of template tracking in [BF06]. *Buchanan* and *Fitzgibbon* propose in [BF06] to perform image patch searches using *k-d trees* to generate multiple trajectories. These trajectories are then merged using *DP* which tries to find the optimal choices of matches over the sequence. This path optimization handles occlusions which are considered as a special kind of match. Finally, a correlation-based localization refines the matches of the optimal trajectories in order to obtain sub-pixel accuracy.

As in the exhaustive case, applying such approach for dense tracking seems difficult due to computational and memory issues, even if *DP* allows to significantly reduce the search space. The extension of *DP* from sparse to dense trajectory estimation is not straightforward since it must not be limited to a simple independent processing for each trajectory. It requires a global approach which incorporates spatial regularization in the underlying model.

9.3.3 Sequential *multi-step* strategy

A third way to establish a trajectory starting from $\boldsymbol{x}_{ref} \in I_{ref}$ consists in sequentially selecting for each frame I_n one candidate position among all the available candidate positions. This sequential *multi-step* strategy is illustrated in Fig. 9.6. To reduce the computational complexity, all the *multi-step* elementary *optical flow* vectors starting from a non-selected candidate position (grey vectors in Fig. 9.6) are not taken into account. Consequently, this limits the number of

Method	$N_{cand}(n)$	$N_{traj}(n)$	$N_{of}(n)$	$N_{cand}(4)$	$N_{traj}(4)$	$N_{of}(4)$	$N_{cand}(10)$	$N_{traj}(10)$	$N_{of}(10)$
Exhaustive	2^{n-1}	$2^{\frac{n(n-1)}{2}}$	$\sum_{i=1}^{n} 2^{i-1}$	8	64	15	256	2^{45}	1023
DP	2^{n-1}	2^{2n-3}	$\sum_{i=1}^{n} 2^{i-1}$	8	32	15	256	131072	1023
Sequential	n	n	$\frac{n(n+1)}{2}$	4	4	10	10	10	55

Table 9.1: Comparisons of the three different *multi-step* strategies to generate trajectories along video sequences in terms of: 1) N_{cand}, the number of candidate positions , 2) N_{traj}, the number of generated trajectories, 3) N_{of}, the number of *multi-step optical flow* vectors involved for the trajectory estimation since I_{ref}. Numerical results are given for $n = 4$ and 10.

candidate positions N_{cand} per frame I_n and the number of possible trajectories N_{traj} at instant n which simply become: $N_{cand}(n) = N_{traj}(n) = n$.

Let us describe how this sequential *multi-step* strategy works. For the first pair $\{I_{ref}, I_{ref+1}\}$, only the *optical flow* vector with step 1 is available which directly gives the corresponding position of $\boldsymbol{x}_{ref}$ in I_{ref+1}. To join I_{ref+2}, a selection must be done between: 1) the candidate obtained using the *optical flow* vector with step 1 starting from the position in I_{ref+1}, 2) the candidate obtained through the *optical flow* vector with step 2 starting from $\boldsymbol{x}_{ref}$. To join I_{ref+3}, a selection must be done between: 1) the candidate obtained using the *optical flow* vector with step 1 starting from the position in I_{ref+2}, 2) the candidate obtained using the *optical flow* vector with step 2 starting from the position in I_{ref+1}, 3) the candidate obtained through the *optical flow* vector with step 3 starting from $\boldsymbol{x}_{ref}$. This sequential selection of the optimal candidate position is repeated this way until the end of the sequence is reached. This finally gives $\boldsymbol{T}(\boldsymbol{x}_{ref})$.

Let us compute $N_{of}(n)$, the number of *multi-step optical flow* vectors effectively involved for the estimation of $\boldsymbol{T}(\boldsymbol{x}_{ref})$ between I_{ref} and I_n. N_{of} can be estimated given that there are many *multi-step optical flow* vectors as there are candidate positions from I_{ref+1} to I_n included. Indeed, each candidate has been obtained through one of these *multi-step optical flow* vectors. Consequently, N_{of} equals to $\sum_{i=1}^{n} N_{cand}(i)$ which gives for the sequential approach:

$$N_{of}(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{9.5}$$

N_{of} is a good cue to see the difference in terms of computational complexity compared to the two previous methods for which: $N_{of}(n) = \sum_{i=1}^{n} 2^{n-1}$.

9.3.4 Overview on *multi-step* strategies

Table 9.1 gives an overview of the three described *multi-step* strategies in terms of number of candidate positions (N_{cand}), number of generated trajectories (N_{traj}), number of involved *multi-step optical flow* vectors (N_{of}). Numerical results for $n = 4$ and 10 show clearly that the sequential approch allows to compute trajectories with a more reasonable complexity. Compared to both exhaustive and *DP*-based approaches, the sequential *multi-step* strategy exploits a lower number of *optical flow* vectors which allows to store in memory a moderate number of resulting candidate positions: 55 *multi-step* vectors against 1023 which leads to respectively 10 and 256 candidate positions for $n = 10$.

9.4 Conclusion

Through this preliminary chapter, we introduced the concept of *multi-step* elementary *optical flow* fields and we described how the combination of these *multi-step* estimates can be useful to both combine the benefits of consecutive *optical flow* concatenation and those of direct matching as well as handle temporary occlusions.

Towards the estimation of accurate dense *from-the-reference* and *to-the-reference* correspondences in long video shots, we gave an overview of three possible *multi-step* integration strategies: exhaustive (Section 9.3.1), *dynamic programming*-based (Section 9.3.2) and sequential (Section 9.3.3). Compared to both exhaustive and *dynamic programming*-based schemes, our study reveals that a reasonable complexity can be reached by considering a sequential *multi-step* strategy. This latter method translates in limiting the process only to *multi-step* estimates starting from previous optimal candidates.

We propose to incorporate such sequential strategy (Section 9.3.3) in a sophisticated dense and long-term motion estimation framework which will be studied in depth in Chapter 10. Through this method, Chapter 10 explores both how to accurately perform a sequential accumulation of *multi-step* elementary *optical flow* vectors and how to select the optimal long-term displacement fields.

An alternative dense and long-term motion estimator will be addressed in Chapter 11 based on the exhaustive strategy of Section 9.3.1. We will in particular study how to take into account a large amount of motion *path* made of *multi-step* elementary *optical flow* vectors while avoiding computational and memory issues through combinatorial integration. In addition, the multiple long-term displacement fields resulting from the generated motion *path* will undergo a statistical-based selection.

Contrary to both sequential and exhaustive schemes, the *dynamic programming*-based strategy of Section 9.3.2 will not be investigated in the following of this Part II since it appears to us not possible to overcome the computational and memory issues caused by the extension of *DP* from sparse to dense motion estimation.

Chapter 10

Sequential *multi-step* flow strategies

Numerous applications related to video processing eventually require determining a *dense* set of trajectories or point correspondences that permit to propagate large amounts of information such as color, disparity, depth, position or any other type of visual information across the sequence. Dense motion information is well represented by *optical flow* fields and pixels can be simply propagated through time by accumulation of *optical flow* vectors [CMP02, BM10, SBK10, WKSL11, WKSL13] using tools such as *Euler* or *Runge-Kutta* integration schemes. However, we have seen in Chapter 8 that such straightforward approach offers to much scope to motion drift since flow estimation errors are inevitably accumulated over time.

In this context, we propose an alternative method to existing frameworks previously described in Chapter 8 such as [WTP+09, SS07, VBVZ11, ST06, ST08, GRA11b, GRA13]. In particular, we propose to robustify the estimation of dense *from-the-reference* (from I_{ref} to I_n) and *to-the-reference* (from I_n to I_{ref}) long-term displacement fields (Section 9.1, Chaper 9) using pre-computed *multi-step* elementary *optical flow* fields (Section 9.2) while relying on the sequential strategy introduced in Section 9.3.3.

We consider a sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\dots,N]\!]}$ including $I_{ref} = I_0$ (i.e. the first frame of the sequence) considered as a reference frame. Let $\boldsymbol{d}_{n,m} : \Omega \to \mathbb{R}^2$ be a displacement field defined on the continuous rectangular domain Ω, such that for every $\boldsymbol{x} \in \Omega$ corresponds a displacement vector $\boldsymbol{d}_{n,m}(\boldsymbol{x}) \in \mathbb{R}^2$ for the ordered pair of images $\{I_n, I_m\}$. Starting from *multi-step* elementary *optical flow* fields pre-computed between arbitrary frames $\{I_i, I_j\}$ with $\{i, j\} \in [\![0, \dots, N]\!]$, our objective is to compute both *from-the-reference* displacement fields $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ for each n where $\boldsymbol{x}_{ref}$ belongs to the image grid Ω of I_{ref} and *to-the-reference* displacement fields $\boldsymbol{d}_{n,ref}(\boldsymbol{x}_n)$ $\forall n$ where $\boldsymbol{x}_n$ belongs to the image grid Ω of I_n.

We propose a process whose general concept follows the two following phases. Considering a pair $\{I_n, I_m\}$, various candidate displacement fields $\boldsymbol{d}_{n,m}$ are computed by concatenating different *multi-step* elementary *optical flow* fields. These *optical flow* concatenations are performed following a variant of the sequential *multi-step* strategy studied in Section 9.3.3. Finally, the optimal long-term displacement field $\boldsymbol{d}^*_{n,m}$ is obtained by merging all the generated candidate displacement fields $\boldsymbol{d}_{n,m}$.

We study in Section 10.1 a first version of our method, called *multi-step* flow via *graph-cuts* (*MS-GC*). Section 10.2 describes the significant improvements we developed with respect to this first approach. The resulting long-term dense motion estimation method is referred to as *multi-step* flow fusion (*MSF*). Finally, we present results for both methods (*MS-GC* and *MSF*) in Section 10.3. These proposed *multi-step* flow estimation techniques are notably evaluated through both point-wise tracking and *pulling* dense information from the reference frame,

as demonstrated through a number of experiments on ground-truth data and through visual assessment. Comparisons with respect to state-of-the-art approaches are provided.

This study led to two publications published in international conferences: [CCRP12] which introduces the *multi-step* elementary *optical flow* estimation concept and describes the *multi-step* flow via *graph-cuts* methods (*MS-GC*) and [CCR+12] which explains the *multi-step* flow fusion (*MSF*) method.

10.1 *Multi-step* flow via *graph-cuts* (*MS-GC*)

Compared to the state-of-the-art, the proposed *multi-step* flow via *graph-cuts* (*MS-GC*) proposes two main contributions. Firstly, we propose a novel sequential method for accumulating elementary motion fields to produce a long term matching through inverse integration (Section 10.1.1). This inverse integration scheme leads to the *multi-step* flow formulation written and detailed in Section 10.1.2. Secondly, we show in Section 10.1.3 how to optimally combine different motion estimation *steps* in order to decide for the best point correspondence between two images.

In the following, we focus more precisely on the estimation of the long-term *to-the-reference* displacement field $\boldsymbol{d}_{n,ref}$ (i.e. $\boldsymbol{d}_{n,0}$ since we assume that $I_{ref} = I_0$). Note that our method can be generalized to any reference frame (and especially I_N, the end of the sequence). Moreover, the application to the *from-the-reference* displacement fields is straightforward.

10.1.1 Sequential displacement field construction via inverse integration

For the time being, we only assume that the elementary *optical flow* fields $\boldsymbol{u}_{n,n-1}$ with $n = 1, \ldots, N$ computed between pairs of consecutive frames are available as input information (i.e. no *multi-step* considerations for the moment).

We have studied in Chapter 10.1.1 that existing point tracking approaches based on *optical flow* generally use a simple 1st-order *Euler* integration. In our *to-the-reference* context, such an approach would be conducted through the three following stages:

1. take a starting grid point $\boldsymbol{x}_n$ of the image grid Ω of I_n,
2. for $m = n, n-1, \ldots, 1$, iteratively obtain:

$$\boldsymbol{x}_{m-1} = \boldsymbol{x}_m + \tilde{\boldsymbol{u}}_{m,m-1}(\boldsymbol{x}_m), \tag{10.1}$$

3. repeat steps (1) and (2) for each $\boldsymbol{x}_n$ to obtain a dense *to-the-reference* displacement field.

This gives an estimate of the positions of pixels $\boldsymbol{x}_n$ in I_0 via concatenation of *backward* consecutive elementary *optical flow* fields from I_n to I_0. We propose an alternative approach based on a different strategy that runs in the inverse direction, i.e. that starts in I_1 and runs *forward* to I_n. It aims at iteratively computing $\boldsymbol{d}_{n,0}(\boldsymbol{x}_n)$ (with $n = 1, 2, 3 \ldots$ and so on) following the iteration described below:

$$\boldsymbol{d}_{n,0}(\boldsymbol{x}_n) = \boldsymbol{u}_{n,n-1}(\boldsymbol{x}_n) + \tilde{\boldsymbol{d}}_{n-1,0}(\boldsymbol{x}_n + \boldsymbol{u}_{n,n-1}(\boldsymbol{x}_n)) \tag{10.2}$$

for each grid location $\boldsymbol{x}_n$ in I_n where $\tilde{.}$ denotes an interpolated displacement. With this new approach, the current long-term displacement field $\boldsymbol{d}_{n,0}$ is obtained by concatenation of the previously computed long-term field $\boldsymbol{d}_{n-1,0}$ and a *backward* elementary *optical flow* field $\boldsymbol{u}_{n,n-1}$.

Note the difference between both approaches (Eq. 10.1 and 10.2). Starting from the grid point $\boldsymbol{x}_n$ at image I_n, and its elementary displacement $\boldsymbol{u}_{n,n-1}(\boldsymbol{x}_n)$, one computes $\boldsymbol{x}_n+\boldsymbol{u}_{n,n-1}(\boldsymbol{x}_n)$. Then, in the former approach (Eq. 10.1), one interpolates the velocity $\boldsymbol{u}_{n-1,n-2}(\boldsymbol{x}_n+\boldsymbol{u}_{n,n-1}(\boldsymbol{x}_n))$ in I_{n-1} (through bilinear interpolation for instace) and continues accumulating *multi-step* elementary *optical flow* vectors in the *backward* direction (as illustrated in Fig. 10.1 (a)). In our approach, the interpolation is applied once on the long term displacement field $\boldsymbol{d}_{n-1,0}(\boldsymbol{x}_n + \boldsymbol{u}_{n,n-1}(\boldsymbol{x}_n))$ directly between I_{n-1} and I_0 (see Fig. 10.1 (b)). This procedure implies that $\boldsymbol{d}_{n-1,0}$ in Eq. 10.2 is available from the previous iteration.

The result is that we sequentially compute the dense *to-the-reference* displacement maps $\boldsymbol{d}_{n,0}$ *forwards* (i.e. starting from $I_{ref+1}, I_{ref+2}, \ldots, I_n$) contrary to the classic approach which runs *backwards*. Consequently, our approach is called *inverse* integration whereas the classical integration scheme is referred to as *direct* integration.

In order to obtain the correspondences between all pixels of all images with respect to the reference frame, it is easy to see that for the standard method the complexity is $O(N^2P)$ while for the proposed method, it is $O(NP)$, where P is the number of pixels for a single image.

10.1.2 *Multi-step* flow formulation

Now, we exploit the previous *inverse* strategy for defining an optimal and sequential way of combining *multi-step* elementary *optical flow* fields in order to perform an improved dense and long-term motion estimation. The reasoning is based on the following. We want to compute $\boldsymbol{d}_{n,0}(\boldsymbol{x}_n)$. Suppose that for a set of Q_n frame *steps* at instant n, say $S_n = \{s_1, s_2, ..., s_{Q_n}\} \subset \{-1, -2, \ldots, -n\}$, the set of corresponding motion fields $\{\boldsymbol{u}_{n,n+s_1}, \boldsymbol{u}_{n,n+s_2}, \ldots, \boldsymbol{u}_{n,n+s_{Q_n}}\}$ is available. For each $s_k \in S_n$, we write:

$$\boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n) = \boldsymbol{u}_{n,n+s_k}(\boldsymbol{x}_n) + \boldsymbol{d}_{n+s_k,0}(\boldsymbol{x}_n + \boldsymbol{u}_{n,n+s_k}(\boldsymbol{x}_n)) \tag{10.3}$$

In this manner, we generate different candidate displacements using *paths* made of two motion fields: a *multi-step* elementary *optical flow* field $\boldsymbol{u}_{n,n+s_k}$ and a previously computed long-term displacement field $\boldsymbol{d}_{n+s_k,ref}$ (Fig. 10.2). Among the generated candidates, we have to decide the optimal candidate for each location $\boldsymbol{x}_n$. Note that with $Q_n = 1\ \forall n$ and $s_1 = -1$, we come back to the formulation of Eq. 10.2.

10.1.3 Optimal *path* selection

To compute the optimal *to-the-reference* long-term displacement vectors $\boldsymbol{d}^*_{n,0}(\boldsymbol{x}_n)$, we have previously defined and computed the Q_n candidates $\boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n)$ for every point $\boldsymbol{x}_n$ in image I_n and now the best one has to be selected at each location. Therefore, we need to define an optimality criterion and an optimization strategy for this selection task. To evaluate the accuracy of the matching, we consider the function $C_{n,0}(\boldsymbol{x}_n, \boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n))$, i.e. the matching cost between location $\boldsymbol{x}_n$ in image I_n and location $\boldsymbol{x}_n + \boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n)$ in I_0 as defined in Eq. 7.16, Section 7.2.2.

Deciding for each location $\boldsymbol{x}_n$ independently by selecting k such that $C_{n,0}(\boldsymbol{x}_n, \boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n))$ is minimized may result in the introduction of an undesired noise in the final displacement field,

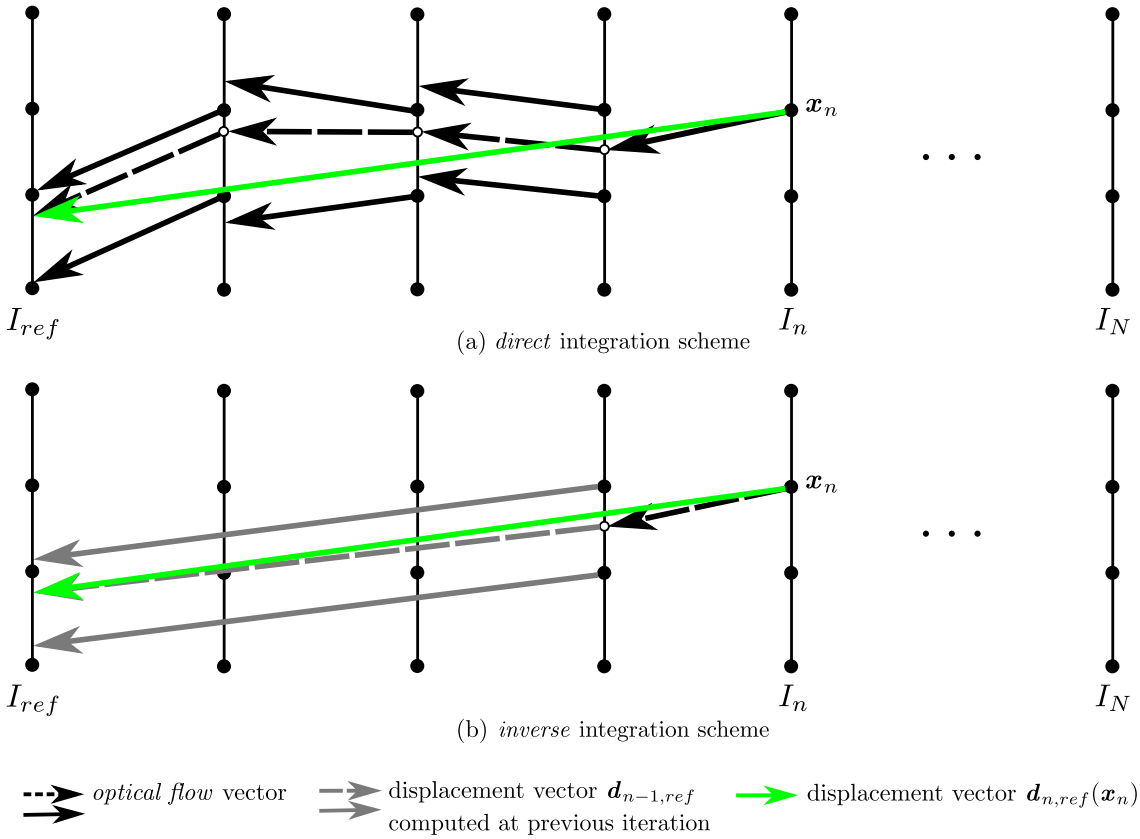


Figure 10.1: Two different integration strategies to estimate the long-term displacement vector $\boldsymbol{d}_{n,ref}(\boldsymbol{x}_n)$: *a*) *direct* integration, i.e. sequential accumulation of *optical flow* vectors starting from I_n (Eq. 10.1), *b*) *inverse* integration where iteratively, a previously estimated long-term displacement vector is interpolated and then accumulated to an *optical flow* vector (Eq. 10.2). Only elementary *optical flow* vectors computed between pairs of consecutive frames are involved in this figure for the sake of simplicity.

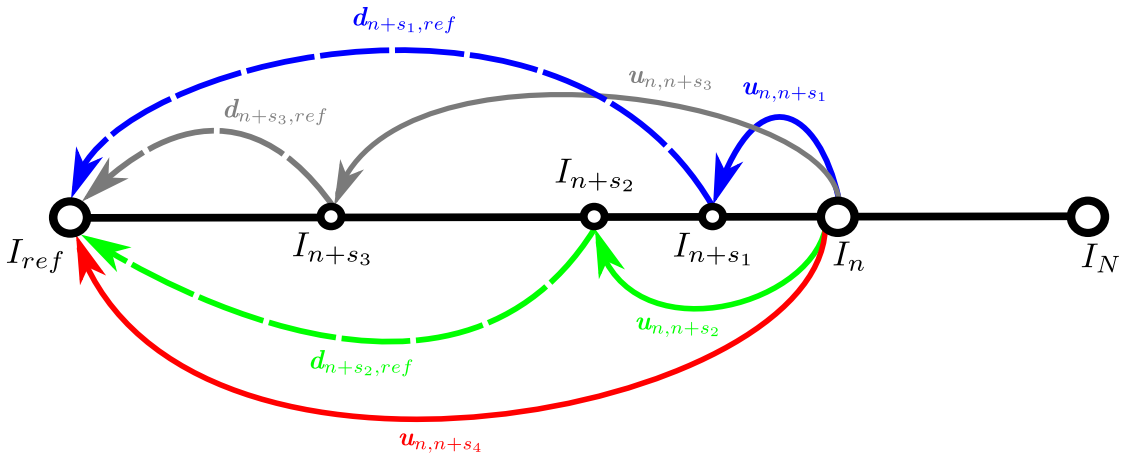


Figure 10.2: *Multi-step* point correspondence involved in *MS-GC*: the displacement from frame I_n to frame I_{ref} is generated following different *paths* according to available *multi-step* elementary *optical flow* fields $\boldsymbol{u}_{n,n+s_k}$ and previously estimated long-term displacements $\boldsymbol{d}_{n+s_k,ref}$.

as neighbouring image points will be frequently assigned with motion values computed with different values of k. Moreover, the proposed cost may not be robust enough. Thus, we improve the result by embedding it together with a spatial *Potts*-like regularization process [WB04]. Let $\boldsymbol{K} = \{k_{\boldsymbol{x}_n}\}$ be a full labelling of the image grid, where each label $k_{\boldsymbol{x}_n}$ indicates one of the available candidate *paths*. We introduce the energy function:

$$E_{n,0}(\boldsymbol{K}) = \sum_{\boldsymbol{x}_n} C_{n,0}(\boldsymbol{x}, \boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n)) - \sum_{<\boldsymbol{x}_n, \boldsymbol{y}_n>} \alpha_{\boldsymbol{x}_n, \boldsymbol{y}_n}.\delta_{k_x = k_y} \tag{10.4}$$

where $< \boldsymbol{x}_n, \boldsymbol{y}_n >$ is a pair of neighbouring image locations according to the 4-point connected neighbourhood, $\delta_{k_{\boldsymbol{x}_n} = k_{\boldsymbol{y}_n}}$ is the *Kronecker* delta and $\alpha_{\boldsymbol{x}_n, \boldsymbol{y}_n}$ the spatial regularization parameter defined in Eq. 10.5.

$$\alpha_{\boldsymbol{x}_n, \boldsymbol{y}_n} = e^{-\frac{1}{\alpha^2} \sum_{c \in \{r,g,b\}} |I^c_n(\boldsymbol{x}_n) - I^c_n(\boldsymbol{y}_n)|^2} \tag{10.5}$$

where $I^c_n(\boldsymbol{x}_n)$ and $I^c_n(\boldsymbol{y}_n)$ are the 3-channel color vectors at locations $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ in image I_n (R,G,B). In practice, α^2 can be estimated locally from the color image or can be set manually. The regularization aims at smoothing the labels assigned to nearby pixels with similar color.

We obtain the optimal $\boldsymbol{K}^*$ by applying a graph-cut-based minimization [BVZ01]. This in turn gives the optimal long-term correspondence vectors:

$$\boldsymbol{d}^*_{n,0}(\boldsymbol{x}_n) = \boldsymbol{d}^{k^*_{\boldsymbol{x}_n}}_{n,0}(\boldsymbol{x}_n) \tag{10.6}$$

10.2 *Multi-step* flow fusion (*MSF*)

We develop significant improvements to the *MS-GC multi-step* flow approach based on three main extensions: 1) we extend the construction of candidate displacement fields by combination of bidirectional (*forward* and *backward*) elementary *optical flows* (Section 10.2.1), 2) we formulate a more robust criterion for fusing flow field candidates (Section 10.2.2), and 3) we develop a new multilateral spatio-temporal filtering method which exploits trajectory-based features to refine long-term correspondence fields (Section 10.2.3).

10.2.1 Inverse integration with bi-directional *paths*

Let us define an initial set of possible step values $S = \{s_1, \ldots, s_Q\}$ with $s_k \in [\![-N, -1]\!] \cup [\![1, N]\!]$. Now, considering the pair $\{I_n, I_0\}$, let $S_n \subset S$ be the plausible subset of steps $S_n = \{s_k \in S | -n \leq s_k \leq N - n\}$ with $|S_n| = Q_n$. As previously, one can compute a displacement field between I_n and I_0 resulting from the combination of the elementary field $\boldsymbol{u}_{n,n+s_k}$ and the displacement $\boldsymbol{d}_{n+s_k,0}$ available between I_{n+s_k} and I_0 as in Eq. 10.3.

The process runs a first pass sequentially from frames I_1 to I_N relying on displacement fields estimated at previous frames. In this case, considered step values are negative (as s_1,s_2,s_3 and s_4 in Fig. 10.3). We propose to extend the set of available steps to positive steps ($s_5 > 0$ in Fig. 10.3), by considering a second pass from frames I_{N-1} to I_1 that takes into account new candidates corresponding to frames m ($m > n$) whose displacement field $\boldsymbol{d}_{m,0}$ was not yet available during the first pass. The novelty is that a correspondence can be built by combining both *forward*

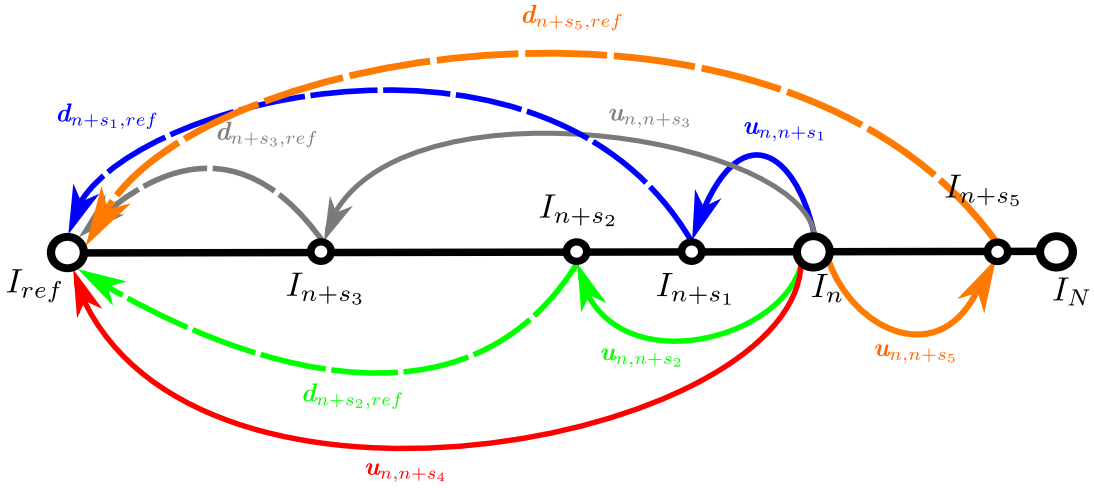


Figure 10.3: Compared to *MS-GC*, the multi-step point correspondence involved in *MSF* extends the set of available *steps* to positive *steps* ($s_5 > 0$). The correspondence is then built by combining both *forward* and *backward* intermediate displacements.

and *backward* intermediate displacements. Not just a matter of adding more candidates, the ability of moving back and forth permits to more appropriately handle temporary occlusions and motion discontinuities.

10.2.2 Optimal *path* selection

The selection of the optimal path for all the points of the grid for a pair $\{I_n, I_0\}$ is achieved via a global optimization stage that fuses all the candidate fields into a single optimal displacement field $\boldsymbol{d}^*_{n,0}$. While a purely discrete model, such as a *Potts*-like energy on the path labels, may seem suitable, such a label-based regularization does not necessarily translate in spatial smoothness of motion. Instead, we propose to minimize Eq. 10.7 instead of Eq. 10.4:

$$E_{n,0}(\boldsymbol{K}) = \sum_{\boldsymbol{x}_n} C_{n,0}(\boldsymbol{x}_n, \boldsymbol{d}^{k_{\boldsymbol{x}_n}}_{n,0}(\boldsymbol{x}_n)) - \sum_{<\boldsymbol{x}_n,\boldsymbol{y}_n>} \alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}.||\boldsymbol{d}^{k_{\boldsymbol{x}_n}}_{n,0}(\boldsymbol{x}_n) - \boldsymbol{d}^{k_{\boldsymbol{y}_n}}_{n,0}(\boldsymbol{y}_n)||_1 \tag{10.7}$$

Compared to Eq. 10.4, the regularization is enforced between the displacement vector values rather than the label values. $\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}$ accounts for colour and motion (motion of *step* 1) spatial similarities:

$$\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n} = K_\alpha.\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{color}.\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{motion} \tag{10.8}$$

where K_α is a positive constant. Both similarities $\alpha^n_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{color}$ and $\alpha^n_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{motion}$ are respectively computed as in Eq. 10.5 and as follows:

$$\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{motion} = exp[-\frac{||\boldsymbol{u}_{n,n\pm1}(\boldsymbol{x}_n) - \boldsymbol{u}_{n,n\pm1}(\boldsymbol{y}_n)||_1}{\eta}] \text{ with } \eta > 0 \tag{10.9}$$

Standard *graph-cut* optimization techniques cannot be applied since the resulting energy does not meet certain necessary conditions [KZ04]. Consequently, we apply the *fusion moves*

algorithm (extension of *α-expansion move* and *$\alpha\beta$-swap move*) recently presented in [LRR08, LRRB10] in the context of instantaneous *optical flow* estimation by flow fusion.

The whole procedure is not applied only toward the computation of *to-the-reference* long-term dense long-term correspondences fields. A exactly similar approach is used to estimate *from-the-reference* long-term dense long-term correspondences fields.

10.2.3 Multilateral spatio-temporal filtering

We propose now to refine the previously obtained long-term correspondences fields through a new multilateral spatio-temporal filtering. Indeed, once *from-the-reference* and *to-the-reference* displacement/trajectory fields exit the *multi-step* fusion stage, they can be advantageously combined in a mutual refinement step. Actually, *from-the-reference* and *to-the-reference* fields $\boldsymbol{d}_{ref,n}$ and $\boldsymbol{d}_{n,ref}$ that have been estimated independently carry complementary, or sometimes contradictory, information. In addition, the trajectory features provided by the *from-the-reference* fields $\boldsymbol{d}_{ref,n}$ $\forall n$ may give useful information about the temporal evolution of the scene content across the sequence. However, these two aspects have not been deeply exploited in the *multi-step* fusion stage (Section 10.2.1 and 10.2.2). That is why we propose a new multilateral spatio-temporal filtering which acts as a motion refinement stage in order to both:

- filter the trajectories in the spatio-temporal domain,
- enforce the consistency between *from-the-reference* and *to-the-reference* long-term displacement fields.

The proposed multilateral spatio-temporal filtering is divided into four steps which are iteratively applied up to convergence or for a given number of times. These four steps are:

1. occlusion detection and inconsistency estimation,
2. a spatio-temporal filtering,
3. a *to-the-reference*/*from-the-reference* multilateral filtering,
4. a *from-the-reference*/*to-the-reference* multilateral filtering.

Let us now describe step by step how our method works.

Occlusion detection and inconsistency estimation The filtering steps are preceded by both occlusion detection and inconsistency estimation which are therefore applied to non-consecutive frames.

Occlusion detection is done following the approach based on *backward* projection illustrated in Fig. 7.1 (Section 7.3) for consecutive *optical flow* vectors. Thus, for a given pair $\{I_{ref} = I_0, I_n\}$, pixels of I_{ref} (resp. I_n) occluded in I_n (resp. I_{ref}) are detected through projection of *to-the-reference* (resp. *from-the-reference*) long-term displacement fields.

Inconsistency estimation is related to the left-right disparity checking (LRC) presented in [EW02] in the context of disparity estimation (Fig. 15.2 in *Appendix A*). Considering an image pair $\{I_{ref}, I_n\}$ and for the *from-the-reference* direction for instance, the inconsistency value assigned to each pixel $\boldsymbol{x}_{ref}$ of I_{ref}, noted as $Inc(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}))$, is measured via a

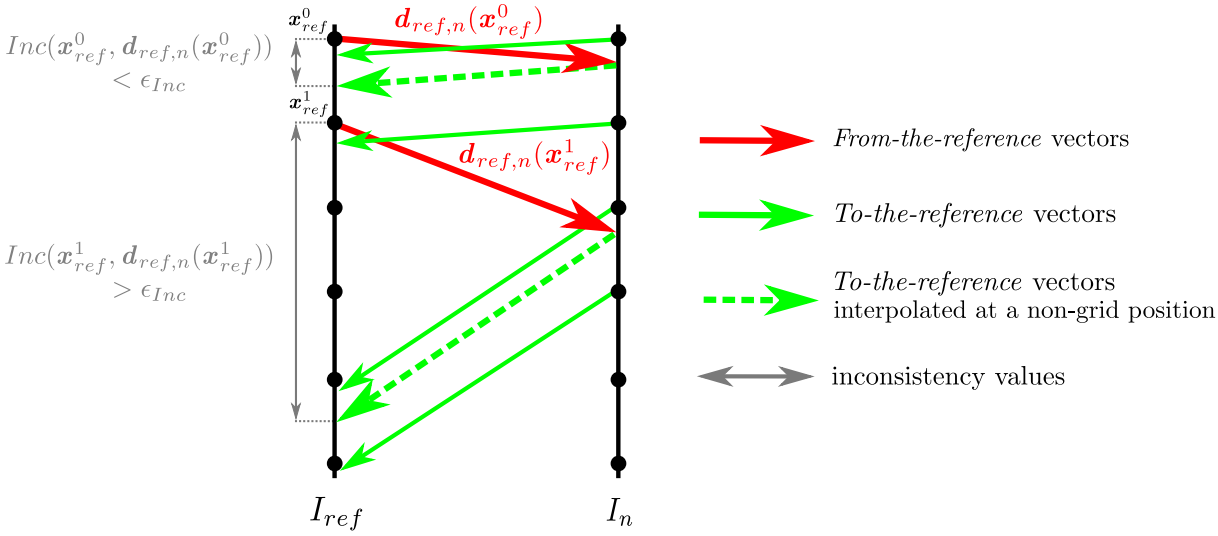


Figure 10.4: Inconsistency estimation: the inconsistency value assigned to each pixel $\boldsymbol{x}_{ref}$ of I_{ref} is measured via a comparison between the *from-the-reference* displacement vector $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ starting from I_{ref} and its corresponding *to-the-reference* displacement vector starting from I_n.

comparison between the *from-the-reference* displacement vector $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ starting from I_{ref} and its corresponding *to-the-reference* displacement vector starting from I_n. As shown in Eq. 10.10, $Inc(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}))$ corresponds more precisely to the *Euclidean* distance between: 1) $\boldsymbol{x}_{ref} \in I_{ref}$, the starting point of the *from-the-reference* displacement vector $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$, 2) the ending point of the *to-the-reference* displacement vector $\boldsymbol{d}_{n,ref}$ starting from the position $\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ in I_n.

$$Inc(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})) = ||[\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}) + \tilde{\boldsymbol{d}}_{n,ref}(\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}))] - \boldsymbol{x}_{ref}||_2 \quad (10.10)$$

The *to-the-reference* displacement vector $\tilde{\boldsymbol{d}}_{n,ref}(\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}))$ is probably interpolated because it may start at a non-grid position. The resulting consistency value can be binarized through the comparison to a threshold ϵ_{inc} which distinguishes consistent and inconsistent pixels. In practice, the interpolation of the *to-the-reference* vectors is done through bilinear interpolation using the *to-the-reference* vectors starting from the four nearest pixels with respect to $\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ in I_n.

Fig. 10.4 illustrates both cases: $\boldsymbol{d}_{ref,n}(\boldsymbol{x}^0_{ref})$ is consistent because $Inc(\boldsymbol{x}^0_{ref}, \boldsymbol{d}_{ref,n}(\boldsymbol{x}^0_{ref})) < \epsilon_{Inc}$ and $\boldsymbol{d}_{ref,n}(\boldsymbol{x}^1_{ref})$ is inconsistent because $Inc(\boldsymbol{x}^1_{ref}, \boldsymbol{d}_{ref,n}(\boldsymbol{x}^1_{ref})) > \epsilon_{Inc}$. For point tracking applications, such inconsistency estimation has proved in [KMM10] to enable a reliable detection of tracking failures. Note that it is called *forward-backward* consistency in [KMM10]. Moreover, the estimation of inconsistency values for *to-the-reference* displacement vectors is done analogously as for *from-the-reference* displacement vectors.

Spatio-temporal filtering For all pairs $\{I_{ref} = I_0, I_n\}$, *from-the-reference* displacement fields $\boldsymbol{d}_{ref,n}$ are spatio-temporally filtered considering the trajectories of spatial neighbouring pixels in the reference frame I_{ref}. This step increases the consistency in terms of trajectory behaviour for neighbouring pixels.

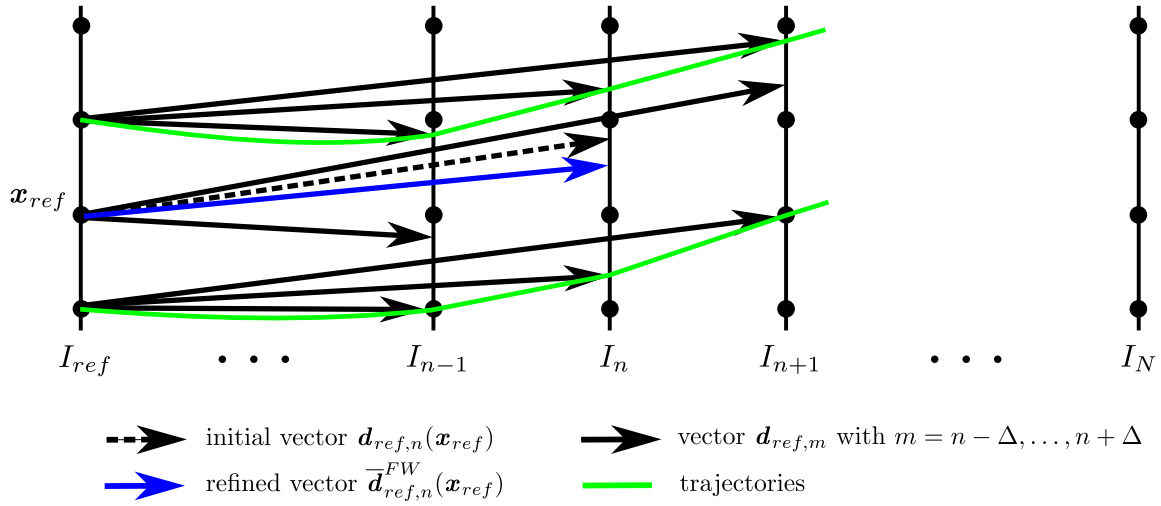


Figure 10.5: Spatio-temporal filtering applied after the *multi-step* fusion stage in order to filter the trajectories in the spatio-temporal domain. Δ defines a temporal window around I_n.

The trajectory aspect of the *from-the-reference* fields is more precisely considered in two ways. First, a trajectory similarity weight between neighbouring pixels in I_{ref} is introduced. Second, as shown in Fig. 10.5, displacement fields with respect to neighbouring frames are taken into account in the filtering process. *From-the-reference* displacement vectors $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$ are iteratively filtered considering the neighbouring *from-the-reference* vectors $\boldsymbol{d}_{ref,m}(\boldsymbol{y}_{ref})|_{m:n-\Delta,\ldots,n+\Delta}$ where Δ defines a temporal window and where $\boldsymbol{y}_{ref}$ indicates a grid position located in the neighborhood of $\boldsymbol{x}_{ref}$.

This first filtering step is finally defined as follows:

$$\overline{\boldsymbol{d}}^{FW}_{ref,n}(\boldsymbol{x}_{ref}) = \frac{\sum_{m=n-\Delta}^{n+\Delta} \sum_{\boldsymbol{y}_{ref}\in\mathcal{N}(\boldsymbol{x}_{ref})} w_{traj}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}}.w_{ref,m}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}}.\frac{n}{m}.\boldsymbol{d}_{ref,m}(\boldsymbol{y}_{ref})}{\sum_{m=n-\Delta}^{n+\Delta} \sum_{\boldsymbol{y}_{ref}\in\mathcal{N}(\boldsymbol{x}_{ref})} w_{traj}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}}.w_{ref,m}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}}} \tag{10.11}$$

where $\mathcal{N}(\boldsymbol{x}_{ref})$ defines a spatial neighbourhood around $\boldsymbol{x}_{ref}$. Each vector in neighbouring frames ($m \neq n$) is weighted by a scaling factor $\frac{n}{m}$ in order to make the displacement fields $\boldsymbol{d}_{ref,m}(\boldsymbol{y}_{ref})|_{m:n-\Delta,\ldots,n+\Delta}$, that correspond to different temporal distances, comparable. $w_{s,t}^{\boldsymbol{xy}}$ is a weight that links points $\boldsymbol{x}$,$\boldsymbol{y}$ at frame I_s based on their motion with respect to frame I_t:

$$w_{s,t}^{\boldsymbol{xy}} = \rho_{s,t}.exp[-(\frac{||\boldsymbol{x}-\boldsymbol{y}||_2^2}{\gamma} + \frac{\sum_{c\in\{r,g,b\}} |I_s^c(\boldsymbol{x}) - I_s^c(\boldsymbol{y})|}{\varphi} + \frac{\sum_{c\in\{r,g,b\}} |I_s^c(\boldsymbol{y}) - I_t^c(\boldsymbol{y}+\boldsymbol{d}_{s,t}(\boldsymbol{y}))|}{\theta})] \tag{10.12}$$

This weight $w_{s,t}^{\boldsymbol{xy}}$ combines spatial distance, colour similarity and matching cost (Section 7.2.2, Chapter 7). It involves $I_s^c(\boldsymbol{x})$ which corresponds to a *RGB* component at location $\boldsymbol{x}$ in image I_s. $\rho_{s,t}$ is a binary value that is 1 if the point is not detected as occluded, 0 otherwise. γ, φ and

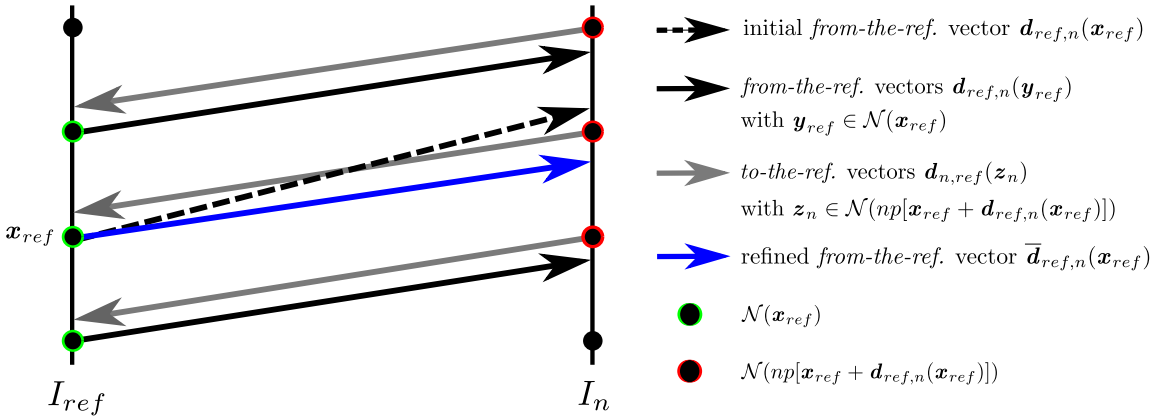


Figure 10.6: Illustration of the *from/to-the-reference* multilateral filtering performed to enforce the consistency between *from-the-reference* and *to-the-reference* long-term displacement fields. $\mathcal{N}(\boldsymbol{x})$ indicates the spatial neighbourhood around $\boldsymbol{x}$. $np[.]$ means *nearest pixel*.

θ are positive constants used to adjust the different components of $w_{s,t}^{\boldsymbol{x}\boldsymbol{y}}$. For uniform areas we set $\gamma \to \infty$ and φ is increased to limit the effect of pixels with a different colour value. Pixels in I_s that belong to uniform areas are those for which:

$$\sum_{\boldsymbol{y} \in \mathcal{N}_{\{\boldsymbol{x}\}}} exp[-\frac{[\sum_{c \in \{r,g,b\}} |I_s^c(\boldsymbol{x}) - I_s^c(\boldsymbol{y})|]^2}{\xi}] > 0.5 \text{ with } \xi > 0 \tag{10.13}$$

The weight $w_{traj}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}}$ in Eq. 10.11 derives from the similarity of the trajectories that support the two currently compared *from-the-reference* vectors. It is defined over the whole sequence, as follows:

$$w_{traj}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}} = exp[-\frac{\sum_{m=1}^{N} ||\boldsymbol{d}_{ref,m}(\boldsymbol{x}_{ref}) - \boldsymbol{d}_{ref,m}(\boldsymbol{y}_{ref})||_2^2}{\psi}] \text{ with } \psi > 0 \tag{10.14}$$

***To/from-the-reference* multilateral filtering** Once *from-the-reference* displacement fields have been spatio-temporally filtered, *from-the-reference* and *to-the-reference* displacement fields $\overline{\boldsymbol{d}}_{ref,n}^{FW}$ and $\boldsymbol{d}_{n,ref}$ are jointly processed via multilateral filtering which propagates iteratively the refinement from the *from-the-reference* direction (Eq. 10.11) to the *to-the-reference* direction. Noting $\boldsymbol{z}_{ref} = \boldsymbol{x}_n + \boldsymbol{d}_{n,ref}(\boldsymbol{x}_n)$, the updated *to-the-reference* displacement field $\overline{\boldsymbol{d}}_{n,ref}$ is:

$$\overline{\boldsymbol{d}}_{n,ref}(\boldsymbol{x}_{ref}) = \frac{\sum_{\boldsymbol{y}_n \in \mathcal{N}(\boldsymbol{x}_n)} w_{n,ref}^{\boldsymbol{x}_n\boldsymbol{y}_n}.\boldsymbol{d}_{n,ref}(\boldsymbol{y}_n) - \sum_{\boldsymbol{y}_{ref} \in \mathcal{N}(np[\boldsymbol{z}_{ref}])} w_{ref,n}^{\boldsymbol{z}_{ref}\boldsymbol{y}_{ref}}.\overline{\boldsymbol{d}}_{ref,n}^{FW}(\boldsymbol{y}_{ref})}{\sum_{\boldsymbol{y}_n \in \mathcal{N}(\boldsymbol{x}_n)} w_{n,ref}^{\boldsymbol{x}_n\boldsymbol{y}_n} + \sum_{\boldsymbol{y}_{ref} \in \mathcal{N}(np[\boldsymbol{z}_{ref}])} w_{ref,n}^{\boldsymbol{z}_{ref}\boldsymbol{y}_{ref}}} \tag{10.15}$$

where $np[\boldsymbol{z}_{ref}]$ (*nearest pixel* with respect to $\boldsymbol{z}_{ref}$) indicates that the spatial neighbourhood is made of grid positions around $\boldsymbol{z}_{ref}$ in I_{ref}. The weights are defined as in Eq. 10.12. However,

a motion vector similarity term replaces the trajectory similarity involved in Eq. 10.14 because trajectories are not available in the *to-the-reference* direction.

***From/to-the-reference* multilateral filtering** The *from-the-reference* vectors from the spatio-temporal filtering are filtered again, this time through a *from/to-the-reference* multilateral filtering. This third stage, illustrated in Fig. 10.6, is similar to the *to/from-the-reference* multilateral filtering we just described in Eq. 10.15 which means that it propagates the refinement from the *to-the-reference* direction to the *from-the-reference* direction. Noting $\boldsymbol{z}_n = \boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref})$, the updated *from-the-reference* displacement field $\overline{\boldsymbol{d}}_{ref,n}$ becomes:

$$\overline{\boldsymbol{d}}_{ref,n}(\boldsymbol{x}_{ref}) = \frac{\sum\limits_{\boldsymbol{y}_{ref} \in \mathcal{N}(\boldsymbol{x}_{ref})} w_{ref,n}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}} . \overline{\boldsymbol{d}}_{ref,n}^{FW}(\boldsymbol{y}_{ref}) - \sum\limits_{\boldsymbol{y}_n \in \mathcal{N}(np[\boldsymbol{z}_n])} w_{n,ref}^{\boldsymbol{z}_n\boldsymbol{y}_n} . \overline{\boldsymbol{d}}_{n,ref}(\boldsymbol{y}_n)}{\sum\limits_{\boldsymbol{y}_{ref} \in \mathcal{N}(\boldsymbol{x}_{ref})} w_{ref,n}^{\boldsymbol{x}_{ref}\boldsymbol{y}_{ref}} + \sum\limits_{\boldsymbol{y}_n \in \mathcal{N}(np[\boldsymbol{z}_n])} w_{n,ref}^{\boldsymbol{z}_n\boldsymbol{y}_n}} \tag{10.16}$$

Finally, at the end of each pass (i.e. spatio-temporal, *to/from-the-reference* or *from/to-the-reference* filtering), an *a posteriori* choice between un-filtered and filtered vectors is performed with respect to the matching cost while encouraging filtering to some extent. For this task, we apply the global optimization proposed in Eq. 10.7 in order to fuse un-filtered and filtered vectors.

Regarding the whole iterative filtering approach, we give more confidence to consistent values in a soft manner. Thus, every 3 iterations, the whole process is applied to the totality of the vectors while, for the other iterations, it is limited to those for which the inconsistency value is above the threshold ϵ_{inc}.

10.3 Experiments

We propose different types of experiments to assess the performance of our methods along with comparisons with state-of-the-art approaches. We compare our *MS-GC* and *MSF multi-step* fusion approaches with respect to classic accumulation (i.e. *direct* integration) of *optical flow* estimated between consecutive frames. The *optical flow* estimators involved in our experiments are the following:

- *TV-L1 optical flow* method [ZPB07] (Section 9.3.2). The classic accumulation of *TV-L1 optical flow* is referred to as *TV-L1 acc*,
- *Large Displacement Optical Flow* (*LDOF*) [BM11] (Section 9.3.4), which gives *LDOF acc*,
- *2D-DE*, an adapted 2D version of the 1D disparity estimator of [RTDC12] which leads to *2D-DE acc* once the consecutive *optical flow* have been accumulated.

We also test *ParticleVideo* (PV), the semi-dense long-term motion estimator from [ST06, ST08] described in Section 10.1.3. The *2D-DE* estimator has been also involved through *inverse* integration of consecutive *optical flows*: *2D-DE inverse*.

The *MS-GC* and *MSF multi-step* fusion approaches have been performed using *multi-step* elementary *optical flow* fields estimated with *2D-DE* as inputs. The corresponding results can be identified by the following denominations: *MS-GC*(*2D-DE*) or *MSF*(*2D-DE*). Note that the *multi-step* fusion flow approach without filtering (*MSF*), with spatial filtering (*MSF+SF*) (i.e. considering only the *to/from-the-reference* and the *from/to-the-reference* multilateral filtering stages of Section 10.2.3) or with the whole multilateral spatio-temporal filtering of Section 10.2.3 (*MSF+STF*) are also compared in the experiments. The set of *steps* will be specified for each sequence. Nevertheless, note that the set of input elementary optical flow fields is manually selected as to handle a rich variety of situations within each video sequence. Therefore, it may depend on the video content.

Before entering into details, let us precise and show which video sequences have been used in this Section 10.3. We focus in particular on the following sequences in order to cover a rich set of characteristics:

- *AmeliaRetro* (courtesy of *Dolby*): 100 frames (1920×1080) containing zooming, occlusions, non-rigid deformations and spatial lighting variations: Fig. 10.7,
- *Newspaper* (*MPEG* sequence): 100 frames (1024×768) featuring temporary occlusions, fixed background, appearing background object, illumination variations, shadows and low colour contrast between different motion regions: Fig. 10.8,
- *Dirk-Hartmut* between I_{96} and I_{110} (1024×540) to focus on temporary occlusion with fixed camera and large motion: Fig. 10.9.
- *Water-Marshall-37* between I_{1329} and I_{1358} (1920×1080) featuring zooming, large uniform areas and spatial lighting variations: Fig. 10.10.

Concerning parameter specification, $C_{n,0}(\boldsymbol{x}, \boldsymbol{d}^k_{n,0}(\boldsymbol{x}_n))$ in Eq. 10.4 and 10.7 is computed as the mean absolute difference (*MAD*) of pixel colour values between image windows of size 5×5. For 8-bit colour components, the value of *MAD* is then truncated to a maximum of 128 in order

(a) I_0 (b) I_{20} (c) I_{40}

(d) I_{60} (e) I_{80} (f) I_{100}

Figure 10.7: Source frames of the *AmeliaRetro* sequence provided by *Dolby*.

(a) I_0 (b) I_{20} (c) I_{40} (d) I_{60} (e) I_{80} (f) I_{100}

Figure 10.8: Source frames from a cropped version of the *Newspaper* sequence.

(a) I_{96} (b) I_{104} (c) I_{110}

Figure 10.9: Source frames of the *Dirk-Hartmut* sequence.

(a) I_{1329} (b) I_{1338} (c) I_{1348} (d) I_{1358}

Figure 10.10: Source frames of the *Water-Marshall*-37 sequence.

to robustify the measurement. Moreover, the effect of illumination variations and shadows is attenuated in the *Newspaper* sequence by normalizing each colour pixel of the input images by a local mean intensity. Moreover, $\alpha = 300$, $K_\alpha = 20$ and $\eta = 10$, respectively in Eq. 10.5, 10.8 and 10.9.

Regarding multilateral filtering, the spatial and temporal windows are respectively of size 7×7 and 3. The number of iterations has been empirically set to 19. Moreover, $\gamma = 200$, $\varphi = 600$ (1000 if the corresponding pixel belongs to a uniform area), $\theta = 600$, $\xi = 200$ and $\psi = 5 \times N$. The threshold for the inconsistency evaluation equals to $\epsilon_{Inc} = 1$ pixel. Δ in Eq. 10.11 equals to 1. Finally, the global optimization described in Eq. 10.7 is applied to fuse unfiltered and filtered vectors with $\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n} = K_\alpha.\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{color}$ where $K_\alpha = 20$ and $\alpha_{\boldsymbol{x}_n,\boldsymbol{y}_n}|_{color}$ is defined in Eq. 10.5.

In order to assess the computation time of the *MSF* process chain, we have conducted an experiment given an input sequence with 100 frames of 400×400 pixels. On average, it takes 2 seconds per frame and per candidate *path* to perform the construction of the energy and the global optimization. That is, with c candidate paths, the fusion process takes $\approx 2.c$ seconds. Regarding multilateral filtering applied with the parameters described above, around 90 seconds per frame are required.

All these methods, sequences and parameters are involved through the following experiments. Quantitative trajectory assessment as well as trajectory visualization are performed in Section 10.3.1. Registration and *PSNR* assessment are involved in Section 10.3.2. Section 10.3.3 illustrates how the parametric motion fields can improve the accuracy of our long-term displacement fields. In a more applicative point of view, the comparisons are carried out through editing tasks (Section 10.3.4) and key-frame based video segmentation (Section 10.3.5). Finally, we give in Section 10.3.6 some clues to improve the results and to introduce the objectives of Chapter 11.

10.3.1 Trajectory quality assessment

Our first experiment in terms of trajectory quality assessment was to pick 8 points in I_0 of the *AmeliaRetro* sequence (Fig. 10.7), carefully selected as to account for textured and non-textured areas. We manually generated the ground truth trajectories along the 100 frames. We then measured the frame-by-frame position error for several methods as depicted in Fig. 10.11. We plot the median error among the 8 points at each instant for *TV-L1 acc*, *LDOF acc*, *2D-DE inverse*, *MS-CG*(*2D-DE*), *MSF*(*2D-DE*), *MSF+SF*(*2D-DE*) and *MSF+STF*(*2D-DE*).

The *multi-step* methods have been performed with the following set of candidate *steps*: $S = \{\pm1, \pm2, \pm5, \pm10, \pm20, \pm30, \pm40, \pm50, \pm100\}$. Due to the predominance of camera motion in the *AmeliaRetro* sequence, we have also complemented the set of non-parametric *optical flow* fields with additional affine motion fields for steps $\{\pm10, \pm20, \pm30, \pm50, \pm80\}$ (see Section 10.3.3 for further details). We draw the following remarks from the plot:

- the three methods based on consecutive *optical flow* integration (*TV-L1 acc*, *LDOF acc*, *2D-DE inverse*) are the worst performing, supporting our claim that high precision in instantaneous motion estimation does not guarantee long-term tracking accuracy,

- *multi-step* methods start to perform better than state-of-the-art methods after $\approx$ 30 frames. This duration is coherent with the maximum track length used in the state-of-the-art methods such as [BM10, LASL11, SBK10],

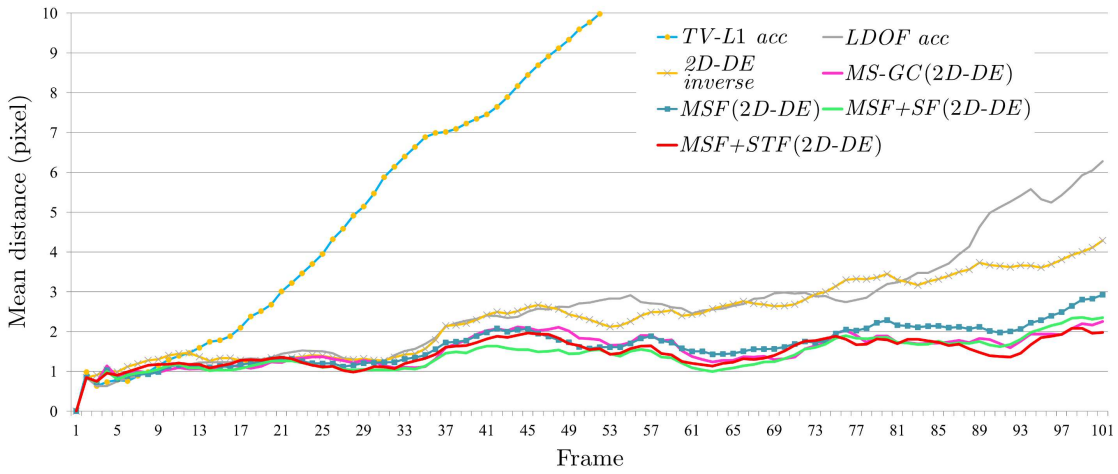


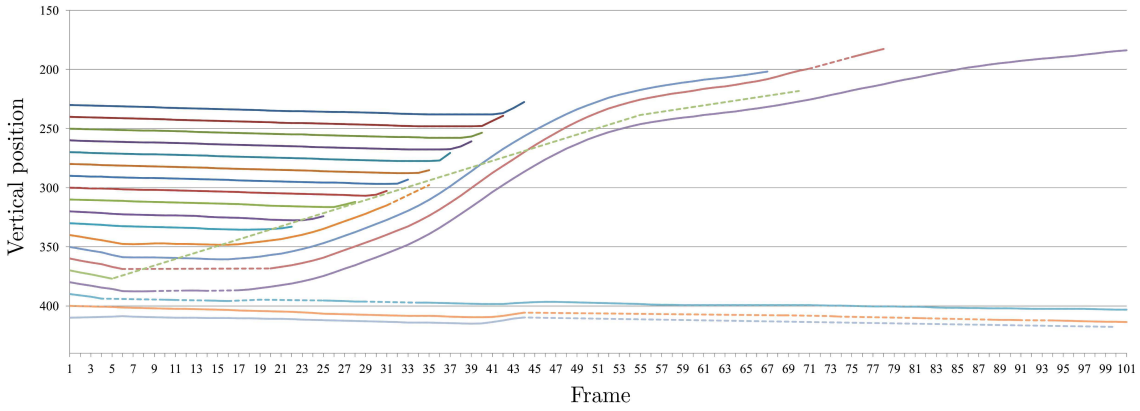
Figure 10.11: Trajectory quality assessment through comparisons with 8 ground-truth trajectories manually tracked in the *AmeliaRetro* sequence. We compare the following methods: *TV-L1 acc* [ZPB07], *LDOF acc* [BM11], *2D-DE inverse*, *MS-GC*(*2D-DE*), *MSF*(*2D-DE*), *MSF+SF*(*2D-DE*) and *MSF+STF*(*2D-DE*).

- the most accurate method is *MSF+STF*(*2D-DE*), specially noting how the position error at frame I_{63} is as small as in frame I_7. The optimal combination of short and long term matching does its job reducing the motion drift.

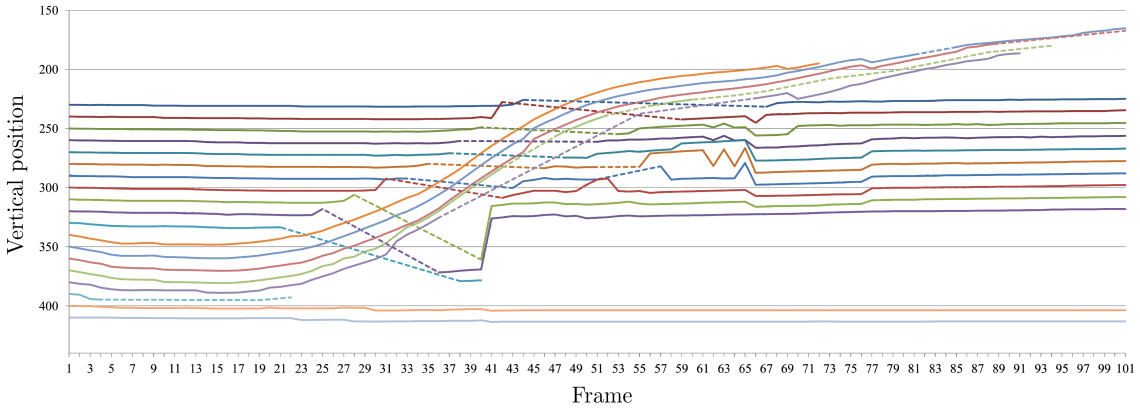
The second experiment consists in analyzing the complex situation of a temporary occlusion in the *Newspaper* sequence where the arm and the cup occlude the chest (see Fig. 10.8). A total of 19 points were tracked, equally spaced by 10 pixels on a vertical line that passes through the chest and the hand. Fig. 10.12 plots the vertical components of successive positions of trajectories estimated with *TV-L1 acc*, *MSF*(*2D-DE*) and *MSF+STF*(*2D-DE*). Since the camera does not move, this experiment is similar to a comparison with ground-truth trajectories as if they were implicitly given. For *Newspaper*, we used as previously $S = \{\pm1, \pm2, \pm5, \pm10, \pm20, \pm30, \pm40, \pm50, \pm100\}$ without additional parametric motion fields.

While for single step methods (Fig. 10.12 (*a*) with *TV-L1 acc*) it is impossible to estimate the trajectories of the occluded pixels after the occlusion (the tracks follow to the motion of the hand), the *MSF* algorithm is able to circumvent the problem thanks to the long-step input motion fields (Fig. 10.12 (b)). Actually, track segments before and after the occlusion are naturally linked together as each position refers to the same reference point. In Fig. 10.12 (c), we notice that the filtering step of *MSF+STF*(*2D-DE*) improves the temporal consistency of trajectories which become smoother.

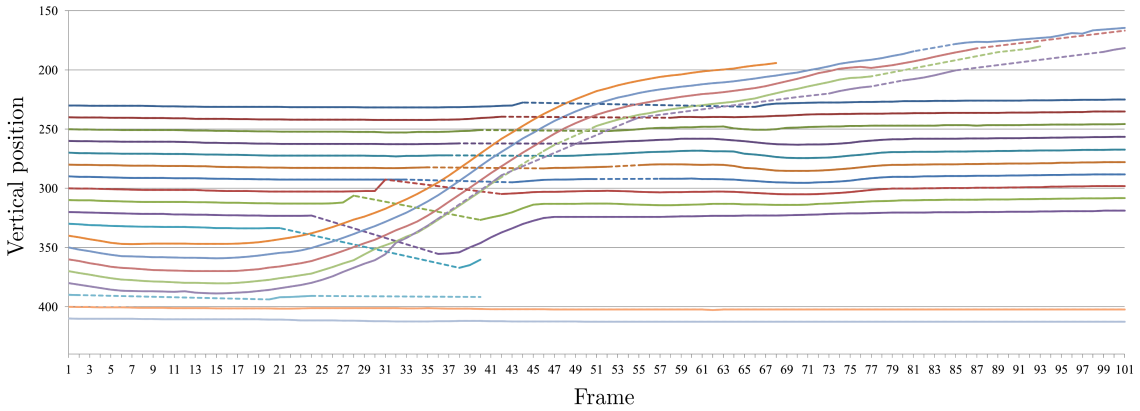
Finally, an ingenious way of quantitatively assessing the quality of the estimated trajectories is by mirroring a sequence in time, i.e., constructing $\{I_n\}_{n=\{0,1,\ldots,N,\ldots,1,0\}}$ and checking the symmetry of the tracks [ST06, ST08]. For a given point, the departing location is known to be identical to the arriving position. We go further by testing the same condition for all the pairs of mirrored instants. The *AmeliaRetro* sequence was cropped to a meaningful area of 768×675 pixels and mirrored taking the first 50 frames to generate a new 100 frame video: *AmeliaRetro-Mirror*. *MSF*(*2D-DE*) is tested on this sequence and compared to *ParticleVideo* and *MS-CG*(*2D-DE*) with $S = \{\pm1, \pm2, \pm5, \pm10, \pm20, \pm30, \pm40, \pm50, \pm100\}$.



(a) *TV-L1 acc*



(b) *MSF(2D-DE)*



(c) *MSF+STF(2D-DE)*

Figure 10.12: Vertical component of 19 estimated trajectories computed on the *Newspaper* sequence with three different methods: (a) *TV-L1 acc* [ZPB07], (b) *MSF(2D-DE)*, (c) *MSF+STF(2D-DE)*. The points were selected equally spaced on a vertical line that passes through the chest and the arm (Fig. 10.8).

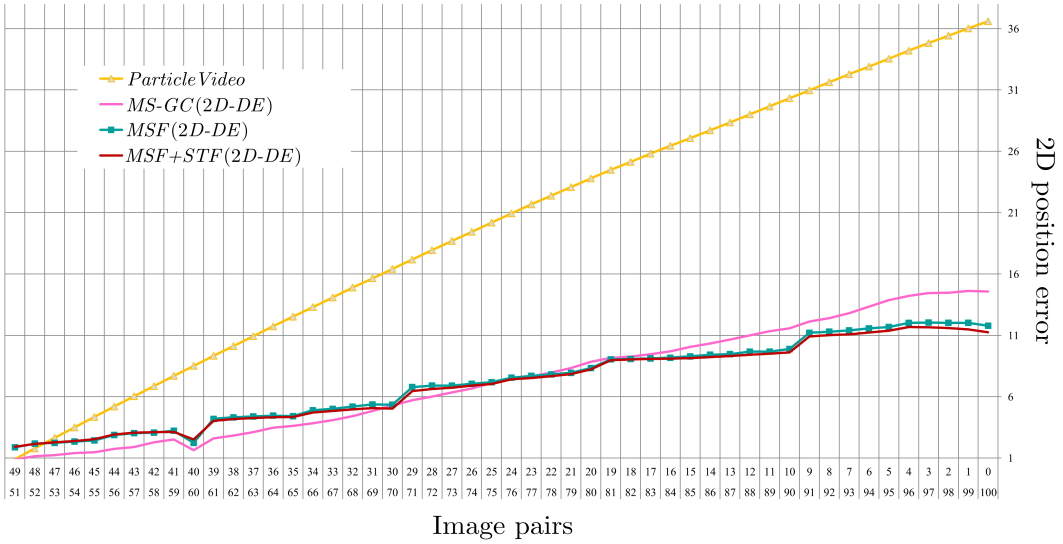


Figure 10.13: Trajectory symmetry checking on the *AmeliaRetro-Mirror* sequence. Position errors are computed by comparing the following pairs: $\{\{49, 51\}, \{48, 52\}, \ldots, \{0, 100\}\}$. We compare *ParticleVideo* [ST06, ST08] with the proposed *MS-GC*(2*D-DE*), *MSF*(2*D-DE*) and *MSF+STF*(2*D-DE*) approaches.

In Fig. 10.13, we observe the improvement obtained in terms of precision. Moreover, *multi-step* methods obtain the full-length tracks for 100% (518400) of the image points in I_0 while *ParticleVideo* is only able to estimate 0.2% full-length tracks, and initially selects only 0.5% (2610) points in the first image. Higher accuracy together with full density clearly shows the benefits of our approach.

10.3.2 Long-term warping

Another experiment consists in performing image registration, i.e. reconstructing the reference image I_0 from each image I_n of a video sequence exploiting dense long-term point correspondences. This is a very challenging task which permits to obtain a global measure of the performance of an algorithm. Indeed, large colour differences between both registred and original images clearly show defective correspondences. This is achieved for *AmeliaRetro* by copying the colour values from I_n according to the displacement field $\boldsymbol{d}_{0,n}(\boldsymbol{x})$, as illustrated in Fig. 10.16 where a block within the dress of *AmeliaRetro* in I_0 is reconstructed from I_{100} with *LDOF acc*, 2*D-DE acc* and *MSF+STF*(2*D-DE*). Note that this corresponds to a *from-the-reference* strategy.

We also compute the color *PSNR* in this area of the dress for each reconstructed reference frame from I_n with $n = 1, 2, \ldots, 100$ with respect to the reference image I_0 (Fig. 10.14).

In Fig. 10.14 and 10.16, we observe that *multi-step* approaches are clearly better for reconstruction than standard *optical flow* integration (see especially the results with *LFOD acc* in Fig. 10.16 (b)). And among them, the improvement of the *MSF* and *MSF+STF* methods is significant especially with respect to *MS-GC*. Although not shown here, the *PSNR* assessment for the *to-the-reference* strategy (reconstruction of I_n from I_0) gives also good results.

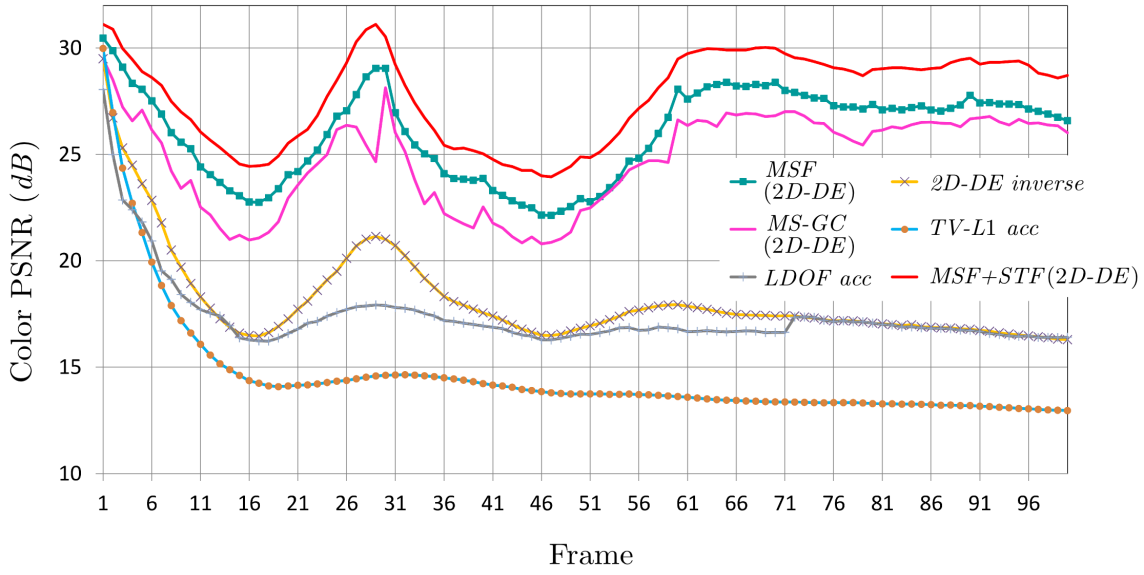


Figure 10.14: Image registration and *PSNR* assessment in a block within the dress of the *AmeliaRetro* sequence. We compare the following methods: *TV-L1 acc* [ZPB07], *LDOF acc* [BM11], *2D-DE inverse*, *MS-GC*(*2D-DE*), *MSF*(*2D-DE*) and *MSF+STF*(*2D-DE*).

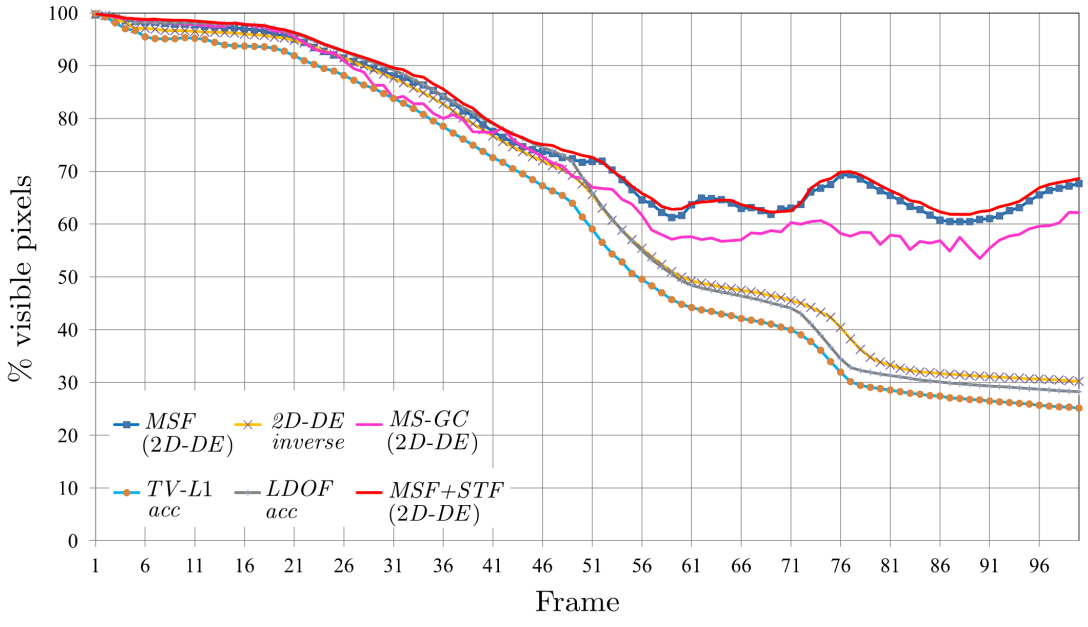


Figure 10.15: Percentage of visible points along the *Newspaper* sequence. We compare the following methods: *TV-L1 acc* [ZPB07], *LDOF acc* [BM11], *2D-DE inverse*, *MS-GC*(*2D-DE*), *MSF*(*2D-DE*) and *MSF+STF*(*2D-DE*).

(a) (b) (c) (d)

Figure 10.16: Reconstruction of the dress of *AmeliaRetro* in I_0 (a) from I_{100} through motion compensation using estimations from: (b) *LDOF acc* [BM11], (c) *2D-DE acc* [RTDC12], (d) *MSF+STF(2D-DE)*.

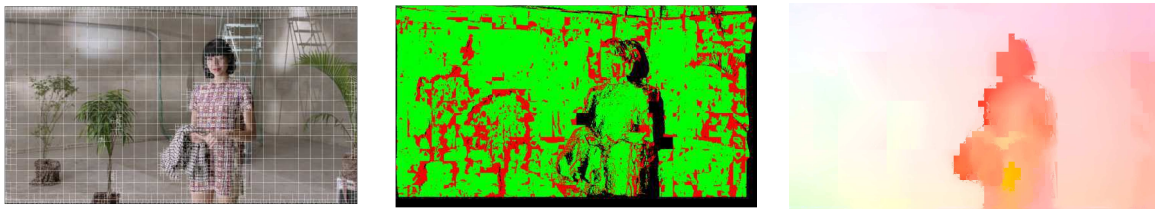
The same experiment was conducted for the *Newspaper* sequence. However, given that the colour of moving regions is basically the same (i.e. blue), the curves were not meaningful for assessing the accuracy of the correspondence estimation. Indeed, relying on registration and *PSNR* assessment for uniform areas has strong limitations since the color similarity prevents from detecting defective motion correspondences. On the other hand, it is interesting to show the behaviour of each method in front of the temporary occlusions caused by the arm and cup. We thus plot in Fig. 10.15 the percentage of visible points detected by each method along the sequence. This illustrates the fact that our method is able to recover reappearing points while for single step methods, the number of visible points decreases monotonically.

10.3.3 Additional parametric motion fields

When necessary, the set of input *multi-step* elementary *optical flow* fields are complemented with a set of parametric motion fields. Indeed, it appears that for some cases, especially when the camera motion is predominant, parametric (affine or homographic in particular) motion fields can perform an more efficient motion estimation than non-parametric *optical flow* fields for planar surfaces. We use in practice both *quadtree* decomposition and *RANSAC* algorithms (Section 7.2.7, Chapter 7) to create parametric motion fields.

Starting from *multi-step* elementary *optical flow* field $\boldsymbol{d}_{s,t}$ between I_s and I_t (see Fig. 10.17 (a,b,d)), we propose to compute an affine (see $\boldsymbol{d}_{s,t}^{aff}$ in Fig. 10.17 (g)) or homographic motion field by first decomposing I_s using a *quadtree* decomposition (Fig. 10.17 (e)) applied on the gradient image corresponding to I_s (Fig. 10.17 (c)). As shown in Fig. 10.17 (e), the *quadtree* decomposition gives patches of varying size (from $l_{min} \times l_{min}$ to $l_{max} \times l_{max}$) which are large for uniform areas and small for high-textured ones. Then, for each rectangular patch p_i, we perform the *RANSAC* algorithm using the *optical flow* vectors $\boldsymbol{d}_{s,t}(\boldsymbol{x}_s)$ with $\boldsymbol{x}_s \in p_i$ in order to compute the 6 (respectively 8) parameters of the affine (resp. homographic) model which best suits the deformation of the patch p_i from I_s to I_t. In practice, the *RANSAC* algorithm randomly takes a subset of all the *optical flow* vectors $\boldsymbol{d}_{s,t}(\boldsymbol{x}_s)$ with $\boldsymbol{x}_s \in p_i$, computes the parameters of the model, applies this model to compute a parametric displacement for each $\boldsymbol{x}_s \in p_i$ and finally estimates a residual error to inform about the warping quality of the whole patch p_i by the current model. After having repeated this process N_{ransac} times, the process selects the model which leads to the best parametric warping. In 10.17 (f), green areas denote *inliers*, i.e. pixels for which the computed model suits the initial *optical flow* vectors with respect to a quality criterion (threshold ϵ_{ransac} on the distance between the end-points of the initial *optical flow*

(a) Original image I_0 (b) Original image I_{10} (c) Gradient image I_0 (d) *Optical flow* $\boldsymbol{d}_{0,10}$



(e) *Quadtree* decomp. in I_0 (f) *RANSAC inliers/outliers* (g) Affine field $\boldsymbol{d}^{aff}_{0,10}$

Figure 10.17: The set of input *multi-step* elementary *optical flow* fields are complemented with a set of parametric (affine or homographic) motion fields through *quadtree* decomposition and RANSAC algorithm.

vector and the parametric displacement vector estimated via the current model). Following the same criterion, red areas indicates *outliers*, i.e. pixels for which the model is not suitable.

Our algorithm is not well adapted in the case of rectangular patches straddling two objects with different displacements. Indeed, we compute only one single motion model which means that, for these patches, only one deformation is modelled in the better case and not two deformations as expected. This explains the presence of red areas near object boundaries in Fig 10.17 (f) and the *blocking effect* in Fig 10.17 (g), especially around the head of the lady. However, for these regions, we expect that non-parametric *optical flows* gain the upper hand against parametric fields during the *fusion* stage of *MSF*, Eq. 10.7.

To conclude, we propose when necessary to compute for each input elementary *optical flow* field a corresponding parametric version which, once computed, acts as an additional input. We assume then that the *fusion* scheme is able to make the right choice between parametric or non-parametric displacements. When a parametric displacement is selected, the regularization tends to impose the corresponding model for the whole object/region. Furthermore, to avoid computing such parametric displacement fields starting from all the available input *multi-step* elementary *optical flow* fields, we suggest to consider only distant or very distant matching for which parametric displacements can offer an efficient alternative.

In our experiments, parametric *optical flow* fields have been involved only for the *AmeliaRetro* sequence (Fig. 10.7). They have been computed with a maximum *RANSAC* iteration number of $N_{ransac} = 100$. In addition, the quality criterion involves the following threshold: $\epsilon_{ransac} = 2$. Regarding the *quadtree* decomposition, the varying size patch are estimated with $l_{min} = 16$ and $l_{max} = 256$.

To illustrate the usefulness to complement non-parametric *optical flow* fields with parametric optical flow fields, we present in Fig. 10.18 some matching results between I_0 and I_{80} of *AmeliaRetro*. Both types of *optical flow* fields are compared: the non-parametric *optical flow* $\boldsymbol{d}_{0,80}$ and $\boldsymbol{d}^{aff}_{0,80}$, the affine flow based on $\boldsymbol{d}_{0,80}$. Despite strong similarities, we notice that affine correspondences are more accurate, especially in the upper left part of the dress. This example

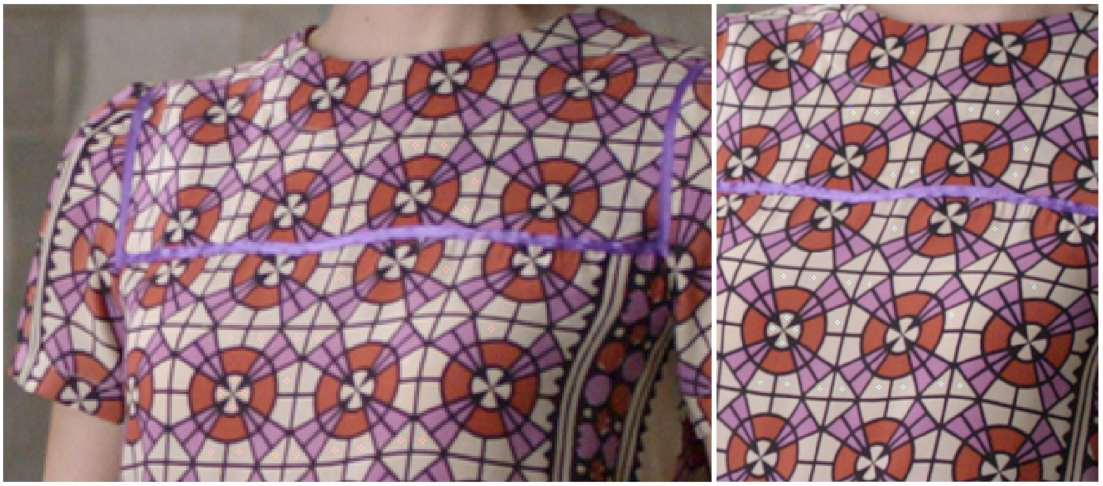


Figure 10.18: Matching results in *AmeliaRetro* between I_0 and I_{80} with an *optical flow* field computed using 2*D-DE* and an affine *optical flow* field estimated via *quadtree* decomposition and *RANSAC* algorithms. The input points (red points in (*a*)) are located at the center of white areas of the dress.

shows that the computation of affine models inside small patches within the dress of *AmeliaRetro* is relevant since the motion of the dress is well described by piecewise parametric deformations.

10.3.4 Video editing

Once we have a set of dense long-term *to-the-reference* correspondences that link every point of the sequence to the reference frame, the applications in the context of video editing are numerous. A typical problem is the insertion of external graphical elements on real surfaces within the video. In this section, we present four results for *AmeliaRetro* (Fig. 10.19), *Newspaper* (Fig. 10.20), *Dirk-Hartmut* (Fig. 10.21) and *Water-Marshall-37* (Fig. 10.22).

The first and the fourth examples consist in changing the colour of one part of the reference frame (the dress of the lady in frame I_{100} for *AmeliaRetro* and the jacket and the tie of the man in I_{1358} for *Water-Marshall-37*) and then propagating this color changes by using the *to-the-reference* long-term displacement fields to the remaining frames (respectively up to I_0 and I_{1329}). In the same spirit, the second and the third examples deal with logo insertion (in I_0 for the *Newspaper* sequence and in I_{96} for *Dirk-Hartmut*) and then logo propagation across the sequence (respectively up to I_{99} and I_{110}). These video editing examples use only *to-the-reference* long-term displacement fields whose end-point in I_{ref} is located in the insertion mask (see the insertion masks displayed in Fig. 10.19 and 10.20) to propagate textures or logos.

The color modifications in *AmeliaRetro* (Fig. 10.19) have been propagated using the proposed *MS-CG*(2*D-DE*) and *MSF+STF*(2*D-DE*) approaches as well as *TV-L1 acc* and 2*D-DE acc* which were the best single step methods in terms of visual quality. The results are clearly more realistic with *MS-CG*(2*D-DE*) compared to *TV-L1 acc* or 2*D-DE acc*. An even better spatial consistency of the structure of the dress is achieved with *MSF+STF*(2*D-DE*).

Texture insertion in I_{100} | Insertion mask in I_{100} | Source I_{100} | Insertion in I_{100}

Prop. to I_{80} *TV-L1 acc* | Prop. to I_{60} *TV-L1 acc* | Prop. to I_{40} *TV-L1 acc* | Prop. to I_{20} *TV-L1 acc* | Prop. to I_0 *TV-L1 acc*

Prop. to I_{80} *2D-DE acc* | Prop. to I_{60} *2D-DE acc* | Prop. to I_{40} *2D-DE acc* | Prop. to I_{20} *2D-DE acc* | Prop. to I_0 *2D-DE acc*

Prop. to I_{80} *MS-GC (2D-DE)* | Prop. to I_{60} *MS-GC (2D-DE)* | Prop. to I_{40} *MS-GC (2D-DE)* | Prop. to I_{20} *MS-GC (2D-DE)* | Prop. to I_0 *MS-GC (2D-DE)*

Prop. to I_{80} *MSF+STF (2D-DE)* | Prop. to I_{60} *MSF+STF (2D-DE)* | Prop. to I_{40} *MSF+STF (2D-DE)* | Prop. to I_{20} *MSF+STF (2D-DE)* | Prop. to I_0 *MSF+STF (2D-DE)*

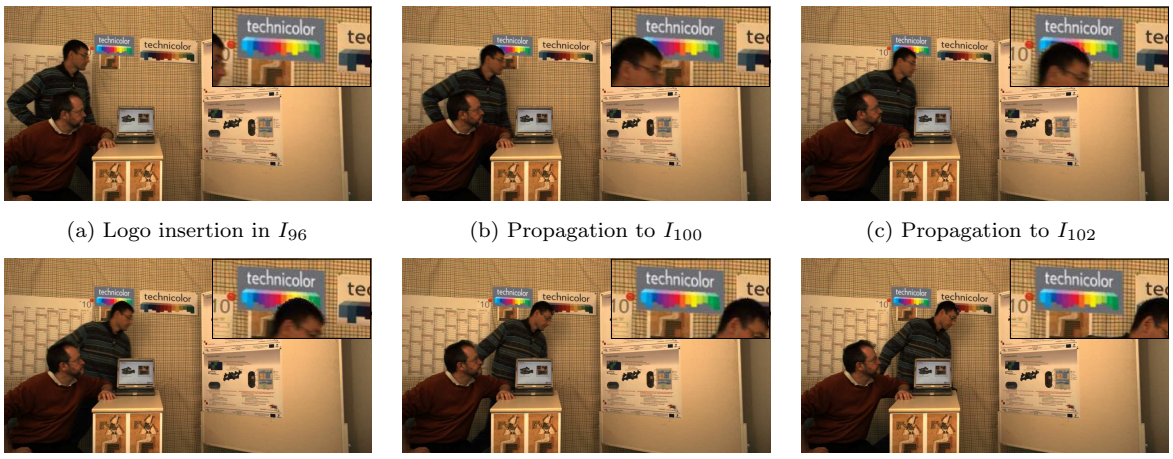
Figure 10.19: Color modifications in I_{100} and propagation up to I_0 (*AmeliaRetro* sequence). We compare: the classic *direct* integration of consecutive *TV-L1* [ZPB07] *optical flow* (*TV-L1 acc*); the classic *direct* integration of consecutive *2D-DE* [RTDC12] *optical flow* (*2D-DE acc*); the proposed *multi-step* via *graph-cuts* approach (Section 10.1) using *multi-step* elementary *2D-DE optical flows*, *MS-GC (2D-DE)*; the proposed *multi-step* flow approach (Section 10.2) using *multi-step* elementary *2D-DE optical flows*, *MSF+STF (2D-DE)*.

Source frame I_0 | Propagation to I_{20} *LDOF acc* | Propagation to I_{36} *LDOF acc* | Propagation to I_{99} *LDOF acc*

Logo insertion in I_0 | Propagation to I_{20} *MS-GC* (2*D-DE*) | Propagation to I_{36} *MS-GC* (2*D-DE*) | Propagation to I_{99} *MS-GC* (2*D-DE*)

Logo insertion in I_0 | Propagation to I_{20} *MSF+STF* (2*D-DE*) | Propagation to I_{36} *MSF+STF* (2*D-DE*) | Propagation to I_{99} *MSF+STF* (2*D-DE*)

Figure 10.20: Logo insertion in I_0 and propagation up to I_{99} (*Newspaper* sequence). We compare: the classic *direct* integration of consecutive *LDOF* [BM11] *optical flow* (*LDOF acc*); the proposed *multi-step* via *graph-cuts* approach (Section 10.1) using *multi-step* elementary 2*D-DE optical flows* [RTDC12], *MS-GC* (2*D-DE*); the proposed *multi-step* flow approach (Section 10.2) using *multi-step* elementary 2*D-DE optical flows*, *MSF+STF* (2*D-DE*).



(a) Logo insertion in I_{96} (b) Propagation to I_{100} (c) Propagation to I_{102}

(d) Propagation to I_{104} (e) Propagation to I_{108} (f) Propagation to I_{110}

Figure 10.21: Logo insertion in I_{96} and propagation up to I_{110} (*Dirk-Hartmut* sequence) with the proposed *multi-step* flow approach (Section 10.2) using *multi-step* elementary 2*D-DE optical flows*: *MSF+STF* (2*D-DE*).

(a) Insertion in I_{1358} (b) Propagation to I_{1348}, *MSF+STF* (2*D-DE*)

(c) Propagation to I_{1338}, *MSF+STF* (2*D-DE*) (d) Propagation to I_{1329}, *MSF+STF* (2*D-DE*)

Figure 10.22: Texture insertion in I_{1358} and propagation up to I_{1329} (*Water-Marshall*-37 sequence) with the proposed *multi-step* flow approach (Section 10.2) using *multi-step* elementary 2*D-DE optical flows*: *MSF+STF* (2*D-DE*).

In Fig. 10.20, note how the large occlusion by the arm can be overcome only by *multi-step* methods. *LDOF acc* is not able to jump the occlusion and stops the propagation once the occlusion occurs. However, the accuracy of *MSF+STF* is clearly better than *MS-GC* as we can see in I_{99} by comparing both methods (artifacts near the cup for *MS-GC*). Note the consistency before and after the occlusion with *MSF+STF*. Moreover, we have taken advantage of the reliable point correspondences in order to compute a brightness gain for each point between the reference and each frame. This permits to insert the element more realistically over a shadowed area.

In the same context, in Fig. 10.21, some regions of the logo are occluded at some instant of the sequence but they can be recovered when they reappear thanks to our *MSF+STF*(2*D-DE*) long-term matching used with $S = \{\pm 1, \pm 2, \pm 5, \pm 10\}$. Moreover, pixels are modified only on disoccluded areas, as one would expect.

Finally, the texture propagation results for the *Water-Marshall-37* sequence (Fig. 10.22) indicate that *MSF+STF*(2*D-DE*) (with $S = \{\pm 1, \pm 2, \pm 5, \pm 10\}$) is able to perform an efficient motion estimation even for large poorly textured areas and in the presence of zooming.

10.3.5 Key-frame based video segmentation

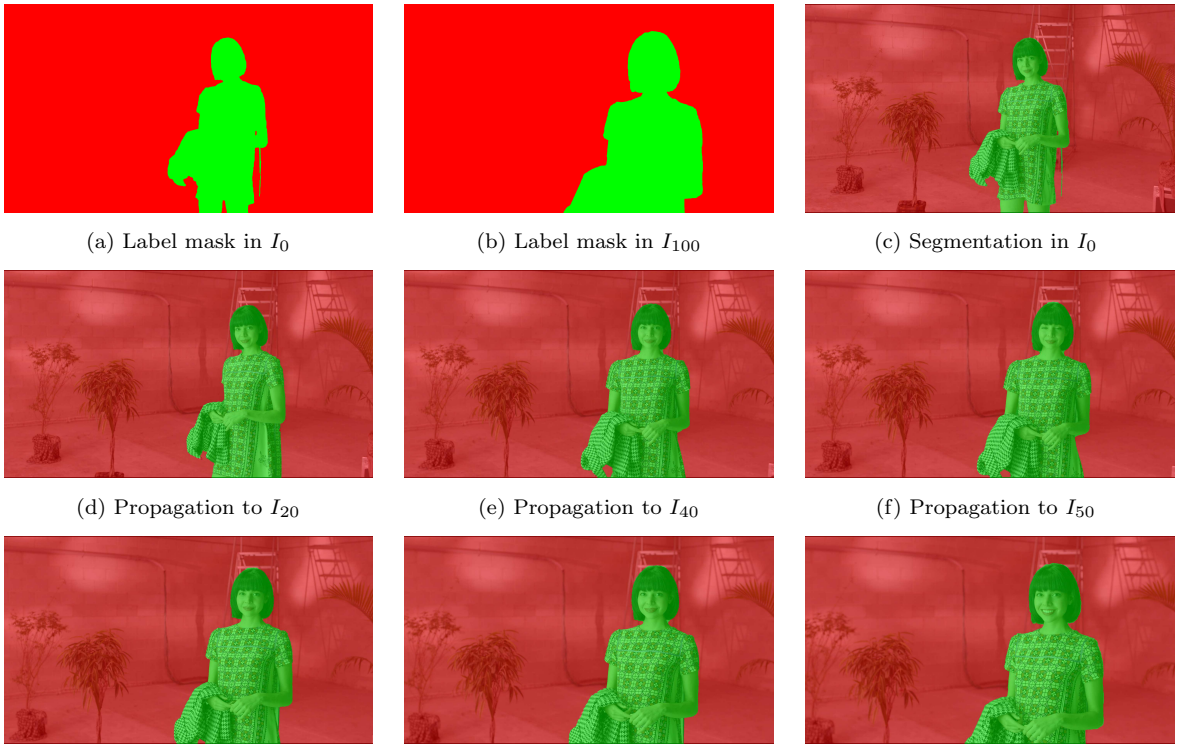
Let us now assume that the user provides a dense segmentation map for a given reference frame I_{ref}. For each grid location $\boldsymbol{x}_n$ of each non-reference frame I_n of the sequence, and if it is not detected as occluded, we determine its corresponding position in the reference frame. If this position is within the image boundaries, the label of the nearest pixel in I_{ref} is given to $\boldsymbol{x}_n$. At this stage, occluded pixels remain unlabelled.

This label propagation process can be easily adapted to use more than one single reference segmentation map. If a conflict appears between the labels propagated at the same pixel $\boldsymbol{x}$ from different reference frames, we simply solve it by assigning $\boldsymbol{x}$ to the label corresponding to the lowest colour matching cost without regularization.

Dense segmentation may then be obtained using standard segmentation tools. Precisely, to refine the maps obtained by label propagation and to assign a label at occluded pixels, we follow the approach of [FPR08b, FPR08a] which consists in performing a *graph-cut* minimization. The cost function to be minimized is the sum of two standard terms. The first term is a colour data penalty term of assigning label l at pixel $\boldsymbol{x}_n$. It is set as the negative log-likelihood of colour distribution of the video region l. This distribution consists of the *Gaussian* mixture model in the *RGB* space computed on the regions l in the reference segmentation maps. The second term is the standard contrast sensitive regularization term defined in [BVZ01].

We illustrate this key-frame based video segmentation through two results obtained for the *AmeliaRetro* (Fig. 10.23) and *Newspaper* (Fig. 10.24) sequences. In both cases, I_0 and I_{100} have been manually segmented. The resulting label masks (see (a,b) of both figures) are then propagated to the whole sequence as previously mentioned using *MSF+STF*(2*D-DE*) long-term displacement fields computed with respect to I_0 and I_{100}. Despite the artifact revealed in Fig. 10.24 (h) due to shadows and a low colour contrast which are difficult to handle accurately, the results are convincing and proves an efficient segmentation of the whole sequence. Note the slight rotation of the girl that results in self-occlusions with respect to both key-frames.

Starting from this key-frame based video segmentation, we could imagine to apply a particular processing for each label across the sequence. In Fig. 10.25, we propose to blur the

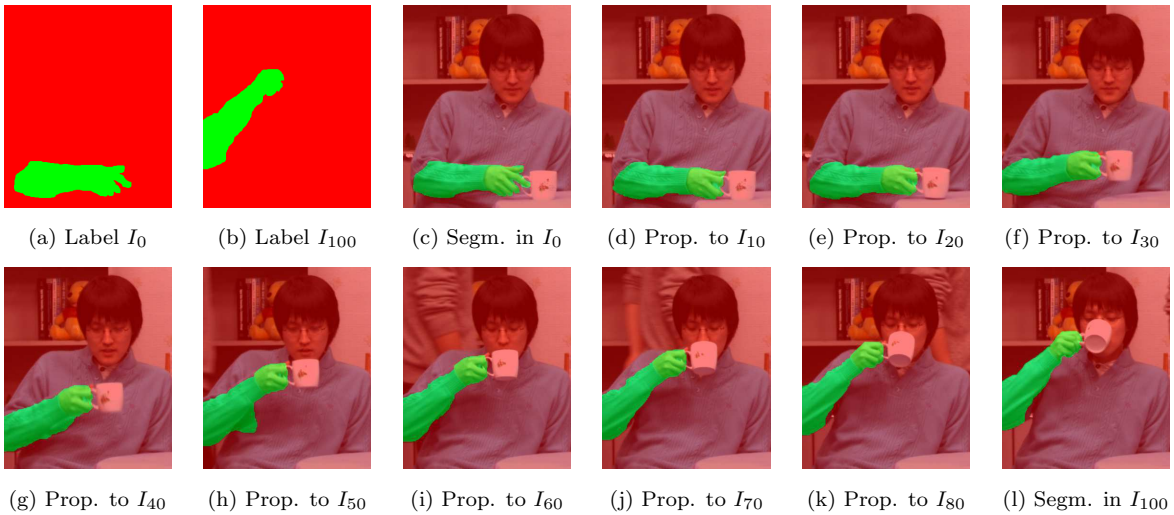


(a) Label mask in I_0 (b) Label mask in I_{100} (c) Segmentation in I_0

(d) Propagation to I_{20} (e) Propagation to I_{40} (f) Propagation to I_{50}

(g) Propagation to I_{60} (e) Propagation to I_{80} (h) Segmentation in I_{100}

Figure 10.23: From label propagation to dense segmentation, *AmeliaRetro* sequence. The user provides the segmentation maps for I_0 (a,c) and I_{100} (b,h) and our *MSF+STF*(2*D-DE*) long-term dense motion estimator performed with respect to I_0 and I_{100} propagates the label to the whole sequence. Segmentation results are shown for I_{20}, I_{40}, I_{50}, I_{60} and I_{80}.



(a) Label I_0 (b) Label I_{100} (c) Segm. in I_0 (d) Prop. to I_{10} (e) Prop. to I_{20} (f) Prop. to I_{30}

(g) Prop. to I_{40} (h) Prop. to I_{50} (i) Prop. to I_{60} (j) Prop. to I_{70} (k) Prop. to I_{80} (l) Segm. in I_{100}

Figure 10.24: From label propagation to dense segmentation, *Newspaper* sequence. The user provides the segmentation maps for I_0 (a,c) and I_{100} (b,l) and our *MSF+STF*(2*D-DE*) long-term dense motion estimator performed with respect to I_0 and I_{100} propagates the label to the whole sequence. Segmentation results are shown for I_{10}, I_{20}, I_{30}, I_{40}, I_{50}, I_{60}, I_{70} and I_{80}.

(a) I_0 original (b) I_0 with blurred background (c) Propagation to I_{40}

(d) Propagation to I_{60} (e) Propagation to I_{80} (f) Propagation to I_{100}

Figure 10.25: Blurring of the background of *AmeliaRetro*. Thanks to the segmentation labels which have been propagated through the sequence via our *MSF+STF(2D-DE)* long-term displacement fields, we are able to identify for each frames of the sequence the pixels which belong to the background in order to finally apply a low-pass filtering.

background of *AmeliaRetro* in order to highlight the lady. Thanks to the segmentation labels which have been propagated through the sequence via our long-term displacement fields, we are able to identify for each frame of the sequence the pixels which belong to the background in order to finally apply a low-pass filtering on them. Any type of local modifications could be considered such as colour modifications on a selected (set of) object(s) in a video editing context (black and white filter for instance).

10.3.6 Towards more accurate dense long-term correspondences

Despite its ability to handle complex scenarios compared to state-of-the-art methods, the proposed *multi-step* strategies has two main drawbacks. First, *MS-GC* and *MSF* perform the selection of displacement fields by relying only on classical *optical flow* assumptions that can sometimes fail between distant frames. Second, we have seen that the candidate displacement fields are based on previous estimations which can sometimes propagate errors across the sequence, until a new available *step* gives a chance to match with a correct location again.

These aspects are illustrated in Fig. 10.26 which focuses on point tracking with color variations in the *AmeliaRetro* sequence from I_0 to I_{100}. A small set of pixels located in the dress is tracked along the sequence using the *MSF+STF(2D-DE)* method. We notice that some of these pixels drift, deviate from the dress where a shadow occurs and finally arrive on the wall (Fig. 10.26 (*b*,*c*)). The two previous considerations can explain why *MSF+STF* fails in this case. Indeed, the tracking failures are due to both the matching cost involved in our selection criterion which is not enough robust to handle color variations and the sequential aspect of our method which encourages single errors to be propagated.

To obtain better results, we have decided to robustify the multilateral spatio-temporal filtering. With this particular example in mind, the following improvements have been done.

1. the spatial window has been increased from 7×7 to 11×11,

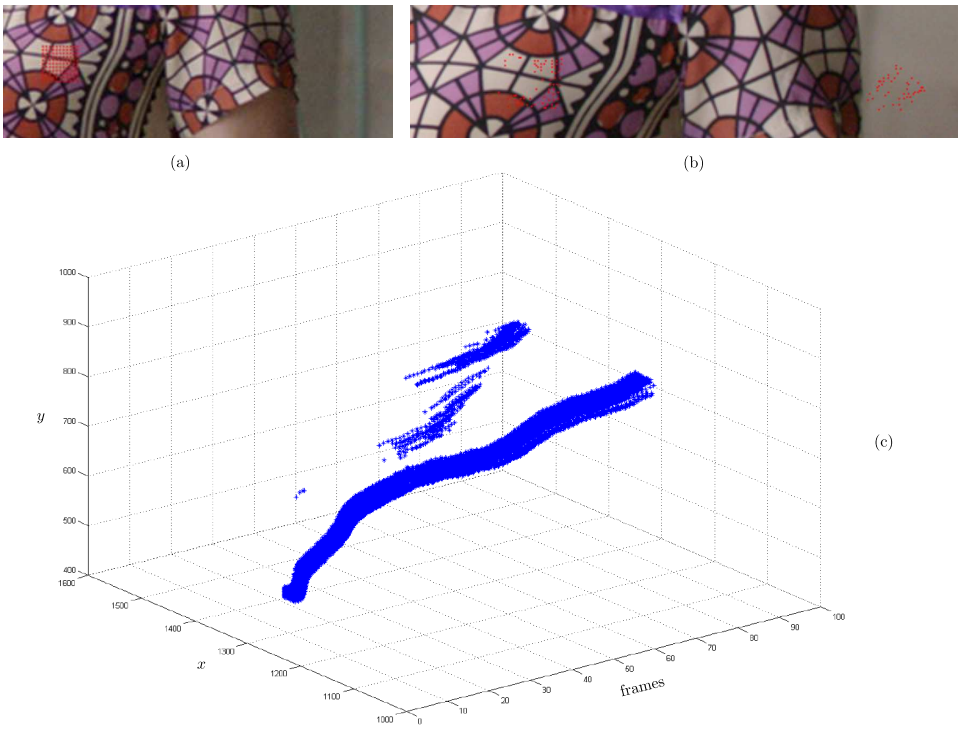


Figure 10.26: Point tracking with color variations in the *AmeliaRetro* sequence using *MSF+STF(2D-DE)* long-term displacement field: (a) grid points in I_0 involved in the tracking, (b) tracking results in I_{100}, (c) 2D+t trajectory visualization.

2. *motion ranges* (i.e. maximal elementary motion values between two consecutive frames for x and y components) have been manually set in order to reject *from-the-reference* displacement vectors whose corresponding locations in I_n do not fulfill the two following conditions across the trajectories:

   - a thresholding condition on the motion vectors between consecutive frames $\{I_{n-1}, I_n\}$ as shown in Eq. 10.17:

$$\begin{cases} |x_n - x_{n-1}| \leq r_x \\ |y_n - y_{n-1}| \leq r_y \end{cases} \tag{10.17}$$

   - a thresholding condition on the accelerations considering a triplet of consecutive frames $\{I_{n-1}, I_n, I_{n+1}\}$. Let r_x and r_y be the $x-$ and $y-$ *motion ranges*. The *from-the-reference* displacement vectors $\boldsymbol{d}_{ref,n}(\boldsymbol{x}_{ref}) = \boldsymbol{x}_n - \boldsymbol{x}_{ref}$ are no longer taken into account during the filtering stage when one of the two following conditions of Eq. 10.18 is not valid. Note that in Eq. 10.18, $[x_{n-1}, y_{n-1}]^t$ and $[x_{n+1}, y_{n+1}]^t$ have been obtained with $\boldsymbol{d}_{ref,n-1}$ and $\boldsymbol{d}_{ref,n+1}$ respectively.

$$\begin{cases} |(x_{n+1} - x_n) - (x_n - x_{n-1})| \leq 2.r_x \\ |(y_{n+1} - y_n) - (y_n - y_{n-1})| \leq 2.r_y \end{cases} \tag{10.18}$$

3. the inconsistency between *from-the-reference* and *to-the-reference* displacement vectors (described in Section 10.2.3) has been added as an additional weight into Eq. 10.12.

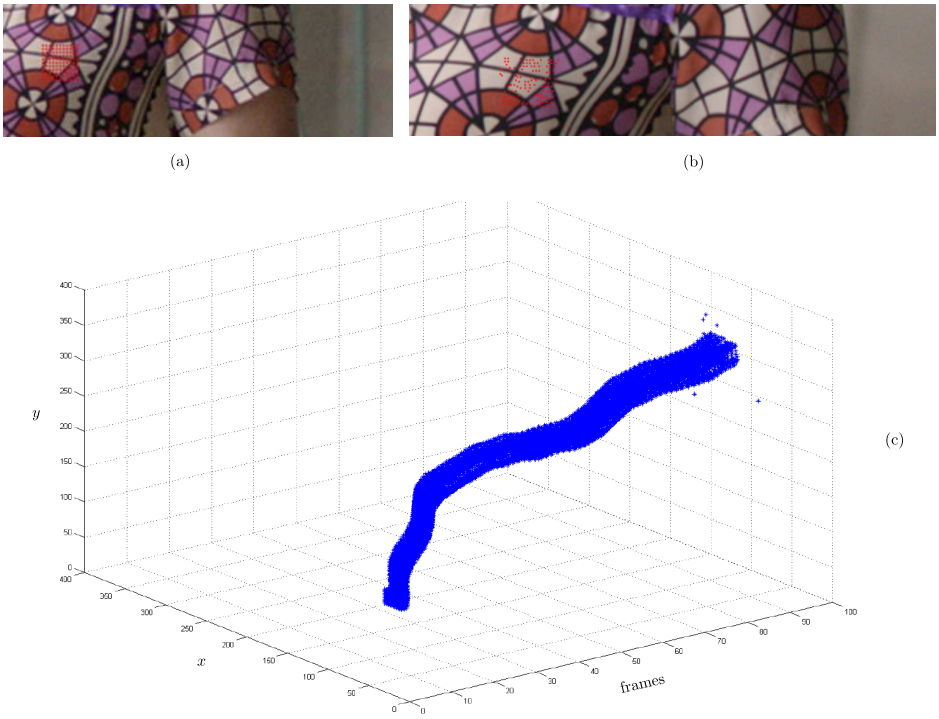


Figure 10.27: Point tracking with color variations in the *AmeliaRetro* sequence using the *MSF+STF*(2*D-DE*) method with an improved version of the multilateral spatio-temporal filtering (see details in the text): (*a*) grid points in I_0 involved in the tracking, (*b*) tracking results in I_{100}, (*c*) 2D+t trajectory visualization.

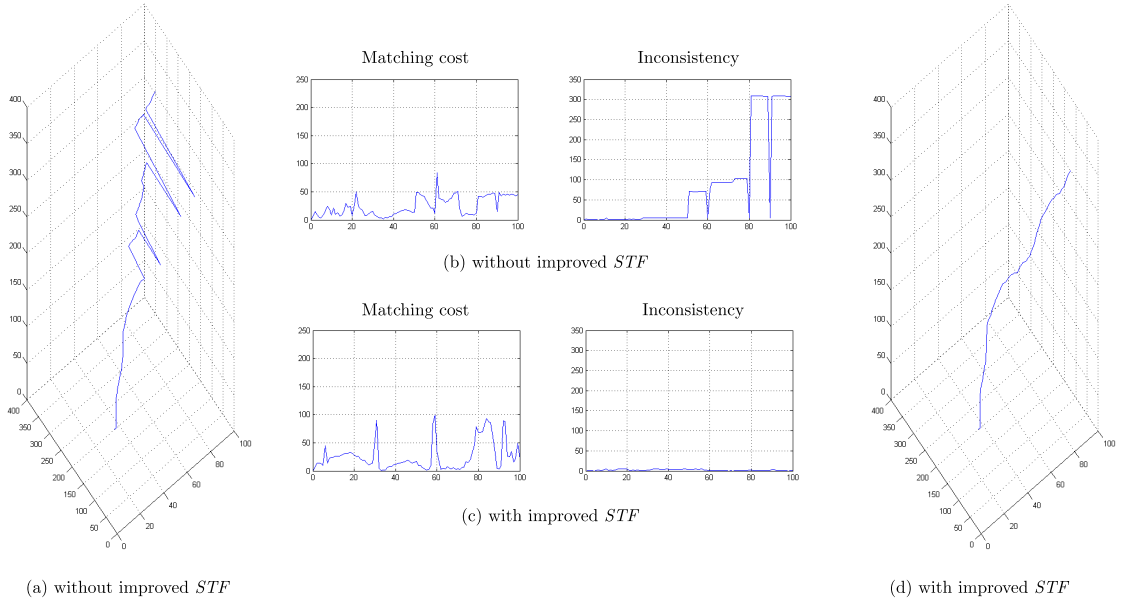


Figure 10.28: Comparisons with (*a*,*b*) / without (*c*,*d*) the improved multilateral spatio-temporal filtering: (*a*,*d*) 2D+t visualization of one single trajectory, (*b*,*c*) matching cost and inconsistency along the trajectory.

All these improvements allow to obtain the results displayed in Fig. 10.27 where all the points remain within the dress with a good accuracy in terms of positions for most of them. Note that the tracked patch contains holes in I_{100} due to the zoom which makes the area described by more pixels in the frame with the highest resolution (i.e. in I_{100}).

Fig. 10.28 focuses on one single point among the initial set and compares the evolution of the matching cost and the inconsistency along:

- the failing trajectory estimated without the described improvements: Fig. 10.28 (a,b),
- the accurate trajectory estimated with the described improvements: Fig. 10.28 (c,d).

This experiment reveals that:

- relying on the matching cost only does not work in this case. Indeed, according to this criterion, it is more relevant to match locations in the wall than continuing tracking the dress. In addition, we can notice that the matching cost for the correct trajectory is relatively high.
- the inconsistency between *from-the-reference* and *to-the-reference* displacement vectors seems to be a better criterion since it is able to efficiently assess the trajectory quality and in particular to predict motion drift.

10.4 Conclusion

Starting from the sequential *multi-step* strategy explored in Section 9.3.3, we built two new frameworks for estimating dense correspondence fields between a reference frame and all the subsequent frames of a long video shot: *multi-step* flow via *graph-cuts* (*MS-GC*) and *multi-step* fusion flow (*MSF*). These methods are based on both: 1) the accumulation of *multi-step* elementary *optical flow* vectors through *inverse* integration, 2) the optimal merge of the resulting candidate long-term displacement fields.

Compared to *MS-GC*, *MSF* performs a combination of bidirectional *multi-step* elementary *optical flow* fields (instead of unidirectional) and a different long-term displacement selection procedure based on the *fusion moves* algorithm proposed in [LRR08, LRRB10] (instead of *graph-cuts*). Moreover, the notion of trajectory is explicitly taken into account in a new multilateral spatio-temporal filtering stage which iteratively refines the long-term displacement fields.

As shown in terms of quantitative trajectory evaluation, registration and *PSNR* assessment or trajectory visualization, the long-term displacement fields resulting from the *MS-GC* and *MSF* approaches present an improved accuracy compared to state-of-the art approaches, particularly for large motions and in presence of temporary occlusions. The proposed techniques perform well both for point-wise tracking (*from-the-reference* strategy) and for *pulling* dense information from a reference frame (*to-the-reference* strategy). In terms of applications, we demonstrate the effectiveness of our methods for graphic element insertion and propagation as well as video volume segmentation in complex video sequences.

A point that would deserve further investigation is the automatic selection of both the reference frames and the input set of candidate *steps* depending on the considered shot. They have indeed to be set properly as each shot contains its own motion peculiarity.

Although the improvements made in Section 10.3.6 on the multilateral spatio-temporal filtering are relevant regarding the described experiment, the proposed approaches still show some limitations. Indeed, *MS-GC* and *MSF* perform the selection of displacement fields by relying only on classical *optical flow* assumptions that can sometimes fail for distant matching. The displacement fields selection criteria should be robustified to reach a better accuracy while rejecting motion *outliers*. In this direction, a first step would be to take into account the inconsistency between *from-the-reference* and *to-the-reference* displacement vectors to complement the matching cost which is not always robust. Moreover, *MS-GC* and *MSF* ensure a certain temporal consistency but can also propagate estimation errors along the following frames of the sequence.

These limitations are addressed in Chapter 11 which will deeply study two main points: 1) How to build long-term dense displacement fields following the exhaustive integration strategy introduced in Section 9.3.1 while avoiding computational and memory issues? 2) How to robustify the long-term displacement fields selection task? In this context and toward our goal of establishing accurate dense correspondences in long video sequences, we propose in Chapter 11 to build and explore another dense long-term motion estimation framework.

Chapter 11

Combinatorial *multi-step* integration and statistical selection

We have proposed in Chapter 10 two sophisticated long-term dense motion estimation frameworks, referred to as *multi-step* flow via *graph-cuts* (*MS-GC*) and *multi-step* flow fusion (*MSF*), which consist in sequentially merging a set of concatenated *multi-step* motion fields at intermediate frames up to the target frame. Despite an improved accuracy compared to state-of-the art approaches, *MS-GC* and *MSF* may fail in some complex situations due to two main points. First, these frameworks strongly rely on *optical flow* assumptions that frequently fail between distant frames. Second, the candidate displacement fields estimated with *MS-GC* or *MSF* are built with respect to previous estimations which ensures a certain temporal consistency but can also propagate errors across the sequence.

Concerning the first limitation, this issue could be partially compensated by complexifying the matching criteria *ad-infinitum*. One can add intrinsic motion quality assessment features as matching criteria such as the inconsistency between *from-the-reference* and *to-the-reference* vectors for instance. However, an uncertainty component seems always present. This argues in favor of a statistical processing which takes into account the random nature of these perturbations among a large set of dense displacement fields. To also take into account the second limitation, we wonder whether it is possible to build a new long-term dense motion estimation framework which relies on robust criteria for the matching task between distant frames while limiting the correlation with previous estimations.

In this context and based on the exhaustive *multi-step* strategy described in Section 9.3.1 (Chapter 9), we propose to study more deeply the concept of exhaustive integration. This translates in generating a large set of long-term displacement fields via combinations of *multi-step* elementary *optical flow* vectors without relying only on the optimal displacement fields computed for previous frames. With this resulting large set of long-term displacement fields, we suggest to study the spatial redundancy of all the resulting candidates through a statistical selection stage in order to provide a more robust indication than classical *optical flow* assumptions for the displacement fields selection task.

We propose in this direction two main contributions to address the dense long-term matching problem across video sequences. Firstly, we present a combinatorial *multi-step* integration method which allows one to get a large set of long-term displacement fields by considering multiple motion *paths* across the sequence while avoiding computational and memory issues. Secondly, once this motion candidate construction stage is performed, we propose to apply a new approach to select the optimal displacement field based on statistics and spatial regularization.

This chapter is organized as follows. In Section 11.1, we present how these combinatorial *multi-step* integration and statistical selection methods can perform an accurate dense motion estimation between a single pair of distant frames. Once cascaded, the combinatorial integration and the statistical selection are referred to as *CISS*. Experiments evaluate the effectiveness of this approach for distant motion estimation and results are presented in the context of video editing (Section 11.2). Then, in Section 11.3, we present *Statistical multi-step Flow* (*StatFlow*), a new two-step framework for computing *from-the-reference* and *to-the-reference* dense long-term displacement fields which extends *CISS* to the whole sequence. Finally, Section 11.4 is devoted to the quantitative and qualitative assessment of *StatFlow* in comparison with several existing methods including the *multi-step* flow fusion (*MSF*) approach described in Chapter 10.

This study led to two publications published in international conferences and one publication published in a national conference. First, [CCRM13a, CCRM13b] focuses on combinatorial integration and statistical selection to perform dense matching between a single pair of two distant frames. Second, [CCRM14] presents the statistical *multi-step* flow (*StatFlow*) algorithm.

11.1 Combinatorial integration and statistical selection between a pair of distant frames (*CISS*)

Let us consider a sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\dots,N]\!]}$ and let I_a and I_b be two distant frames of this sequence with $0 \leq a < b \leq N$. In this section, we focus on the frame pair $\{I_a, I_b\}$ and our goal is to accurately estimate a dense displacement field between these two frames. Let us study how the proposed combinatorial *multi-step* integration and the statistical selection stages are able to perform such a task.

The combinatorial *multi-step* integration aims at building a large set of candidate displacement fields between I_a and I_b by considering multiple motion *paths* across the sequence built via concatenations of *multi-step* elementary *optical flows*. This motion candidate construction stage is presented in Section 11.1.1. Once all the candidate displacement fields have been estimated, we aim at selecting the optimal motion field based on statistics and spatial regularization. The motion vector selection on large sets is described in Section 11.1.2.

11.1.1 Motion candidate construction via combinatorial integration

First, let us introduce the concept of motion *path* between the two distant frames I_a and I_b. A motion *path* deals with multiple concatenations of *multi-step* elementary *optical flow* fields across the video sequence. It links each pixel $\boldsymbol{x}_a$ of I_a to a corresponding position in I_b. Elementary *optical flow* fields can be computed between consecutive frames or with different frame *steps* as shown previously. Let $S_n = \{s_1, s_2, \dots, s_{Q_n}\} \subset \{1, \dots, N-n\}$ be the set of Q_n available *steps* at instant n. For recall, this means that the set of *optical flow* fields $\{\boldsymbol{u}_{n,n+s_1}, \boldsymbol{u}_{n,n+s_2}, \dots, \boldsymbol{u}_{n,n+s_{Q_n}}\}$ is available.

Step sequences generation

Our objective is to produce a large set of motion maps between I_a and I_b as to form a significative set of samples upon which a statistical processing would be meaningful and advantageous. Given this objective, we propose to initially generate all the possible *step sequences* (i.e. combinations of *steps*) in order to join I_b from I_a. Each *step sequence* will define a motion *path*.

Let $\mathbf{\Gamma}_{a,b} = \{\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{K-1}\}$ be the set of K possible *step sequences* $\boldsymbol{\gamma}_i$ between I_a and I_b. A *step sequence* $\boldsymbol{\gamma}_i = \{s_1^i, s_2^i, \ldots, s_{K_{\boldsymbol{\gamma}_i}}^i\}$ is defined by a set of $K_{\boldsymbol{\gamma}_i} - 1$ *steps* s_k^i which once cascaded join I_b from I_a. The set of K possible *step sequences* $\mathbf{\Gamma}_{a,b}$ is computed by building a tree structure (Fig. 11.1) where each node corresponds to a *optical flow* field assigned to a given frame for a given *step* value (corresponds to the node value).

In practice, the construction of the tree is done recursively starting from I_a: we create for each node as many children as the number of *steps* available at the current instant. A child node is not generated when I_b has already been reached (therefore, the current node is considered as a leaf) or if I_b is passed given the considered *step* (i.e. when we have exceeded I_b without passing through I_b). Finally, once the tree has been built, going from the root node to leaf nodes gives $\mathbf{\Gamma}_{a,b}$, the set of K possible *step sequences* between I_a and I_b. For illustration, the tree in Fig. 11.1 indicates the four *step sequences* that can be generated going from I_0 to I_3 with *steps* 1, 2 and 3: $\mathbf{\Gamma}_{0,3} = \{\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3\} = \{\{1,1,1\}, \{1,2\}, \{2,1\}, \{3\}\}$.

From *step sequences* to motion *path*

Once all the possible *step sequences* $\boldsymbol{\gamma}_i \; \forall i \in [\![0, \ldots, K-1]\!]$ between I_a and I_b have been generated, the corresponding motion *paths* can be constructed through 1st-order *Euler* integration. Starting from each pixel $\boldsymbol{x}_a \in I_a$ and for each *step sequence* $\boldsymbol{\gamma}_i$, this integration performs the accumulation of *optical flow* fields following the *steps* which form the current *step sequence*, i.e. $s_1^i, s_2^i, \ldots, s_{K_{\boldsymbol{\gamma}_i}}^i$.

Thus, with *steps* 1, 2 and 3, Fig. 11.2 illustrates the construction of the four possible motion *paths* (one for each *step sequence* of $\mathbf{\Gamma}_{0,3}$) between I_0 and I_3. Let $f_j^i = a + \sum_{k=0}^{j} s_k^i$ be the current frame number during the construction of motion *path* i from I_a where j is the *step* index within the *step sequence* $\boldsymbol{\gamma}_i$. For each $\boldsymbol{\gamma}_i \in \mathbf{\Gamma}_{a,b}$ and for each *step* $s_j^i \in \boldsymbol{\gamma}_i$, we start from x_a in order to iteratively compute:

$$\boldsymbol{x}_{f_j^i}^i = \boldsymbol{x}_{f_{j-1}^i}^i + \boldsymbol{u}_{f_{j-1}^i, f_j^i}(\boldsymbol{x}_{f_{j-1}^i}^i) \tag{11.1}$$

Once all the *steps* $s_j^i \in \boldsymbol{\gamma}_i$ have been run through, we obtain $\boldsymbol{x}_b^i$, the corresponding position in I_b of $\boldsymbol{x}_a$ of I_a obtained with the *step sequence* $\boldsymbol{\gamma}_i$. By considering all the *step sequences*, we finally get a large set of candidate positions in I_b and this for each pixel $\boldsymbol{x}_a$ of I_a. In our simple example, the construction of the motion *paths* corresponding to all the possible *step sequences* of $\mathbf{\Gamma}_{0,3}$ allows to obtain $\boldsymbol{x}_0^3, \boldsymbol{x}_1^3, \boldsymbol{x}_2^3$ and $\boldsymbol{x}_3^3$ in I_3 (Fig. 11.2).

The occlusion maps attached to input *multi-step* elementary *optical flow* fields are used to possibly stop the motion *path* construction. Considering an intermediate point $\boldsymbol{x}_{f_j^i}^i$ during the construction, a *step* can be added only if the closest pixel to $\boldsymbol{x}_{f_j^i}^i$ is considered as un-occluded for this *step*. Otherwise, the motion *path* is removed.

In the following, the large set of candidate positions in I_b is defined as $T_{a,b}(\boldsymbol{x}_a) = \{\boldsymbol{x}_b^i\}$ $\forall i \in [\![0, \ldots, K_{\boldsymbol{x}_a} - 1]\!]$ where $K_{\boldsymbol{x}_a}$ is the cardinal of $T_{a,b}(\boldsymbol{x}_a)$.

Avoiding computational and memory issues

Up to now, we have considered an exhaustive generation of *step sequences* for clarity. However, for very distant frames and for a large set of *steps*, it is not possible to consider all possible *step sequences* due to computational and memory issues. For instance, for a distance of 30 frames and with *steps* 1, 2, 5 and 10, the number of possible motion *paths* is 5877241. Therefore, the

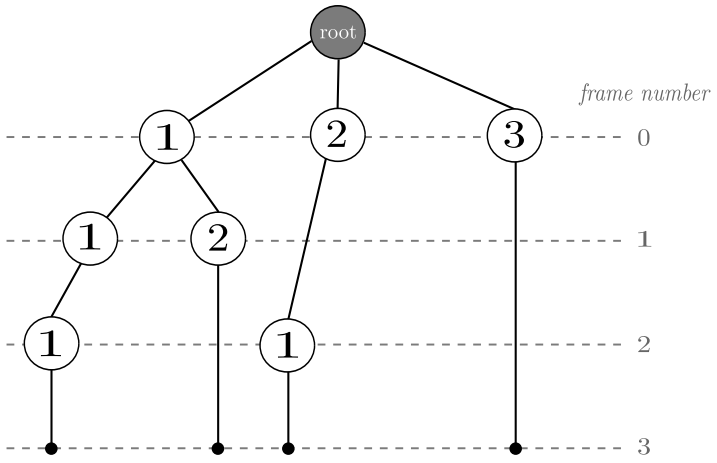


Figure 11.1: Generation of *step sequences* by building a tree structure where each node corresponds to a *optical flow* field assigned to a given frame for a given *step* value (node value). Going from the root node to leaf nodes of this tree structure gives $\mathbf{\Gamma}_{a,b}$, the set of K possible *step sequences* from I_a to I_b. In this example, four *step sequences* are generated from I_0 to I_3 with *steps* 1, 2, and 3: $\mathbf{\Gamma}_{0,3} = \{\{1,1,1\},\{1,2\},\{2,1\},\{3\}\}$.

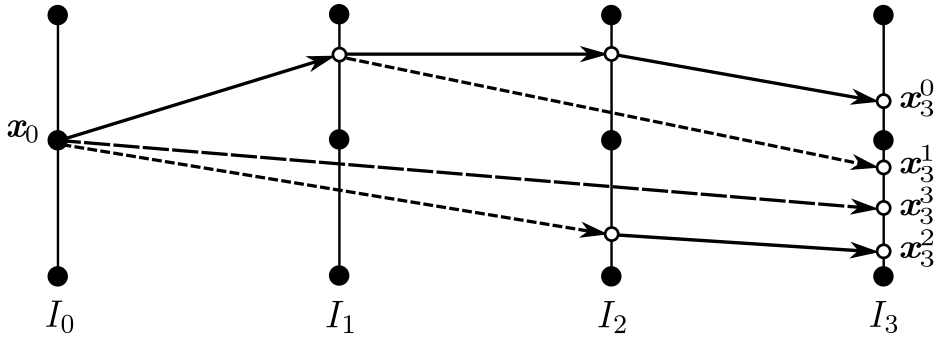


Figure 11.2: Generation of motion *paths* following all the *step sequences* generated in Fig. 11.1 between I_0 and I_3. For each pixel $\boldsymbol{x}_0$ of I_0, this gives a set of candidate positions in I_3: $\boldsymbol{x}_0^3, \boldsymbol{x}_1^3, \boldsymbol{x}_2^3$ and $\boldsymbol{x}_3^3$.

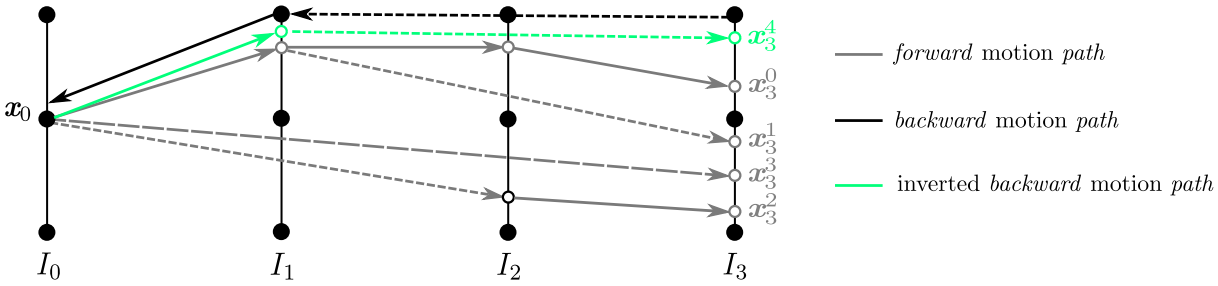


Figure 11.3: *Backward* motion *paths* (i.e. from I_b to I_a) are inversed into *forward* motion *paths* in order to enrich $T_{a,b}(\boldsymbol{x}_a)$.

procedure described above is performed on a reasonable number of *step sequences* and not for all as previously assumed. Let us describe the two conditions which limit the number of *step sequences*:

1. Limited number of concatenations: we limit the number of elementary *optical flow* vectors composing the motion *paths* by providing a maximum number of concatenations N_c. Indeed, the concatenation of numerous vectors may lead to an important drift.

2. Guided random selection: we randomly select N_s motion *paths* among the remaining motion *paths* where N_s is determined by storage capacity. This selection is constrained by the fact that the candidate vectors should not be highly correlated. The frequency of appearance of a given *step* at a given frame must be uniform among all the possible *steps* arising from this frame in order to avoid a systematic bias towards the more populated branches of the tree.

In the very simple example of Fig. 11.2 and assuming that only 3 motion *paths* can be selected, our guided random selection would select only one single candidate position among $\boldsymbol{x}_3^0$ and $\boldsymbol{x}_3^1$ because the corresponding motion *paths* have been generated using the same *optical flow* vector between I_0 and I_1. This selection would aim at reducing the influence of $\boldsymbol{u}_{0,1}(\boldsymbol{x}_0)$ which in case of bad estimation would lead to two poor candidate positions in I_3.

Joint *forward* and *backward* processing

We claim that motion estimation can be enhanced by considering both *forward* and *backward* motion fields. Similarly to the *forward* direction, the set of *backward* displacement fields from each pixel $\boldsymbol{x}_b$ of I_b to I_a can be computed by considering *multi-step backward* motion *paths*. These *backward* motion fields can be inversed into *forward* motion fields in order to enrich $T_{a,b}(\boldsymbol{x}_a)$.

More precisely, as illustrated in Fig. 11.3, *backward* motion vectors from pixels of I_b are projected into I_a. For each one, we identify the nearest pixel of the arrival position in I_a. The corresponding vector from I_b to I_a is reversed and started from the previously identified nearest pixel in I_a which gives a new candidate for $T_{a,b}(\boldsymbol{x}_a)$.

In the following, candidates of $T_{a,b}(\boldsymbol{x}_a)$ which have been obtained through *backward* estimation and inversion are defined as *reverse*. Otherwise, the candidate positions coming from *forward* motion *paths* are called *direct*.

11.1.2 Motion vector selection on large sets

Given $T_{a,b}(\boldsymbol{x}_a) = \{\boldsymbol{x}_b^i\}_{i\in[\![0,\ldots,K_{\boldsymbol{x}_a}-1]\!]}$, the set of candidate positions in I_b obtained for each pixel $\boldsymbol{x}_a$ of I_a from the displacement field construction stage described in Section 11.1.1, the objective is now to select the optimal candidate position $\boldsymbol{x}_b^*$ in the set $T_{a,b}(\boldsymbol{x}_a)$. The selection of the optimal candidate position is performed following a statistical processing (SP) which is then combined to a global optimization method (GO). Before explaining how both tools are cascaded into the proposed motion vector selection framework (SP+GO), let us describe them separately.

Statistical processing (SP) for motion vector selection

To select the optimal candidate position, we propose first of all a new statistical processing which exploits the statistical information on the point distribution and the quality of each candidate.

The idea is to assume a *Gaussian* model for the distribution of $T_{a,b}(\boldsymbol{x}_a)$ and to identify its central value, $\boldsymbol{x}_b^*$. Consequently, we suppose that the position candidates in $T_{a,b}(\boldsymbol{x}_a)$ follow a *Gaussian* probability density with mean μ and variance σ^2. The probability density function for $\boldsymbol{x}_b^i$ is thus given by:

$$\pi(\boldsymbol{x}_b^i|\mu,\sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}}.exp[-\frac{1}{2}(\frac{\boldsymbol{x}_b^i-\mu}{\sigma})^2] \tag{11.2}$$

Supposing that all the candidate positions $\boldsymbol{x}_b^i$ are independent, the probability density function of $T_{a,b}(\boldsymbol{x}_a)$ can be written as follows:

$$\pi(T_{a,b}(\boldsymbol{x}_a)|\mu,\sigma^2) = \prod_{i=0}^{K_{\boldsymbol{x}_a}-1}\pi(\boldsymbol{x}_b^i|\mu,\sigma^2) = \prod_{i=0}^{K_{\boldsymbol{x}_a}-1}(2\pi\sigma^2)^{-\frac{1}{2}}.exp[-\frac{1}{2}(\frac{\boldsymbol{x}_b^i-\mu}{\sigma})^2] \tag{11.3}$$

The estimation of the mean μ and the variance σ^2 through the maximum likelihood estimator (MLE) is performed by maximizing Eq. 11.4:

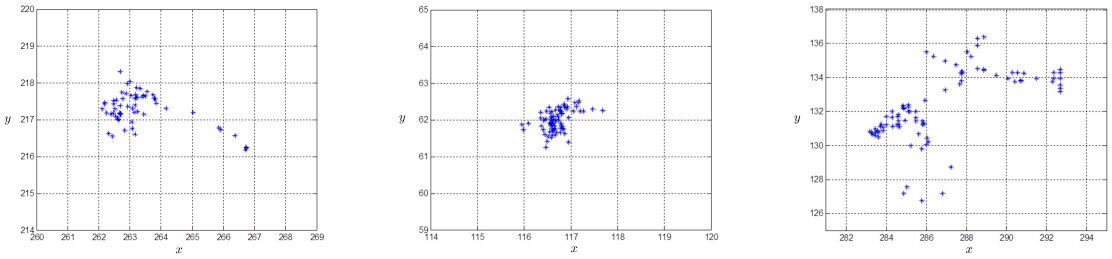
$$ln(\pi(T_{a,b}(\boldsymbol{x}_a)|\mu,\sigma^2)) = -K_{\boldsymbol{x}_a}.ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=0}^{K_{\boldsymbol{x}_a}-1}(\boldsymbol{x}_b^i-\mu)^2 \tag{11.4}$$

We are interested in the central value, which in the case of a *Gaussian* distribution coincides with the mean value, the median value and the mode. Thus, we seek for estimating μ, regardless of the value of σ^2. Using the MLE criterion and imposing the selection among elements of $T_{a,b}(\boldsymbol{x}_a)$, the choice of the optimal candidate position $\boldsymbol{x}_b^*$ is defined by:

$$\boldsymbol{x}_b^* = arg\min_{\boldsymbol{x}_b^i}\sum_{\substack{j=0\\j\neq i}}^{K_{\boldsymbol{x}_a}-1}\left\|\boldsymbol{x}_b^j-\boldsymbol{x}_b^i\right\|_2^2 \tag{11.5}$$

The assumption of *Gaussianity* can be largely perturbed by outliers. Consequently, a robust estimation of the distribution central value is necessary. For this sake, the mean operator is replaced by the median operator which leads to:

$$\boldsymbol{x}_b^* = arg\min_{\boldsymbol{x}_b^i} med_{j\neq i}\left\|\boldsymbol{x}_b^j-\boldsymbol{x}_b^i\right\|_2^2 \tag{11.6}$$



(a) (b) (c)

Figure 11.4: Examples of candidate distributions given as inputs of the statistical processing. The candidates come from the combinatorial *multi-step* integration described in Section 11.1.1.

Fig. 11.4 shows examples of candidate distributions coming from the combinatorial *multi-step* integration described in Section 11.1.1. In the better case, $T_{a,b}(\boldsymbol{x}_a)$ actually looks like a *Gaussian* distribution with optionally some *outlier* motion candidates, as shown in Fig. 11.4 (a,b). Our formulation is able to cope with such situation since the median operator automatically rejects the *outliers*. However, if a significant number of *optical flow* vectors involved in the motion *path* generated during the motion candidate construction stage are incorrect, $T_{a,b}(\boldsymbol{x}_a)$ may consist of two (or more) clusters (see Fig. 11.4 (c)). Let us assume two clusters, one correct cluster which houses the optimal candidate position $\boldsymbol{x}_b^*$ and one erroneous cluster made of *outliers*. If the wrong cluster is more populated than the correct cluster, the statistical criterion as described previously will unfortunately select a candidate position belonging to the erroneous cluster. That is why propose to involve a candidate quality assessment stage within the proposed processing in order to both discredit *outliers* and encourage the selection of a candidate position within the correct cluster. Let us study more precisely how the quality assessment works.

Each candidate position $\boldsymbol{x}_b^i$ receives a corresponding quality score $Q(\boldsymbol{x}_b^i)$ computed using both the matching cost $C(\boldsymbol{x}_a, \boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a))$ (Section 7.2.2, Chapter 7) and $Inc(\boldsymbol{x}_a, \boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a))$, the inconsistency (Section 10.2.3, Chapter 10), where $\boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a) = \boldsymbol{x}_b^i - \boldsymbol{x}_a$. $Inc(\boldsymbol{x}_a, \boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a))$ corresponds to the *Euclidean* distance to the nearest *reverse* (resp. *direct*) candidate among the distribution if $\boldsymbol{x}_b^i$ is *direct* (resp. *reverse*).

The underlying idea is that we aim at promoting candidates in the neighborhood of high quality candidates. Thus, $Q(\boldsymbol{x}_b^i)$ is used as a voting mechanism [YYGN96]: while computing the medians in Eq. 11.6, each sample $\boldsymbol{x}_b^j$ is considered $Q(\boldsymbol{x}_b^j)$ times to set the occurrence of elements $\left\| \boldsymbol{x}_b^j - \boldsymbol{x}_b^i \right\|_2^2$. Since the quality scores $Q(\boldsymbol{x}_b^i)$ involves matching cost and inconsistency, this voting procedure enforces the color similarity between the corresponding positions in I_a and I_b as well as the motion consistency.

Let us explain more precisely how we compute $Q(\boldsymbol{x}_b^i)$ for each candidate position $\boldsymbol{x}_b^i$. In practice, we combine matching cost and inconsistency to obtain a candidate error through a weighted sum of both terms: $w_{MC}.C(\boldsymbol{x}_a, \boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a)) + w_{Inc}.Inc(\boldsymbol{x}_a, \boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a))$ where w_{MC} and w_{Inc} are defined between 0 and 1 with $w_{MC} + w_{Inc} = 1$. The matching cost aims at performing the selection based on the classic *brightness constancy assumption* whereas with the inconsistency, we aim at choosing a candidate position for which the corresponding displacement field $\boldsymbol{d}_{a,b}^i(\boldsymbol{x}_a)$ between I_a and I_b is consistent with a displacement field between I_b and I_a.

Starting from this candidate error estimation, the quality score $Q(\boldsymbol{x}_b^i)$ is computed for each candidate position $\boldsymbol{x}_b^i$ as follows. The maximum value and the minimum value of the candidate errors among all candidates are mapped from 0 to a predefined integer Q_{max}. Intermediate candidate errors are mapped to the line defined by these two values and the result is rounded to the nearest integer: $Q(\boldsymbol{x}_b^i) \in [\![0, \ldots, Q_{max}]\!]$. The higher $Q(\boldsymbol{x}_b^i)$ is, the smaller the candidate error is.

The statistical processing being applied for each pixel $x_a \in I_a$ independently, we introduce in what follows a global optimization method which includes a regularization process.

Global optimization (*GO*) for motion vector selection

To introduce spatial regularization in the candidate selection process, we perform a global optimization stage that fuses for each pixel motion candidates into a single optimal motion field, following the approach of [LRR08]. We introduce $\boldsymbol{L} = \{l_{\boldsymbol{x}_a}\}$ as a labelling of pixels $\boldsymbol{x}_a$ of I_a where each label indicates one of the candidates of $T_{a,b}(\boldsymbol{x}_a)$. Let $d_{a,b}^{l_{\boldsymbol{x}_a}}$ be the corresponding motion vectors of candidates of $T_{a,b}(\boldsymbol{x}_a)$. We define the following energy and minimize it with *fusion moves* [LRR08, LRRB10].

$$\begin{aligned} E_{a,b}(\boldsymbol{L}) &= \sum_{\boldsymbol{x}_a} \rho_d(w_{MC}.C(\boldsymbol{x}_a, d_{a,b}^{l_{\boldsymbol{x}_a}}(\boldsymbol{x}_a)) + w_{Inc}.Inc(\boldsymbol{x}_a + d_{a,b}^{l_{\boldsymbol{x}_a}}(\boldsymbol{x}_a))) \\ &+ \sum_{<\boldsymbol{x}_a, \boldsymbol{y}_a>} \alpha_{\boldsymbol{x}_a, \boldsymbol{y}_a}.\rho_r(\left\| d_{a,b}^{l_{\boldsymbol{x}_a}}(\boldsymbol{x}_a) - d_{a,b}^{l_{\boldsymbol{y}_a}}(\boldsymbol{y}_a) \right\|_1) \end{aligned} \tag{11.7}$$

The data term involves the matching cost $C(\boldsymbol{x}_a, d_{a,b}^{l_{\boldsymbol{x}_a}})$ and the inconsistency value $Inc(\boldsymbol{x}_a + d_{a,b}^{l_{\boldsymbol{x}_a}})$ which is introduced as previously to make the energy be more robust to motion *outliers*. The regularization term involves motion similarities with neighboring positions. α_{x_a, y_a} accounts for local color similarities in frame I_a, as done in Chapter 10 (Eq. 10.5, Section 10.1.3).

Pixels considered as occluded (i.e. pixels for which it has not been possible to find any candidate positions) are not involved within the regularization process. Functions ρ_d and ρ_r are respectively the negative log of a *Student-t* distribution and the *Geman-McClure* penalty function [LRR08].

The *fusion moves* algorithm fuses candidates pair by pair up to getting an optimal field $d_{a,b}^*$ but its application to a large set of motion fields is however limited by the computational load.

Motion vector selection framework (*SP*+*GO*)

Therefore, as previously mentioned, we propose to combine statistical processing and the above global optimization stage to simultaneously take into account:

- information about the point distribution (as done in the statistical processing),
- a robust selection based on the intrinsic motion field quality (matching cost and inconsistency),
- a spatial regularization, involved in the energy formulation Eq. 11.7.

The combination of these two selection tools is done as follows. For each $\boldsymbol{x}_a \in I_a$, the statistical processing is applied to the whole set $T_{a,b}(\boldsymbol{x}_a)$ in order to select the N_{opt} best candidates of

the distribution with the criterion of median minimization of Eq. 11.6. Then, the *fusion moves* algorithm fuses by pairs the resulting N_{opt} candidate displacement fields, following the previously described global optimization method (i.e. minimization of Eq. 11.7), up to obtaining the optimal displacement field between the distant frames I_a and I_b.

11.2 Experimental results between a pair of distant frames

The experiments of the combinatorial *multi-step* integration and the statistical selection (*CISS*) proposed in Section 11.1 have been conducted as follows. In Section 11.2.1, we compare the three different selection procedures previously detailed: the statistical processing, the global optimization method (Eq. 11.7 minimized by *fusion moves*) and finally the statistical processing combined with the global optimization, as suggested. Section 11.2.2 explores the trade-off between the number of candidate positions and the quality of the results. Finally, the whole *CISS* framework is compared to state-of-the-art methods in Section 11.2.3.

Our experiments focus on pairs of frames taken from four cropped sequences: *MPI-S*1-25-55 [GKT+], *Hope*, *Newspaper*-2 and *Walking-Couple*-0-60. The characteristics of these sequences are described below:

- *MPI-S*1-25-55: 950×800 frames containing temporary occlusions (kiosk in the background), non-rigid deformations, large uniform areas and strong spatial lighting variations (coat and hair of the woman): Fig. 11.5,
- *Hope*: 800×500 frames containing large motion, very large uniform areas (wall and foreground characters) and zooming: Fig. 11.6,
- *Newspaper*-2: 600×700 frames with fixed background and large uniform areas: Fig. 11.7,
- *Walking-Couple*-0-60: 700×700 frames featuring periodic textures, large uniform areas, spatial lighting variations and large motion: Fig. 11.8.

For the selected pairs, the combinatorial *multi-step* integration has been performed taking input elementary *optical flow* fields estimated with the 2*D-DE optical flow* estimator [RTDC12]. steps 1, 2, 3, 4, 5, 15 and 30 (not for *Walking-Couple*-0-60) have been used. For all the experiments, the parameters are as follows. The maximum number of concatenations and the number of randomly selected motion *paths* are respectively $N_c = 7$ and $N_s = 100$. The maximum integer quality score value is $Q_{max} = 2$. In addition, the weights which make the balance between matching cost and inconsistency are defined such as: $w_{MC} = w_{Inc} = 0.5$. Finally, the number of candidate positions selected by the statistical processing is $N_{opt} = 3$.

11.2.1 Comparisons between the selection procedures

Given a large amount of candidate displacement fields obtained through the proposed combinatorial *multi-step* integration stage, we have compared three selection procedures: 1) the

(*a*) I_{25} (*b*) I_{35} (*c*) I_{45} (*d*) I_{55}

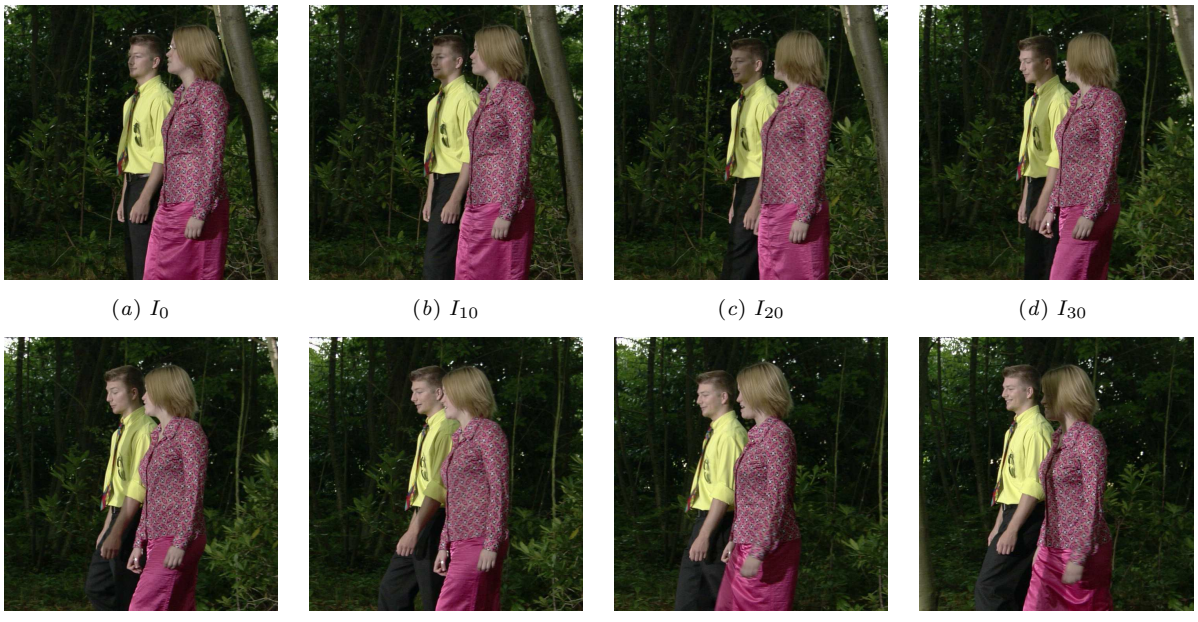
Figure 11.5: Source frames from the *MPI-S*1-*25-55* sequence.

(*a*) I_{5036} (*b*) I_{5054} (*c*) I_{5063}

Figure 11.6: Source frames from the *Hope* sequence.

(*a*) I_{160} (*b*) I_{180} (*c*) I_{200} (*d*) I_{220} (*e*) I_{230}

Figure 11.7: Source frames from the *Newspaper*-2 sequence.



(*a*) I_0 (*b*) I_{10} (*c*) I_{20} (*d*) I_{30}

(*e*) I_{35} (*f*) I_{40} (*g*) I_{50} (*h*) I_{60}

Figure 11.8: Source frames from the *Walking-Couple-0-60* sequence.

(*a*) I_{46} (*b*) *SP* (*c*) *GO* (*d*) *SP*+*GO*

Figure 11.9: Reconstruction of the kiosk of I_{46} (*a*) from I_{25} (*MPI-S1-25-55* sequence [GKT+]) with: *b*) the statistical processing (*SP*), *c*) the global optimization (*GO*) method solved by *fusion moves* [LRRB10], *d*) both combined (*SP*+*GO*).

Frame pairs	{25,45}	{25,46}	{25,47}	{25,48}	{25,49}	{25,50}
SP	12.72	15.27	21.7	25.33	**24.48**	24.7
GO	11.19	14	11.14	13.7	21.7	22.22
SP*+*GO	**12.84**	**16.11**	**24.75**	**25.55**	24	**24.79**

Table 11.1: Comparison through registration and *PSNR* assessment between: 1) the statistical processing (*SP*), 2) the global optimization (*GO*), 3) the statistical processing combined to the global optimization (*SP*+*GO*), as suggested. *PSNR* scores are computed on the kiosk of the *MPI-S*1-25-55 sequence (Fig. 11.5). Low *PSNR* for first pairs are due to the foreground object which degrades the estimation.

statistical processing (*SP*), 2) the global optimization (*GO*) method solved by *fusion moves* [LRRB10], 3) the statistical processing combined with the global optimization (*SP*+*GO*).

The final displacement fields of each one of these methods have been compared through registration and *PSNR* assessment. For a given pair $\{I_a, I_b\}$, the final fields are used to reconstruct I_a from I_b through motion compensation and color *PSNR* scores are computed between I_a and the registered frame $I_a^{'}$ for non-occluded pixels.

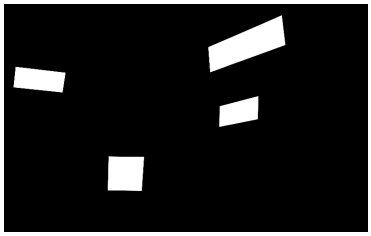
Tab. 11.1 and Tab. 11.2 (see page 187) show quantitative comparisons through *PSNR* computed for various distances between I_a and I_b respectively on the kiosk of *MPI-S*1-25-55 and on whole images of *Newspaper*-2. An example of registration of the kiosk for a distance of 21 frames is provided Fig. 11.9.

Results show that *SP* is better than *GO* for all pairs. The main differences occur from the pairs $\{25, 47\}$ and $\{25, 48\}$ for which *SP* reaches $21.7dB$ and $25.33dB$ against $11.14dB$ and $13.7dB$ for *GO*. The low diversity of candidates at the output of *SP* limits the effect of regularization and explains the slight improvement between *SP* and *SP*+*GO*. Moreover, note that low *PSNR* scores for first pairs in Tab. 11.1 are due to the foreground character which degrades motion estimation. The example of Fig. 11.9 is interesting due to the temporary occlusion of the kiosk which is jumped by *multi-step* motion *paths*. For this complex situation, *SP*+*GO* is more adapted than *GO* despite some artifacts still visible.

In the context of video editing, we evaluate the accuracy of *SP*+*GO* and *GO* by motion compensating in I_b logos manually inserted in I_a. Fig. 11.10 and 11.11 present results for *Hope*

(a) Original image I_{5036}

(b) Insertion in I_{5036}



(c) Insertion mask, I_{5036}

(d) Prop. to I_{5054}, *GO*

(e) Prop. to I_{5054}, *SP+GO*

Figure 11.10: Logo insertion in I_{5036} (a,b,c) and propagation to I_{5054} (*Hope* sequence, Fig. 11.6). We compare the global optimization (*GO*) method (d) with the statistical processing (*SP*) combined to *GO* (*SP+GO*) (e).

(a) Original image I_{230}

(b) Insertion in I_{230}

(c) Prop. to I_{160} *GO*

(d) Prop. to I_{160} *SP+GO*

Figure 11.11: Logo insertion in I_{230} (a,b) and propagation to I_{160} (*Newspaper*-2 sequence, Fig. 11.7). We compare the global optimization (*GO*) method (c) with the statistical processing (*SP*) combined to *GO* (*SP+GO*) (d).

and *Newspaper*-2 with a distance of 18 and 70 frames respectively. For both cases, *SP+GO* shows a clear improvement compared to *GO*. Indeed, *GO* distorts the texture structure and creates shadow artifacts whereas *SP+GO* performs texture propagation without any visible artifact.

11.2.2 How many motion *paths*?

In the previous experiments, we focused on the accuracy of the dense matching task between distant frames. Up to now, a relatively large amount of candidate positions have been considered (around 100). Note that this amount is limited by the memory capability. Let us study now how the quality of the matching behaves with respect to the number of candidates. Moreover, we want to explore the variability of the results given the fact that the motion *paths* are chosen randomly. In this direction, our experiment consists in performing *CISS*, i.e. the combinatorial *multi-step* integration and the statistical selection (*SP+GO* as suggested), on a single pair of frames ($\{I_0, I_{25}\}$) taken from the *Walking-Couple-0-60* sequence using different number of candidates positions. In addition, each test (i.e. *CISS* for a given number of candidates) has been repeated 6 times in order to see in what extend the random aspect of the *CISS* framework affects the quality of the results.

The quality evaluation has been done through registration and *PSNR* assessment on an area located in I_0 within the dress of the woman of *Walking-Couple-0-60*. As shown in Fig. 11.14, this area (surrounded by a white square Fig. 11.14 (a) and displayed in Fig. 11.14 (c)) is reconstructed via our displacement vectors using color values of I_{25} (Fig. 11.14 (b)). A result of registration is given in Fig. 11.14 (d) with $N_s = 90$. Note that the *PSNR* score for this example is $17.82dB$. Although the registration is performed satisfactorily, the *PSNR* score is not very large due to illumination changes which occur between I_0 and I_{25}.

Following this protocol, Fig. 11.12 shows all the *PSNR* scores obtained with different number of candidates (from 9 to 120) and with 6 different executions for each. For a given small number of candidates, we can notice that a large range of *PSNR* scores is obtained. For $N_s = 18$ candidate positions, the *PSNR* interval ranges from $16.2dB$ to $17.67dB$ which represents a very large dynamic. When increasing the number of candidates, the *PSNR* range decreases. Moreover, the obtained *PSNR* scores are generally better. Thus, for $N_s = 120$ for instance, the six *PSNR* scores obtained are all between $17.7dB$ and $17.77dB$. Although a high reconstruction quality can be achieved with only $N_s = 9$ candidates (the maximum *PSNR* score for $N_s = 9$ is $17.77dB$), the opposite situation is also possible (the minimum *PSNR* is $17.07dB$, i.e. $0.7dB$ less). Therefore, it is safe to consider a more substantial number of candidates to avoid the situation where the random selection does not select good motion *paths*. By increasing N_s, we increase the probability to base the matching between distant frames on good candidates.

Fig. 11.13 justifies these finding by plotting the average and the standard deviation computed on *PSNR* values with respect to N_s, the number of candidate positions. The average *PSNR* values tend to increase when considering a wider distribution $T_{a,b}(\boldsymbol{x}_a)$ (from $17.10dB$ with $N_s = 24$ to $17.74dB$ with $N_s = 120$). On the contrary, the standard deviation decreases from 0.38 with $N_s = 24$ to 0.03 with $N_s = 120$.

It is important to note that the guided random selection selects a reasonable number of *step sequences* which are common to all the pixels $\boldsymbol{x}_a$ of I_a. This explains the fact that the quality of the corresponding motion *paths* (or the resulting candidate positions in I_b) cannot be involved in the selection. Indeed, this selection task is done once for the whole image. Performing a guided random selection based on matching quality would require both motion *paths* generation

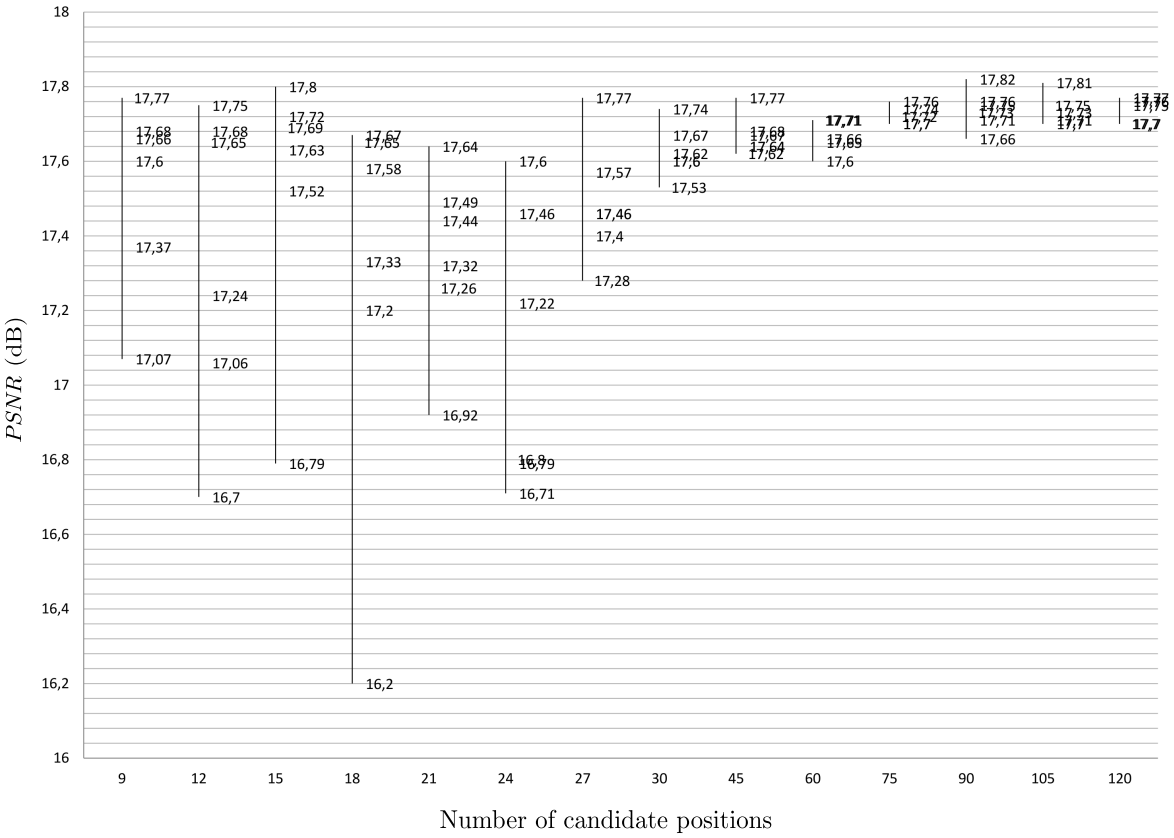


Figure 11.12: *PSNR* scores obtained with different number of candidates (from 9 to 120) and with 6 different executions for each. The quality assessment has been performed by comparing a block within the dress of the woman in I_0 and the corresponding registered block from I_{25} (*Walking-Couple*-0-60 sequence, Fig. 11.14).

and selection procedures for each pixel taken independently. It would be hardly feasible in our context of dense motion estimation due to computational and memory issues. Although it has not been tested in this thesis, an hybrid solution could be considered. It deals with multiple successive random selections which would finally translate in generating several sets $T_{a,b}(\boldsymbol{x}_a)$ of candidate positions. A final set could be computed using the best candidates of all the previously generated sets based on matching quality this time.

Finally, considering $N_s = 60$, the plots displayed in Fig. 11.13 suggest that a set of proposals which is 2 times bigger would result approximately in the same quality. This is in line with [LRR08] whose authors highlight an asymptotic quality value with respect to the number of proposals as input.

11.2.3 Performance of *CISS*

We propose two experiments to assess the quality of the whole *CISS* framework with respect to state-of-the-art methods. First, Tab. 11.2 gives *PSNR* scores related to the registration of whole images of *Newspaper*-2. The *CISS* framework with *GO* or *SP+GO* selection procedures are compared to the *multi-step* fusion (*MSF*) method described in Section 10.2. Second, a logo insertion and propagation example is provided in Fig. 11.15. The classic accumulation (i.e.

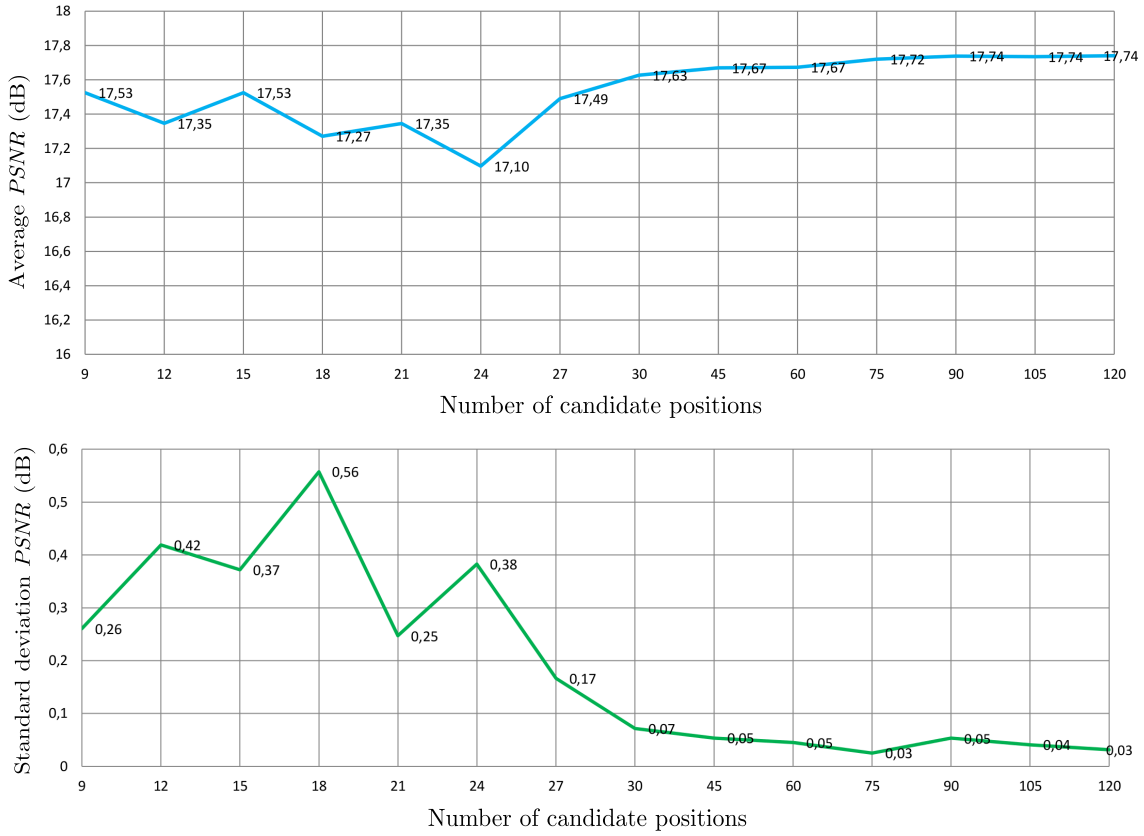


Figure 11.13: Average and standard deviation computed for each number of candidate position using the *PSNR* scores displayed in Fig. 11.12.

(*a*) (*b*) (*c*) (*d*)

Figure 11.14: Reconstruction (d) of a block of the dress in I_0 (*a*,*c*) from I_{25} (*b*) (*Walking-Couple-0-60* sequence) using displacement vectors coming from the proposed combinatorial *multi-step* integration and the statistical selection framework.

Frame pairs	{160,190}	{160,200}	{160,210}	{160,220}	{160,230}
CISS with *GO* (Section 11.1)	21.11	19.33	18.11	17.06	16.29
CISS* with *SP+GO (Section 11.1)	**21.42**	**19.53**	**18.3**	**17.74**	**17.09**
MSF (Section 11.2)	20.5	18.22	17.8	16.95	16.6

Table 11.2: Registration and *PSNR* assessment with: the combinatorial integration followed by the global optimization (*GO*); by the statistical processing combined to *GO* (*SP+GO*); the multi-step fusion (*MSF*) method described in Section 10.2. *PSNR* scores are computed on the whole images of *Newspaper*-2.

(*a*) Source frame I_0 (*b*) I_0 with logo (*c*) Insertion mask, I_0

(*d*) Prop. to I_{25}, *LDOF acc* (*e*) Prop. to I_{25}, *MSF*(2*D-DE*) (*f*) Prop. to I_{25}, *CISS*(2*D-DE*)

Figure 11.15: Texture insertion in I_0 and propagation in I_{25} (*Walking-Couple-0-60* sequence). We compare: *Euler* integration using *LDOF* [BM11] elementary *optical flow* fields computed between consecutive frames: *LDOF acc*; *multi-step* flow fusion (*MSF*) described in Section 10.2 using 2*D-DE multi-step* elementary *optical flow* fields: *MSF*(2*D-DE*); the combinatorial integration followed by the statistical processing combined to the global optimization (*CISS*) using 2*D-DE multi-step* elementary *optical flow* fields: *CISS*(2*D-DE*).

direct integration) of *optical flow* estimated between consecutive frames using *LDOF* (*LDOF acc*) is involved as well as *MSF* and *CISS* with 2*D-DE multi-step* elementary *optical flow* fields as inputs.

The proposed combinatorial integration combined to *SP+GO* gives better performance compared to the *multi-step* fusion (*MSF*) method (Section 10.2) according to *PSNR* scores of Table 11.2. This comparison is encouraging since the *MSF* method itself has been shown in Section 10.3 to outperform state-of-the-art methods such as [ZPB07, SBK10, BM11]. Moreover, *CISS* performs a better logo propagation than *LDOF acc* or *MSF* in the context illustrated in Fig. 11.15, i.e. where non-rigid periodic structures follow a large displacement.

11.2.4 Conclusion

To conclude, the combinatorial *multi-step* integration and statistical selection (*CISS*) method proposed in Section 11.1 is able to perform an accurate dense motion estimation between a single pair of distant frames. Motivated by the good results obtained through the presented experiments, we suggest in the next section, Section 11.3, to extend the *CISS* approach to the whole sequence in order to generate long-term dense *from-the-reference* and *to-the-reference* displacement fields.

11.3 Statistical *multi-step* Flow (*StatFlow*)

Toward our goal of dense motion estimation in long video sequences, we now present *Statistical multi-step Flow* (*StatFlow*), a new long-term dense motion estimation framework. This framework aims at computing *from-the-reference* and *to-the-reference* long-term dense displacement fields by extending to the whole sequence the combinatorial *multi-step* integration and statistical selection, *CISS*, described in Section 11.1 for a pair of distant frames.

As in Section 10, we consider a sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\dots,N]\!]}$ including I_{ref} considered as a reference frame. Moreover, we focus on dense motion estimation between the reference frame I_{ref} and each frame I_n of the sequence by addressing the computation of *from-the-reference* and *to-the-reference* displacement fields, respectively $\boldsymbol{d}_{ref,n}$ and $\boldsymbol{d}_{n,ref}$ $\forall n \in [\![0,\dots,N]\!] \neq ref$.

The proposed *StatFlow* estimator is based first of all on a slightly different version of *CISS*, called *CISS-K*, which is applied independently for each pair of frames $\{I_{ref}, I_n\}$. By this way, we generate K *from-the-reference* and *to-the-reference* dense motion correspondences between the reference frame I_{ref} and each of the subsequent images I_n. Significant improvements of *CISS* have been developed, especially to reduce erroneous motion estimates by focusing on the inconsistency between *from-the-reference* and *to-the-reference* displacement vectors. Once all these pairs of frames have been processed through *CISS-K*, we propose in a second stage to provide an accurate final dense matching by applying a new iterative motion refinement (*IMR*) step which involves temporal smoothness constraints. The whole procedure is schematized in Fig. 11.16.

Compared to the sequential approaches *MS-GC* and *MSF* (Chapter 10), *StatFlow* is related to the exhaustive *multi-step* strategy introduced in Section 9.3.1 (Chapter 9). Indeed, by relying on combinatorial integration independently for each pair of frames $\{I_{ref}, I_n\}$, *StatFlow* aims at manipulating a large amount of motion *paths* which are built without only relying on the optimal displacement fields computed for previous frames. Through this property, we limit the sequential propagation of errors which may occur with *MS-GC* and *MSF*. Moreover, the statistical selection involved within *StatFlow* gives a more robust displacement field selection tool than classical *optical flow* assumptions that frequently fail between distant frames.

In the following, we describe precisely the two main stages (Fig. 11.16) performed by the proposed *statistical multi-step flow* (*StatFlow*) method:

1. temporally independent generation of K *from-the-reference* and *to-the-reference* dense motion correspondences for each pair of frames $\{I_{ref}, I_n\}$ through *CISS-K*: Section 11.3.1,

2. iterative motion refinement (*IMR*) via temporal consistency constraints to obtain an accurate final dense matching: Section 11.3.2.

11.3.1 Motion candidates generation through *CISS-K*

This initial motion candidates generation focuses on each pair of frames $\{I_{ref}, I_n\}$ independently. The goal is to compute for each pixel $\boldsymbol{x}_{ref}$ (resp. $\boldsymbol{x}_n$) of I_{ref} (resp. I_n) K candidate positions in I_n (resp. I_{ref}). Since the processing is applied analogously to both *from-the-reference* and *to-the-reference* directions, we focus our explanations in this Section 11.3.1 only on the estimation of *from-the-reference* displacement fields, i.e. from I_{ref} to I_n.

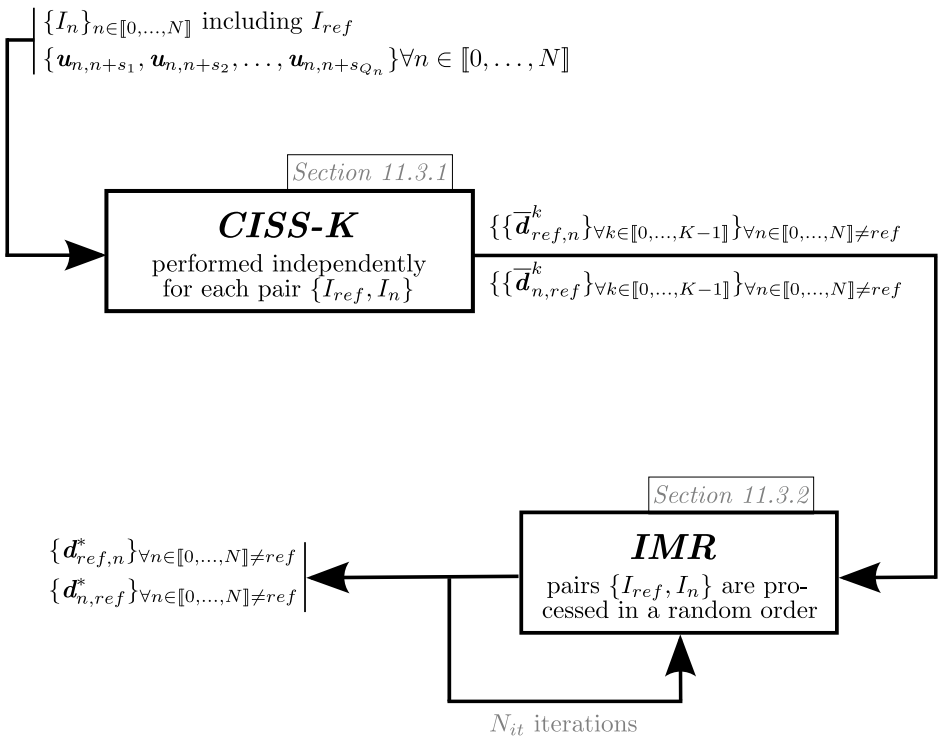


Figure 11.16: Overview of the proposed long-term dense motion estimation framework: *Statistical multi-step Flow* (*StatFlow*). Two main stages are involved: 1) temporally independent generation of K initial dense motion correspondences for each pair of frames $\{I_{ref}, I_n\}$ through *CISS-K* (Section 11.3.1) 2) iterative motion refinement (*IMR*) via temporal consistency constraints to obtain an accurate final dense matching (Section 11.3.2).

As previously mentioned, our motion candidates generation extends to the whole sequence the combinatorial *multi-step* integration and the selection procedure (*CISS*) introduced in Section 11.1 for dense motion estimation between a single pair of distant frames. Indeed, the idea is to apply *CISS* for each pair of frames $\{I_{ref}, I_n\}$ in order to obtain a set $T_{ref,n}(\boldsymbol{x}_{ref}) = \{\boldsymbol{x}_n^i\}_{i\in[\![0,\ldots,K_{\boldsymbol{x}_{ref}}-1]\!]}$ of $K_{\boldsymbol{x}_{ref}} >> K$ candidate positions in I_n for each pixel $\boldsymbol{x}_{ref}$ of I_{ref} via multiple motion *paths*. The idea is then to select, for each $\boldsymbol{x}_{ref}$, K candidate positions among $T_{ref,n}(\boldsymbol{x}_{ref})$ and not only one as it was the case in Section 11.1. This explains the denomination *CISS-K*. In addition, the *CISS* approach itself has been significantly improved.

In this context, we described in the following the input data before focusing on how *CISS* has been improved and extended to generate K accurate displacement fields for each pair $\{I_{ref}, I_n\}$.

Input *optical flows* fields

As inputs, our method takes into account a set of *multi-step* elementary *optical flow* fields estimated from each frame of the sequence including I_{ref}. As previously, $S_n = \{s_1, s_2, \ldots, s_{Q_n}\} \subset \{1, \ldots, N-n\}$ is the set of Q_n possible *steps* at instant n which means that the following set of *optical flow* fields starting from I_n is available: $\{\boldsymbol{u}_{n,n+s_1}, \boldsymbol{u}_{n,n+s_2}, \ldots, \boldsymbol{u}_{n,n+s_{Q_n}}\}$.

Input *multi-step* elementary *optical flow* fields are provided with attached occlusion and inconsistency masks. For the pair $\{I_n, I_{n+s_i}\}$ with $s_i \in \{1, \ldots, N-n\}$, the occlusion mask attached to the elementary *optical flow* field $\boldsymbol{u}_{n,n+s_i}$ indicates the visibility of each pixel of I_n in I_{n+s_i}. The inconsistency mask attached to $\boldsymbol{u}_{n,n+s_i}$ distinguishes consistent and inconsistent pixels among the pixels marked as visible. For recall, a pixel $\boldsymbol{x}_n \in I_n$ is defined as inconsistent (resp. consistent) if the *optical flow* vector $\boldsymbol{u}_{n,n+s_i}(\boldsymbol{x}_n)$ starting from I_n is intrinsically inconsistent (resp. consistent) with respect to the corresponding *optical flow* vector $\boldsymbol{u}_{n+s_i,n}$ starting from I_{n+s_i} and coming back to I_n (threshold ϵ_{Inc} applied on $Inc(\boldsymbol{x}_n, \boldsymbol{u}_{n,n+s_i}(\boldsymbol{x}_n))$).

Improvements: from *CISS* to *CISS-K*

First of all, compared to the baseline *CISS* method presented in Section 11.1, our displacement fields selection procedure differently combines the statistical selection (*SP*) step and the global optimization (*GO*). For each $\boldsymbol{x}_{ref} \in I_{ref}$, we select $K_{sp} = 2 \times K$ candidate positions through statistical selection among the large distribution $\boldsymbol{T}_{ref,n}(\boldsymbol{x}_{ref})$, with $K_{sp} < K_{\boldsymbol{x}_{ref}}$. Then, we randomly group by pairs these K_{sp} candidates and choose the K best candidates $\overline{\boldsymbol{x}}_n^k$ $\forall k \in [\![0, \ldots, K-1]\!]$ by pair-wise fusing these candidates following the global flow fusion approach of Eq. 11.7. Finally, this same global optimization method fuses these K best candidates to obtain an optimal one: $\boldsymbol{x}_n^*$. In other words, these two last steps give a set of K candidate displacement fields $\overline{\boldsymbol{d}}_{ref,n}^k$ $\forall k \in [\![0, \ldots, K-1]\!]$ and finally $\boldsymbol{d}_{ref,n}^*$, the optimal displacement field.

For pairs of frames relatively close or in case of temporary occlusions, the statistical selection is not well adapted due to the small amount of initial candidates. Therefore, between 1 and K candidate positions, we do not perform any selection and all the candidates are kept. Between $K+1$ and K_{sp} candidates, we only use the global optimization method up to obtain the K best candidate fields. If the number of candidates exceeds K_{sp}, the statistical processing and the global optimization method are applied as explained above.

Another improvement with respect to the initial *CISS* method deals with the fact that we now provide further focus to inconsistency reduction between *from-the-reference* and *to-the-reference* motion vectors. Consequently, stronger consistency constraints are proposed:

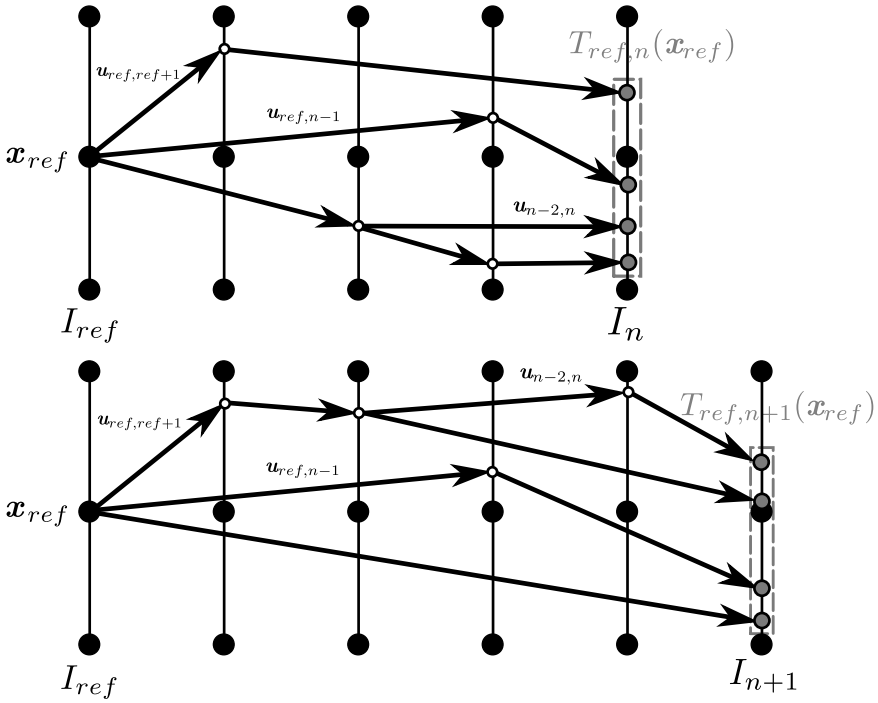


Figure 11.17: Multiple motion candidates are generated via a guided-random selection among all possible motion *paths*. This combinatorial integration (Section 11.1) is done independently for each pair $\{I_{ref}, I_n\}$ which limits the correlation between candidates selected for neighbouring frames.

1. Only input *multi-step* elementary *optical flow* vectors considered as consistent according to their inconsistency masks are used to generate motion *paths* between I_{ref} and I_n.
2. We introduce an outlier removal step before the statistical selection. It consists in ordering all the candidates of the distribution with respect to their inconsistency values. Then, a percentage $R_{\%}$ of bad candidates is removed and the selection is performed on the remaining ones.
3. The intrinsic candidate quality involved in the statistical selection computes $Q(\boldsymbol{x}_n^i)$, the quality score assigned to each candidate $\boldsymbol{x}_n^i$ (Section 11.1.2), based on inconsistency only.
4. At the end of the combinatorial integration and the selection procedure between I_{ref} and I_n, the optimal displacement field $\boldsymbol{d}^*_{ref,n}$ is incorporated into the processing between I_n and I_{ref} which aims at enforcing the motion consistency between *from-the-reference* and *to-the-reference* fields.

Extention to the whole sequence

CISS-K processes independently all the pairs $\{I_{ref}, I_n\}$. Only N_c, the maximum number of concatenations, changes with respect to the temporal distance between the considered frames. In practice, we determine N_c from Eq. 11.8:

$$N_c(n) = \begin{cases} \mid n - ref \mid \text{ if } \mid n - ref \mid \leq 5 \\ \alpha_0 . \log 10(\alpha_1 . \mid n - ref \mid) \text{ otherwise} \end{cases} \tag{11.8}$$

This function, empirically built, is a good compromise between a too large number of concatenations which leads to large propagation errors and a too small number N_c which limits the effectiveness of the statistical processing due to an insufficient total number of candidate positions.

Discussion on the correlation between candidates

The guided-random selection which selects for each pair of frames $\{I_{ref}, I_n\}$ one part of all the possible motion *paths* limits the correlation between candidates respectively estimated for neighbouring frames. This avoids the situation in which a single estimation error is propagated and therefore badly influences the whole trajectory, as it may occur for sequential methods such as *MS-GC* or *MSF* (Chapter 10).

The example Fig. 11.17 shows the motion *paths* selected by the guided-random selection for the pairs $\{I_{ref}, I_n\}$ and $\{I_{ref}, I_{n+1}\}$. We can notice that motion *paths* between I_{ref} and I_{n+1} are not highly correlated with those between I_{ref} and I_n. Indeed, the sets of elementary *optical flow* vectors involved in both cases are not the same except for $\boldsymbol{u}_{ref,ref+1}$ and $\boldsymbol{u}_{ref,n-1}$ which are then concatenated with different vectors. $\boldsymbol{v}_{n-2,n}$ contributes for both cases but the considered vectors do not start from the same position.

These considerations about the statistical independence of the resulting displacement fields are not addressed by state-of-the-art methods for which a strong temporal correlation is generally inescapable.

11.3.2 Iterative motion refinement (*IMR*)

The previous stage guarantees a low correlation between the K motion candidates respectively estimated for pairs $\{I_{ref}, I_n\}$. Without losing this key characteristic, this second stage aims at iteratively refining these motion estimates while enforcing the temporal smoothness along the sequence, up to obtain an accurate final bi-directional dense matching for each pair $\{I_{ref}, I_n\}$.

We propose to question the matching between each pixel $\boldsymbol{x}_{ref}$ (resp. $\boldsymbol{x}_n$) of I_{ref} (resp. I_n) and the selected position $\boldsymbol{x}^*_n$ (resp. $\boldsymbol{x}^*_{ref}$) in I_n (resp. I_{ref}) established during the previous iteration (or the initial motion candidates generation stage if the current iteration is the first one). For this task, we generate several competing candidates which are compared to $\boldsymbol{x}^*_n$ (resp. $\boldsymbol{x}^*_{ref}$) through a global optimization approach.

Competing candidates

The competing candidates used to question $\boldsymbol{x}^*_n$ (resp. $\boldsymbol{x}^*_{ref}$) are illustrated in Fig. 11.18 for the *from-the-reference* direction. These competing candidates are:

- the K initial candidate positions $\overline{\boldsymbol{x}}^k_n$ (resp. $\overline{\boldsymbol{x}}^k_{ref}$) $\forall k \in [\![0, \dots, K-1]\!]$ obtained during the initial motion candidates generation stage (Section 11.3.1),
- a candidate position coming from the previous estimation of $\boldsymbol{d}^*_{n,ref}$ (resp. $\boldsymbol{d}^*_{ref,n}$) which is inverted to obtain $\boldsymbol{x}^r_n$ (resp. $\boldsymbol{x}^r_{ref}$), as illustrated in Fig. 11.18,
- candidates from neighbouring frames to enforce temporal smoothing. Let W be the temporal window of width w centered around I_n. In the *from-the-reference* direction, let $\boldsymbol{x}^*_m \in I_m$ with $m \in [\![n - \frac{w}{2}, \dots, n + \frac{w}{2}]\!]$ and $m \neq n$ be the motion candidate corresponding to $\boldsymbol{x}_{ref}$ of I_{ref} obtained at previous iteration in the neighbouring image I_m. We use the

elementary *optical flow* fields $\boldsymbol{u}_{m,n}$ between I_m and I_n to obtain from $\boldsymbol{x}_m^* \in I_m$ the new candidate $\boldsymbol{x}_n^m$ in I_n:

$$\begin{aligned} \boldsymbol{x}_n^m &= \boldsymbol{x}_m^* + \boldsymbol{u}_{m,n}(\boldsymbol{x}_m^*) \\ &= \boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,m}^*(\boldsymbol{x}_{ref}) + \tilde{\boldsymbol{u}}_{m,n}(\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,m}^*(\boldsymbol{x}_{ref})) \end{aligned} \tag{11.9}$$

Similarly, in the *to-the-reference* direction, the elementary *optical flow* field $\boldsymbol{u}_{n,m}$ is concatenated with $\boldsymbol{d}_{m,ref}^*$ to propose new candidates in I_{ref}.

Global optimization approach

We perform a global optimization method in order to fuse the previously described competing candidates into a single optimal displacement field, similarly to [LRRB10]. For this task, a new energy has been built and two formulations are proposed depending on the type (*from-the-reference* or *to-the-reference*) of the displacement fields to be refined.

In the *from-the-reference* case, we introduce $\boldsymbol{L} = \{l_{\boldsymbol{x}_{ref}}\}$ as a labelling of pixels $\boldsymbol{x}_{ref}$ of I_{ref} where each label indicates $\boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}}$, one of the candidates listed above. Let $\boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}$ be the corresponding motion vector. We define the energy in Eq. 11.10 and we use the *fusion moves* algorithm [LRR08, LRRB10] to minimize it with respect to $\boldsymbol{L}$:

$$\begin{aligned} E_{ref,n}(\boldsymbol{L}) &= E_{ref,n}^d(\boldsymbol{L}) + E_{ref,n}^r(\boldsymbol{L}) = \sum_{\boldsymbol{x}_{ref}} \rho_d(\epsilon_{ref,n}^d) \\ &+ \sum_{<\boldsymbol{x}_{ref},\boldsymbol{y}_{ref}>} \alpha_{\boldsymbol{x}_{ref},\boldsymbol{y}_{ref}} \rho_r(\left\| \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref}) - \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{y}_{ref}}}(\boldsymbol{y}_{ref}) \right\|_1) \end{aligned} \tag{11.10}$$

The data term $E_{ref,n}^d$, detailed in Eq. 11.11, involves the matching cost $C(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}})$ and the inconsistency value $Inc(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}})$ with respect to $\boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}$ as done in Chapter 11.1. In addition, we propose to introduce temporal smoothness constraints, $T_{sc}(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref}))$, into the energy formulation in order to efficiently guide the motion refinement. The weights w_{MC}, w_{Inc} and $w_{T_{sc}}$ are defined between 0 and 1 such as $w_{MC} + w_{Inc} + w_{Tsc} = 1$

$$\begin{aligned} \epsilon_{ref,n}^d &= w_{MC}.C(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref})) + w_{Inc}.Inc(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref})) \\ &+ w_{T_{sc}}.T_{sc}(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref})) \end{aligned} \tag{11.11}$$

The temporal smoothness constraints $T_{sc}(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref}))$ translate into three new terms which are computed with respect to each neighbouring candidate $\boldsymbol{x}_m^*$ defined for the frames inside the temporal window W of width w:

$$T_{sc}(\boldsymbol{x}_{ref}, \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref})) = \sum_{\substack{m=n-\frac{w}{2} \\ m \neq n}}^{n+\frac{w}{2}} \left[C(\boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}}, \boldsymbol{x}_m^* - \boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}}) + ed_{m,n} + ed_{n,m} \right] \tag{11.12}$$

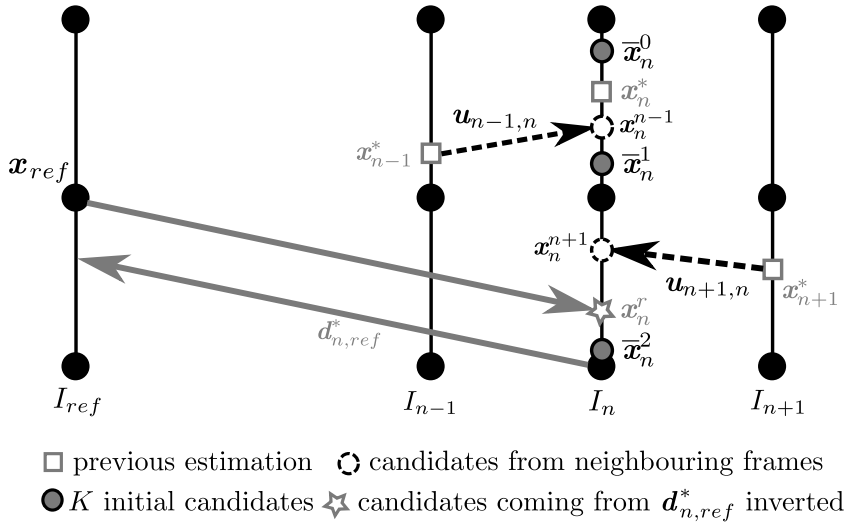


Figure 11.18: The displacement field $\boldsymbol{d}_{ref,n}^*$ is questioned by considering for each pixel $\boldsymbol{x}_{ref}$ of I_{ref} the following candidate positions in I_n: candidates coming from neighbouring frames, the K initial candidates and a candidate obtained via $\boldsymbol{d}_{n,ref}^*$ inverted.

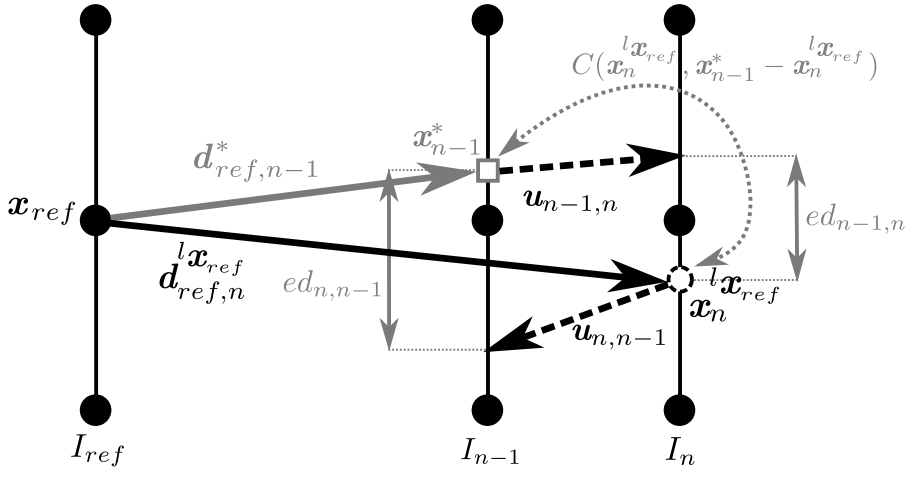


Figure 11.19: Matching cost and *Euclidean* distances $ed_{n,m}$ and $ed_{m,n}$ defined with respect to each temporal neighboring candidate $\boldsymbol{x}_m^*$ and involved in the proposed energy. These three terms act in Eq. 11.11 as strong temporal smoothness constraints.

These three terms are illustrated in Fig. 11.19 and deal more precisely with:

- the matching cost $C(\boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}}, \boldsymbol{x}_m^* - \boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}})$ between $\boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}} \in I_n$ and $\boldsymbol{x}_m^*$ of I_m which encourages color consistency along the trajectory,

- the *euclidean* distance $ed_{m,n}$ between $\boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}}$ and the ending point of the elementary *optical flow* vector $\boldsymbol{u}_{m,n}$ starting from $\boldsymbol{x}_m^*$ (see Eq. 11.13). $ed_{m,n}$ encourages the selection of $\boldsymbol{x}_n^m$, the candidate coming from the neighbouring frame I_m via the elementary *optical flow* field $\boldsymbol{u}_{m,n}$ and therefore tends to strengthen the temporal motion smoothness. Indeed, for $\boldsymbol{x}_n^m$ (Eq. 11.9), the *euclidean* distance $ed_{m,n}$ is equal to 0.

$$ed_{m,n} = \left\| [\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref})] - [\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,m}^*(\boldsymbol{x}_{ref}) + \tilde{\boldsymbol{u}}_{m,n}(\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,m}^*(\boldsymbol{x}_{ref}))] \right\|_2 \quad (11.13)$$

- the *euclidean* distance $ed_{n,m}$ between $\boldsymbol{x}_m^*$ and the ending point of the *optical flow* vector $\boldsymbol{u}_{n,m}$ starting from $\boldsymbol{x}_n^{l_{\boldsymbol{x}_{ref}}}$ (see Eq. 11.14). If $\boldsymbol{u}_{m,n}$ is consistent, i.e. $\boldsymbol{u}_{m,n} \approx -\boldsymbol{u}_{n,m}$, $ed_{n,m}$ is close to 0 for $\boldsymbol{x}_n^m$, the candidate coming from I_m, whose selection is again promoted.

$$ed_{n,m} = \left\| [\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,m}^*(\boldsymbol{x}_{ref})] - [\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref}) + \tilde{\boldsymbol{u}}_{n,m}(\boldsymbol{x}_{ref} + \boldsymbol{d}_{ref,n}^{l_{\boldsymbol{x}_{ref}}}(\boldsymbol{x}_{ref}))] \right\|_2 \quad (11.14)$$

The spatial regularization term $E_{ref,n}^r$ involves motion similarities with neighbouring positions, as shown in Eq. 11.10. $\alpha_{\boldsymbol{x}_{ref},\boldsymbol{y}_{ref}}$ accounts for local color similarities in the reference frame I_{ref} where $\boldsymbol{y}_{ref}$ is in the neighbourhood of $\boldsymbol{x}_{ref}$. The robust functions ρ_d and ρ_r are respectively the negative log of a *Student-t* distribution and the *Geman-McClure* penalty function [LRR08].

Compared to the *from-the-reference* case, the energy for the refinement of *to-the-reference* displacement fields is similar except for the data term, Eq. 11.15, which involves through $T_{sc}(\boldsymbol{x}_n, \boldsymbol{d}_{n,ref}^{l_{\boldsymbol{x}_n}}(\boldsymbol{x}_n))$ neither the matching cost between the current candidate and the temporal neighboring one nor the *euclidean* distance $ed_{m,n}$. This is due to trajectories which can not be explicitly handled in this direction.

$$\begin{aligned} \epsilon_{n,ref}^d &= w_{MC}.C(\boldsymbol{x}_n, \boldsymbol{d}_{n,ref}^{l_{\boldsymbol{x}_n}}(\boldsymbol{x}_n)) + w_{Inc}.Inc(\boldsymbol{x}_n, \boldsymbol{d}_{n,ref}^{l_{\boldsymbol{x}_n}}(\boldsymbol{x}_n)) \\ &+ w_{T_{sc}}.T_{sc}(\boldsymbol{x}_n, \boldsymbol{d}_{n,ref}^{l_{\boldsymbol{x}_n}}(\boldsymbol{x}_n)) \end{aligned} \quad (11.15)$$

Nevertheless, as detailed in Eq. 11.16 and 11.17, the term corresponding to $ed_{m,n}$ in Eq. 11.13 remains. Thus, we compute the *euclidean* distance between the ending points of $\boldsymbol{d}_{n,ref}^*$ starting from $\boldsymbol{x}_n \in I_n$ and $\boldsymbol{u}_{n,m}$ concatenated with $\boldsymbol{d}_{m,ref}^*$:

$$T_{sc}(\boldsymbol{x}_n, \boldsymbol{d}_{n,ref}^{l_{\boldsymbol{x}_n}}(\boldsymbol{x}_n)) = \sum_{\substack{m=n-\frac{w}{2} \\ m \neq n}}^{n+\frac{w}{2}} ed_{m,n} \quad (11.16)$$

$$ed_{m,n} = \left\| [\boldsymbol{x}_n + \boldsymbol{d}_{n,ref}^{l_{\boldsymbol{x}_n}}(\boldsymbol{x}_n)] - [\boldsymbol{x}_n + \boldsymbol{u}_{n,m}(\boldsymbol{x}_n) + \tilde{\boldsymbol{d}}_{m,ref}^*(\boldsymbol{x}_n + \boldsymbol{u}_{n,m}(\boldsymbol{x}_n))] \right\|_2 \quad (11.17)$$

The global optimization method fuses the displacement fields by pairs and finally chooses to update or not the previous estimations with one of the previously described candidates. The motion refinement phase consists in applying this technique for each pair of frames $\{I_{ref}, I_n\}$ in *from-the-reference* and *to-the-reference* directions. The pairs $\{I_{ref}, I_n\}$ are processed in a random order in order to encourage temporal smoothness without introducing a sequential correlation between the resulting displacement fields.

This motion refinement phase is repeated iteratively N_{it} times where one iteration corresponds to the processing of all the pairs $\{I_{ref}, I_n\}$.

11.3.3 Overview on *StatFlow*

As shown in Fig. 11.16, the proposed *statistical multi-step flow* (*StatFlow*) long-term dense motion estimator has been fully applied once the motion candidates generation has processed each pair $\{I_{ref}, I_n\}$ through *CISS-K* (Section 11.1) and once the motion refinement of Section 11.2 has then processed N_{it} times all the pairs $\{I_{ref}, I_n\}$ in a random order. We finally obtain *from-the-reference* and *to-the-reference* long-term dense displacement fields across the whole sequence.

11.4 Experimental results for long video shots

Our experiments focus on the following sequences: *MPI-S*1-*25-55* [GKT+] (Fig. 11.5), *MPI-S*1-115-175 [GKT+] (Fig. 11.20), *Hope* (Fig. 11.6), *Newspaper*-2 (Fig. 11.7) and *Walking-Couple*-0-60 (Fig. 11.8). The additional *MPI-S*1-*115-175* sequence deals with non-rigid deformations, large un-textured areas and illumination changes.

For the experiments, the proposed *statistical multi-step flow* method (referred to as *StatFlow*) has been performed with the following parameters. The number of randomly selected motion *paths* is $N_s = 100$. The percentage of bad candidates to be remove corresponds to $R_{\%} = 50\%$. The motion candidates generation computes $K = 3$ *from-the-reference* and *to-the-reference* dense motion correspondences for each pair of frames $\{I_{ref}, I_n\}$. In addition, the inconsistency threshold, ϵ_{Inc}, equals to 1. In Eq. 11.8, $\alpha_0 = 3$, $\alpha_1 = 15$. The weights involved in Eq. 11.11 and 11.15 are defined as follows: $w_{MC} = w_{Inc} = 0.25$ and $w_{T_{sc}} = 0.5$. Finally, the width of the temporal window W centered around each frame I_n is $w = 5$. The set of *steps* and input *optical flow* estimators will be specified for each experiment and each sequence.

Experiments have been conducted as follows. In Section 11.4.1, we evaluate the performance of our extended version of the combinatorial integration and the statistical selection (*CISS*) initially introduced in Section 11.1 through registration and *PSNR* assessment. The effects of the iterative motion refinement stage described in Section 11.3.2 are also studied. Then, we compare the proposed *StatFlow* algorithm to state-of-the-art methods through both qualitative assessment via point tracking (Section 11.4.2) and texture insertion and propagation (Section 11.4.3) and quantitative assessment using the *Flag* benchmark dataset provided in [GRA11a, GRA11b] (Section 11.4.4) and the *Hopkins* groundtruth sparse trajectories given in [TV07] (Section 11.4.5). Finally, Section 11.4.6 focuses on block matching estimations as inputs of our *StatFlow* method to show that any type of *optical flow* estimators can be considered to compute long-term dense displacement fields.

In terms of computation time, we performed the *StatFlow* framework on a sequence with 60 frames of 500×500 pixels. The initial phase (*CISS-K*, Section 11.3.1) takes around 5 minutes per pair of frames. The iterative refinement stage (Section 11.3.2) takes approximately 15 seconds per pair of frames and per iteration.

11.4.1 Long-term warping

The first experiment consists in showing how the improvements we made with respect to the combinatorial integration and the statistical selection introduced in Section 11.1 impact the quality of the final displacement fields. For this task, we focus on pairs of frames taken from the

Frame pairs	{25,45}	{25,46}	{25,47}	{25,48}	{25,49}	{25,50}	{25,51}	{25,52}
CISS (Section 11.1)	21.83	24.98	25.56	25.83	25.04	24.83	24.48	24.3
CISS-K (Section 11.3.1)	**29.02**	**28.4**	**27.27**	**27.23**	**26.84**	**26.33**	**26.1**	**25.69**

Table 11.3: Registration and *PSNR* assessment with the combinatorial integration and the statistical selection (*CISS*) introduced in Section 11.1 and the proposed extended version (*CISS-K*, i.e. the initial phase of *StatFlow*) described in Section 11.3.1. *PSNR* scores are computed on the kiosk of the *MPI-S1-25-55* sequence.

(a) I_{115} (b) I_{125} (c) I_{135} (d) I_{145}

(e) I_{155} (f) I_{165} (g) I_{170} (h) I_{175}

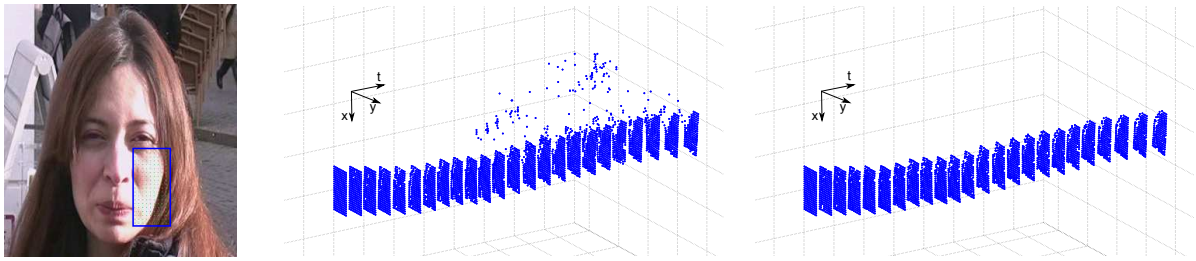
Figure 11.20: Source frames from a cropped version of the *MPI-S*1*-115-175* sequence.

(a) I_{25} (b) I_{45} (c) I_{25} (d) *CISS* (e) *StatFlow* init.

Figure 11.21: Source frames of the *MPI-S1-25-55* sequence [GKT+] and reconstruction of the kiosk of I_{25} from I_{45} with: e) the combinatorial integration and the statistical selection (*CISS*) introduced in Section 11.1, f) the proposed extended version (initial phase of *StatFlow*) described in Section 11.3.1. Black boxes focus on differences between both methods.

Frame pairs	{160,180}	{160,190}	{160,200}	{160,210}	{160,220}	{160,230}
CISS (Section 11.1)	22.50	21.21	18.59	17.12	15.87	15.76
CISS-K (Section 11.3.1)	22.70	21.39	19.28	18.21	17.12	16.58
StatFlow, i.e. *CISS-K+IMR* (Section 11.3)	**22.93**	**22.18**	**20.25**	**18.68**	**17.40**	**16.81**

Table 11.4: Registration and *PSNR* assessment with: 1) combinatorial integration and statistical selection (*CISS*) introduced in Section 11.1, 2) proposed extended version (*CISS-K*) described in Section 11.3.1, 3) whole *StatFlow* framework (Section 11.3). *PSNR* scores are computed on whole images of the *Newspaper*-2 sequence.



(a) I_{115} with tracking area

(b) Point tracking from I_{115} to I_{138} *MSF* (2*D-DE*)

(c) Point tracking from I_{115} to I_{138} *StatFlow* (2*D-DE*)

Figure 11.22: Point tracking from I_{115} up to I_{138}, *MPI-S*1-115-175 sequence [GKT+] (Fig. 11.20). We compare: b) *multi-step* flow fusion (*MSF*) described in Section 11.2 using 2*D-DE* *multi-step* elementary optical flow fields: *MSF*(2*D-DE*); c) the proposed *statistical multi-step flow* (*StatFlow*) method proposed in Section 11.3 using 2*D-DE* *multi-step* elementary optical flow fields: *StatFlow*(2*D-DE*).

*MPI-S*1-25-55 and *Newspaper*-2 sequences. The sets of *steps* are respectively 1, 2, 3, 4, 5, 15 and 1, 2, 3, 4, 5, 10, 20, 30. The tested algorithms have been performed taking 2*D-DE* *multi-step* elementary *optical flow* fields as inputs.

We compare the optimal displacement fields obtained in output of our motion estimates generation (*CISS-K*, described in Section 11.3.1) with those resulting from *CISS* (Section 11.1). The comparison is done through registration and *PSNR* assessment. For a given pair $\{I_{ref}, I_n\}$, the final fields are used to reconstruct I_{ref} from I_n through motion compensation and color *PSNR* scores are computed between I_{ref} and the registered frame for non-occluded pixels.

Tables 11.3 and 11.4 show the *PSNR* scores for various distances between I_{ref} and I_n respectively on the kiosk of *MPI-S*1-25-55 (Fig. 11.21) and on whole images of *Newspaper*-2. Results on *MPI-S*1-25-55 show that *CISS-K* outperforms *CISS* for all pairs ($29.02dB$ instead of $21.83dB$ for $\{I_{25}, I_{45}\}$). An example of registration of the kiosk for a distance of 20 frames is given Fig. 11.21. It appears that *CISS-K* allows a better reconstruction (Fig. 11.21 (e)) than *CISS* (Fig. 11.21 (d)), especially within the black rectangle. As previously, we notice that *multi-step* estimations deal satisfactorily with the temporary occlusion. Experiments on *Newspaper*-2 (Tab. 11.4) reveal the same finding: the novelty in terms of inconsistency reduction improves the displacement fields quality. Moreover, the iterative motion refinement (*IMR*) stage (the number of iterations is $N_{it} = 9$ in this case) allows to obtain better *PSNR* scores for all pairs compared to *CISS-K*, the initial stage of *StatFlow* ($20.25dB$ instead of $19.28dB$ for $\{I_{160}, I_{200}\}$).

11.4.2 Point tracking

StatFlow and *MSF* have been assessed through point tracking (using 2*D-DE* with *steps* 1, 2, 5, 10 and 15) within the *MPI-S*1-115-175 sequence (Fig. 11.20). In Fig. 11.22, the bottom right part of the face of the woman in *MPI S*1 is tracked from I_{115} to I_{138}. The $2D+t$ trajectory visualization indicates that some trajectories drift to another part of the sequence with *MSF*(2*D-DE*). This illustrates the inherent issue of *MSF* which sometimes propagates estimation errors due to the sequential processing. On the contrary, *StatFlow*(2*D-DE*) ($N_{it} = 9$) provides accurate displacement fields while limiting the temporal correlation between displacement fields respectively estimated for neighbouring frames.

(a) Original image I_{115} (c) Propagation to I_{125} *MSF* (2*D-DE*) (d) Propagation to I_{130} *MSF* (2*D-DE*) (e) Propagation to I_{137} *MSF* (2*D-DE*)

(b) Texture insertion in I_{115} (f) Propagation to I_{125} *StatFlow* (2*D-DE*) (g) Propagation to I_{130} *StatFlow* (2*D-DE*) (h) Propagation to I_{137} *StatFlow* (2*D-DE*)

Figure 11.23: Texture insertion in I_{115} and propagation along the *MPI-S*1 sequence [GKT+] up to I_{137}. We compare: c-e) *multi-step* flow fusion (Section 11.2) using 2*D-DE multi-step* elementary *optical flow* fields: *MSF*(2*D-DE*); f-h) the *statistical multi-step flow* method (Section 11.3) using 2*D-DE multi-step* elementary *optical flow* fields: *StatFlow*(2*D-DE*).

11.4.3 Video editing

In this part, we aim at showing that our method provides satisfying results in a wide set of complex scenes. Moreover, we focus on the comparison between the proposed *StatFlow* ($N_{it} = 9$) and *MSF* to prove that *StatFlow* can perform a more efficient integration and selection procedure compared to *MSF* using the same *multi-step* elementary *optical flows* as inputs. For this sake, experiments have been conducted in the context of video editing. More precisely, we evaluate the accuracy of both methods by considering textures or logos manually inserted in the reference frame I_{ref} and by motion compensating them into I_n.

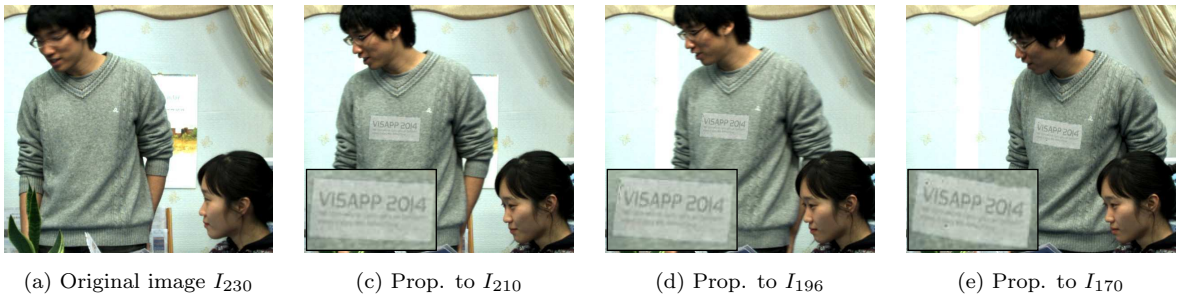
In Fig. 11.23 and 11.26, textures have been respectively inserted in I_{115} of *MPI-S*1*-115-175* and I_0 of *Walking-Couple-0-60*. *To-the-reference* displacement fields computed with *StatFlow (2D-DE)* and *MSF (2D-DE)* serve to propagate the textures up to respectively I_{137} and I_{40}. 2*D-DE* has been chosen for its good results for these video editing tasks. The logo propagation examples, Fig. 11.24 and 11.25, deal respectively with *Hope* from I_{5036} up to I_{5063} and *Newspaper*-2 from I_{230} up to I_{170}. The *steps* involved in these experiments are $1-5$, 10, 15, 30 for *MPI-S*1*-115-175*, $1-5$, 10, 15 for *Walking-Couple-0-60*, $1-5$, 8, 10, 15, 20 for *Hope* and $1-5$, 10, 20, 30 for *Newspaper*-2.

In these results, we notice that *MSF* sometimes distorts structures (bottom left zoom in Fig.11.23 (*c-e*), Fig.11.24 (*d,e*), Fig.11.25 (*d*)), makes shadow textures appear (bottom right zoom in Fig.11.23 (*c-e*)) and does not estimate motion with high accuracy (top right zoom in Fig.11.23 (*e*)). Visual results with *StatFlow* reveal a better long-term propagation. Fig. 11.26

(a) Original image I_{5036} (c) Propagation to I_{5046} *MSF* (2*D-DE*) (d) Propagation to I_{5052} *MSF* (2*D-DE*) (e) Propagation to I_{5063} *MSF* (2*D-DE*)

(b) Logo insertion in I_{5036} (f) Propagation to I_{5046} *StatFlow* (2*D-DE*) (g) Propagation to I_{5052} *StatFlow* (2*D-DE*) (h) Propagation to I_{5063} *StatFlow* (2*D-DE*)

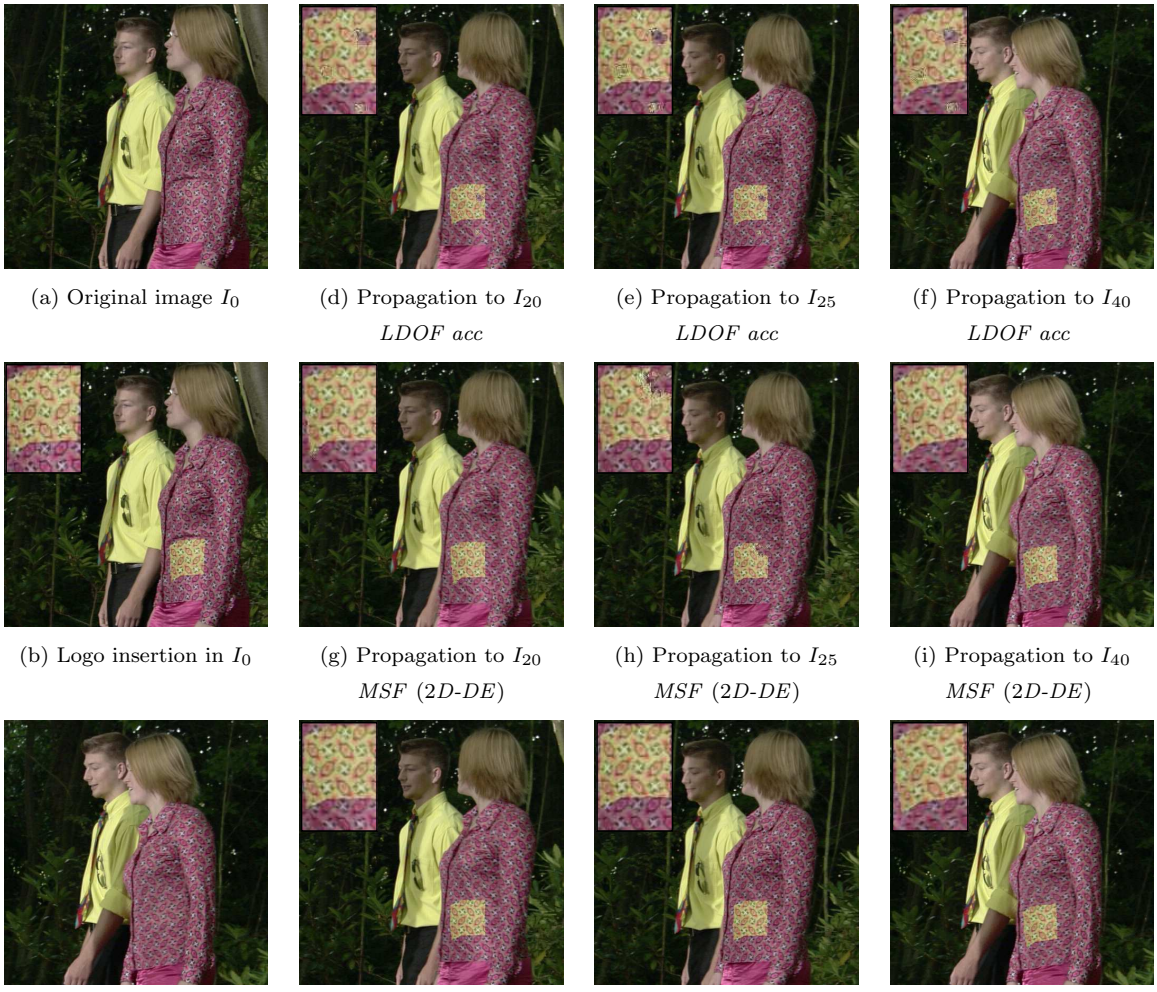
Figure 11.24: Logo insertion in I_{5036} and propagation along the *Hope* sequence up to I_{5063}. We compare: c-e) *multi-step* flow fusion (Section 11.2) using 2*D-DE multi-step* elementary *optical flow* fields: *MSF*(2*D-DE*); f-h) the *statistical multi-step flow* method (Section 11.3) using 2*D-DE multi-step* elementary *optical flow* fields: *StatFlow*(2*D-DE*).



(a) Original image I_{230} (c) Prop. to I_{210} *MSF* (2*D-DE*) (d) Prop. to I_{196} *MSF* (2*D-DE*) (e) Prop. to I_{170} *MSF* (2*D-DE*)

(b) Logo insertion in I_{230} (f) Prop. to I_{210} *StatFlow* (2*D-DE*) (g) Prop. to I_{196} *StatFlow* (2*D-DE*) (h) Prop. to I_{170} *StatFlow* (2*D-DE*)

Figure 11.25: Logo insertion in I_{230} and propagation along the *Newspaper*-2 sequence up to I_{170}. We compare: c-e) *multi-step* flow fusion (Section 11.2) using 2*D-DE multi-step* elementary *optical flow* fields: *MSF*(2*D-DE*); f-h) the *statistical multi-step flow* method (Section 11.3) using 2*D-DE multi-step* elementary *optical flow* fields: *StatFlow*(2*D-DE*).



(a) Original image I_0 (d) Propagation to I_{20} *LDOF acc* (e) Propagation to I_{25} *LDOF acc* (f) Propagation to I_{40} *LDOF acc*

(b) Logo insertion in I_0 (g) Propagation to I_{20} *MSF* (2*D-DE*) (h) Propagation to I_{25} *MSF* (2*D-DE*) (i) Propagation to I_{40} *MSF* (2*D-DE*)

(c) Original image I_{40} (j) Propagation to I_{20} *StatFlow* (2*D-DE*) (k) Propagation to I_{25} *StatFlow* (2*D-DE*) (l) Propagation to I_{40} *StatFlow* (2*D-DE*)

Figure 11.26: Texture insertion in I_0 and propagation up to I_{40} (*Walking-Couple-0-60* sequence). We compare: d-f) *Euler* integration using *LDOF* [BM11] elementary *optical flow* fields computed between consecutive frames: *LFOF acc*; g-i) *multi-step* flow fusion (Section 11.2) using 2*D-DE multi-step* elementary *optical flow* fields: *MSF*(2*D-DE*); j-l) the *statistical multi-step flow* (Section 11.3) using 2*D-DE multi-step* elementary *optical flow* fields: *StatFlow*(2*D-DE*).

compares *StatFlow(2D-DE)* and *MSF(2D-DE)* with a third method: the *Euler* integration of *LDOF* elementary *optical flows* computed between consecutive frames (*LDOF acc*). We observe that *LDOF acc* badly performs motion estimation for periodic structures. *MSF* also encounters matching issues (I_{25}) whereas *StatFlow* performs propagation without any visible artifacts.

11.4.4 Quantitative results using the *Flag* benchmark dataset

Quantitative results have been obtained using the dense ground-truth *optical flow* data provided by the *Flag* benchmark dataset [GRA11a]. This dataset is based on sparse motion capture data estimated on a flag waving in the wind. Sparse estimates have been interpolated to create a dense 3D surface which has been then projected into the image plane to provide dense ground-truth *optical flow* data. The original version of the resulting *Flag* sequence, displayed in Fig. 11.27, has been used to test *StatFlow* and state-of-the-art methods. Experiments more exactly focus on:

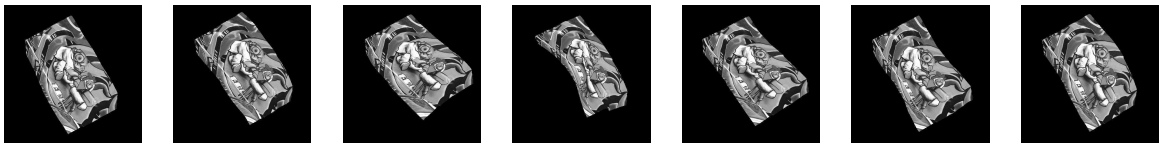
- Direct motion estimation between each pair $\{I_{ref}, I_n\}$ using the following estimators:
  - *LDOF* [BM11]: *LDOF direct*,
  - *ITV-L1* [WPZ+09], an improved duality based *TV-L1 optical flow*: *ITV-L1 direct*,
  - the keypoint-based non-rigid registration algorithm of [PB12]: [PB12] *direct*.
- Classical *Euler* integration, i.e. concatenation of *optical flows* computed between consecutive frames using *LDOF*: *LDOF acc*
- The *multi-frame subspace flow* (*MFSF*) algorithm proposed in [GRA11b] and its extended version detailed in [GRA13] using the PCA (*principal component analysis*) or the DCT (*discrete cosine transform*) motion basis: *MFSF-PCA* or *MFSF-DCT*
- The *multi-step* flow fusion (*MSF*) method using *LDOF multi-step* elementary *optical flow* fields as inputs: *MSF* (*LDOF*)
- The proposed *statistical multi-step flow* (*StatFlow*) method (with $N_{it} = 3$) using *LDOF multi-step* elementary *optical flow* fields: *StatFlow* (*LDOF*).

For the comparison task, Tab. 11.5 gives the *RMS* (*root mean square*) endpoint errors between the respective obtained displacement fields and the ground-truth data for all the previously described methods. The *RMS* errors are estimated for all the foreground pixels and for all the pairs of frames $\{I_{ref}, I_n\}$ together. For each $\boldsymbol{x}_{ref}$ of I_{ref}, let $\boldsymbol{x}_n^{GT}$ and $\boldsymbol{x}_n$ be respectively the corresponding positions in I_n obtained via the groundtruth displacement field $\boldsymbol{d}_{ref,n}^{GT}$ and the estimated one $\boldsymbol{d}_{ref,n}$ such as:

$$\left\{ \begin{array}{l} \boldsymbol{x}_n = \boldsymbol{x}_{ref} + \boldsymbol{d}_n \\ \boldsymbol{x}_n^{GT} = \boldsymbol{x}_{ref} + \boldsymbol{d}_n^{GT} \end{array} \right. \tag{11.18}$$

The *RMS* error is computed in Eq. 11.19 where $N+1$, L and W are respectively the number of frames in the sequence, the height and the width of the frames.

$$RMS = \sqrt{\frac{1}{(N+1).L.W} \sum_{\substack{n=0 \\ n \neq ref}}^{N} \sum_{\boldsymbol{x}_{ref} \in I_{ref}} ||\boldsymbol{x}_n - \boldsymbol{x}_n^{GT}||_2} \tag{11.19}$$



(a) I_1 (b) I_{10} (c) I_{20} (d) I_{30} (e) I_{40} (f) I_{50} (g) I_{60}

Figure 11.27: Source frames of the *Flag* sequence [GRA11a].

Method	RMS endpoint error (pixels)
***StatFlow* (*LDOF*)** (Section 11.3)	**0.69**
MSF (*LDOF*) (Section 10.2)	1.41
LDOF direct [BM11]	1.74
LDOF acc [BM11]	4
MFSF-PCA [GRA13]	**0.69**
MFSF-DCT [GRA13]	0.80
MFSF-PCA [GRA11b]	0.98
MFSF-DCT [GRA11b]	1.06
[PB12] *direct*	1.24
ITV-L1 direct [WPZ+09]	1.43

Table 11.5: RMS endpoint errors for different methods on the *Flag* benchmark dataset provided in [GRA11a].

RMS endpoint errors computed for each pair of frames are also shown in Fig.11.28 for all the methods based on *LDOF*: *LDOF direct*, *LDOF acc*, *MSF* (*LDOF*) and *StatFlow* (*LDOF*). Note that the last two *multi-step* strategies have considered as inputs the steps 1, 2, 3, 4, 5, 8, 10, 15, 20, 25, 30, 40 and 50.

We can firstly observe that *LDOF acc*, the method based on *Euler* integration of consecutive *LDOF optical flows* rapidly diverge. This is mainly due to both estimation errors which are propagated along trajectories and accumulation errors which are inherent to the interpolation process. In addition, the results obtained through direct motion estimation are reasonably good, especially for [PB12]. *LDOF direct* gives a lower *RMS* endpoint error than the consecutive accumulations of *LDOF acc* (1.74 against 4). However, it is not possible to draw conclusions in the light of the *Flag* sequence because the flag comes back approximately to its initial position at the end of the sequence (Fig.11.27 (*a*,*g*)). Motion estimation for complex scenes cannot generally rely only on direct estimation and combining *optical flow* accumulations and direct matching is clearly a more suitable strategy.

Tab. 11.5 and Fig. 11.28 also show that with the same *multi-step* elementary *optical flow* fields as inputs (*LDOF* in this case), the proposed *StatFlow* method shows a clear improvement compared to the *multi-step* flow fusion (*MSF*) approach (0.69 against 1.41). Although both methods achieve the same quality for first pairs or for some pairs which coincide with existing *steps*, other displacement fields are computed with a better accuracy using *StatFlow*. Moreover, our *StatFlow* (*LDOF*) approach reaches the same *RMS* endpoint error with respect to *MFSF-PCA* [GRA13], the best one of the *MFSF* approaches, with 0.69. This proves that *StatFlow* is competitive compared to challenging state-of-the-art methods.

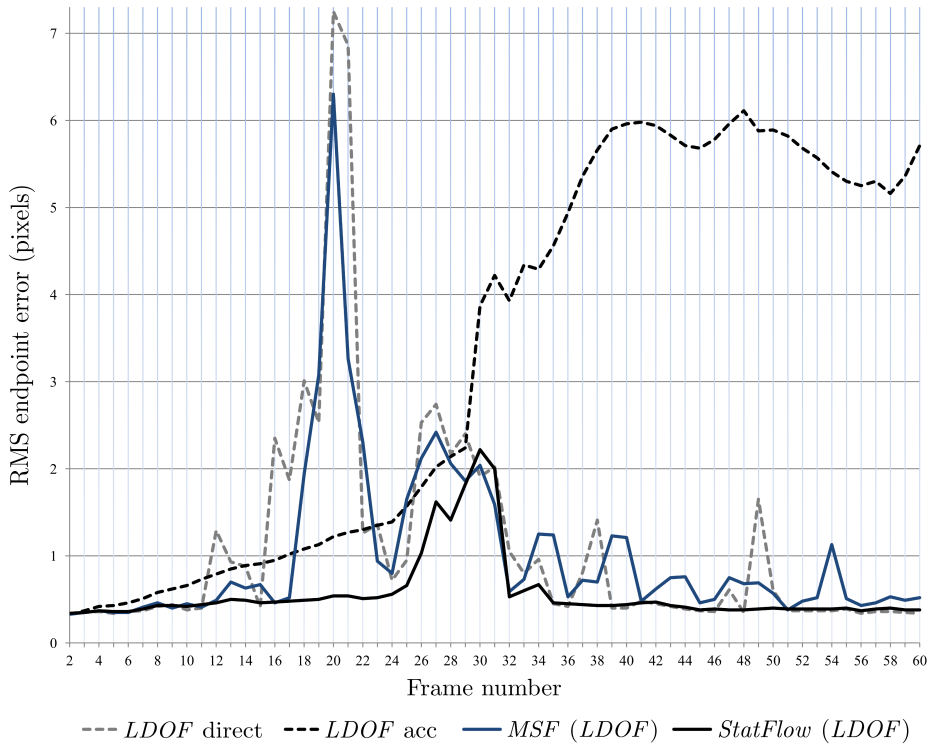


Figure 11.28: *RMS* endpoint errors for each pair $\{I_{ref}, I_n\}$ along *Flag* sequence [GRA11a] with different methods: *LDOF direct*, *LDOF acc*, *MSF*$(LDOF)$ and *StatFlow*$(LDOF)$.

11.4.5 Quantitative results using *Hopkins* ground-truth data

The *Hopkins*-155 dataset [TV07] has been originally proposed as a benchmark for testing motion-based segmentation algorithms. It provides several real-world scenes together with valuable information on point trajectories, which makes it of interest for our purpose. In particular, the dataset provides ground-truth trajectories starting from a sparse set of pixels in the first image. All these pixels are always visible along the image sequence. Among all the proposed 155 video sequences, we distinguished 6 sequences among the longest:

- *Hopkins-people1* where a woman is walking,
- *Hopkins-cars4* which deals with a car turning at a crossing,
- *Hopkins-two-cranes*, a sequence of two bulldozers moving independently (one of the two bulldozers moves articulately with its arm rotating),
- *Hopkins-cars9* where a car and a van are starting from a traffic light,
- *Hopkins-head*, a sequence of a person moving with his head rotating around the neck,
- *Hopkins-truck2* which shows a truck at a crossing.

These sequences are shown in Fig. 11.29 with additional groundtruth masks which indicate the positions of the groundtruth trajectory starting points in the first frame. Sequence length is between 22 and 61 frames, and the number of tracked points in each sequence is between 94 and

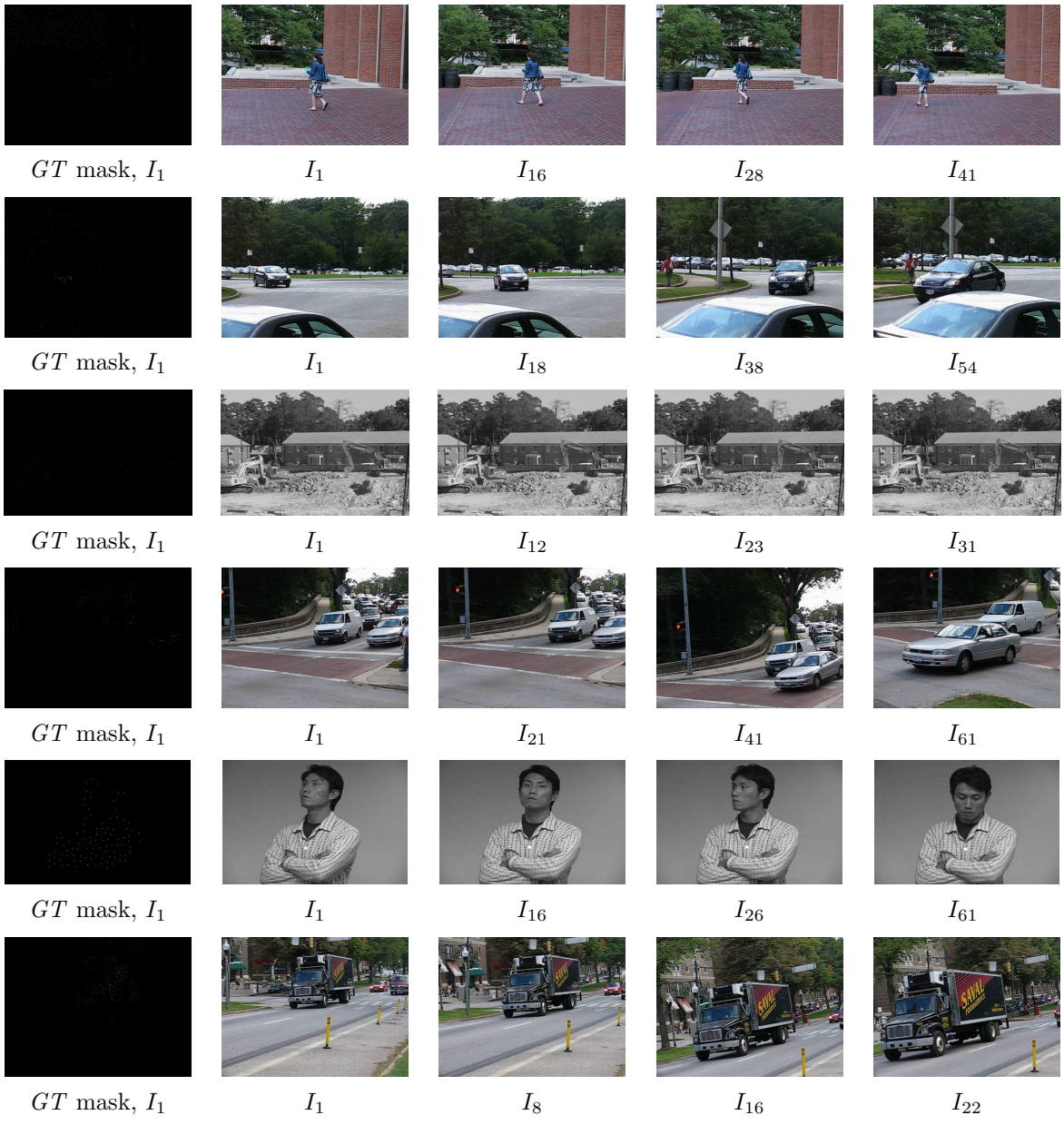


Figure 11.29: Six video sequences taken from the *Hopkins*-155 dataset [TV07] with associated groundtruth (GT) mask in I_1 which indicates the starting point of the sparse *groundtruth* trajectories. From top to bottom: *Hopkins-people1*, *Hopkins-cars4*, *Hopkins-two-cranes*, *Hopkins-cars9*, *Hopkins-head* and *Hopkins-truck2*.

Sequences	Sequence length	Number of GT trajectories
Hopkins-people1	41	504
Hopkins-cars4	54	147
Hopkins-two-cranes	30	94
Hopkins-cars9	61	220
Hopkins-head	60	99
Hopkins-truck2	22	331

Table 11.6: Characteristics of the 6 selected sequences from the *Hopkins*-155 dataset [TV07] in terms of sequence length and number of groundtruth (GT) trajectories.

504. Tab. 11.6 specifies the characteristics of each sequence in terms of sequence length and number of groundtruth trajectories.

For each sequence, our experiments focus on the comparison between the corresponding sparse groundtruth trajectories and the trajectories starting from the same input points in the first frame (therefore considered as the reference frame I_{ref}) and obtained using the *from-the-reference* displacement fields computed via:

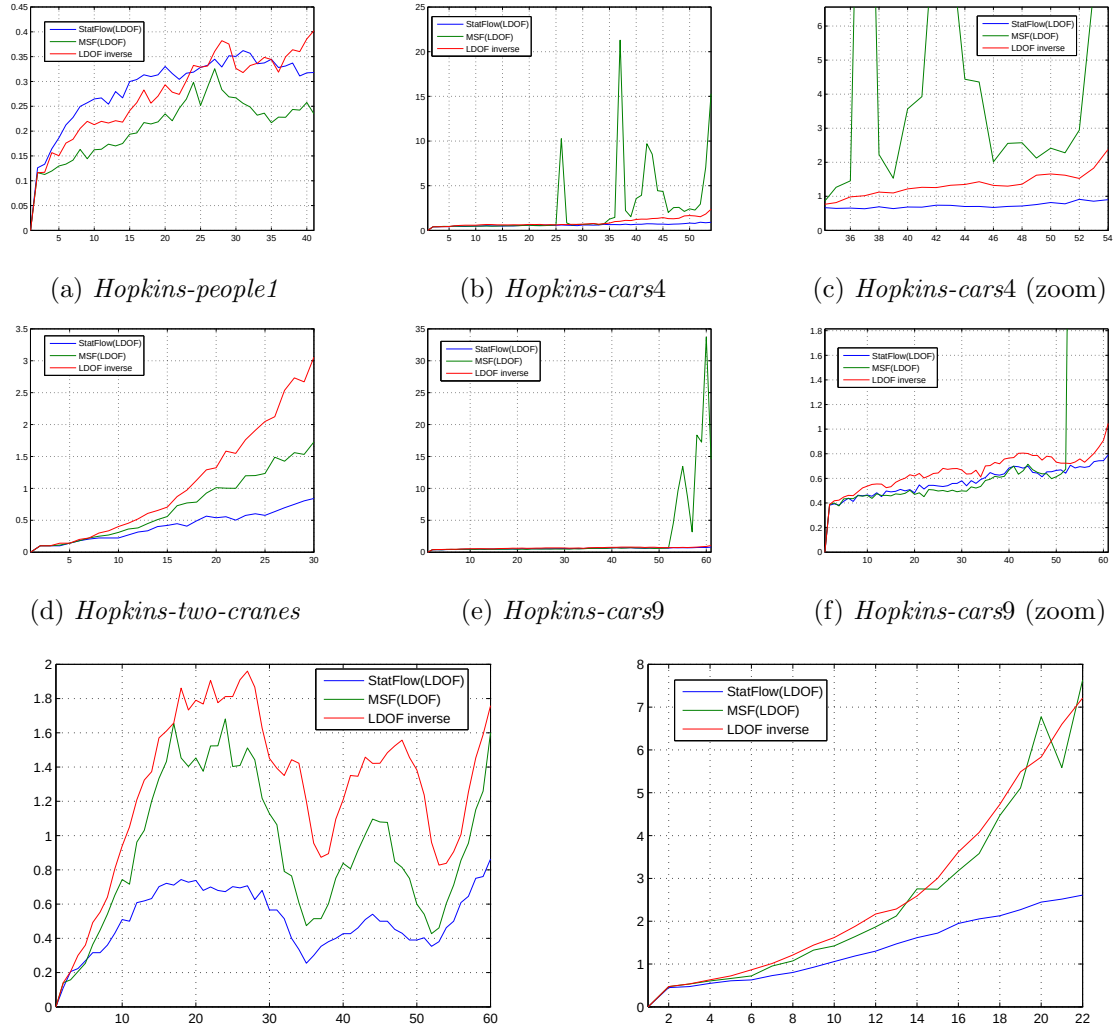
- *LDOF inverse*: inverse integration of *optical flows* computed between consecutive frames using *LDOF*,
- *MSF*(*LDOF*): the *multi-step* flow fusion (*MSF*) method using *LDOF multi-step* elementary *optical flow* fields as inputs,
- *StatFlow*(*LDOF*): the proposed *statistical multi-step flow* (*StatFlow*) method (with $N_{it} = 9$) using *LDOF multi-step* elementary *optical flow* fields.

The same *multi-step* elementary *optical flows* have been given as inputs of *MSF* and *StatFlow*. These algorithms have been performed with the following set of *steps*: 1, 2, 3, 4, 5, 10, 20 and 40. This last *step* has been used only for the *Hopkins-two-cranes* and *Hopkins-truck2* sequences.

For each $\boldsymbol{x}_{ref}$ of I_{ref}, $\boldsymbol{x}_n^{GT}$ and $\boldsymbol{x}_n$ are respectively the corresponding groundtruth and estimated positions in I_n. The comparisons between groundtruth and estimated trajectories involve two error measures. In Fig. 11.30, we focus on the position *median absolute error* (*MedE*) described in Eq. 11.20. In Fig. 11.31, we take into account the percentage of points $\boldsymbol{x}_{ref}$ whose location $\boldsymbol{x}_n$ in I_n is distant of maximum 1 pixel with respect to $\boldsymbol{x}_n^{GT}$.

$$MedE(n) = med_{\boldsymbol{x}_{ref} \in I_{ref}}(||\boldsymbol{x}_n - \boldsymbol{x}_n^{GT}||_2) \tag{11.20}$$

The plots in Fig. 11.30 indicate that in terms of *MedE*, *StatFlow* outperforms *LDOF inverse* and *MSF* for all the sequences except for *Hopkins-people1* where *MSF* results in a lower error. After 60 frames, the *MedE* for the *Hopkins-head* sequence is 0.86 pixel for *StatFlow* whereas it is around twice as much for *MSF* (1.6 pixel) and *LDOF inverse* (1.76 pixel). According to the results, the *MSF* method is generally better than *LDOF inverse* except for *Hopkins-cars4* and *Hopkins-cars9* for which we can notice very strong errors promptly. *MSF* strongly diverges during the tracking in the *Hopkins-cars9* sequence from I_{53}. In *Hopkins-cars4*, *MSF* oscillates between accurately estimated positions and motion *outliers*.

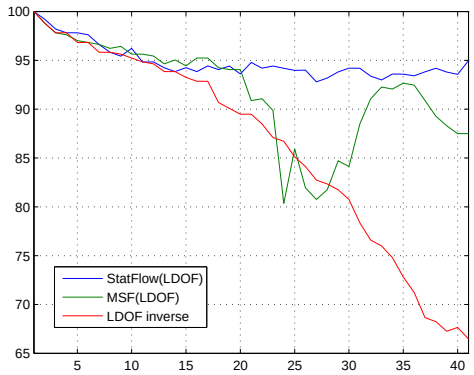


(a) *Hopkins-people1* (b) *Hopkins-cars4* (c) *Hopkins-cars4* (zoom)

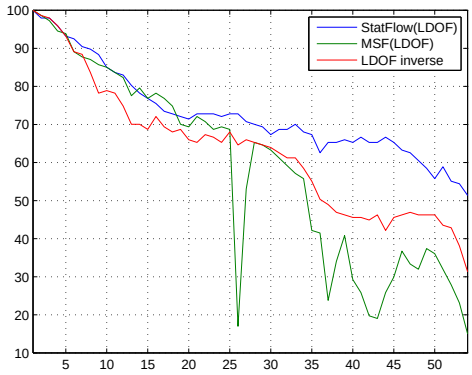
(d) *Hopkins-two-cranes* (e) *Hopkins-cars9* (f) *Hopkins-cars9* (zoom)

(g) *Hopkins-head* (h) *Hopkins-truck2*

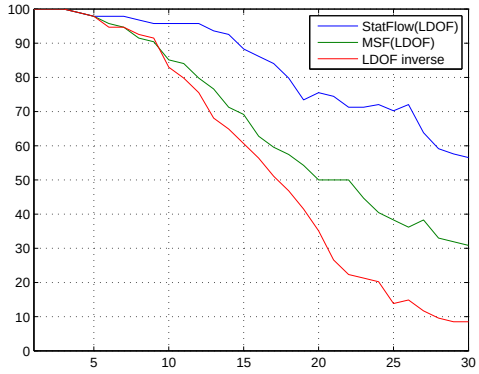
Figure 11.30: Position *median absolute error* (*MedE*) along six *Hopkins* sequences between the groundtruth trajectories and the trajectories obtained with the following methods: inverse integration of consecutive *optical flow* estimated with *LDOF* [BM11]: *LDOF inverse*; the *multi-step* flow fusion (*MSF*) described in Section 11.2 using *LDOF multi-step* elementary *optical flow* fields: *MSF*(*LDOF*); the *statistical multi-step flow* (*StatFlow*) method proposed in Section 11.3 using *LDOF multi-step* elementary *optical flow* fields: *StatFlow*(*LDOF*).



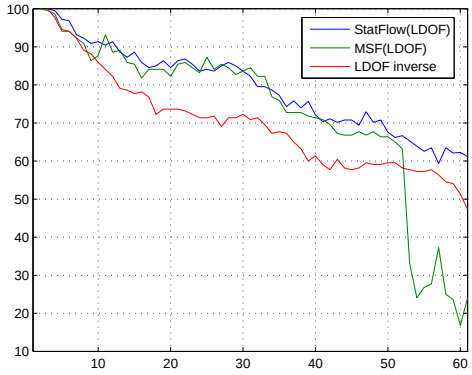
(a) *Hopkins-people1*



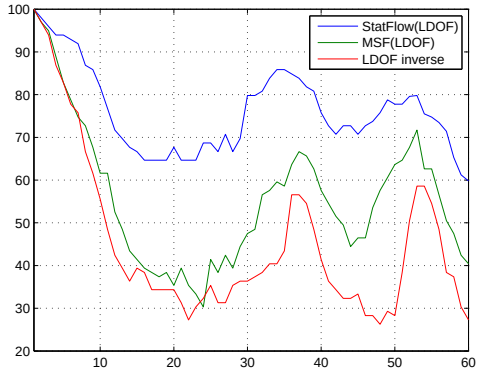
(b) *Hopkins-cars4*



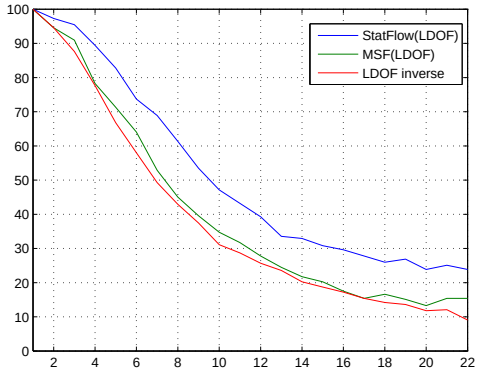
(c) *Hopkins-two-cranes*



(d) *Hopkins-cars9*



(e) *Hopkins-head*



(f) *Hopkins-truck2*

Figure 11.31: Proportion of points whose location is distant of maximum 1 pixel with respect to the corresponding groundtruth position. Six *Hopkins* sequences are involved. We compare: inverse integration of consecutive *optical flow* estimated with *LDOF* [BM11]: *LDOF inverse*; the *multi-step* flow fusion (*MSF*) described in Section 11.2 using *LDOF multi-step* elementary *optical flow* fields: *MSF*(*LDOF*); the *statistical multi-step flow* (*StatFlow*) method proposed in Section 11.3 using *LDOF multi-step* elementary *optical flow* fields: *StatFlow*(*LDOF*).

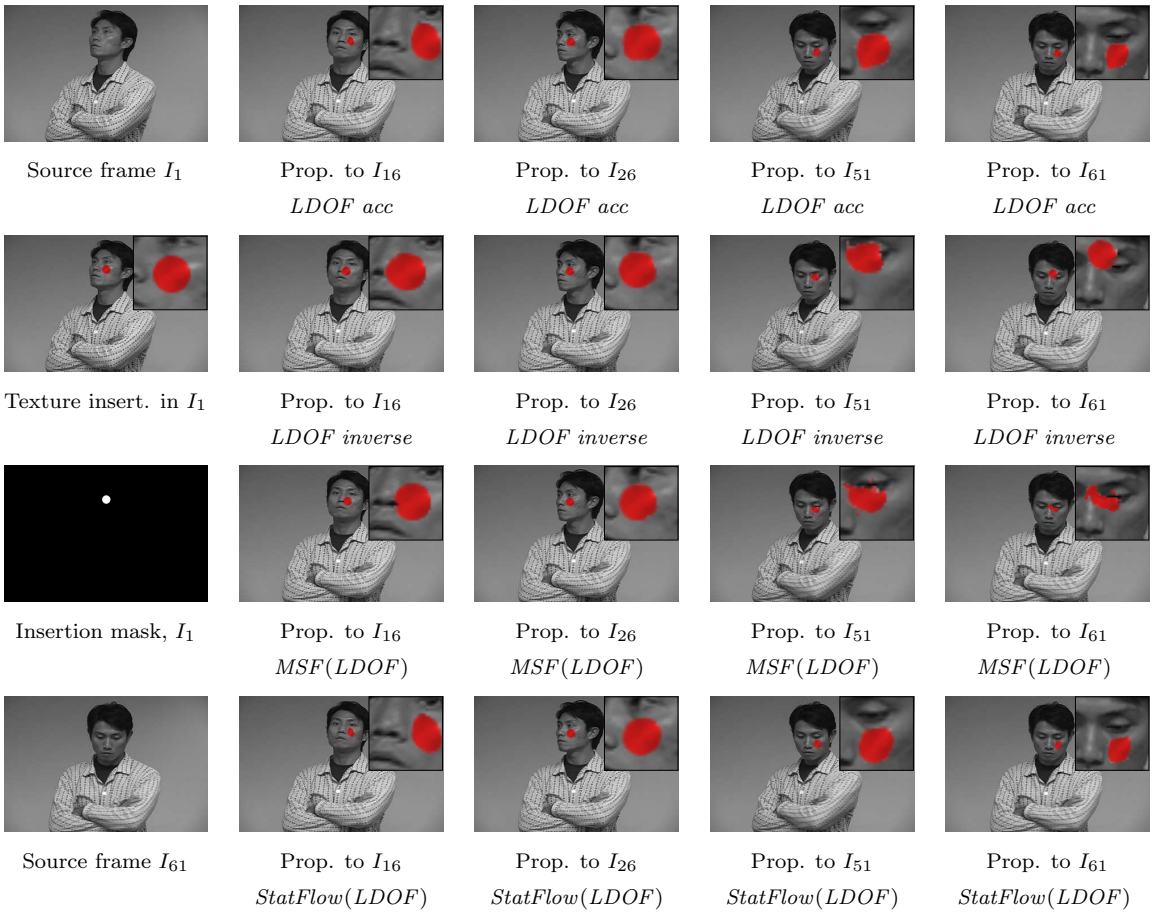


Figure 11.32: Texture insertion in I_1 and propagation up to I_{61} (*Hopkins-head* sequence). We compare: *Euler* integration using *LDOF* [BM11] consecutive elementary *optical flow* fields: *LDOF acc*; inverse integration of *LDOF* consecutive elementary *optical flow* fields: *LDOF inverse*; *multi-step* flow fusion (*MSF*) described in Section 11.2 using *LDOF multi-step* elementary optical flow fields: *MSF*(*LDOF*); *statistical multi-step flow* (*StatFlow*) proposed in Section 11.3 using *LDOF multi-step* elementary *optical flow* fields: *StatFlow*(*LDOF*) .

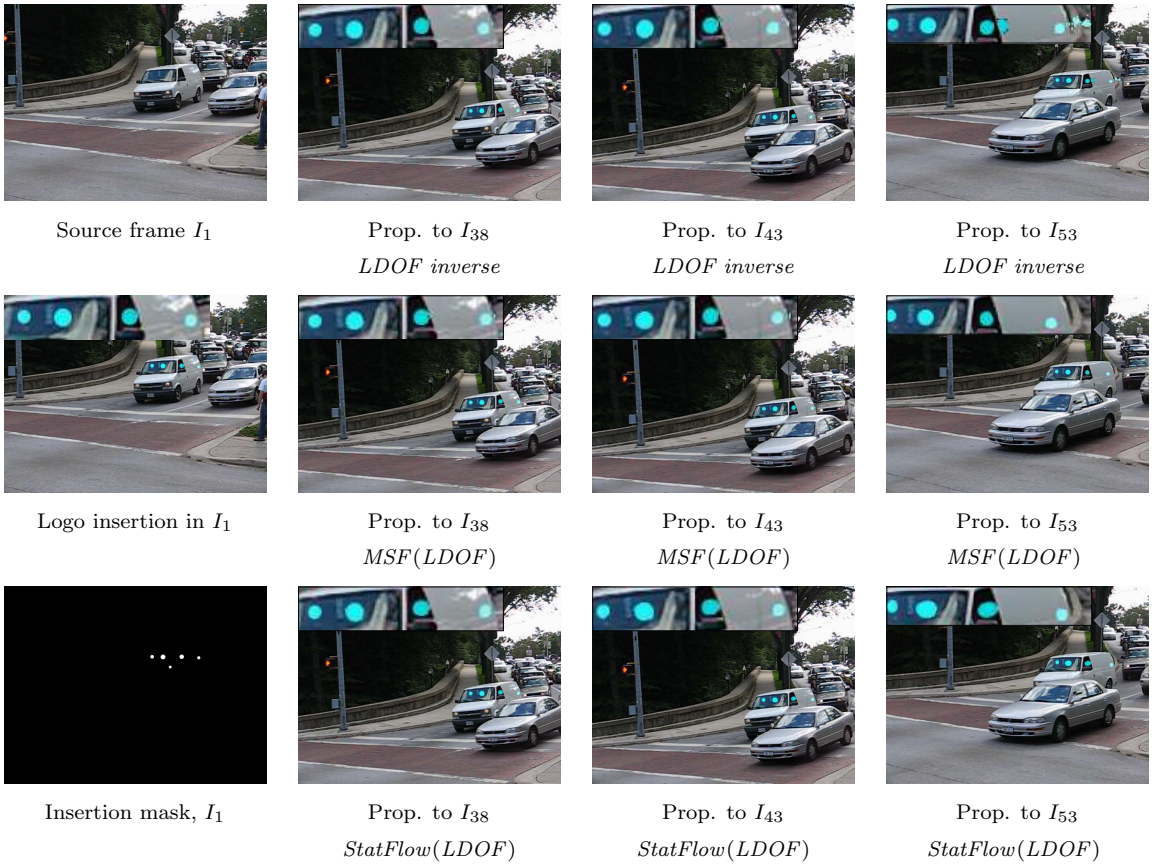


Figure 11.33: Texture insertion in I_1 and propagation up to I_{53} (*Hopkins-cars9* sequence). We compare: inverse integration of *LDOF* consecutive elementary *optical flow* fields: *LDOF inverse*; *multi-step* flow fusion (*MSF*) described in Section 11.2 using *LDOF multi-step* elementary optical flow fields: *MSF*(*LDOF*); *statistical multi-step flow* (*StatFlow*) proposed in Section 11.3 using *LDOF multi-step* elementary *optical flow* fields: *StatFlow*(*LDOF*) .

The study of the results in Fig. 11.31 allows to draw the same conclusions: the proposed *StatFlow* framework is likely to improve and at worst, does not degrade the quality of the tracking with respect to *MSF*. It is interesting to see that for *Hopkins-two-cranes* for instance, 56.52% of the input points are still well tracked in I_{30} against 30.85% and 8.5% for *MSF* and *LDOF inverse*.

To illustrate all these results, we propose two texture insertion and propagation examples in order to visually assess the performance of each algorithm. Fig. 11.32 and Fig. 11.33 respectively deal with texture insertion in the first frame of *Hopkins-head* and *Hopkins-cars9* and propagation along the sequence. Fig. 11.32 involves *LDOF acc* in addition to the three previously described methods.

In Fig. 11.32, the red circular texture propagated with *LDOF inverse* and *MSF*(*LDOF*) knows significant distortions. The circular shape is not maintained due to the strong rotation of the head and we denote a large drift leading to the eye of the character from the cheek where the texture has been originally inserted. On the contrary, the compactness of the initial texture is respected with *LDOF acc* and *StatFlow*(*LDOF*) which reveal better results compared to *LDOF inverse* and *MSF*(*LDOF*). Finally, the long-term *to-the-reference* displacement fields obtained via *StatFlow* reach the best tracking among the four tested methods.

Concerning Fig. 11.33, it appears that *LDOF inverse* causes duplications of the initial texture. The best propagation is in this case performed by *MSF*. Indeed, the blue circles tend to spread with *StatFlow* (see the propagation in I_{53} especially).

11.4.6 Experiments with input block matching motion estimation

In order to demonstrate that any *optical flow* estimator can be used as input of our method, we present video editing results based on long-term displacement fields computed via *StatFlow* using a block matching algorithm as input. Of course, the quality of *StatFlow* depends on the input data quality. As shown in Fig. 11.34 (b) where the elementary *optical flow* $\boldsymbol{u}_{155,130}$ is displayed, the *multi-step* elementary *optical flow* fields propose one single motion vector per bloc of 8×8 pixels.

The editing task of Fig. 11.34 has been performed on the *MPI-S*1-155-175 sequence from I_{115} to I_{134}. The results show that *StatFlow* ($N_{it} = 9$) allows to densify the flow starting from block matching data. Although the input *multi-step* elementary *optical flow* fields are not very accurate, *StatFlow* is able to find and to select the best motion *paths* to perform the long-term matching task. Thus, a good accuracy can be reached as evidenced by the propagation results and the output displacement map $\boldsymbol{d}_{115,130}$ in Fig. 11.34 (c) to be compared with $\boldsymbol{u}_{115,130}$ Fig. 11.34 (b).

(a) Original image I_{115} (b) *BM*-based $\boldsymbol{u}_{115,130}$ (c) $\boldsymbol{d}_{115,130}$ after *StatFlow* (d) Texture insertion in I_{115}

(e) Propagation to I_{120} *StatFlow* (*BM*) (f) Propagation to I_{125} *StatFlow* (*BM*) (g) Propagation to I_{130} *StatFlow* (*BM*) (h) Propagation to I_{134} *StatFlow* (*BM*)

Figure 11.34: Logo insertion in I_{115} and propagation along the *MPI-S1-155-175* sequence up to I_{134} with the *statistical multi-step flow* (*StatFlow*) method proposed in Section 11.3 using block matching (*BM*) *multi-step* elementary *optical flow* fields.

11.5 Conclusion and perspectives

In this chapter, we firstly performed dense matching between two distant frames by considering multiple *multi-step* motion *paths* across the sequence. Given the resulting large set of motion candidates, we introduced a selection procedure where a global optimization stage is preceded by a new statistical processing which exploits the spatial distribution and the intrinsic quality of candidates. In terms of displacement vector selection, the proposed selection procedure, mainly based on statistics, leads to better results compared to state-of-the-art methods. The whole combinatorial *multi-step* integration and statistical selection framework (*CISS*) has shown its effectiveness for distant motion estimation. This is why we decided to extend this method to the whole sequence.

Consequently, we presented *Statistical multi-step Flow* (*StatFlow*), a two-step framework which performs dense and long-term motion estimation along long video shots. Our method starts by generating initial dense motion correspondences with a strong focus on motion inconsistency reduction. For this task, we rely on an improved and extended version of *CISS*, *CISS-K*, which is applied independently between a reference frame and each of the subsequent images of the sequence. It guarantees a low temporal correlation between the resulting correspondences respectively estimated for each of these pairs. We then propose to enforce temporal smoothness through a new iterative motion refinement (*IMR*) step. It considers several motion candidates including candidates from neighboring frames and it involves a new energy formulation with temporal smoothness constraints.

Experiments have studied the performance of the *StatFlow* approach compared to state-of-the-art methods and the *multi-step* flow fusion method (*MSF*) introduced in Chapter 10. These comparisons have been performed through both quantitative evaluation via registration and *PSNR* assessment and by comparing the obtained trajectories with dense ground-truth trajectories from the *Flag* [GRA11a] and *Hopkins* [TV07] datasets and qualitative evaluation via texture propagation and point tracking for a wide set of complex scenes. With respect to state-of-the-art methods and *MSF*, our conclusions reveal that *StatFlow* reaches a significant improvement of the results in terms of accuracy and robustness for both *from-the-reference* and *to-the-reference* displacement fields estimation.

In the context of our study, four points would deserve further investigation. First, it would be interesting to robustify the intrinsic quality value assigned to each candidate within the large distributions resulting from the combinatorial *multi-step* integration. In the same spirit, we could conceive to provide a more robust intrinsic quality assessment for each displacement vector involved within the global optimization stage. For this sake, one could imagine to involve gain factors to handle strong variations of illumination more accurately through gain-compensated matching cost or gain-based regularization. The median minimization formulation itself could be performed on gain values additionally to spatial positions. Features such as gradient similarity or correlation computation could also be considered.

Second, different types of motion information could be combined as inputs of the combinatorial integration such as block matching based estimates, parametric motion fields, sparse features using tools like *KLT*, *SIFT*, *SURF*... Following the approach of [LRR08], another idea could consist in simultaneously taking into account multiple *optical flow* estimators as inputs of *StatFlow*. Thus, it becomes possible to build hybrid motion *paths* made of *multi-step* elementary *optical flow* coming from different estimators. Even if the considered estimators may fail in some regions, the idea is to pool the strengths of each one.

Third, one can wonder if the accumulation of consecutive *optical flows* should or should not be kept within the combinatorial *multi-step* integration stage. As described in Section 11.3.1, our method limits the construction of motion *paths* by providing a maximum number of concatenations. However, it appears that the accumulation of *steps* 1 may be relevant in certain situations, especially dealing with large homogeneous areas, periodic structures or slow color variations. In this latter case, consecutive *optical flows* can prevent to jump from one structure to another.

Fourth, another version of motion candidate construction could consist in considering both *forward* and *backward multi-step* elementary *optical flow* fields in the concatenation for a given direction (*from-the-reference* or *to-the-reference*). This may have advantages in particular in case of occlusions. For the same reasons, we can extend the motion candidate construction using elementary *optical flow* fields that join frames which are outside the interval delimited by the pair of frames under consideration. The introduction of such additional *optical flow* fields may allow compensating the break of motion concatenations due to temporary occlusion.

Up to now, our frameworks have based the processing with respect to a single reference frame, I_{ref}. In the next chapter, Chapter 12, we will study how the introduction of additional reference frames can further robustify the dense and long-term matching process.

Chapter 12

Multi-reference frames long-term dense motion estimation

In Chapters 10 and 11, we described two different approaches to perform long-term dense motion estimation with respect to one single reference frame: *multi-step flow fusion* and *statistical multi-step flow*. Despite the robustness of these sophisticated methods, trajectories starting from the selected reference frame may diverge after a while for scenes whose motion is very complex or for very long video shots (or both). To repair the tracking failures and to push the motion estimation process as far as possible temporally, we suggest to rely on multi-reference frames strategies.

Recently, multi-reference frames motion estimation has been involved within video encoding standards such as H.264/AVC. For each macroblock, these video encoders select among multiple reference frames (16 concurrent reference frames maximum for H.264) which one it can rely on. Multiple reference frames estimation can considerably increase the encoding time but it has been shown that it improves the accuracy of the motion compensation and therefore provides significant coding gain [BDRB07].

We propose in this chapter to exploit the concept of multi-reference frames estimation to the purpose of very long-term dense motion estimation. Instead of relying only on one single reference frame, the basic idea behind this is to insert new reference frames along the sequence each time the motion estimation process fails and then to apply the long-term dense motion estimator with respect to each of these inserted reference frames. To avoid the motion drift and to correct the single reference frame estimation issues, we claim that we can advantageously combine the displacement vectors with good quality among all the generated multi-reference displacement vectors.

Such approach follows the same spirit of [RLF12] whose aim is to re-correlate short-range *tracklets* (i.e. pieces of trajectories) estimated with respect to different starting frames in order to go towards longer long-range motion trajectories. Indeed, our contributions in the context of multi-reference frames motion estimation deal with robust combinations of accurate *tracklets* estimated with respect to multiple reference frames in either *forward* or *backward* direction.

Contrary to [RLF12], we propose to exploit this concept of *tracklets* combinations in the context of dense motion estimation. Moreover, we aim at giving a central role to the trajectory quality assessment aspects. We claim that a good evaluation of the intrinsic quality of the displacement vectors can both guide the insertion of new reference frames and identify which vectors must be refined and which ones can contribute toward this refinement task.

In particular, two complementary frameworks are studied in the following. First, we detail in Section 12.1 how to perform multi-reference frame dense motion estimation through concatenation of multi-reference displacement fields. We focus especially on the insertion of new reference frames based on trajectory quality assessment. This multi-reference strategy is then assessed in Section 12.2 via both qualitative and quantitative experiments. Second, we present in Section 12.3 a new two-reference frames motion refinement which consists in refining when necessary the motion estimates within each temporal segment located between two reference frames. Experimental results of this refinement stage are given in Section 12.4. Finally, Section 12.5 concludes this chapter.

12.1 Multi-reference frames strategy through trajectory quality assessment

Toward very long-term dense motion estimation, we propose to add new reference frames once the trajectories fail. By this way, we continue the motion estimation from an intermediate sound frame in order to finally extend temporally the trajectory estimation process as far as possible.

Contrary to modern coding frameworks such as H.264/AVC which rely on periodically inserted reference frames, we suggest to insert new reference frames based on a quality assessment tool able to automatically judge how the quality of the trajectories are evolving along the sequence.

The *reference frame* terminology is ambiguous and it is important before explaining our contributions to dissociate *reference frame* in the point of view of user interaction and *reference frame* considered as an algorithmic tool. In the context of video editing for instance, the user will insert the texture/logo in one single reference frame and run the multi-reference frames algorithm described below. The other reference frames we propose to insert will just be an algorithmic way to perform a better motion estimation without any user interaction.

Let us focus on the estimation of the trajectory $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ along a sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\ldots,N]\!]}$ with $I_{ref_0} = I_0$ considered as reference frame. $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ starts from the grid point $\boldsymbol{x}_{ref_0}$ of I_{ref_0} and is defined by a set of *from-the-reference* displacement vectors $\{\boldsymbol{d}_{ref_0,n}(\boldsymbol{x}_{ref_0})\}\ \forall n \in [\![ref_0+1,\ldots,N]\!]$.

Let us assume that the long-term dense motion estimation involved for the estimation of $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ fails before I_N and more precisely at I_{fail_0} with $fail_0$ (see Fig. 12.1). We propose to introduce a new reference frame at I_{fail_0-1}, i.e. at the instant which precedes the tracking failure and for which $\boldsymbol{d}_{ref_0,fail_0-1}(\boldsymbol{x}_{ref_0})$ has been accurately estimated.

Once this new reference frame (referred to as I_{ref_1}) has been inserted, we run new motion estimations starting from the position $\boldsymbol{x}_{ref_0} + \boldsymbol{d}_{ref_0,ref_1}(\boldsymbol{x}_{ref_0})$ in $I_{ref_1} = I_{fail_0-1}$ between I_{ref_1} and each subsequent frames I_n with $n \in [\![ref_1+1,\ldots,N]\!]$. Thus, we obtain the set of displacement vectors $\{\tilde{\boldsymbol{d}}_{ref_1,n}\}\ \forall n \in [\![ref_1+1,\ldots,N]\!]$. These estimates allow to obtain a new version of the displacement vectors we would like to correct: $\{\boldsymbol{d}_{ref_0,n}(\boldsymbol{x}_{ref_0})\}_{n\in[\![ref_1+1,\ldots,N]\!]}$. Indeed, each initial estimate of these displacement vectors can be replaced by the vector obtained through concatenation of $\boldsymbol{d}_{ref_0,ref_1}$ estimated with respect to I_{ref_0} and $\boldsymbol{d}_{ref_1,n}$ we just computed with respect to I_{ref_1}:

$$\boldsymbol{d}_{ref_0,n}(\boldsymbol{x}_{ref_0}) = \boldsymbol{d}_{ref_0,ref_1}(\boldsymbol{x}_{ref_0}) + \tilde{\boldsymbol{d}}_{ref_1,n}(\boldsymbol{x}_{ref_0} + \boldsymbol{d}_{ref_0,ref_1}(\boldsymbol{x}_{ref_0})) \tag{12.1}$$

If this resulting new version of $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ fails again, at I_{fail_1} for instance (with $fail_0 < fail_1 < N$), we insert a new reference frame I_{ref_2} at I_{fail_1-1} and we perform the long-term estimator starting from I_{ref_2} (Fig. 12.1). Thus, we can obtain new estimates of the displacement vectors $\{\boldsymbol{d}_{ref_0,n}(\boldsymbol{x}_{ref_0})\}$ with $n \in [\![ref_2+1,\ldots,N]\!]$ as follows:

$$\begin{aligned}\boldsymbol{d}_{ref_0,n}(\boldsymbol{x}_{ref_0}) &= \boldsymbol{d}_{ref_0,ref_1}(\boldsymbol{x}_{ref_0}) + \tilde{\boldsymbol{d}}_{ref_1,ref_2}(\boldsymbol{x}_{ref_0} + \boldsymbol{d}_{ref_0,ref_1}) \\ &+ \tilde{\boldsymbol{d}}_{ref_2,n}(\boldsymbol{x}_{ref_0} + \boldsymbol{d}_{ref_0,ref_1} + \tilde{\boldsymbol{d}}_{ref_1,ref_2})\end{aligned} \tag{12.2}$$

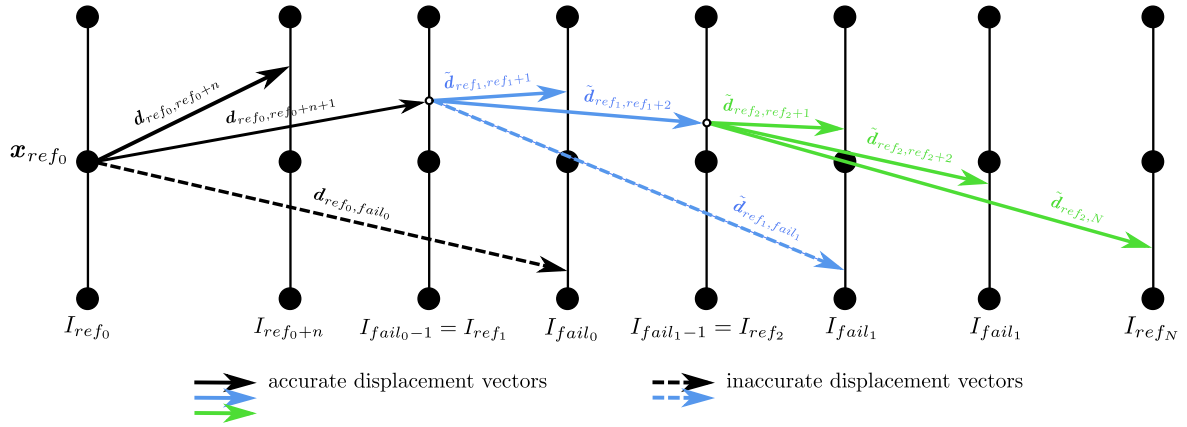


Figure 12.1: Multi-reference frames algorithm towards very long-term dense motion estimation. We propose to add new reference frames once the trajectories under consideration fail to push the motion estimation process as far as possible temporally along the image sequence.

We apply an exactly similar processing each time $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ fails again, up to the end of the sequence. How to justify the improvement of this multi reference frames motion estimation compared to classic single reference frame approach? It appears that the displacement selection criteria (including the *brightness constancy assumption*) are more valid when we rely on a reference frame which is closer from the current frame than the initial reference frame (I_{ref_0}). In case of strong color variations especially, the matching can be more easily performed.

In practice, we can judge the quality of $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ through the study of the binary inconsistency values assigned to each displacement vectors $\{\boldsymbol{d}_{ref_0,n}(\boldsymbol{x}_{ref_0})\}\ \forall n \in [\![ref_0 + 1, \ldots, N]\!]$. For recall, the binary inconsistency values are obtained after having thresholded with ϵ_{Inc} the continuous inconsistency values, as explained in Section 10.2.3, Chapter 10. If one of these vectors is inconsistent, the process automatically adds a new reference frame at the instant which precedes the matching issue and runs the procedure described above.

If the long-term motion estimator is the *statistical multi-step flow* algorithm (described in Chapter 11), an alternative to judge the quality of $\boldsymbol{T}(\boldsymbol{x}_{ref_0})$ consists in studying the variance of the candidate distribution $T_{ref_0,n}(\boldsymbol{x}_{ref_0})$ obtained via combinatorial integration by the multiple *motion paths*. The more the distribution is spread out, the higher the probability of a bad motion estimate is.

Let us now extend this processing to a set of trajectories starting from I_{ref_0}. The underlying idea remains the same but applying the previously described procedure is not straightforward since the quality must be studied globally for the whole set of trajectories and does not focus only on one single trajectory. If the insertion of new reference frames is not decided manually (i.e. by the user), an automatic study of the temporal evolution of a global motion confidence score is required. For this task, several criteria can be considered, such as:

- mean of continuous displacement inconsistency values,
- percentage of pixels whose corresponding displacement vector is inconsistent (according to the binary inconsistency value),
- mean of motion-compensated absolute differences,

- reconstruction quality through registration and *PSNR* assessment on the area to be tracked.

Whatever the criteria, a motion quality threshold must be set according to the quality requirements to determine from which instant a new reference frame is needed. A local assessment which focuses only on the region of interest may be relevant when the whole images are not involved. The quality of the motion estimation process highly depends on the area under consideration and studying the motion vector quality for the whole image could badly influence the reference frame insertion process in this case.

If the application under consideration requires the estimation of *to-the-reference* displacement vectors $\boldsymbol{d}_{n,ref_0}(\boldsymbol{x}_n)\ \forall n$ (texture insertion and propagation for instance), it seems difficult to apply this multi-reference frames processing starting from each frame I_n to I_{ref_0} for computational issues. We propose to keep the processing in the *from-the-reference* direction from I_{ref_0} and therefore to decide the introduction of new reference frames with respect to the quality of *from-the-reference* displacement vectors.

To-the-reference displacement vectors can benefit from the introduction of these new reference frames anyway. If we come back to the previous example where I_{ref_1} and I_{ref_2} have been inserted, inaccurate displacement vectors $\boldsymbol{d}_{n,ref_0}(\boldsymbol{x}_n)$ starting from the grid point $\boldsymbol{x}_n$ of I_n with $n \in [\![ref_2+1,\ldots,N]\!]$ can be refined by considering the following concatenations:

$$\begin{aligned} \boldsymbol{d}_{n,ref_0}(\boldsymbol{x}_n) &= \boldsymbol{d}_{n,ref_2}(\boldsymbol{x}_n) + \tilde{\boldsymbol{d}}_{ref_2,ref_1}(\boldsymbol{x}_n + \boldsymbol{d}_{n,ref_2}) \\ &+ \tilde{\boldsymbol{d}}_{ref_1,n}(\boldsymbol{x}_n + \boldsymbol{d}_{n,ref_2} + \tilde{\boldsymbol{d}}_{ref_2,ref_1}) \end{aligned} \tag{12.3}$$

To ensure a certain correlation between the quality assessment of *from-the-reference* displacement vectors and the effective quality of *to-the-reference* displacement vectors, we propose to select the percentage of pixels whose corresponding displacement vector is inconsistent among the previously described criteria for the insertion of new reference frames. We explain this choice by the fact that the inconsistency involved in this criterion deals with *forward-backward* consistency and therefore simultaneously addresses the quality of both *from-the-reference* and *to-the-reference* displacement vectors.

Despite its ability to achieve very long-term motion estimation, the multi-reference frame strategy works only if the area for which it estimates the displacement vectors is visible in each inserted reference frame. Indeed, each inserted reference frame will serve as a relay to continue the estimation for the subsequent frames. In other words, if any part of the area to be tracked is occluded at a frame considered as a reference frame by the algorithm, it will be impossible to track it later along the subsequent frames of the video sequence.

12.2 Experimental evaluation of the multi-reference estimation

Our experiments focus on the following sequences: *Walking-Couple*-0-60 (Fig. 11.8), *MPI-S*1-115-175 [GKT+] (Fig. 11.20), *Flag* [GRA11a, GRA11b] (Fig. 11.27), *Hopkins-head* [TV07] (Fig. 11.29 *(e)*) and *Hopkins-two-cranes* [TV07] (Fig. 11.29 *(c)*).

We propose to evaluate the performance of the multi-reference frames strategy described in Section 12.1. For this sake, Section 12.2.1 shows in details how this strategy works through a simulation of the processing chain applied to the *Walking-Couple*-0-60 sequence. A stong focus is given to the trajectory quality assessment aspects. A video editing example is finally provided to qualitatively assess the good performance of the processing chain in comparison with a classic single reference frame motion estimation. In Section 12.2.2, we demonstrate that it provides also satisfying results for the *MPI-S*1-115-175 sequence. Finally, Section 12.2.3 is dedicated to a quantitative assessment via comparisons with respect to groundtruth data provided with the *Flag* [GRA11a, GRA11b] and *Hopkins* [TV07] datasets.

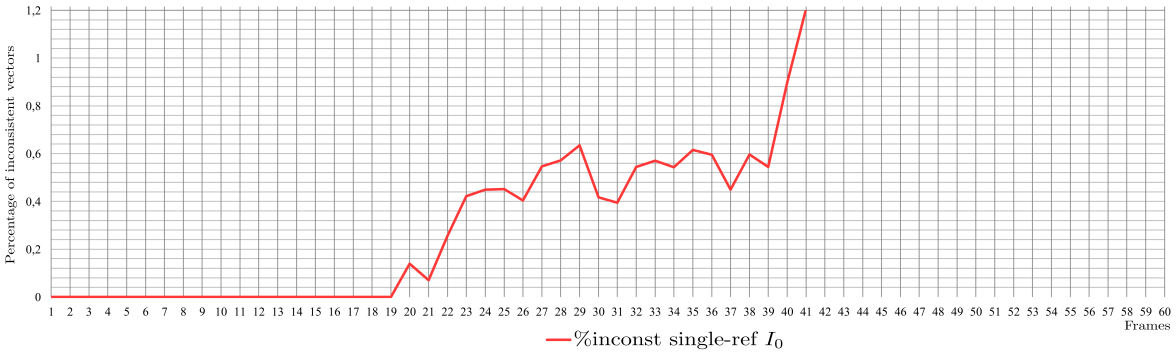
12.2.1 Simulation of the multi-reference processing chain

For this experiment, we focus on the yellow patch located on the woman shirt (Fig. 12.3 *(a)*) in frame I_0 of *Walking-Couple-0-60*. We aim at performing a long-term motion estimation starting from all the pixels belonging to this area. In our example, the resulting *to-the-reference* displacement vectors will serve to propagate the yellow texture from I_0, up to I_{60}.

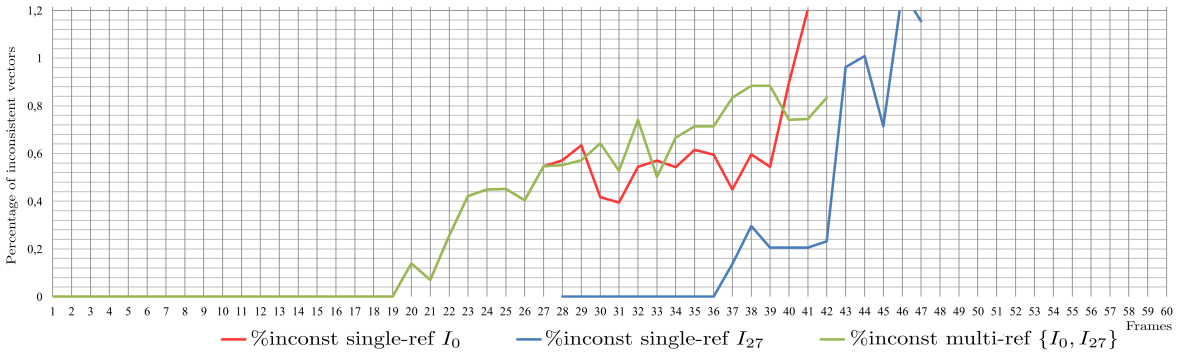
We start the process by performing the *statistical multi-step flow* (*StatFlow*) long-term dense motion estimator from I_0 with *LDOF* [BM11] *multi-step* elementary *optical flow* vectors estimated with the following *steps*: 1-5, 10 and 15. Then, for each pair $\{I_0, I_n\}$, we compute the percentage of pixels whose displacement vector $\boldsymbol{d}_{0,n}$ is inconsistent (threshold of $\epsilon_{Inc} = 1$ pixel applied on the continuous inconsistency values). The temporal evolution of this percentage of inconsistent displacement vectors is displayed in Fig. 12.2 *(a)*. Let us assume that the quality requirements dictate a quality threshold of 0.5%. Thus, the algorithm determines from which frame I_n the threshold of 0.5% is exceeded. According to Fig. 12.2 *(a)*, it deals with I_{28} with a percentage of 0.55%. Therefore, the algorithm inserts a reference frame at the previous instant, i.e. at I_{27}.

Then, *StatFlow*(*LDOF*) is applied again from I_{27} (*steps* 1-5, 10, 15) and one studies how the percentage of pixels of I_{27} whose displacement vector $\boldsymbol{d}_{27,n}$ is inconsistent evolves (Fig. 12.2 *(b)*, blue curve). The pixels of I_{27} involved for this quality assessment task are defined as the nearest grid points with respect to the ending positions of the displacement vectors $\boldsymbol{d}_{0,n}$ starting from each pixel of the yellow patch in I_0. According to Fig. 12.2 *(b)* (blue curve), we notice that the quality threshold is again exceeded at I_{43} (with 0.96%). Consequently, the algorithm automatically inserts a new reference frame at instant $n = 42$ and performs *StatFlow*(*LDOF*) with respect to I_{42} (*steps* 1-5, 10, 15) .

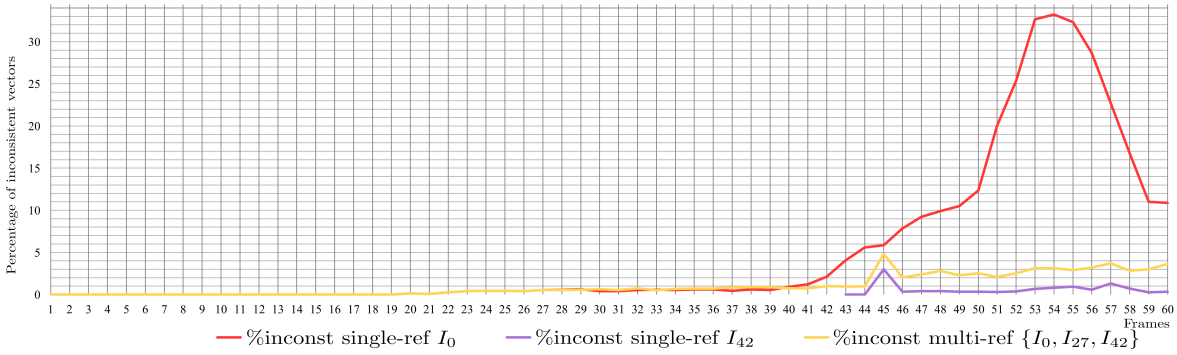
These three *StatFlow* estimations (from I_0, I_{27} and I_{42}) allow to compute *from-the-reference* displacement vectors $\boldsymbol{d}_{0,n}$ and *to-the-reference* displacement vectors $\boldsymbol{d}_{n,0}$ $\forall n \in [\![1, \ldots, 60]\!]$ through concatenations of multi-reference frames displacement vectors following respectively Eq. 12.2 and Eq. 12.3 (Section 12.1). Note that we made the choice to stop inserting new reference frames from I_{42} even if the threshold is exceeded from I_{45} (see the purple curve in Fig. 12.2 *(c)* which deals with the percentage of pixels in I_{42} whose displacement vector $\boldsymbol{d}_{42,n}$ is inconsistent).



(a) *StatFlow* estimation from I_0



(b) *StatFlow* estimation from the new reference frame inserted at I_{27}



(c) *StatFlow* estimation from the new reference frame inserted at I_{42}

Figure 12.2: Simulation of the multi-reference frames processing chain described in Section 12.1 for the purpose of dense long-term motion estimation of a patch located within the woman shirt in frame I_0 of *Walking-Couple-0-60* (see Fig. 12.3 *(a)*). New reference frames are inserted through the study of the percentage of inconsistent *from-the-reference* vectors starting from the pixels which belong to the propagated patch.

(a) Texture insertion in I_0 (b) Propagation to I_{20} *StatFlow*(*LDOF*) (c) Propagation to I_{27} *StatFlow*(*LDOF*) (d) Propagation to I_{28} *StatFlow*(*LDOF*)

(e) Propagation to I_{31} Single-ref *StatFlow*(*LDOF*) (f) Propagation to I_{39} Single-ref *StatFlow*(*LDOF*) (g) Propagation to I_{42} Single-ref *StatFlow*(*LDOF*) (h) Propagation to I_{48} Single-ref *StatFlow*(*LDOF*)

(i) Propagation to I_{31} **Multi-ref *StatFlow*(*LDOF*)** (j) Propagation to I_{39} **Multi-ref *StatFlow*(*LDOF*)** (k) Propagation to I_{42} **Multi-ref *StatFlow*(*LDOF*)** (l) Propagation to I_{48} **Multi-ref *StatFlow*(*LDOF*)**

(m) Propagation to I_{51} Single-ref *StatFlow* (*LDOF*) (n) Propagation to I_{55} Single-ref *StatFlow* (*LDOF*) (o) Propagation to I_{57} Single-ref *StatFlow* (*LDOF*) (p) Propagation to I_{60} Single-ref *StatFlow* (*LDOF*)

(q) Propagation to I_{51} **Multi-ref *StatFlow* (*LDOF*)** (r) Propagation to I_{55} **Multi-ref *StatFlow* (*LDOF*)** (s) Propagation to I_{57} **Multi-ref *StatFlow* (*LDOF*)** (t) Propagation to I_{60} **Multi-ref *StatFlow* (*LDOF*)**

Figure 12.3: Texture insertion in I_0 and propagation along the *Walking-Couple-0-60* sequence up to I_{60}. Through the *statistical multi-step flow* method (Section 11.3) using *LDOF* [BM11] *multi-step* elementary *optical flows* (*StatFlow*(*LDOF*)), we compare: 1) a single-ref. frame strategy from I_0, 2) a multi-ref. frames strategy (Section 12.1) using $\{I_0, I_{27}, I_{42}\}$ as reference frames.

Fig. 12.2 gives more information and shows especially the curves of percentage of inconsistent displacement vectors once the multi-reference frames strategy has been performed, i.e. after the multi-reference frames displacement vector concatenations. The multi-reference frames estimations respectively performed with respect to $\{I_0, I_{27}\}$ and $\{I_0, I_{27}, I_{42}\}$ are assessed via the green curve of Fig. 12.2 *(b)* and the yellow curve of Fig. 12.2 *(c)*. We notice that the green curve of Fig. 12.2 *(b)* (multi-reference $\{I_0, I_{27}\}$) gives slightly poorer results compared to the red curve of Fig. 12.2 *(b)* (single-reference $\{I_0\}$) between I_{30} and I_{32} and between I_{34} and I_{39}. This is in fact due to pixels which were considered as occluded with the single-reference frame estimation and which become visible and inconsistent. Therefore, this artificially increases the number of inconsistent vectors. Nevertheless, the yellow curve of Fig. 12.2 *(c)* (multi-reference $\{I_0, I_{27}, I_{42}\}$) compared to the red curve of Fig. 12.2 *(b)* (single-reference $\{I_0\}$) shows clearly the performance of our multi-reference frames estimation. At I_{60}, only 3.65% of the displacement vectors starting from the yellow patch of I_0 are inconsistent with our multi-reference frames approach against 10.86% with a single-reference motion estimation from I_0.

To show qualitatively the effectiveness of the proposed multi-reference processing chain, we propose to involve the resulting *to-the-reference* displacement vectors $\boldsymbol{d}_{n,0}$ $\forall n \in [\![1, \ldots, 60]\!]$ to propagate the yellow texture across the sequence, from I_0 to I_{60} (Fig. 12.3).

The two first rows and the fourth row show how the single-reference frame motion estimation from I_0 performs the propagation. We notice that a hole appears in I_{28} (upper right part of the texture, Fig. 12.3 *(d)*) and grows gradually due to a bad motion estimation of the periodic structures. It appears also that the compacity of the initial texture is lost from in I_{48} (Fig. 12.3 *(h)*). The texture diverge abnormally above (Fig. 12.3 *(n)*) and to the right (Fig. 12.3 *(p)*) of the correct texture position. This quality loss from I_{28} has been identified when studying the temporal evolution of the percentage of inconsistent vectors (Fig. 12.3 *(a)*). One can realize that the visual results and our automatic quality assessment process coincide.

The third and the fifth rows illustrate how the multi-reference frames motion estimation performed with respect to $\{I_0, I_{27}, I_{42}\}$ propagates the texture up to I_{60}. Despite small holes (Fig. 12.3 *(t)*), the results appear to be much better than the ones obtained with the single reference frame estimation. The propagation is clearly performed without any disturbing artifacts. We can notice also that the occlusion due to the arm of the woman is well handled (Fig. 12.3 *(r)*) by our method. Occluded parts of the texture are not propagated, as one expects.

12.2.2 Long-term warping

To illustrate that the multi-reference frame strategy based on trajectory quality assessment provides satisfying results in a wide set of complex scenes, we performed the same type of experiment on the *MPI-S1-115-175* sequence. The idea here is to propagate a logo inserted in I_{115} across the sequence, up to I_{165}. The logo is inserted on an un-textured area which undergoes strong illumination variations as well as a non-rigid transformation due to the rotation of the woman (Fig. 12.4 *(a)*).

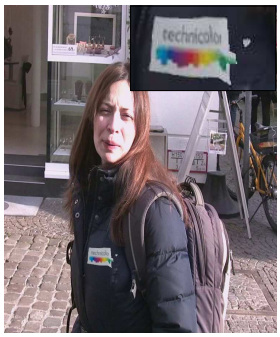
The study of the percentage of inconsistent vectors belonging to the area where the logo has been inserted in I_{115} has lead in this case to the insertion of three additional reference frames ($\{I_{135}, I_{155}, I_{160}\}$) from which *StatFlow* has been performed. In Fig. 12.4, the resulting *to-the-reference* displacement vectors are compared through logo propagation to *to-the-reference* displacement vectors coming from a single reference frame motion estimation achieved from I_{115}. Whatever the reference frame under consideration, the *StatFlow* long-term motion estimator has

(a) Logo insertion in I_{115} *StatFlow(2D-DE)*

(b) Propagation to I_{121} *StatFlow(2D-DE)*

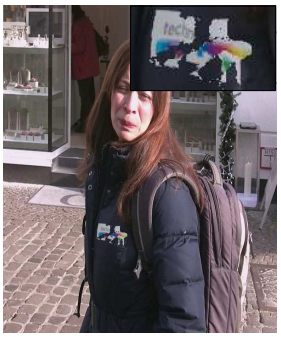
(c) Propagation to I_{126} *StatFlow(2D-DE)*

(d) Propagation to I_{131} *StatFlow(2D-DE)*



(e) Propagation to I_{139} Single-ref *StatFlow(2D-DE)*

(f) Propagation to I_{144} Single-ref *StatFlow(2D-DE)*



(g) Propagation to I_{151} Single-ref *StatFlow(2D-DE)*

(h) Propagation to I_{155} Single-ref *StatFlow(2D-DE)*

(i) Propagation to I_{139} **Multi-ref *StatFlow(2D-DE)***

(j) Propagation to I_{144} **Multi-ref *StatFlow(2D-DE)***

(k) Propagation to I_{151} **Multi-ref *StatFlow(2D-DE)***

(l) Propagation to I_{155} **Multi-ref *StatFlow(2D-DE)***

(m) Propagation to I_{157} **Multi-ref *StatFlow (2D-DE)***

(n) Propagation to I_{160} **Multi-ref *StatFlow (2D-DE)***

(o) Propagation to I_{163} **Multi-ref *StatFlow (2D-DE)***

(p) Propagation to I_{165} **Multi-ref *StatFlow (2D-DE)***

Figure 12.4: Logo insertion in I_{115} and propagation along the *MPI-S*1-115-*175* sequence up to I_{165}. Through the *statistical multi-step flow* method (Section 11.3) using 2*D-DE multi-step* elementary *optical flows* (*StatFlow*(2*D-DE*)), we compare: 1) a single-ref. frame strategy from I_{115}, 2) a multi-ref. frames strategy (Section 12.1) using $\{I_{115}, I_{135}, I_{155}, I_{160}\}$ as reference frames.

used 2*D-DE* [RTDC12] *multi-steps* elementary *optical flow* vectors as inputs. The following *steps* have been considered: 1-5, 10, 15 and 30.

The first row of Fig. 12.4 shows good results for the 16 first frames. Then, by comparing the second and the third row, we notice that the single reference frame estimation makes the logo progressively distorted (Fig. 12.4 *(f)*) and finally not at all recognizable (Fig. 12.4 *(h)*). On the contrary, the multi-reference frames estimation keeps the logo in a compact form (Fig. 12.4 *(i-l)*) and follows accurately the non-rigid motion of the woman. Finally, the fourth row indicates that it is possible to rely on good motion estimates for a temporal distance of 50 frames, up to I_{165}.

12.2.3 Quantitative results with groundtruth data

This section aims at quantitatively assessed *from-the-reference* displacement vectors computed with the multi-reference frames strategy. This evaluation is done via comparisons with respect to groundtruth data provided with the *Flag* [GRA11a, GRA11b] (Section 12.2.3.1) and *Hopkins* [TV07] (Section 12.2.3.2) datasets.

12.2.3.1 Quantitative results using the *Flag* benchmark dataset

Following the same procedure as the one described in Section 12.1 and illustrated in Section 12.2.1, we performed a multi-reference frames motion estimation based on *StatFlow*(*LDOF*) with $\{I_1, I_{20}\}$ as reference frames on the original version of the *Flag* [GRA11a, GRA11b] sequence (Fig. 11.27). Using its associated groundtruth dense trajectory data, the *from-the-reference* displacement vectors obtained with the multi-reference frames procedure are compared to *from-the-reference* displacement vectors estimated with the single reference frame motion estimators previously described in Section 11.4.4: *LDOF direct* [BM11], *ITV-L1 direct* [WPZ+09], [PB12] *direct*, *LDOF acc*, *MFSF-PCA* [GRA11b], *MFSF-DCT* [GRA11b], *MFSF-PCA* [GRA13], *MFSF-DCT* [GRA13], *MSF* (*LDOF*) and the single reference frame *StatFlow* (*LDOF*) performed with respect to I_0. The *multi-step* strategies including the multi-reference frames estimation based on *StatFlow*(*LDOF*) have considered as inputs the steps $1-5$, 8, 10, 15, 20, 25, 30, 40 and 50.

Tab. 12.1 gives for all the previously described methods the *RMS* (*root mean square*) endpoint errors between the respective obtained displacement fields and the ground-truth data (Eq. 11.19 in Section 11.4.4). The multi-reference frames *StatFlow*(*LDOF*) outperforms the single-reference frame *StatFlow*(*LDOF*) with a global *RMS* error of 0.58 pixels against 0.69. A fortiori, the multi-reference frames *StatFlow*(*LDOF*) algorithm gives more accurate displacement fields than all the single reference frame methods including the challenging *MFSF-PCA* [GRA13].

When studying the *RMS* endpoint errors computed for each pair of frames with *LDOF direct*, *LDOF acc*, *MSF* (*LDOF*), the single reference frame *StatFlow*(*LDOF*) and the multi-reference frame *StatFlow*(*LDOF*) (Fig. 12.5), we observe that the multi-reference frames strategy strongly reduces the matching issues around I_{30} which coincides with the maximum deformation of the flag (Fig. 11.27 *(d)*). Indeed, the single reference frame StatFlow(*LDOF*) gives a *RMS* error of 2.22 pixels for $\{I_0, I_{29}\}$ whereas the multi-reference frame strategy leads to 1.37 pixels.

However, the multi-reference frames strategy method gives slightly worse results from I_{35} to I_{60} due to the fact that the flag comes back approximately to its initial position at the end of the sequence (Fig. 11.27 *(g)*). In this very particular context of symmetric sequence, it appears that the matching criteria are more valid with respect to I_0 than with respect to I_{20}.

Method	RMS endpoint error (pixels)
Multi-ref.$\{I_1, I_{20}\}$ ***StatFlow(LDOF)*** (Section 12.1)	**0.58**
Single-ref. *StatFlow*(*LDOF*) (Section 11.3)	0.69
MSF(*LDOF*) (Section 11.2)	1.41
LDOF direct [BM11]	1.74
LDOF acc [BM11]	4
MFSF-PCA [GRA13]	0.69
MFSF-DCT [GRA13]	0.80
MFSF-PCA [GRA11b]	0.98
MFSF-DCT [GRA11b]	1.06
[PB12] *direct*	1.24
ITV-L1 direct [WPZ+09]	1.43

Table 12.1: RMS endpoint errors for different methods on the *Flag* benchmark dataset provided in [GRA11a].

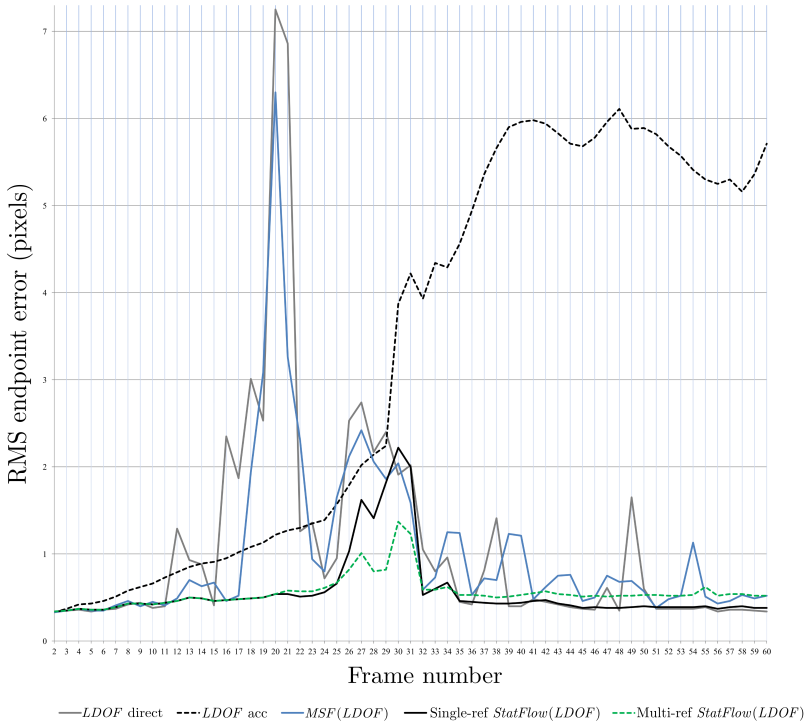
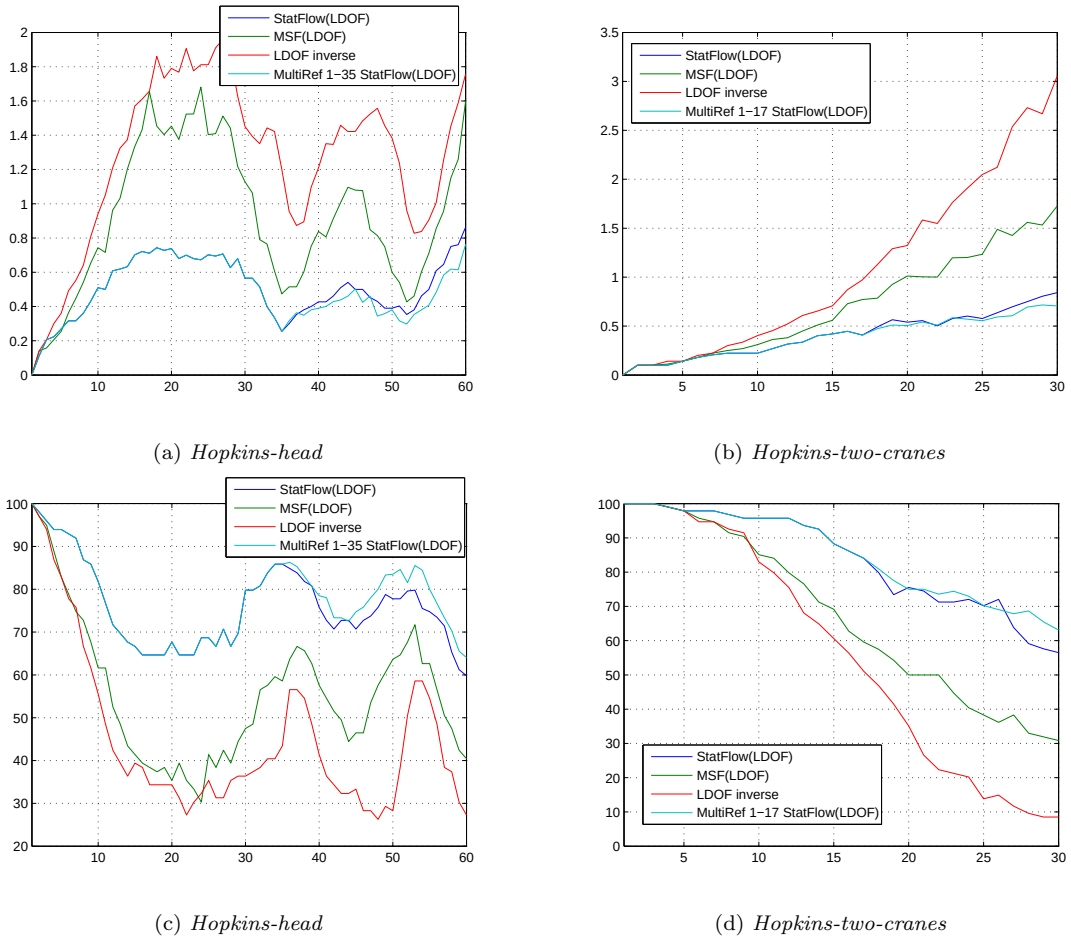


Figure 12.5: *RMS* endpoint errors for each pair $\{I_{ref}, I_n\}$ along *Flag* sequence [GRA11a] with: *LDOF direct*, *LDOF acc*, *MSF*(*LDOF*), *StatFlow*(*LDOF*) and our multi-reference frames strategy based on *StatFlow*(*LDOF*) with $\{I_1, I_{20}\}$ as reference frames.



(a) *Hopkins-head* (b) *Hopkins-two-cranes*

(c) *Hopkins-head* (d) *Hopkins-two-cranes*

Figure 12.6: Position *median absolute error* (*MedE*) (*a,b*) and proportion of points whose location is distant of maximum 1 pixel (*c,d*) with respect to the corresponding groundtruth positions (*Hopkins-head* and *Hopkins-two-cranes* sequences). We compare: 1) single-ref. frame strategies from I_0 (*LDOF inverse*, *MSF*(*LDOF*), *StatFlow*(*LDOF*)), 2) our multi-ref. frames strategy (Section 12.1) based on *StatFlow*(*LDOF*) using respectively $\{I_1, I_{35}\}$ (*Hopkins-head*) and $\{I_1, I_{17}\}$ (*Hopkins-two-cranes*) as reference frames.

12.2.3.2 Quantitative results using the *Hopkins* ground-truth data

To finish the quantitative assessment task, we focus on two sequences of the *Hopkins-155* dataset [TV07]: *Hopkins-head* (Fig. 11.29 *(e)*) and *Hopkins-two-cranes* (Fig. 11.29 *(c)*). The associated groundtruth sparse trajectories are involved to compare the results of: 1) single-reference frame strategies from I_0 (*LDOF inverse*, *MSF*(*LDOF*), *StatFlow*(*LDOF*)), 2) our multi-reference frames strategy (Section 12.1) based on *StatFlow*(*LDOF*) using respectively $\{I_1, I_{35}\}$ (*Hopkins-head*) and $\{I_1, I_{17}\}$ (*Hopkins-head*) as reference frames. The *multi-step* strategies including the multi-reference frames estimation based on *StatFlow*(*LDOF*) use the following steps: $1-5$, 10, 20 and 40.

As in Section 11.4.5, we use the position *MedE* (Fig. 12.6 (*a,b*)), i.e. the position *median absolute error* and the percentage of points whose location is distant of maximum 1 pixel with respect to the corresponding groundtruth positions (Fig. 12.6 (*c,d*)) as comparison criteria.

In Fig. 12.6 (*a,b*), we observe that the results are slightly improved for both sequences using an additional reference frame. Indeed, for the last frame I_{60} of the *Hopkins-head* sequence for instance, the multi-reference frames *StatFlow*(*LDOF*) algorithm leads to a *MedE* of 0.74 pixel to be compared with the 0.86 pixel obtained with the single-reference frame version. Fig. 12.6 (*c,d*) reveals the same finding. The multi-reference frames *StatFlow*(*LDOF*) approach outperforms the three single reference frame strategies and especially the single reference frame *StatFlow*(*LDOF*) method with, for example, a gain of about 8% for the frame I_{30} of the *Hopkins-two-cranes* sequence.

12.3 Two-reference frames motion refinement

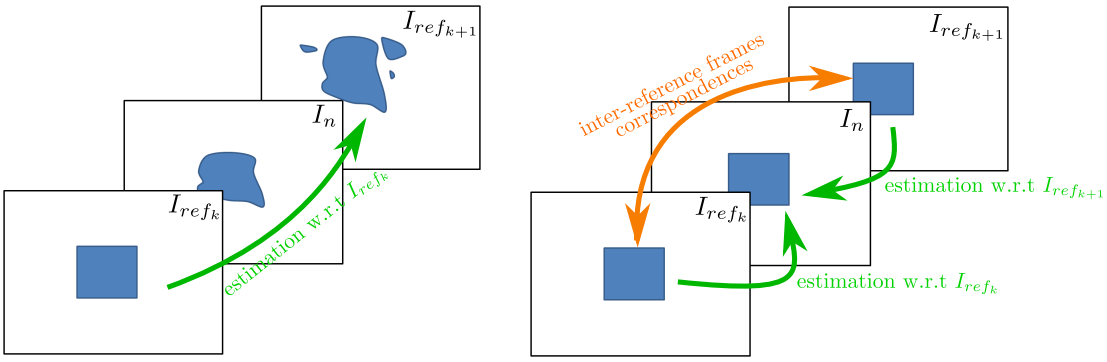
We previously studied how the motion estimation process can be enhanced through the insertion of multiple reference frames. Additionally to this method, we describe in what follows a new technique to improve the quality of the displacement vectors between two reference frames I_{ref_k} and $I_{ref_{k+1}}$ with $ref_k < ref_{k+1}$. The underlying idea is to combine the previously described multi-reference frames processing (presented in Section 12.1) with a two-reference frames motion refinement approach performed when necessary within the small temporal segments $[\![I_{ref_k}, I_{ref_{k+1}}]\!]$, i.e. between each reference frame and the next one.

This two-reference frames motion refinement approach has been designed with in mind complex situations such as temporary occlusion or strong illumination changes for which classic single-reference frame-based motion estimation may fail. As illustrated in Fig. 12.7, if we are able to accurately match the two reference frames I_{ref_k} and $I_{ref_{k+1}}$, we can benefit from the additional information coming from a *backward* motion estimation performed with respect to $I_{ref_{k+1}}$ in order to refine *forward* motion estimates computed with respect to I_{ref_k}.

More precisely, *forward* trajectories from I_{ref_k} can be refined using *backward* trajectories from $I_{ref_{k+1}}$ by advantageously combining their high-quality displacement vectors while rejecting motion *outliers*. Fig. 12.8 illustrates how the *forward* trajectory starting from $\boldsymbol{x}_{ref_k}$ in I_{ref_k} can be refined using *backward* displacement vectors from $I_{ref_{k+1}}$. Thus, the example of Fig. 12.8 shows that the badly estimated *forward* displacement vector $\boldsymbol{d}_{ref_k,n+2}$ can be refined using the *backward* displacement vector $\boldsymbol{d}_{ref_{k+1},n+2}$. We claim that such combination of motion estimates defined from both reference frames (Fig. 12.7 (b)) can lead to better results than a classic single-reference frame processing (Fig. 12.7 (a)).

Our method requires as inputs dense *from-the-reference* and *to-the-reference* motion fields computed through any long-term dense motion (such as *multi-step* flow fusion or *statistical multi-step flow*) with respect to both reference frames I_{ref_k} and $I_{ref_{k+1}}$:

- $\boldsymbol{d}_{ref_k,n}$ and $\boldsymbol{d}_{n,ref_k}$ with respect to I_{ref_k} with $n \in [\![ref_k + 1, \ldots, ref_{k+1}]\!]$,
- $\boldsymbol{d}_{ref_{k+1},n}$ and $\boldsymbol{d}_{n,ref_{k+1}}$ with respect to $I_{ref_{k+1}}$ with $n \in [\![ref_k, \ldots, ref_{k+1} - 1]\!]$.



(a) single-reference frame processing (b) two-reference frames processing

Figure 12.7: Schematic comparison between a classic single-reference frame processing and the proposed two-reference frames approach.

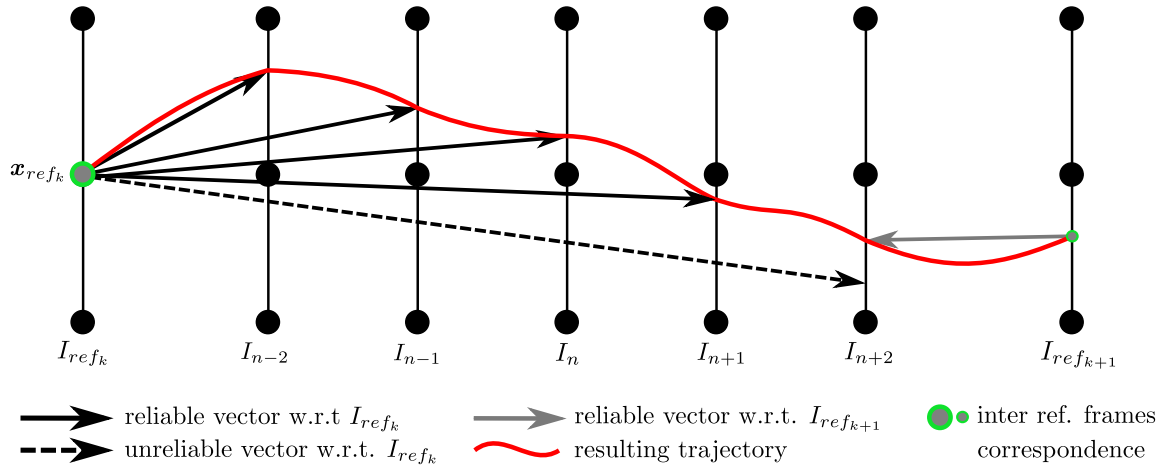


Figure 12.8: General concept of the proposed two-reference frames motion refinement framework: combination of reliable motion estimates defined from both reference frames.

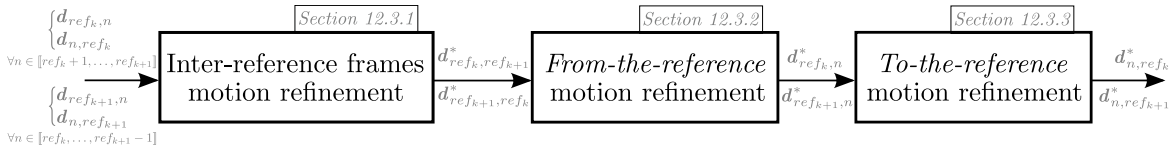


Figure 12.9: The three stages of the proposed two-reference frames motion refinement framework: 1) inter-reference frames motion refinement (Section 12.3.1), 2) *from-the-reference* motion refinement (Section 12.3.2), 3) *to-the-reference* motion refinement (Section 12.3.3).

We assume that we have also occlusion information which indicates which pixels of I_{ref_k} and $I_{ref_{k+1}}$ become occluded in I_n $\forall n$ as well as binary inconsistency information [RTDC12] which denotes the intrinsic quality of each one of the vectors belonging to the previously described input displacement fields.

Refining *forward* trajectories using *backward* trajectories requires as first step an inter-reference frame motion processing. As described in Section 12.3.1, this first step aims at establishing accurate dense inter-reference frame correspondences, i.e. correspondences between the reference frames I_{ref_k} and $I_{ref_{k+1}}$. The idea here is to benefit from the long-term dense motion estimations which have been already performed with respect to both reference frames I_{ref_k} and $I_{ref_{k+1}}$ to improve the inter-reference frames correspondences.

The resulting improved inter-reference frames correspondences are then used to refine intermediate displacement fields, i.e. *from-the-reference* displacement fields $\boldsymbol{d}_{ref_k,n}$ and $\boldsymbol{d}_{ref_{k+1},n}$ as well as the *to-the-reference* displacement fields $\boldsymbol{d}_{n,ref_k}$ and $\boldsymbol{d}_{n,ref_{k+1}}$ with $n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$. More precisely, the improved inter-reference frames correspondences can be used to identify for each *forward* trajectory $\boldsymbol{T}$ starting from $\boldsymbol{x}_{ref_k} \in I_{ref_k}$ which *backward* trajectory starting from $I_{ref_{k+1}}$ must be involved for the refinement of $\boldsymbol{T}$. This trajectory refinement task, which can be also defined as a *from-the-reference* motion refinement task (since the trajectories are made of *from-the-reference* displacement vectors) is presented in Section 12.3.2. In the same spirit, we can refine *to-the-reference* displacement fields by relying on the improved inter-reference frames correspondences. This last point is described in Section 12.3.3.

These three stages (inter-reference frames motion refinement, *from-the-reference* motion refinement, *to-the-reference* motion refinement) form the proposed two-reference frames motion refinement framework. They are summarized into the diagram displayed in Fig. 12.9.

The explanations in Sections 12.3.1 and 12.3.2 focus respectively on the establishment of *forward* inter-reference dense correspondences and on the refinement of *forward* trajectories $\boldsymbol{T}$ starting from $\boldsymbol{x}_{ref_k}$ of I_{ref_k} where $\boldsymbol{T}$ is defined through the set of *from-the-reference* displacement vectors $\boldsymbol{d}_{ref_k,n}$ $\forall n \in [\![ref_k + 1, \ldots, ref_{k+1}]\!]$. Note that an exactly similar processing can refine the trajectories running *backward* from $I_{ref_{k+1}}$ after having computed *backward* inter-reference frames dense correspondences, i.e. between $I_{ref_{k+1}}$ and I_{ref_k}. Section 12.3.3 shows how to perform a combined refinement, i.e. how to refine simultaneously *to-the-reference* vectors $\boldsymbol{d}_{n,ref_k}$ and $\boldsymbol{d}_{n,ref_{k+1}}$ $\forall n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$.

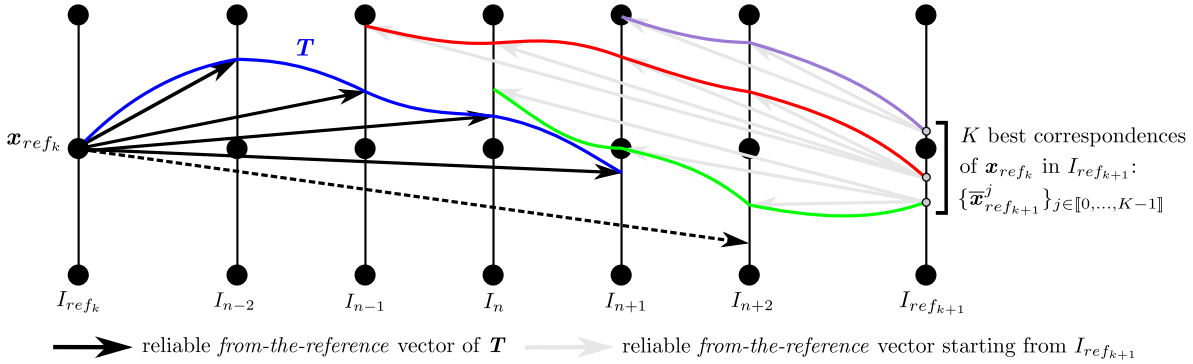


Figure 12.10: Overview of the inter-reference frames motion refinement: the establishment of improved inter-reference correspondences is based on trajectory similarity criteria between the *forward* trajectories and the candidate *backward* ones. The final inter-reference frames correspondences link each $\boldsymbol{x}_{ref_k}$ of I_{ref_k} to the starting point of the *backward* trajectory starting from $I_{ref_{k+1}}$ which is the more compatible with the *forward* trajectory starting from $\boldsymbol{x}_{ref_k}$.

12.3.1 Inter-reference frames motion refinement

The inter-reference frames motion refinement stage, whose general concept is illustrated in Fig. 12.10, aims at improving the motion correspondences between I_{ref_k} and $I_{ref_{k+1}}$. Toward this goal, three steps are involved:

1. construction of multiple inter-reference frames correspondences (Section 12.3.1.1),
2. selection of the K best inter-reference frames candidates among all the generated ones (Section 12.3.1.2),
3. for each *forward* trajectory $\boldsymbol{T}$ starting from $\boldsymbol{x}_{ref_k}$ (blue trajectory in Fig. 12.10), identification of the corresponding *backward* trajectory starting from $I_{ref_{k+1}}$ among the *backward* trajectories starting from each one of the K best correspondences of $\boldsymbol{x}_{ref_k}$ in $I_{ref_{k+1}}$ (i.e. among the green, red and purple trajectories of Fig. 12.10). This step, described in Section 12.3.1.3, is performed through a new trajectory-based global optimization method.

At the end of these three steps, we obtain an improved inter-reference frames dense matching by establishing motion correspondences between each pixel $\boldsymbol{x}_{ref_k}$ and the starting point of the selected corresponding *backward* trajectory starting from $I_{ref_{k+1}}$ (Fig. 12.10). These three steps are described in detail in the next sub-sections.

Notice that in a more simple version of the algorithm, the process is composed of the two first steps and the selection in step 2 is limited to $K = 1$. This version is dedicated to motion estimation improvement between the two reference frames only.

12.3.1.1 Generation of inter-reference frames motion candidates

For each grid point $\boldsymbol{x}_{ref_k}$ of the reference frame I_{ref_k}, we define a set $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ of K_{ref_k} candidate positions $\boldsymbol{x}^i_{ref_{k+1}}$ $\forall i \in [\![0,\ldots,K_{ref_k}-1]\!]$ in $I_{ref_{k+1}}$: $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k}) = \{\boldsymbol{x}^i_{ref_{k+1}}\}_{i\in[\![0,\ldots,K_{ref_k}-1]\!]}$. These candidate positions are obtained via concatenation of two displacement fields: *from-the-reference* displacement fields $\boldsymbol{d}_{ref_k,n}$ estimated with respect to I_{ref_k} and *to-the-reference* displacement fields $\tilde{\boldsymbol{d}}_{n,ref_{k+1}}$ computed with respect to $I_{ref_{k+1}}$ with $n \in [\![ref_k+1,\ldots,ref_{k+1}-1]\!]$,

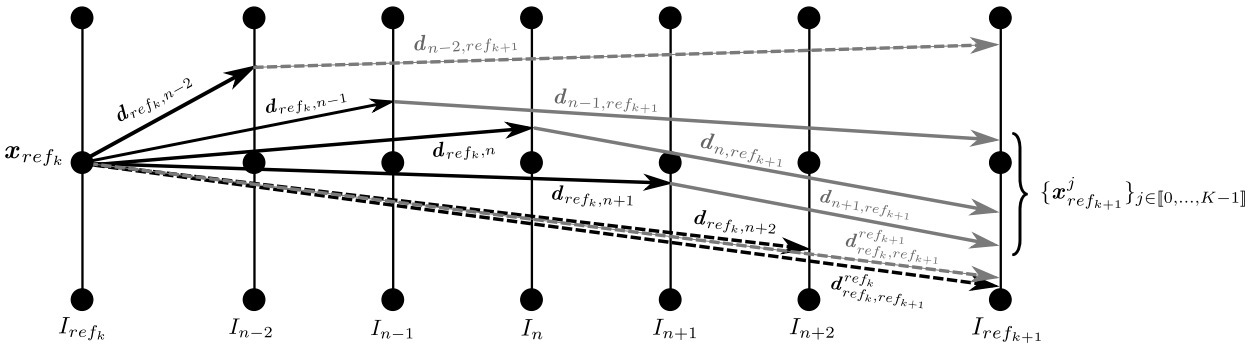


Figure 12.11: Inter-reference frames motion refinement: construction of motion *paths* made of un-occluded and consistent displacement vectors computed with respect to both reference frames $\boldsymbol{d}_{ref_k,n}$ (in black) and $\boldsymbol{d}_{n,ref_{k+1}}$ (in grey). These motion *paths* give a set of correspondences for each grid point $\boldsymbol{x}_{ref_k}$ of I_{ref_k} in $I_{ref_{k+1}}$. Solid lines indicate consistent and un-occluded vectors. Conversely, dashed lines denote inconsistent or occluded ones.

as shown in Eq. 12.4. The notation $\tilde{.}$ indicates a displacement field probably interpolated at a non-grid position. These two *forward* displacement fields form *forward* motion *paths*.

$$\begin{aligned} \boldsymbol{x}^n_{ref_{k+1}} = \boldsymbol{x}_{ref_k} \quad &+ \quad \boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k}) \\ &+ \quad \tilde{\boldsymbol{d}}_{n,ref_{k+1}}(\boldsymbol{x}_{ref_k} + \boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k})) \end{aligned} \tag{12.4}$$

A candidate position is added into $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ only if the corresponding *forward* motion *path* is made of two displacement vectors ($\boldsymbol{d}_{ref_k,n}$ and $\boldsymbol{d}_{n,ref_{k+1}}$) un-occluded and intrinsically consistent [RTDC12]. This criterion demonstrates the reliability of the displacement vectors. Let $\boldsymbol{d}^{ref_k}_{ref_k,ref_{k+1}}$ and $\boldsymbol{d}^{ref_{k+1}}_{ref_k,ref_{k+1}}$ be the direct input displacement fields respectively computed with respect to I_{ref_k} (*from-the-reference*) and $I_{ref_{k+1}}$ (*to-the-reference*). The direct displacement vectors $\boldsymbol{d}^{ref_k}_{ref_k,ref_{k+1}}$ and $\boldsymbol{d}^{ref_{k+1}}_{ref_k,ref_{k+1}}$ starting from $\boldsymbol{x}_{ref_k}$ are also used to propose new candidates to $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ if they are un-occluded and consistent.

Similarly to the combinatorial *multi-step* integration procedure described in Section 11.1.1, *backward* motion paths from the reference frame $I_{ref_{k+1}}$ can be also used to enrich the candidate set $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$. These new candidates are obtained through concatenation of *from-the-reference* $\boldsymbol{d}_{ref_{k+1},n}$ displacement fields and *to-the-reference* $\tilde{\boldsymbol{d}}_{n,ref_k}$ displacement fields with $n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$. Then, the resulting *backward* motion *paths* are inverted to become *forward* motion *paths*. As for *forward* motion paths, these *backward* motion paths must be built with two intrinsically un-occluded and consistent displacement vectors. As previously, the direct un-occluded and consistent displacement vectors $\boldsymbol{d}^{ref_k}_{ref_{k+1},ref_k}$ and $\boldsymbol{d}^{ref_{k+1}}_{ref_{k+1},ref_k}$ can be considered to provide additional candidates to $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ after inversion.

This procedure of candidates generation, illustrated in Fig. 12.11, proposes different possible inter reference frames correspondences for each grid point $\boldsymbol{x}_{ref_k}$ of I_{ref_k}. For clarity, only *forward* motion paths are displayed in Fig. 12.11. Solid lines indicate un-occluded and consistent vectors. Conversely, dashed lines denote occluded or inconsistent vectors to be refined.

12.3.1.2 Selection of inter-reference frames motion candidates

Once all these different possible inter-reference frames correspondences have been built, we aim at selecting the K best candidates $\{\overline{\boldsymbol{x}}^j_{ref_{k+1}}\} \in \boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ with $j \in [\![0, \ldots, K-1]\!]$.

The selection of the K best candidates $\{\overline{\boldsymbol{x}}^j_{ref_{k+1}}\}_{j\in[\![0,\ldots,K-1]\!]}$ is based on the statistical processing of Section 11.1.2. This translates in performing the median minimization criterion (Eq. 11.6) in order to choose the best candidates according to both statistics and intrinsic quality assessment.

In a first version of the algorithm, the best candidate ($K = 1$) is selected. At this stage, a new *forward* motion field linking the reference frames I_{ref_k} and $I_{ref_{k+1}}$ has been obtained that is generally more accurate than the one estimated via single-reference frame methods such as *multi-step flow fusion* (Chapter 10) or *statistical multi-step flow* (Chapter 11). In the same way, the *backward* motion field linking the reference frame $I_{ref_{k+1}}$ to I_{ref_k} can be improved with, in addition, the new *forward* motion field as input.

We propose to go further to improve the robustness of the selection process. $K > 1$ best candidates are first selected and then the selection among these K candidates is made via similarity distance of the *backward* trajectories starting from these candidates with the *forward* trajectory. The more robust process is described in the next section, Section 12.3.1.3.

For $K = 1$, the selection of the best candidate is more precisely performed via statistical processing followed by a global optimization method in order to involve spatial regularization into the process. Thus, as described in Section 11.1 (Chapter 11), $N_{opt} < K_{ref_k}$ candidates are selected through statistical processing and then merged by pairs via global optimization to obtain the best one. In the case $K > 1$, only the statistical processing is involved to select the K best candidates because the rest of the proposed robust procedure (Section 12.3.1.3) includes itself a regularization process.

12.3.1.3 Selection of the best inter-reference frames correspondences via similarity between *forward* and *backward* trajectories

Once the K inter reference frames correspondences among $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ have been obtained, we propose to select the best correspondence via the comparison of the *forward* trajectory with the *backward* trajectories starting from these selected points. This method selects among the K *backward* trajectories $\boldsymbol{T}_j \; \forall j \in [\![0, \ldots, K-1]\!]$ starting from $\{\overline{\boldsymbol{x}}^j_{ref_{k+1}}\}_{j\in[\![0,\ldots,K-1]\!]}$ in $I_{ref_{k+1}}$ (i.e. among the green, red and purple *backward* trajectories of Fig. 12.10) the *backward* trajectory which is more compatible with $\boldsymbol{T}$ (i.e. the blue *forward* trajectory of Fig. 12.10), the current trajectory running *forward* from $\boldsymbol{x}_{ref_k} \in I_{ref_k}$ to be refined. In other words, we aim at identifying two pieces of trajectory (usually called *tracklets*), one *forward* from I_{ref_k} and one *backward* from $I_{ref_{k+1}}$, which translate the motion behavior of the same physical point.

Therefore, starting from the selection of the K correspondences for each pixel $\boldsymbol{x}_{ref_k}$ of I_{ref_k}, K *backward* trajectory fields are built. Then, the selection of a final *backward* trajectory field is carried out via a trajectory-based global optimization method described below.

At the end of this trajectory identification task, the final inter-reference frames correspondences will link each $\boldsymbol{x}_{ref_k}$ of I_{ref_k} to the starting point of the selected corresponding *backward* trajectory starting from $I_{ref_{k+1}}$. In addition, note that the *backward* trajectory which is identi-

fied among $\{\boldsymbol{T}_j\}_{j\in[\![0,\ldots,K-1]\!]}$ as the most similar compared to $\boldsymbol{T}$ will be used to refine $\boldsymbol{T}$ during the *from-the-reference* refinement stage, Section 12.3.2.

To identify the most similar *backward* trajectories, our global optimization method fuses pair by pair the K *backward* trajectory fields. Each of these candidate trajectories is characterized by a set of positions defined for each frame n of the temporal interval $[I_{ref_k}, I_{ref_{k+1}}]$ and defined as follows: $\boldsymbol{T}_j.\boldsymbol{x}(n) = \overline{\boldsymbol{x}}^j_{ref_{k+1}} + \boldsymbol{d}_{ref_{k+1},n}(\overline{\boldsymbol{x}}^j_{ref_{k+1}})$. Occlusion information as well as inconsistency information assigned respectively to each position $\boldsymbol{T}_j.\boldsymbol{x}(n)$ and displacement vector $\boldsymbol{T}_j.\boldsymbol{d}(n) = \boldsymbol{d}_{ref_{k+1},n}(\overline{\boldsymbol{x}}^j_{ref_{k+1}})$ are also considered. In order to fuse two *backward* trajectory fields, our selection criteria quantifies and compares the similarity of each of these *backward* trajectories with respect to the current *forward* trajectory $\boldsymbol{T}$. Analogously to $\boldsymbol{T}_j$, note that $\boldsymbol{T}$ is also defined by sets of positions $\boldsymbol{T}.\boldsymbol{x}(n) = \boldsymbol{x}_{ref_k} + \boldsymbol{T}.\boldsymbol{d}(n)$ with $\boldsymbol{T}.\boldsymbol{d}(n) = \boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k})$, occlusion and inconsistency information.

Let us describe how we fuse two *backward* trajectory fields and more precisely what are the similarity criteria with respect to *forward* trajectories involved for this task. We introduce $L = \{l_{\boldsymbol{x}_{ref_k}}\}$ as a labelling of all the pixels $\boldsymbol{x}_{ref_k}$ of I_{ref_k} where each label indicates a candidate *backward* trajectory $\boldsymbol{T}_j$ with $j \in [\![0,\ldots,K-1]\!]$. We introduce the energy defined in Eq. 12.5 (detailed in Eq. 12.6 and Eq. 12.10) and we minimize it with respect to L via *fusion moves* [LRR08, LRRB10]. λ makes the balance between the data term and the regularization term.

$$E_{ref_k,ref_{k+1}}(L) = \sum_{\boldsymbol{x}_{ref_k}} \epsilon_{data} + \lambda \sum_{<\boldsymbol{x}_{ref_k},\boldsymbol{y}_{ref_k}>} \epsilon_{reg} \tag{12.5}$$

This energy gives an energy value to each candidate *backward* trajectory $\boldsymbol{T}_j$. The most similar *backward* trajectory with respect to the current *forward* trajectory will be the one with the lowest energy value.

The data term, Eq. 12.6, involves $\alpha(n)$ which is equal to 1 when both $\boldsymbol{T}_j.\boldsymbol{d}(n)$ and $\boldsymbol{T}.\boldsymbol{d}(n)$ are consistent (and non-occluded) and 0 otherwise. Conversely, $\beta(n)$ equals 1 when $\boldsymbol{T}_j.\boldsymbol{d}(n)$ is consistent (and non-occluded) whereas $\boldsymbol{T}.\boldsymbol{d}(n)$ is inconsistent (and non-occluded), 0 otherwise.

$$\epsilon_{data} = \frac{1}{N_{ref_k,ref_{k+1}}}.[\sum_{n=ref_k+1}^{ref_{k+1}} \alpha(n).\beta(n).\epsilon_{inc}(n) + (1-\alpha(n)).\beta(n).\epsilon_{MC}(n)] \tag{12.6}$$

For frames n for which $\boldsymbol{T}_j.\boldsymbol{d}(n)$ and $\boldsymbol{T}.\boldsymbol{d}(n)$ are consistent (and non-occluded) (i.e. $\alpha(n) = 1$), we compute the *euclidean* distance $ed(n)$ between the positions $\boldsymbol{T}_j.\boldsymbol{x}(n)$ and $\boldsymbol{T}.\boldsymbol{x}(n)$ as described in Eq. 12.7. This quantifies the similarity between the two trajectories at common consistent instants (n and $n+1$ in the example Fig. 12.12).

$$\epsilon_{inc}(n) = \rho_d(ed(n)) = \rho_d(\|\boldsymbol{T}_j.\boldsymbol{x}(n) - \boldsymbol{T}.\boldsymbol{x}(n)\|_2) \tag{12.7}$$

In the second situation, i.e. $\boldsymbol{T}_j.\boldsymbol{d}(n)$ consistent (and non-occluded) and $\boldsymbol{T}.\boldsymbol{d}(n)$ inconsistent (and non-occluded) ($\beta(n) = 1$), we want to know if it is relevant to replace $\boldsymbol{T}.\boldsymbol{x}(n)$ by $\boldsymbol{T}_j.\boldsymbol{x}(n)$, as for frames I_{n+2} and $I_{ref_{k+1}}$ in the example Fig. 12.12. For this task, we define a matching cost with respect to both reference frames, as shown in Eq. 12.8. Thus, we compute the mean of the matching cost between the starting point of $\boldsymbol{T}$, $\boldsymbol{x}_{ref_k} \in I_{ref_k}$ and the position $\boldsymbol{T}_j.\boldsymbol{x}(n)$

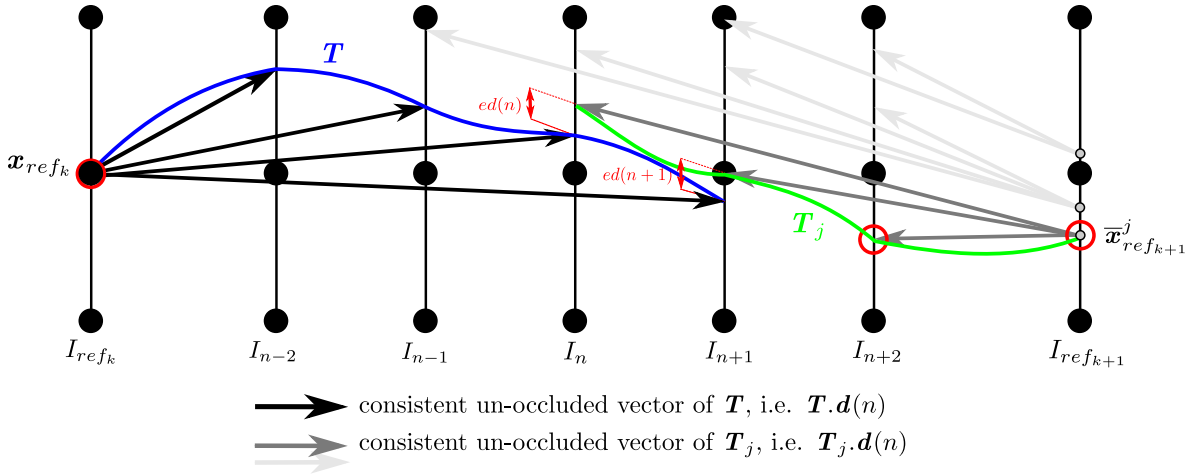


Figure 12.12: Inter-reference frames motion refinement: illustration of the similarity criteria involved for the choice of the best candidate *backward* trajectories with respect to the current *forward* trajectory $\boldsymbol{T}$ during the trajectory-based global optimization method (Section 12.3.1.3). Trajectory similarities are computed via *euclidean* distances between end-point displacement vectors (red segments) and matching costs with respect to both reference frames (red circles).

and the matching cost between the starting point of $\boldsymbol{T}_j$ in $I_{ref_{k+1}}$ ($\boldsymbol{T}_j.\boldsymbol{x}(ref_{k+1})$, which can be also written $\overline{\boldsymbol{x}}^j_{ref_{k+1}}$) and the position $\boldsymbol{T}_j.\boldsymbol{x}(n)$.

$$\epsilon_{MC}(n) = \rho_d\left(\frac{C(\boldsymbol{x}_{ref_k}, \boldsymbol{T}.\boldsymbol{d}(n)) + C(\overline{\boldsymbol{x}}^j_{ref_{k+1}}, \boldsymbol{T}_j.\boldsymbol{d}(n))}{2}\right) \tag{12.8}$$

In Eq. 12.7 and Eq. 12.8, ρ_d corresponds to the *Geman-McClure* robust penalty function, as in [LRR08]. The data term is normalized by $N_{ref_k,ref_{k+1}}$ defined in Eq. 12.9. $N_{ref_k,ref_{k+1}}$ corresponds to the number of occurrence of both situations (i.e. $\alpha(n) = 1$ and $\beta(n) = 0$ or $\alpha(n) = 0$ and $\beta(n) = 1$) . Moreover, note that only *backward* trajectories with at least N_{cf} common frames with the current *forward* trajectory are considered.

$$N_{ref_k,ref_{k+1}} = \sum_{n=ref_k+1}^{ref_{k+1}} \alpha(n).\beta(n) + [1-\alpha(n)].\beta(n) \tag{12.9}$$

The regularization term, Eq. 12.10, involves the similarity in terms of displacement vector between the current inter-reference frame correspondence (from $\boldsymbol{x}_{ref_k}$) and the one of the neighboring pixels $\boldsymbol{y}_{ref_k}$ according to the 8-point connectivity. $\boldsymbol{T}_j^{\boldsymbol{y}_{rek_k}}$ is the current candidate corresponding to $\boldsymbol{y}_{ref_k}$ in $I_{ref_{k+1}}$. Note that $\boldsymbol{T}_j.\boldsymbol{x}(ref_{k+1})$, the starting point of $\boldsymbol{T}_j$ in $I_{ref_{k+1}}$, corresponds to $\overline{\boldsymbol{x}}^j_{ref_{k+1}}$. $\gamma_{\boldsymbol{x}_{ref_k},\boldsymbol{y}_{ref_k}}$ accounts for color similarity between $\boldsymbol{x}_{ref_k}$ and $\boldsymbol{y}_{ref_k}$. ρ_r corresponds to the negative log of a *Student-t* distribution [LRR08].

$$\epsilon_{reg} = \gamma_{\boldsymbol{x}_{ref_k},\boldsymbol{y}_{ref_k}}.\rho_r\left(\left\|(\boldsymbol{T}_j.\boldsymbol{x}(ref_{k+1}) - \boldsymbol{x}_{ref_k}) - (\boldsymbol{T}_j^{\boldsymbol{y}_{ref_k}}.\boldsymbol{x}(ref_{k+1}) - \boldsymbol{y}_{ref_k})\right\|_1\right) \tag{12.10}$$

To conclude, this global optimization method selects the best compatible *backward* trajectory field with respect to the *forward* trajectory field. In other words, this method is able to identify

among the *backward* trajectories starting from $\{\overline{\boldsymbol{x}}^{j}_{ref_{k+1}}\}_{j\in[\![0,\ldots,K-1]\!]}$ in $I_{ref_{k+1}}$ which one is more compatible with $\boldsymbol{T}$, the current *forward* trajectory.

This simple selection of the best correspondence in the previous step ($K = 1$) is in practice less robust than our approach. Indeed, it is better to use an inter-reference frames matching criterion which relies on the identification on two trajectories (one *forward* from I_{ref_k}, one *backward* from $I_{ref_{k+1}}$) describing the behavior of the same physical point than focusing only on two single positions (one in I_{ref_k}, one in $I_{ref_{k+1}}$). We claim that the trajectories involve more suitable information. Moreover, toward the goal of refining *forward* trajectories from I_{ref_k} using *backward* trajectories from $I_{ref_{k+1}}$ (Section 12.3.2), it seemed to us more appropriate to take into account such trajectory similarity features as soon as the inter-reference frames motion refinement is under study.

The *backward* trajectory selected for a given *forward* trajectory $\boldsymbol{T}$ will be referred to as $\boldsymbol{T}^{*}_{j}$ in Section 12.3.2. Through this trajectory mapping procedure, we succeed in refining the corresponding position of $\boldsymbol{x}_{ref_k}$ in $I_{ref_{k+1}}$. This position, referred to as $\boldsymbol{x}^{*}_{ref_{k+1}}$ in the following, corresponds to the starting point of the *backward* trajectory $\boldsymbol{T}^{*}_{j}$.

In terms of displacement fields, this inter-reference frames motion refinement stage produces the optimal inter-reference frames displacement field $\boldsymbol{d}^{*}_{ref_k,ref_{k+1}}$. An exactly similar processing from $I_{ref_{k+1}}$ to I_{ref_k} leads to the computation of $\boldsymbol{d}^{*}_{ref_{k+1},ref_k}$.

12.3.1.4 Limitation

If it is not possible to find any un-occluded and consistent motion *paths* from $\boldsymbol{x}_{ref_k} \in I_{ref_k}$ to $I_{ref_{k+1}}$, $\boldsymbol{S}_{ref_{k+1}}(\boldsymbol{x}_{ref_k})$ is empty, i.e. with $K_{ref_k} = 0$. Therefore, our method cannot refine the trajectory starting from $\boldsymbol{x}_{ref_k}$ since no inter-reference frame correspondence has been found. In other words, the proposed two-reference frames processing works only for areas which are visible in both reference frames. Two reasons can explain the absence of inter-reference frames motion correspondences:

- $\boldsymbol{x}_{ref_k}$ is occluded in $I_{ref_{k+1}}$,
- $\boldsymbol{x}_{ref_k}$ is un-occluded in $I_{ref_{k+1}}$ but the input displacement fields have not been able to perform the inter-reference frames matching.

12.3.2 *From-the-reference* motion refinement

This step concerns the refinement of the *forward* trajectory field (i.e. all the displacement fields between the reference frame I_{ref_k} and the other frames of the sequence) using the selected *backward* trajectory field (attached to the second reference frame $I_{ref_{k+1}}$).

Once the best inter-reference frames correspondence $\boldsymbol{x}^*_{ref_{k+1}} \in I_{ref_{k+1}}$ has been established for each $\boldsymbol{x}_{ref_k} \in I_{ref_k}$ using the previously described trajectory similarity criteria between *forward* and *backward* trajectories, we aim at refining the *forward* trajectory $\boldsymbol{T}$ starting from $\boldsymbol{x}_{ref_k}$ using $\boldsymbol{T}^*_j$, the corresponding *backward* trajectory starting from $\boldsymbol{x}^*_{ref_{k+1}}$ and identified in the previous section. In particular, we use the consistent displacement vectors of the *backward* trajectory $\boldsymbol{T}^*_j$ to refine the inconsistent displacement vectors of the current *forward* trajectory $\boldsymbol{T}$ (dashed black vector for instant $n+2$ in Fig. 12.10).

For a given frame pair $\{I_{ref}, I_n\}$ with $n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$, a global optimization method is used to perform for each $\boldsymbol{x}_{ref_k} \in I_{ref_k}$ the selection of the best correspondence in the current frame I_n between:

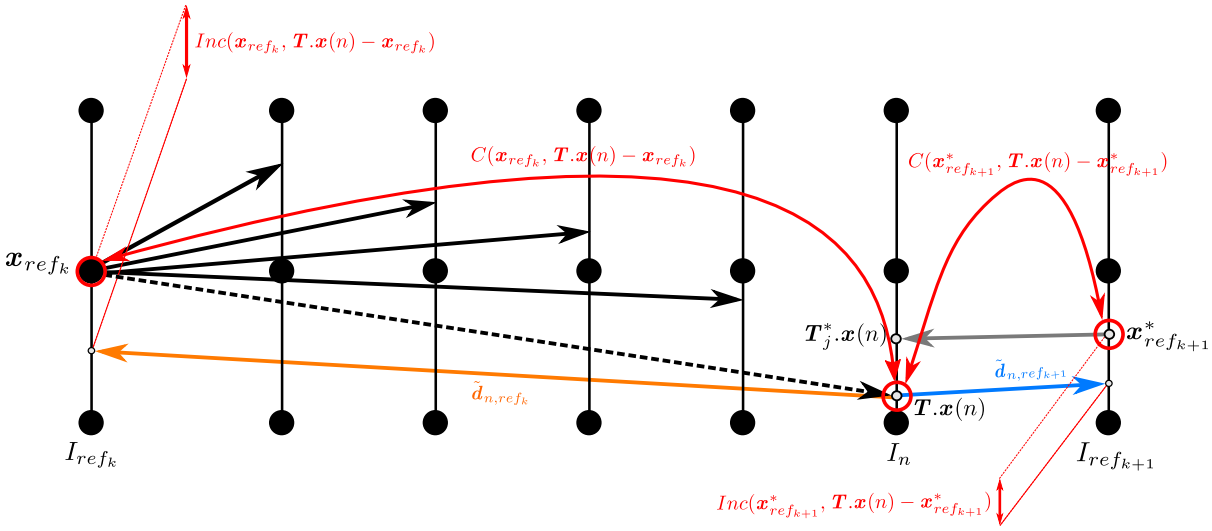
- $\boldsymbol{T}.\boldsymbol{x}(n)$ which means keeping the already existing position (i.e. the position of the *forward* trajectory $\boldsymbol{T}$),
- $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$, the position obtained via the *backward* trajectory $\boldsymbol{T}^*_j$.

Such choice is adressed only if $\boldsymbol{T}.\boldsymbol{d}(n)$ is inconsistent (and un-occluded) and $\boldsymbol{T}^*_j$ consistent (and un-occluded).

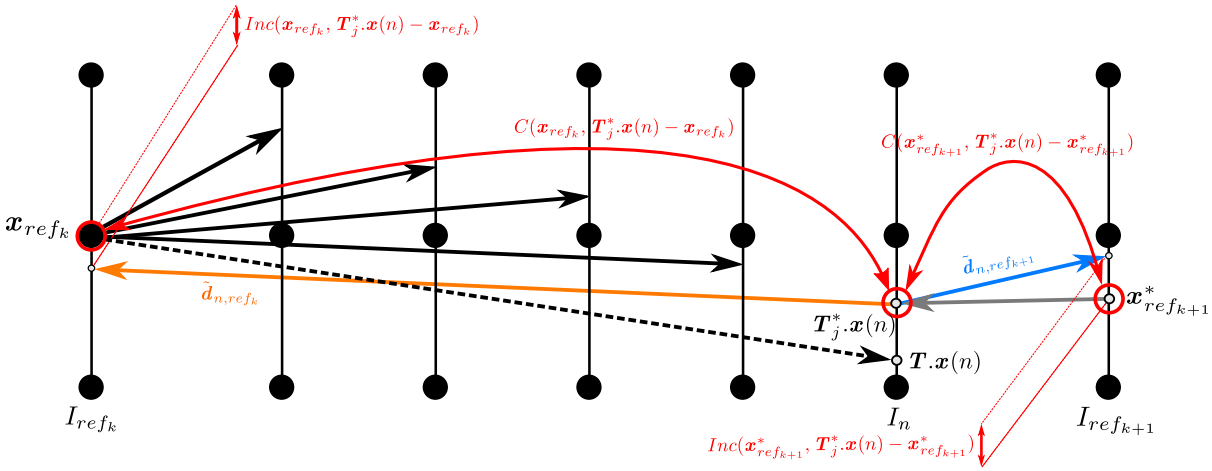
We introduce $L^b = \{l^b_{\boldsymbol{x}_{ref_k}}\} \subset \{l_0, l_1\}$ as a binary labelling of all the pixels $\boldsymbol{x}_{ref_k}$ of I_{ref_k} where the label l_0 indicates that we keep the already existing position $\boldsymbol{T}.\boldsymbol{x}(n)$ and where the label l_1 indicates that $\boldsymbol{T}.\boldsymbol{x}(n)$ has to be replaced with the position obtained via the *backward* trajectory $\boldsymbol{T}^*_j$. We introduce the energy defined from Eq. 12.11 to Eq. 12.17 for each pair $\{I_{ref}, I_n\}$ $\forall n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$ and as previously, we minimize it with respect to L^b via *fusion moves* [LRR08, LRRB10]. The trajectory under consideration ($\boldsymbol{T}$ or $\boldsymbol{T}^*_j$) is written generically $\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}$ in the following equations.

$$E_{ref,n}(L^b) = \sum_{\boldsymbol{x}_{ref_k}} \epsilon_{data} + \lambda \sum_{<\boldsymbol{x}_{ref_k}, \boldsymbol{y}_{ref_k}>} \epsilon_{reg} \text{ with } \epsilon_{data} = \frac{\rho_d(\epsilon_{MC,Inc})}{2} \tag{12.11}$$

The data term, Eq. 12.12, involves both matching costs and inconsistency values defined with respect to the two reference frames. This leads to four terms which are illustrated in Fig. 12.13 in the two possible situations: data term computation for label l_0 (i.e. with respect to $\boldsymbol{T}.\boldsymbol{x}(n)$) and for label l_1 (i.e. with respect to $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$). More precisely, two matching costs are estimated respectively between $\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n)$ and $\boldsymbol{x}_{ref_k}$ and between $\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n)$ and $\boldsymbol{x}^*_{ref_{k+1}}$. Moreover, we take into account the inconsistency of the two following displacement vectors: 1) $\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}_{ref_k}$ (which involves $\tilde{\boldsymbol{d}}_{n,ref_k}(\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n))$) and 2) $\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}^*_{ref_{k+1}}$ (which involves $\tilde{\boldsymbol{d}}_{n,ref_{k+1}}(\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n))$).



(a) Data term computation with respect to label l_0: we evaluate the quality of the matching between $\boldsymbol{x}_{ref_k}$ and $\boldsymbol{T}.\boldsymbol{x}(n)$.



(b) Data term computation with respect to label l_1: we evaluate the quality of the matching between $\boldsymbol{x}_{ref_k}$ and $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$.

Figure 12.13: *From-the-reference* motion refinement: the data term involved into the energy functional Eq. 12.11 is computed through both matching costs and inconsistency values defined with respect to the two reference frames I_{ref_k} and $I_{ref_{k+1}}$ (Eq. 12.12). Via this global optimization method, we aim at choosing between: 1) keeping the existing matching between I_{ref} and I_n (label l_0, *(a)*) or 2) updating it using the proposal coming from the *backward* trajectory (label l_1, *(b)*).

$$\begin{aligned}\epsilon_{MC,Inc} &= w_{ref_k}(n).C(\boldsymbol{x}_{ref_k}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}_{ref_k}) \\ &+ w_{ref_{k+1}}(n).C(\boldsymbol{x}^*_{ref_{k+1}}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}^*_{ref_{k+1}}) \\ &+ w_{ref_k}(n).Inc(\boldsymbol{x}_{ref_k}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}_{ref_k}) \\ &+ w_{ref_{k+1}}(n).Inc(\boldsymbol{x}^*_{ref_{k+1}}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}^*_{ref_{k+1}}) \end{aligned} \tag{12.12}$$

Following the definition of the inconsistency (Eq. 10.10, Chapter 10) and as shown in Fig. 12.13, the two terms $Inc(\boldsymbol{x}_{ref_k}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}_{ref_k})$ and $Inc(\boldsymbol{x}^*_{ref_{k+1}}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}^*_{ref_{k+1}})$ are computed as follows:

$$\begin{aligned} Inc(\boldsymbol{x}_{ref_k}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) &- \boldsymbol{x}_{ref_k}) = \\ ||\boldsymbol{x}_{ref_k} &- [\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) + \tilde{\boldsymbol{d}}_{n,ref_k}(\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n))]||_2 \end{aligned} \tag{12.13}$$

$$\begin{aligned} Inc(\boldsymbol{x}^*_{ref_{k+1}}, \boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) &- \boldsymbol{x}^*_{ref_{k+1}}) = \\ ||\boldsymbol{x}^*_{ref_{k+1}} &- [\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) + \tilde{\boldsymbol{d}}_{n,ref_{k+1}}(\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n))]||_2 \end{aligned} \tag{12.14}$$

The matching costs and the inconsistency values are weighted with respect to the temporal distances to the reference frames I_{ref_k} (w_{ref_k}, Eq. 12.15) and $I_{ref_{k+1}}$ ($w_{ref_{k+1}}$, Eq. 12.16) with $w_{ref_k}(n) + w_{ref_{k+1}}(n) = 1\ \forall n$. The respective weights are defined as follows:

$$w_{ref_k}(n) = \frac{n - ref_k}{ref_{k+1} - ref_k} \tag{12.15}$$

$$w_{ref_{k+1}}(n) = \frac{ref_{k+1} - n}{ref_{k+1} - ref_k} \tag{12.16}$$

As usual, the regularization term, Eq. 12.17, involves the similarity in terms of displacement vector between the current displacement vector from $\boldsymbol{x}_{ref_k}$ and the one of the neighboring pixels $\boldsymbol{y}_{ref_k}$ according to the 8-point connectivity. $\gamma_{\boldsymbol{x}_{ref_k},\boldsymbol{y}_{ref_k}}$ accounts for color similarity between $\boldsymbol{x}_{ref_k}$ and $\boldsymbol{y}_{ref_k}$.

$$\epsilon_{reg} = \gamma_{\boldsymbol{x}_{ref_k},\boldsymbol{y}_{ref_k}}.\rho_r\left(\left\|(\boldsymbol{T}^{l^b_{\boldsymbol{x}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{x}_{ref_k}) - (\boldsymbol{T}^{l^b_{\boldsymbol{y}_{ref_k}}}.\boldsymbol{x}(n) - \boldsymbol{y}_{ref_k})\right\|_1\right) \tag{12.17}$$

In the same spirit as in Section 10.3.6, a thresholding condition on the motion vectors between consecutive frames $\{I_{n-1}, I_n\}$ following Eq. 10.17 is added to the framework. More precisely, we rely on the positions selected for I_{n-1} and we compute the elementary motion to both candidate positions in I_n ($\boldsymbol{T}.\boldsymbol{x}(n)$ or $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$). If the elementary motion computed with $\boldsymbol{T}.\boldsymbol{x}(n)$ (resp. $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$) does not fulfill the motion range condition contrary to the one obtained with $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$ (resp. $\boldsymbol{T}.\boldsymbol{x}(n)$), $\boldsymbol{T}^*_j.\boldsymbol{x}(n)$ (resp. $\boldsymbol{T}.\boldsymbol{x}(n)$) is automatically chosen.

Depending on this thresholding condition and on the energy values obtained for label l_0 and l_1, we choose to keep the existing matching or to question it using the proposal coming from the *backward* trajectory. This global optimization is applied for each pair of frames $\{I_{ref_k}, I_n\}$ which allows to obtain new *from-the-reference* displacement fields and therefore new trajectories along the temporal interval $[I_{ref_k}, I_{ref_{k+1}}]$.

Finally, this *from-the-reference* motion refinement stage leads to the computation of the optimal displacement fields $\boldsymbol{d}^*_{ref_k,n}$ $\forall n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$. An exactly similar processing performed in the *backward* direction from $I_{ref_{k+1}}$ produces refined *from-the-reference* displacement fields $\boldsymbol{d}^*_{ref_{k+1},n}$.

12.3.3 *To-the-reference* motion refinement

Up to now, this two reference frames processing has performed an efficient dense matching between the reference frames I_{ref_k} and $I_{ref_{k+1}}$ (Section 12.3.1) and has been able to refine *forward* trajectories starting from I_{ref_k} within the temporal interval $[I_{ref_k}, I_{ref_{k+1}}]$ using *backward* trajectories starting from $I_{ref_{k+1}}$ (Section 12.3.2). The same processing has been applied in the *backward* direction, i.e. between $I_{ref_{k+1}}$ and I_{ref_k}. In terms of displacement vectors, this bidirectional framework has for the time being already produced:

- optimal inter-reference frames displacement fields: $\boldsymbol{d}^*_{ref_k,ref_{k+1}}$ and $\boldsymbol{d}^*_{ref_{k+1},ref_k}$ (Section 12.3.1),
- optimal *from-the-reference* displacement fields: $\boldsymbol{d}^*_{ref_k,n}$ and $\boldsymbol{d}^*_{ref_{k+1},n}$ (Section 12.3.2).

Let us now refine *to-the-reference* displacement fields with respect to both reference frames: $\boldsymbol{d}_{n,ref_k}$ and $\boldsymbol{d}_{n,ref_{k+1}}$. Via the input *to-the-reference* displacement fields, we have for each grid point $\boldsymbol{x}_n$ of I_n two initial matches:

- one between $\boldsymbol{x}_n \in I_n$ and $\boldsymbol{x}^0_{ref_k} = \boldsymbol{x}_n + \boldsymbol{d}_{n,ref_k}$ in I_{ref_k},
- one between $\boldsymbol{x}_n$ and $\boldsymbol{x}^0_{ref_{k+1}} = \boldsymbol{x}_n + \boldsymbol{d}_{n,ref_{k+1}}$ in $I_{ref_{k+1}}$.

We propose to use the consistent inter-reference frames correspondences established in Section 12.3.1 to create competing candidates to $\boldsymbol{x}^0_{ref_k}$ and $\boldsymbol{x}^0_{ref_{k+1}}$ (see Fig. 12.14):

- $\boldsymbol{x}^1_{ref_k} = \boldsymbol{x}^0_{ref_{k+1}} + \tilde{\boldsymbol{d}}^*_{ref_{k+1},ref_k}(\boldsymbol{x}^0_{ref_{k+1}})$ defined in I_{ref_k},
- $\boldsymbol{x}^1_{ref_{k+1}} = \boldsymbol{x}^0_{ref_k} + \tilde{\boldsymbol{d}}^*_{ref_k,ref_{k+1}}(\boldsymbol{x}^0_{ref_k})$ defined in $I_{ref_{k+1}}$.

To make the right decision among $\{\boldsymbol{x}^0_{ref_k}, \boldsymbol{x}^1_{ref_k}\}$ in I_{ref_k} and $\{\boldsymbol{x}^0_{ref_{k+1}}, \boldsymbol{x}^1_{ref_{k+1}}\}$ in $I_{ref_{k+1}}$, we introduce the concept of candidate *doublets*. The idea is to gather by pairs these candidates and then to perform the selection task for both reference frames simultaneously. In this context, a selection must be performed between the following candidate *doublets*: $\{\boldsymbol{x}^0_{ref_k}, \boldsymbol{x}^1_{ref_{k+1}}\}$ and $\{\boldsymbol{x}^1_{ref_k}, \boldsymbol{x}^0_{ref_{k+1}}\}$ (Fig. 12.14).

We introduce $L^{dbl} = \{l^{dbl}_{\boldsymbol{x}_n}\} \subset \{l_{0,1}, l_{1,0}\}$ as a labelling of all the pixels $\boldsymbol{x}_n$ of I_n. Each label corresponds to one of the previously described candidate *doublets*. More precisely, we assign to each label $l_{i,j}$ with $(i,j) \in \{(0,1),(1,0)\}$ the candidate *doublet* $\{\boldsymbol{x}^i_{ref_k}, \boldsymbol{x}^j_{ref_{k+1}}\}$ and therefore its corresponding displacement vector *doublet* $\{\boldsymbol{d}^i_{n,ref_k}(\boldsymbol{x}_n), \boldsymbol{d}^j_{n,ref_{k+1}}(\boldsymbol{x}_n)\}$ with $\boldsymbol{d}^i_{n,ref_k}(\boldsymbol{x}_n) = \boldsymbol{x}^i_{ref_k} - \boldsymbol{x}_n$ and $\boldsymbol{d}^j_{n,ref_{k+1}}(\boldsymbol{x}_n) = \boldsymbol{x}^j_{ref_{k+1}} - \boldsymbol{x}_n$. We propose to follow again a global optimization approach and more precisely, we involve the following energy to fuse these two possible candidate *doublets*:

$$E_n(L^{dbl}) = \sum_{\boldsymbol{x}_n} \epsilon_{data} + \lambda \sum_{<\boldsymbol{x}_n,\boldsymbol{y}_n>} \epsilon_{reg} \text{ with } \epsilon_{data} = \frac{\rho_d(\epsilon_{MC,||.||_2})}{2} \tag{12.18}$$

Following the approach pursued in Eq. 12.12, the data term in this context of *doublet fusion* involves matching costs (Fig. 12.15) defined with respect to I_{ref_k} (between $\boldsymbol{x}_n$ and $\boldsymbol{x}^i_{ref_k}$) and

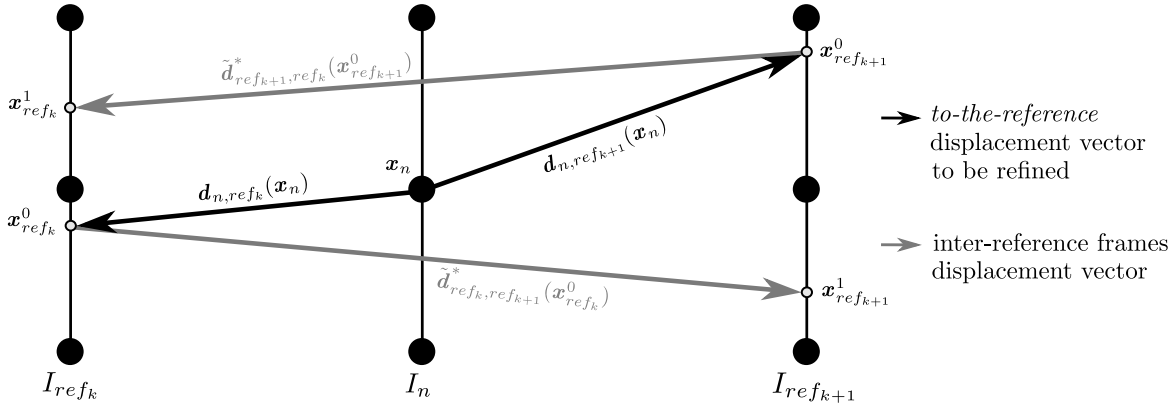


Figure 12.14: *To-the-reference* motion refinement: through the optimal inter-reference frames correspondences established in Section 12.3.1, we propose competing candidates ($\boldsymbol{x}^1_{ref_k}$ and $\boldsymbol{x}^1_{ref_{k+1}}$) to initial *to-the-reference* motion correspondences ($\boldsymbol{x}^0_{ref_k}$ and $\boldsymbol{x}^0_{ref_{k+1}}$) in the reference frames I_{ref_k} and $I_{ref_{k+1}}$.

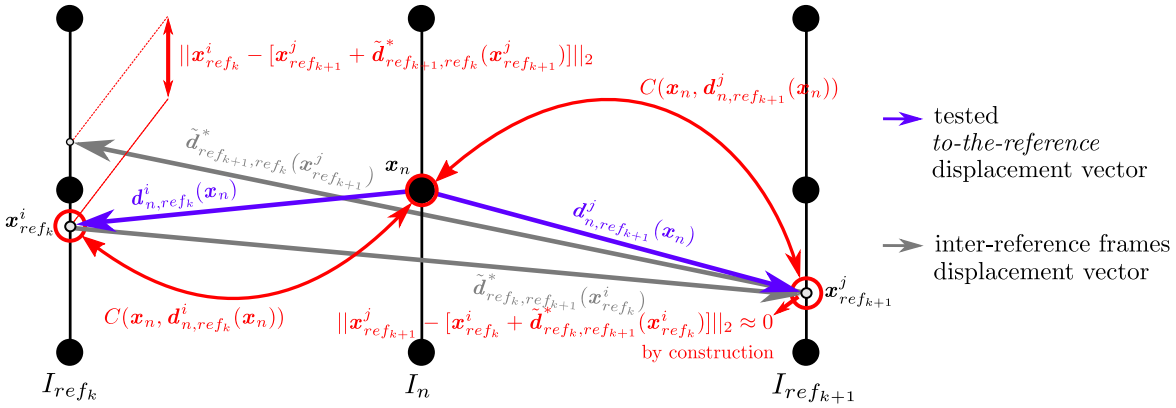


Figure 12.15: *To-the-reference* motion refinement: the data term (Eq. 12.19) of the proposed *doublet fusion* energy formulation involves matching costs defined with respect to I_{ref_k} (between $\boldsymbol{x}_n$ and $\boldsymbol{x}^i_{ref_k}$) and $I_{ref_{k+1}}$ (between $\boldsymbol{x}_n$ and $\boldsymbol{x}^j_{ref_{k+1}}$) as well as distance between $\boldsymbol{x}^i_{ref_k}$ (resp. $\boldsymbol{x}^j_{ref_{k+1}}$) and the ending point of the inter-reference frames vector $\tilde{\boldsymbol{d}}^*_{ref_{k+1},ref_k}$ (resp. $\tilde{\boldsymbol{d}}^*_{ref_k,ref_{k+1}}$) starting from $\boldsymbol{x}^j_{ref_{k+1}}$ (resp. $\boldsymbol{x}^i_{ref_k}$) in $I_{ref_{k+1}}$ (resp. I_{ref_k}).

$I_{ref_{k+1}}$ (between $\boldsymbol{x}_n$ and $\boldsymbol{x}^j_{ref_{k+1}}$). In addition, we compute also the distances (Fig. 12.15) between $\boldsymbol{x}^i_{ref_k}$ (resp. $\boldsymbol{x}^j_{ref_{k+1}}$) and the ending point of the inter-reference frames vector $\tilde{\boldsymbol{d}}^*_{ref_{k+1},ref_k}$ (resp. $\tilde{\boldsymbol{d}}^*_{ref_k,ref_{k+1}}$) starting from $\boldsymbol{x}^j_{ref_{k+1}}$ (resp. $\boldsymbol{x}^i_{ref_k}$) in $I_{ref_{k+1}}$ (resp. I_{ref_k}). Through these two distances, we aim at selecting a candidate *doublet* $\{\boldsymbol{x}^i_{ref_k}, \boldsymbol{x}^j_{ref_{k+1}}\}$ such as the *forward* inter-reference frames displacement vector starting from $\boldsymbol{x}^i_{ref_k}$ is the exact opposite of the *backward* inter-reference frames displacement vector starting from $\boldsymbol{x}^j_{ref_{k+1}}$. We manipulate here the notion of double consistency: the *backward* vector which is used to evaluate the consistency of a given *forward* vector must be intrinsically consistent itself. Schematically, we aim at obtaining a *closed triangle* when considering all the displacement vectors linking the three vertices $\{\boldsymbol{x}_n, \boldsymbol{x}^i_{ref_k}, \boldsymbol{x}^j_{ref_{k+1}}\}$. In this context, $\epsilon_{MC,||.||_2}$ is estimated for each candidate *doublet* as follows:

$$\begin{aligned}
\epsilon_{MC,||.||_2} &= w_{ref_k}(n).C(\boldsymbol{x}_n, \boldsymbol{d}^i_{n,ref_k}(\boldsymbol{x}_n)) \\
&+ w_{ref_{k+1}}(n).C(\boldsymbol{x}_n, \boldsymbol{d}^j_{n,ref_{k+1}}(\boldsymbol{x}_n)) \\
&+ w_{ref_k}(n).||\boldsymbol{x}^i_{ref_k} - [\boldsymbol{x}^j_{ref_{k+1}} + \tilde{\boldsymbol{d}}^*_{ref_{k+1},ref_k}(\boldsymbol{x}^j_{ref_{k+1}})]||_2 \\
&+ w_{ref_{k+1}}(n).||\boldsymbol{x}^j_{ref_{k+1}} - [\boldsymbol{x}^i_{ref_k} + \tilde{\boldsymbol{d}}^*_{ref_k,ref_{k+1}}(\boldsymbol{x}^i_{ref_k})]||_2
\end{aligned} \tag{12.19}$$

The regularization term, Eq. 12.20, involves two displacement similarity terms: 1) the similarity ϵ^{n,ref_k}_{reg} between the current displacement vector $\boldsymbol{d}^i_{n,ref_k}(\boldsymbol{x}_n)$ and the one from the neighboring pixels $\boldsymbol{y}_n$ according to the 8-point connectivity (Eq. 12.21), 2) the similarity $\epsilon^{n,ref_{k+1}}_{reg}$ between the current displacement vector $\boldsymbol{d}^j_{n,ref_{k+1}}(\boldsymbol{x}_n)$ and the one from the neighboring pixels $\boldsymbol{y}_n$. $\gamma_{\boldsymbol{x}_n,\boldsymbol{y}_n}$ accounts for color similarity between $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$.

$$\epsilon_{reg} = \gamma_{\boldsymbol{x}_n,\boldsymbol{y}_n}.\rho_r\left(w_{ref_k}(n).\epsilon^{n,ref_k}_{reg} + w_{ref_{k+1}}(n).\epsilon^{n,ref_{k+1}}_{reg}\right) \tag{12.20}$$

$$\epsilon^{n,ref_k}_{reg} = \left\|(\boldsymbol{d}^i_{n,ref_k}(\boldsymbol{x}_n) - \boldsymbol{d}^i_{n,ref_k}(\boldsymbol{y}_n))\right\|_1 \tag{12.21}$$

$$\epsilon^{n,ref_{k+1}}_{reg} = \left\|(\boldsymbol{d}^j_{n,ref_{k+1}}(\boldsymbol{x}_n) - \boldsymbol{d}^j_{n,ref_{k+1}}(\boldsymbol{y}_n))\right\|_1 \tag{12.22}$$

Finally, the whole proposed *doublet fusion* procedure is repeated for each pair of frames $\{I_{ref_k}, I_n\}$ in order to obtain refined *to-the-reference* displacement fields $\boldsymbol{d}^*_{n,ref_k}$ and $\boldsymbol{d}^*_{n,ref_{k+1}}$ $\forall n \in [\![ref_k + 1, \ldots, ref_{k+1} - 1]\!]$.

On the scale of the whole two-reference frames motion refinement framework (Fig. 12.9), the three previously described stages (inter-reference frames motion refinement, *from-the-reference* motion refinement, *to-the-reference* motion refinement) can be iteratively applied up to convergence or for a given number of times. Indeed, the *from-the-reference* and *to-the-reference* displacement fields which have been refined can be considered as inputs of the inter-reference frames motion refinement stage. Then, the resulting refined correspondences between I_{ref_k} and $I_{ref_{k+1}}$ can be used to improve again *from-the-reference* and *to-the-reference* displacement fields and so on.

12.4 Experimental evaluation of the two-reference refinement

In the previous section, Section 12.3, we described a new two-reference frames motion refinement made on three main steps. First, an inter-reference frames motion refinement (Section 12.3.1) which aims at establishing accurate correspondences between two distant reference frames I_{ref_k} and $I_{ref_{k+1}}$. Second, a *from-the-reference* motion refinement (Section 12.3.2) whose goal is to refine *from-the-reference* displacement fields between each reference frame and the intermediate frames. Third, a *to-the-reference* motion refinement (Section 12.3.3) which consists in correcting *to-the-reference* displacement fields between each intermediate frame and the reference frames.

In what follows, we propose to evaluate the performance of the two first steps. Thus, the inter-reference frames motion refinement is qualitatively assessed in Section 12.4.1 via the visualization of both inter-reference frames displacement maps and motion inter-reference frames correspondences. In Section 12.4.2, we study the performance of the *from-the-reference* motion refinement through label mask visualization, 2D+t trajectory visualization and point tracking. Finally, Section 12.4.3 tests the extension of the inter-reference frames motion refinement (Section 12.3.1) to the whole sequence (i.e. for each pair of frames $\{I_{ref_k}, I_n\}$ with $n \in [\![ref_k + 1, \ldots, ref_{k+1}]\!]$) through two video editing examples.

The experiments have mainly focused on temporary occlusion and illumination variations through the three following sequences: *Walking-Couple*-72-92 (Fig. 12.16) where the couple is temporary occluded by the foreground tree, *Walking-Couple*-61-72 (Fig. 12.17) where strong illumination changes occur and *MPI-S*1-25-55 [GKT+] (Fig. 11.5) where the background kiosk is temporary occluded by the woman.

Concerning parameter specification, $K = 3$ best inter-reference frames candidates are selected using the statistical processing (Section 12.3.1.2). The minimum number of common instants between *forward* and *backward* trajectories, N_{cf}, corresponds to 10% of the temporal distance between I_{ref_k} and $I_{ref_{k+1}}$ rounded to the nearest integer. Moreover, in Eq. 12.5 and Eq. 12.11, λ equals to 1. Finally, the color similarity term $\gamma_{\boldsymbol{x}_{ref_k}, \boldsymbol{y}_{ref_k}}$ involved in Eq. 12.10 and in Eq. 12.17 is computed following Eq. 10.5 (Chapter 10).

12.4.1 Evaluation of the inter-reference frames motion refinement

To evaluate the performance of the inter-reference frames motion refinement, we propose to visualize *forward* and *backward* inter-reference frames displacement maps for *Walking-Couple*-72-92 (Fig. 12.18) and *MPI-S*1-25-55 (Fig. 12.19). In particular, we compare the single-reference frame strategy based on *StatFlow*(*2D-DE*) performed from I_{72} (resp. I_{25}) and the inter-reference frames motion refinement proposed in Section 12.3.1 and based on *StatFlow*(*2D-DE*) achieved in *forward* from I_{72} (resp. I_{25}) and in *backward* from I_{92} (resp. I_{55}). 2D-DE [RTDC12] *multi-step* elementary *optical flow* fields have been used as inputs of both strategies with *steps* $1 - 20$ for *Walking-Couple*-72-92, $1 - 5$, 10, 15, 22, 24, 26, 28 and 30 for *MPI-S*1-25-55.

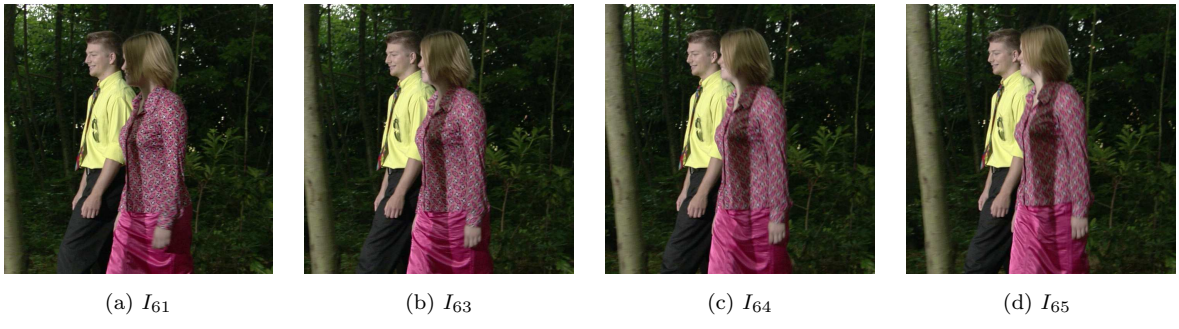
The visualization of the displacement fields is done in the HSV color space. In this standard representation, illustrated in Fig. 12.18 (a), the hue and the saturation indicate respectively the direction of the displacement and its magnitude.

In Fig. 12.18, we notice that the inter-reference frames refinement allows a better motion estimation of the left arm of the woman in both directions (*forward* in Fig. 12.18 (c) and *backward* in Fig. 12.18 (f) compared respectively to Fig. 12.18 (b) and Fig. 12.18 (e)). In

(a) I_{72} (b) I_{78} (c) I_{82}

(d) I_{86} (e) I_{88} (f) I_{92}

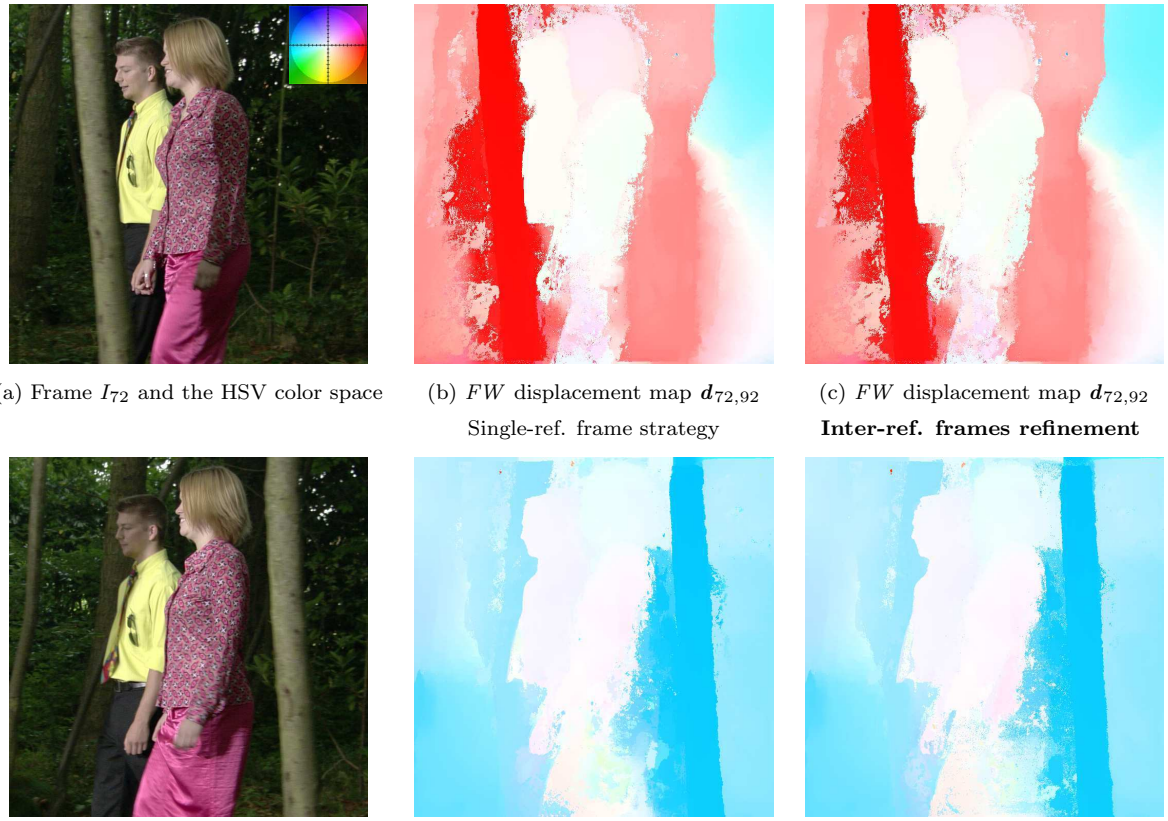
Figure 12.16: Source frames from a cropped version of the *Walking-Couple-72-92* sequence.



(a) I_{61} (b) I_{63} (c) I_{64} (d) I_{65}

(e) I_{67} (f) I_{69} (g) I_{71} (h) I_{72}

Figure 12.17: Source frames from a cropped version of the *Walking-Couple-61-72* sequence.



(a) Frame I_{72} and the HSV color space

(b) FW displacement map $\boldsymbol{d}_{72,92}$ Single-ref. frame strategy

(c) FW displacement map $\boldsymbol{d}_{72,92}$ **Inter-ref. frames refinement**

(d) Frame I_{92}

(e) BW displacement map $\boldsymbol{d}_{92,72}$ Single-ref. frame strategy

(f) BW displacement map $\boldsymbol{d}_{92,72}$ **Inter-ref. frames refinement**

Figure 12.18: Visualization of the *forward* (FW) and *backward* (BW) inter-reference frames displacement maps (*Walking-Couple-72-92* sequence). We compare: 1) a single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{72}, 2) the inter-reference frames motion refinement (first step of the two-reference frames motion refinement) proposed in Section 12.3.1 and based on *StatFlow*(*2D-DE*) performed in *forward* from I_{72} and in *backward* from I_{92}.

(a) Source frame I_{25} (b) *Forward* displacement map $\boldsymbol{d}_{25,55}$ Single-reference frame strategy (c) *Forward* displacement map $\boldsymbol{d}_{25,55}$ **Inter-reference frames refinement**

(d) Source frame I_{55} (e) *Backward* displacement map $\boldsymbol{d}_{55,25}$ Single-reference frame strategy (f) *Backward* displacement map $\boldsymbol{d}_{55,25}$ **Inter-reference frames refinement**

Figure 12.19: Visualization of the *forward* and *backward* inter-reference frames displacement maps (*MPI-S1-25-55* sequence). We compare: 1) a single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{25}, 2) the inter-reference frames motion refinement (first step of the two-reference frames motion refinement) proposed in Section 12.3.1 and based on *StatFlow*(*2D-DE*) performed in *forward* from I_{25} and in *backward* from I_{55}.

(a) Points in I_{25} (b) Matching from I_{25} to I_{55} Single-reference frame strategy (c) Matching from I_{25} to I_{55} **Inter-reference frames refinement**

Figure 12.20: Motion correspondences between I_{25} and I_{55} (*MPI-S1-25-55* sequence). We compare: 1) a single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{25}, 2) the inter-reference frames motion refinement (first step of the two-reference frames motion refinement) proposed in Section 12.3.1 and based on *StatFlow*(*2D-DE*) performed in *forward* from I_{25} and in *backward* from I_{55}.

addition, some spots within the woman shirt are erased by the proposed refinement which makes this area more homogeneous in term of displacement. We observe also that the segmentation of the foreground tree has been improved, especially when comparing Fig. 12.18 (*f*) and Fig. 12.18 (*e*) (i.e. in the *backward* direction).

The improvements allowed by the inter-reference frames motion refinement are more obvious for the *MPI-S*1-25-55 sequence (Fig. 12.19). When looking at the results, it appears clearly that the displacement estimates starting from the pixels which belong to the woman have been refined. Some of these displacement vectors do not have the background motion anymore as it was the case without the inter-reference frames refinement (see especially Fig. 12.19 (*f*) compared to Fig. 12.19 (*e*)). This gives a displacement map spatially smoother in this area for both directions (Fig. 12.19 (*c,f*) compared to Fig. 12.19 (*b,e*)).

As justified by the 11 motion correspondences illustrated in Fig. 12.20, some of the pixels located in the woman's hair (Fig. 12.20 (*a*)) were previously matched to the cardboard boxes placed on the kiosk due to strong illuminations changes (Fig. 12.20 (*b*)). These variations made the matching cost lower with respect to the cardboard boxes than with respect to the hair. With the proposed motion refinement stage, the corresponding displacement vectors now follow more accurately the woman displacement (Fig. 12.20 (*c*)). For illustration, note that the curve drawn by the 11 inputs points in Fig. 12.20 (*a*) approximately keeps the original layout in Fig. 12.20 (*c*) contrary to Fig. 12.20 (*b*). This has been permitted thanks to accurate inter-reference displacement vectors and robust *from-the-reference* displacement vectors from I_{55} which have been able to propose better matching proposals.

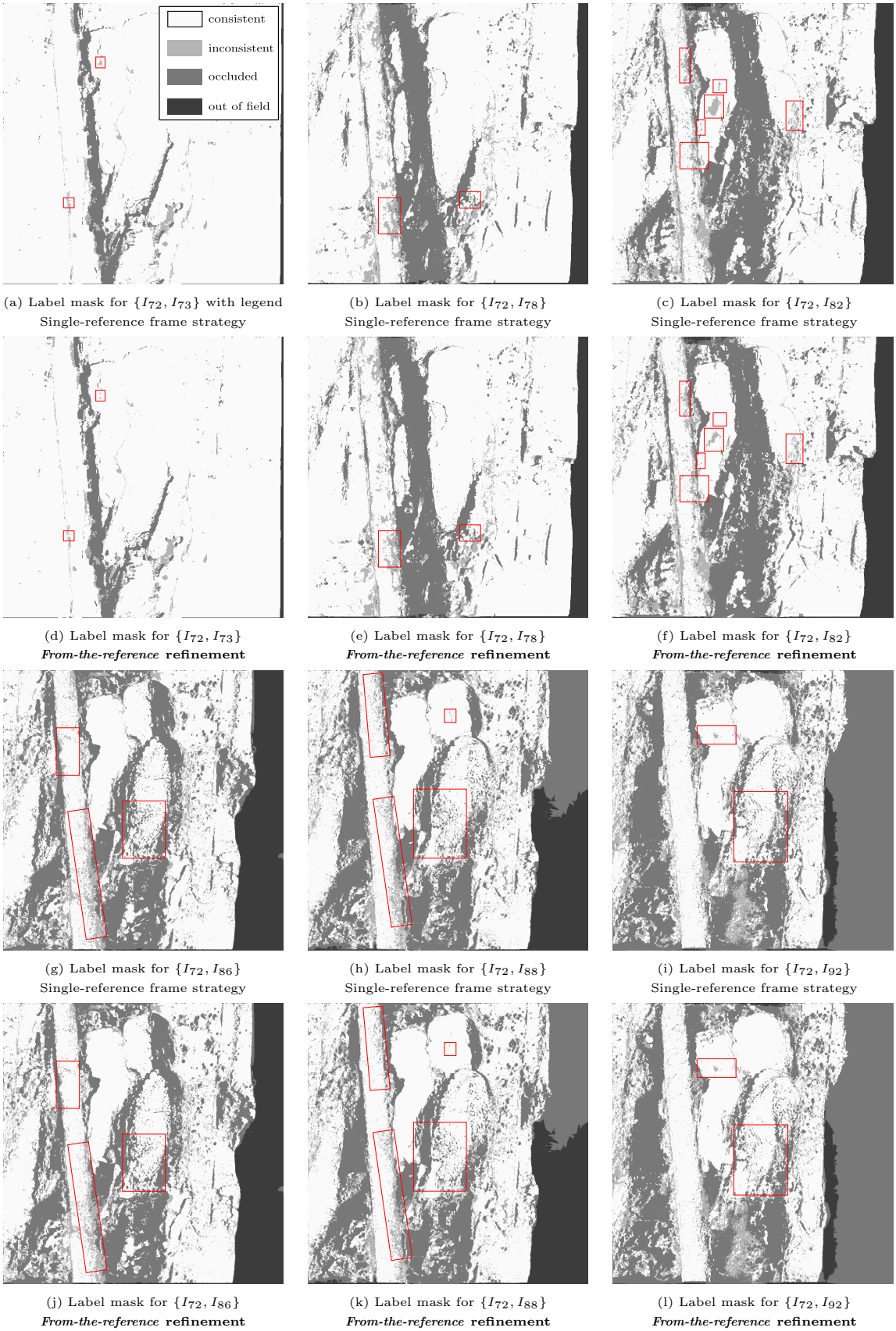
12.4.2 Evaluation of the *from-the-reference* motion refinement

Let us now provide the results obtained via the *from-the-reference* motion refinement described in Section 12.3.2. First of all, we suggest to compare the label masks obtained for the *Walking-Couple*-72-92 sequence with the single-reference frame strategy based on *StatFlow*(*2D-DE*) performed from I_{72} and the *from-the-reference* motion refinement proposed in Section 12.3.2 and based on *StatFlow*(*2D-DE*) achieved in *forward* from I_{72} and in *backward* from I_{92}.

The label masks displayed in Fig. 12.21 are associated to *from-the-reference* displacement fields $\boldsymbol{d}_{72,n}$ with $n = \{73, 78, 82, 86, 88, 92\}$ and indicate both occlusion and inconsistency. In particular, the label masks distinguish occluded pixels (out of frame or real occlusions) from visible pixels. This latter category is divided into consistent and inconsistent pixels, i.e. pixels for which the displacement vector is consistent/inconsistent. The legend in Fig. 12.21 (*a*) specifies the grayscale levels associated to each one of these categories. Additionally to this label information, the binary masks of Fig. 12.22 indicate in white the pixels of I_{72} for which the *from-the-reference* displacement vectors have been modified by the proposed *from-the-reference* motion refinement.

According to the results, it seems that the label masks have been slightly improved especially when the current frame I_n (of the pair $\{I_{ref_k}, I_n\}$) is near to the second reference frame $I_{ref_{k+1}} = I_{92}$. The tree and the shirts of the two characters are slightly more consistent in terms of displacement vectors. Red blocks of Fig. 12.21 focus on the differences between the label masks.

However, it appears that only a small subset of all the displacement vectors is actually refined by the *from-the-reference* motion refinement. This can be explained by the successive conditions which take place before the refinement itself. More precisely, let $\boldsymbol{x}_{ref_k}$ be a grid



(a) Label mask for $\{I_{72}, I_{73}\}$ with legend
Single-reference frame strategy

(b) Label mask for $\{I_{72}, I_{78}\}$
Single-reference frame strategy

(c) Label mask for $\{I_{72}, I_{82}\}$
Single-reference frame strategy

(d) Label mask for $\{I_{72}, I_{73}\}$
***From-the-reference* refinement**

(e) Label mask for $\{I_{72}, I_{78}\}$
***From-the-reference* refinement**

(f) Label mask for $\{I_{72}, I_{82}\}$
***From-the-reference* refinement**

(g) Label mask for $\{I_{72}, I_{86}\}$
Single-reference frame strategy

(h) Label mask for $\{I_{72}, I_{88}\}$
Single-reference frame strategy

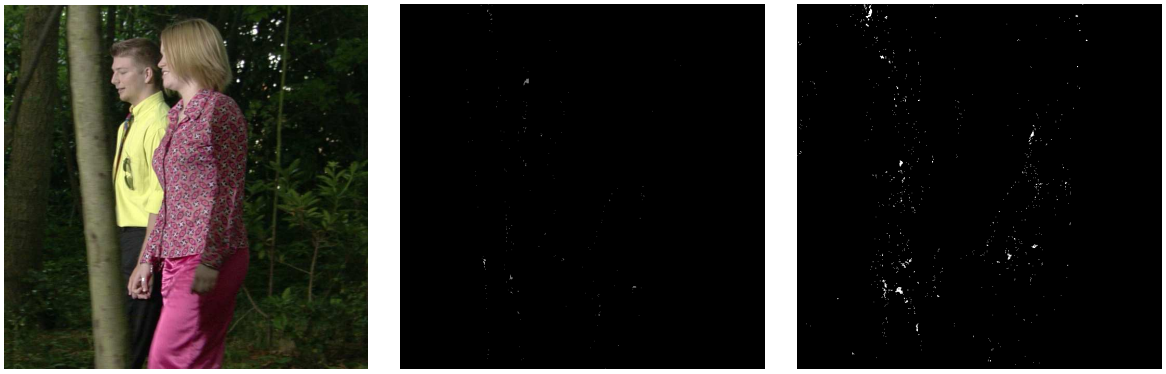
(i) Label mask for $\{I_{72}, I_{92}\}$
Single-reference frame strategy

(j) Label mask for $\{I_{72}, I_{86}\}$
***From-the-reference* refinement**

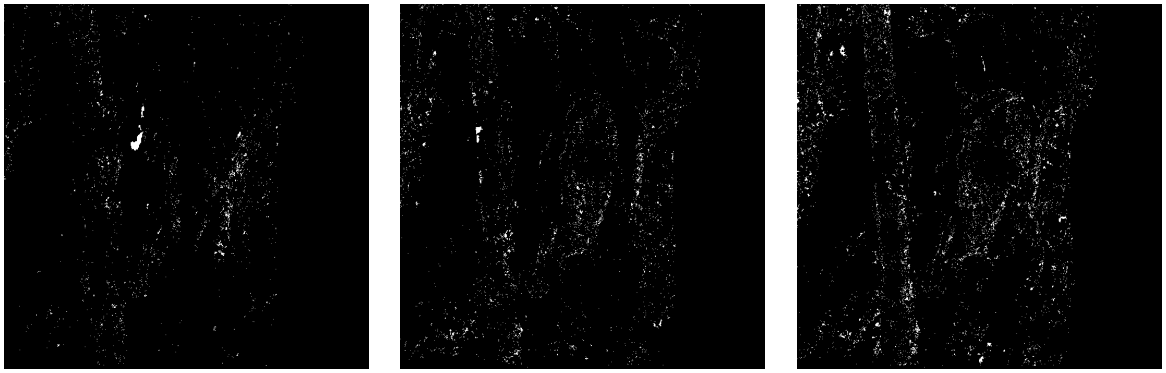
(k) Label mask for $\{I_{72}, I_{88}\}$
***From-the-reference* refinement**

(l) Label mask for $\{I_{72}, I_{92}\}$
***From-the-reference* refinement**

Figure 12.21: Label masks associated to *from-the-reference* displacement maps (*Walking-Couple-72-92* sequence). The label masks indicate both occlusion and inconsistency (see legend in *(a)*). We compare: 1) a single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{72}, 2) the *from-the-ref.* motion refinement (second step of the two-ref. frames motion refinement) proposed in Section 12.3.2 and based on *StatFlow*(*2D-DE*) achieved from I_{72} and I_{92}.

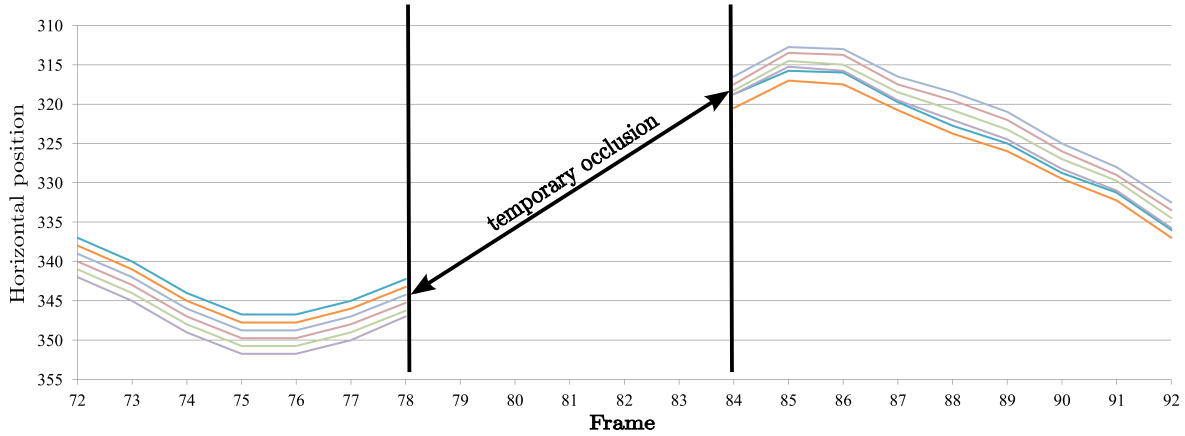


(a) Source frame I_{72} (b) Modified vectors, $\boldsymbol{d}_{72,73}$ (c) Modified vectors, $\boldsymbol{d}_{72,78}$

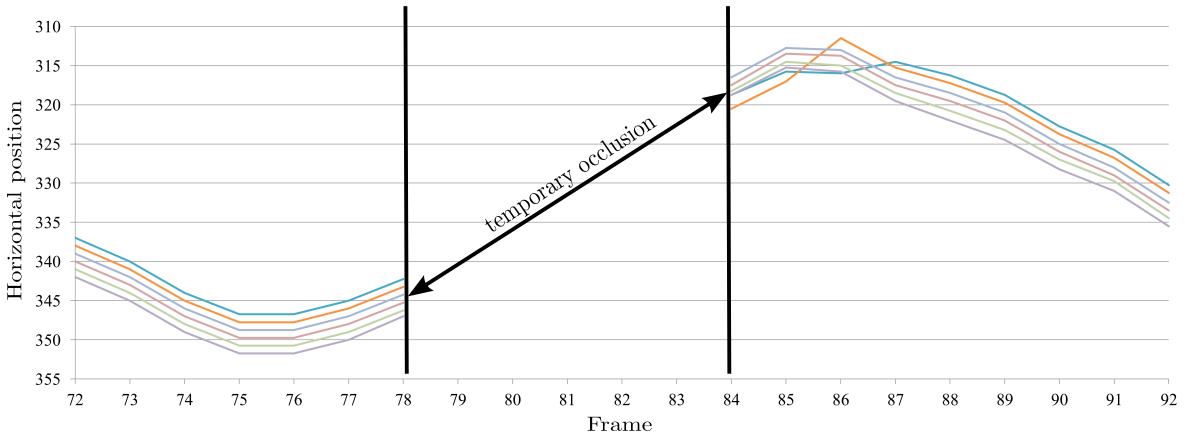


(d) Modified vectors, $\boldsymbol{d}_{72,82}$ (e) Modified vectors, $\boldsymbol{d}_{72,86}$ (f) Modified vectors, $\boldsymbol{d}_{72,88}$

Figure 12.22: Pixels of I_{72} (*Walking-Couple-72-92* sequence) for which the *from-the-reference* displacement vector has been modified by the *from-the-reference* motion refinement described in Section 12.3.2 and involved in the two-reference frames motion refinement.

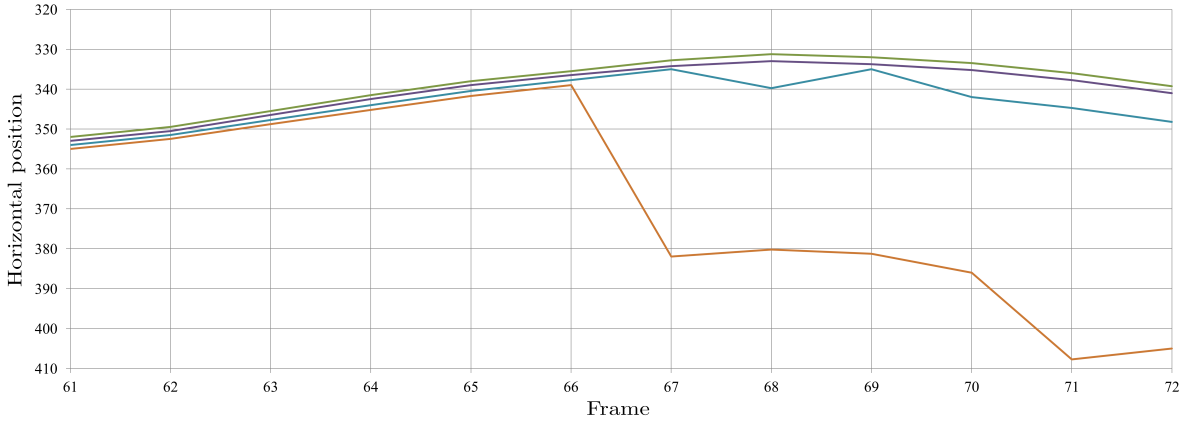


(a) Single-reference frame strategy

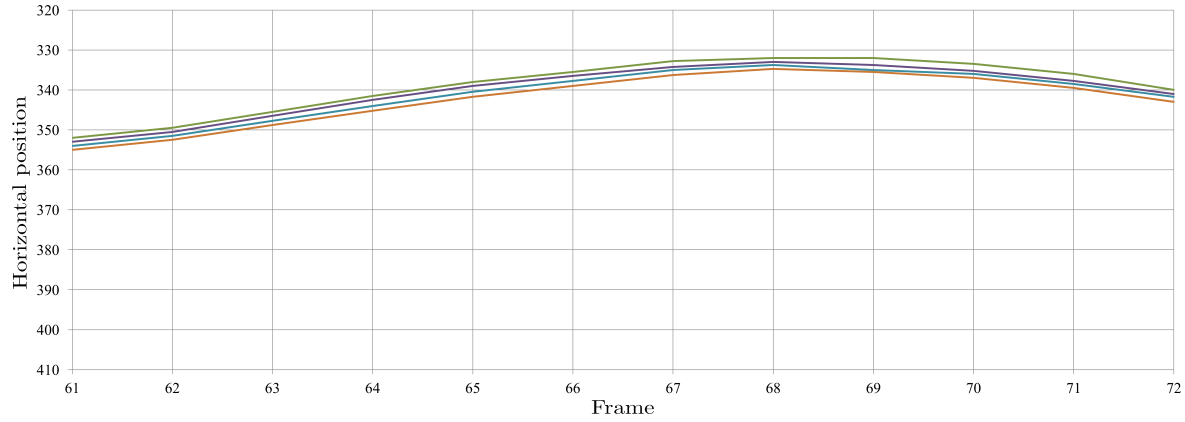


(b) ***From-the-reference* refinement**

Figure 12.23: Horizontal component (with initial $y = 404$ in I_{72}) of 6 estimated trajectories estimated for visible instants (i.e. when the starting points of I_{72} are visible in I_n) on the *Walking-Couple-72-92* sequence with: 1) the single-ref. frame strategy based on *StatFlow(2D-DE)* performed from I_{72}, 2) the *from-the-reference* motion refinement (second step of the two-reference frames motion refinement) proposed in Section 12.3.2 and based on *StatFlow(2D-DE)* from I_{72} and I_{92}.



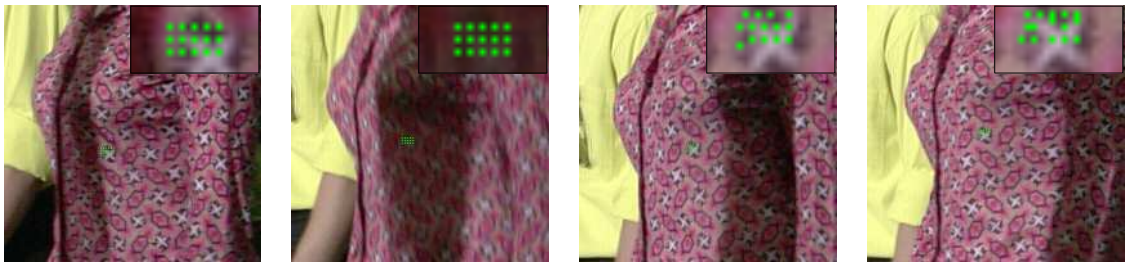
(a) Single-reference frame strategy



(b) ***From-the-reference* refinement**

Figure 12.24: Horizontal component (with initial $y = 440$ in I_{61}) of 4 trajectories estimated on the *Walking-Couple-61-72* sequence with: 1) the single-ref. frame strategy based on *StatFlow(2D-DE)* performed from I_{61}, 2) the *from-the-reference* motion refinement (second step of the two-reference frames motion refinement) proposed in Section 12.3.2 and based on *StatFlow(2D-DE)* from I_{61} and I_{72}.

(a) Source frame I_{61} (c) Point tracking in I_{65} Single-ref. frame strategy (d) Point tracking in I_{69} Single-ref. frame strategy (e) Point tracking in I_{72} Single-ref. frame strategy



(b) Grid points in I_{61} (f) Point tracking in I_{65} ***From-the-ref.*** **refinement** (g) Point tracking in I_{69} ***From-the-ref.*** **refinement** (h) Point tracking in I_{72} ***From-the-ref.*** **refinement**

Figure 12.25: Point tracking from I_{61} (*Walking-Couple-61-72* sequence) with: 1) the single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{61}, 2) the *from-the-reference* motion refinement (second step of the two-reference frames motion refinement) proposed in Section 12.3.2 and based on *StatFlow*(*2D-DE*) from I_{61} and I_{72}.

pixel of the reference frame I_{ref_k} with its associated *from-the-reference* displacement vector $\boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k})$ (given as input and to be refined) and inter-reference frames displacement vector $\boldsymbol{d}^*_{ref_k,ref_{k+1}}(\boldsymbol{x}_{ref_k})$ (computed during the inter-reference frames motion refinement stage, Section 12.3.1). The successive conditions to be allowed to refine $\boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k})$ are as follows. First, the inter-reference frames correspondence must exist and must be consistent (i.e. $\boldsymbol{d}^*_{ref_k,ref_{k+1}}(\boldsymbol{x}_{ref_k})$ visible and consistent). Second, $\boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k})$ must be inconsistent (if consistent, no need to correction) and the *backward* displacement vector starting from the correspondence of $\boldsymbol{x}_{ref_k}$ in $I_{ref_{k+1}}$ must be visible and consistent. Even if these conditions are fulfilled, it does not guarantee the refinement of $\boldsymbol{d}_{ref_k,n}(\boldsymbol{x}_{ref_k})$ in all cases. This depends on the energy values obtained with respect to the energy functional displayed in Eq. 12.11.

In the matter of trajectory, the algorithm is still able to refine some trajectories by sequentially correcting their constitutive *from-the-reference* displacement vectors. Fig. 12.23 and Fig. 12.24 prove this finding by providing two examples of 2D+t trajectory visualization. Respectively 6 and 4 grid points belonging to I_{72} (*Walking-Couple*-72-92) and I_{61} (*Walking-Couple*-61-72) have been tracked using both the single-reference frame strategy and the *from-the-reference* motion refinement based on *StatFlow* estimations achieved from the two reference frames ($\{I_{72}, I_{92}\}$ and $\{I_{61}, I_{72}\}$). Each single *StatFlow* estimation has been performed using 2D-DE [RTDC12] *multi-step* elementary *optical flow* fields with the *steps* $1-20$ for *Walking-Couple*-72-92, $1-5$ and 10 for *Walking-Couple*-61-72. The tracked points are initially located on an horizontal line within the shirt of the woman in I_{72} (resp. I_{61}).

Fig. 12.23 displays the 6 estimated trajectories only for visible instants, i.e. when the starting points of I_{72} are not occluded by the foreground tree in I_n. We observe that the

blue trajectory (starting from $(337, 404)$) and the orange one (starting from $(338, 404)$) are not accurately estimated using the single-reference frame strategy because they appear below the four other ones after the temporary occlusion whereas they were initially above (Fig. 12.23 (a)). On the contrary, the *from-the-reference* motion refinement refines these two trajectories, even in presence of strong illumination variations, except in I_{84} and I_{85} for the orange trajectory and from I_{84} to I_{86} for the blue one (see Fig. 12.23 (b)).

In the same spirit, we notice in Fig. 12.24 that the two trajectories (blue and orange respectively starting from $(354, 440)$ and $(355, 440)$) which diverge with the single-reference frame strategy (Fig. 12.24 (a)) are properly corrected with the proposed refinement (Fig. 12.24 (b)). Once again, *backward from-the-reference* displacement vectors starting from I_{72} have succeeded in refining the *forward* trajectories estimated with respect to I_{61}.

Finally, single-reference frame and *from-the-reference* strategies are assessed through point tracking in Fig. 12.25. A grid made of 15 pixels is tracked from I_{61} of the *Walking-Couple*-61-72 sequence, up to I_{72} going through intermediate frames where strong shadows occur. It is interesting to note in Fig. 12.25 (d,e) that the points diverge to the dark areas of the black and white star-shaped structure with the single-reference frame estimation. They have been badly influenced by the shadow which darkens the colour of the pixels for previous frames. On the contrary, the layout of the grid is better preserved using *from-the-reference* displacement vectors coming from the *from-the-reference* refinement (Fig. 12.25 (g,h)).

To conclude, the experimental results have shown that the *from-the-reference* strategy is able to repair some tracking failures and to improve the spatio-temporal consistency of neighboring trajectories. However, it has been noticed that the proposed approach finally refines only a small subset of all the *from-the-reference* displacement vectors for the reasons explained above. This strong limitation explains why we did not pursue the study and the assessment of the *to-the-reference* motion refinement strategy described in Section 12.3.3. Indeed, as for the *from-the-reference* case, the *to-the-reference* refinement strategy performs the refinement only under certain consistency conditions and requires consistent inter-reference frames correspondences.

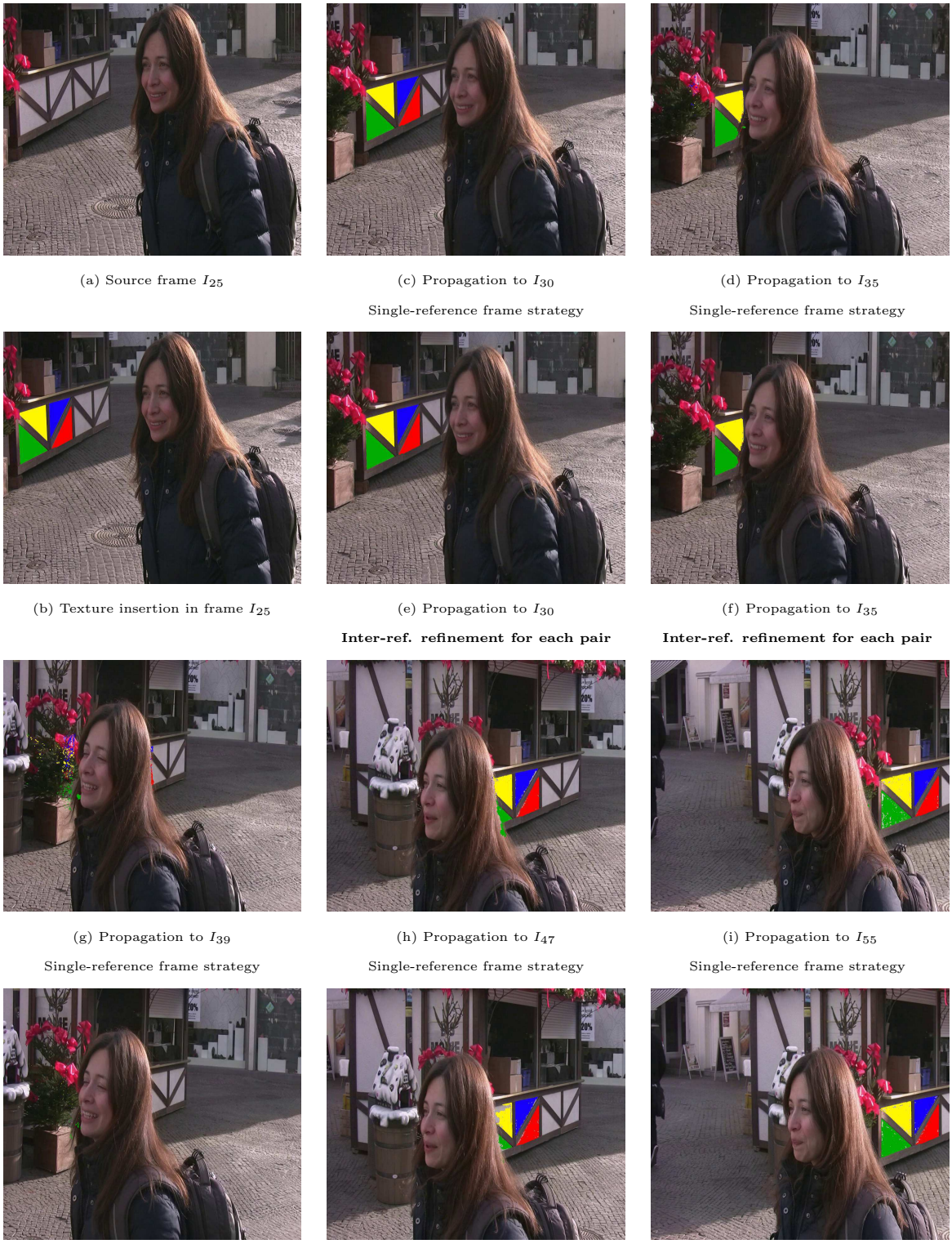
However, we chose in the next section to propose and to assess an extension to the whole video of the inter-reference frames motion refinement stage (Section 12.3.1) which has shown encouraging results between a pair of two reference frames in Section 12.4.2.

12.4.3 Extension of the inter-reference frames refinement to the whole video

To finish this experimental part, we propose to extend to the whole sequence the inter-reference frames motion refinement (Section 12.3.1) which has given good results according to the experiments of Section 12.4.1 for a pair of reference frames $\{I_{ref_k}, I_{ref_{k+1}}\}$. In this context, we consider a video sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\dots,N]\!]}$ with $I_{ref} = I_0$ considered as reference frame. The idea is to consider not only I_{ref} as reference frame but also each frame I_n of the sequence as reference frame with $n \in [\![1, \dots, N]\!]$. Then, we aim at performing a long-term dense motion estimator such as *StatFlow* from:

- the initial reference frame I_{ref} in the *forward* direction,
- each intermediate frame I_n in the *backward* direction.

Once all these multiple *StatFlow* estimations have been achieved, the goal is to finally apply the inter-reference frame motion refinement for each pair $\{I_{ref}, I_n\}$. Of course, this approach is very



(a) Source frame I_{25}

(c) Propagation to I_{30}
Single-reference frame strategy

(d) Propagation to I_{35}
Single-reference frame strategy

(b) Texture insertion in frame I_{25}

(e) Propagation to I_{30}
Inter-ref. refinement for each pair

(f) Propagation to I_{35}
Inter-ref. refinement for each pair

(g) Propagation to I_{39}
Single-reference frame strategy

(h) Propagation to I_{47}
Single-reference frame strategy

(i) Propagation to I_{55}
Single-reference frame strategy

(j) Propagation to I_{39}
Inter-ref. refinement for each pair

(k) Propagation to I_{47}
Inter-ref. refinement for each pair

(l) Propagation to I_{55}
Inter-ref. refinement for each pair

Figure 12.26: Texture insertion in I_{25} and propagation along the *MPI-S*1-25-55 sequence up to I_{55}. We compare: 1) the single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{25}, 2) the inter-ref. refinement (Section 12.3.1) applied to each pair $\{I_{ref}, I_n\}$ which assumes that *StatFlow*(*2D-DE*) has been applied in *forward* from I_{25} and in *backward* from each frame of the sequence.

(a) Source frame I_{72}

(c) Propagation to I_{77} Single-reference frame strategy

(d) Propagation to I_{80} Single-reference frame strategy

(e) Propagation to I_{82} Single-reference frame strategy

(b) Texture insertion in I_{72}

(f) Prop. to I_{72}, **Inter-ref refinement for each pair**

(g) Prop. to I_{80}, **Inter-ref refinement for each pair**

(h) Prop. to I_{82}, **Inter-ref refinement for each pair**

(i) Propagation to I_{84} Single-reference frame strategy

(j) Propagation to I_{87} Single-reference frame strategy

(k) Propagation to I_{92} Single-reference frame strategy

(l) Prop. to I_{84}, **Inter-ref refinement for each pair**

(m) Prop. to I_{87}, **Inter-ref refinement for each pair**

(n) Prop. to I_{92}, **Inter-ref refinement for each pair**

Figure 12.27: Texture insertion in I_{72} and propagation up to I_{92} (*Walking-Couple-72-92*). We compare: 1) single-ref. frame strategy based on *StatFlow*(*2D-DE*) performed from I_{72}, 2) inter-ref. refinement (Section 12.3.1) applied to each pair $\{I_{ref}, I_n\}$ which assumes that *StatFlow*(*2D-DE*) has been applied in *forward* from I_{72} and in *backward* from each frame of the sequence.

computationally expensive. Nevertheless, it seems relevant to consider such method having in mind the future improvements in terms of computation speed we can expect. Moreover, it can be easily performed by considering a low-computational long-term motion estimator as input or by reducing the tracking area.

In the following, two examples of texture propagation are provided for *MPI-S*1-25-55 (Fig. 12.26) and *Walking-Couple*-72-92 (Fig. 12.27) in order to assess the extension of the inter-reference frames motion refinement to the whole sequence. Each single *StatFlow* estimation has been performed using 2*D-DE* [RTDC12] *multi-step* elementary *optical flow* fields with the *steps*: $1-5$, 10, 15, 22, 24, 26, 28 and 30 for *MPI-S*1-25-55, $1-20$ for *Walking-Couple*-72-92.

Through these complex video editing examples, we compare the single-reference frame strategy using *StatFlow*(2*D-DE*) from I_{25} (resp. I_{72}) and the inter-reference frames motion refinement applied to each pair $\{I_{ref}, I_n\}$ using *StatFlow*(2*D-DE*) estimations established in *forward* from I_{ref} and in *backward* from I_n.

Fig. 12.26 shows that before the temporary occlusion, the extended inter-reference frames refinement is able to perform a better texture propagation than the single-reference frame strategy. The artifacts located on the christmas tree or on both background sides of the woman's face (Fig. 12.26 (*d,g*)) are removed in I_{30} (Fig. 12.26 (*f*)) and clearly reduced for I_{39} (Fig. 12.26 (*j*)). After the temporary occlusion, the extended inter-reference frames refinement does not propagate green parts of the initial texture on the hair (Fig. 12.26 (*k*)) as wrongly performed by the single-reference frame strategy (Fig. 12.26 (*h*)). Unfortunately, the propagation inside the panel of the kiosk is deteriorated, except for I_{55} (Fig. 12.26 (*l*)).

Given the texture propagation provided in Fig. 12.27, it appears that the extended inter-reference frames refinement improves the results by apprehending more accurately the occlusions made by the foreground tree (Fig. 12.27 (*d,g*)). By comparing Fig. 12.27 (*e*) and Fig. 12.27 (*h*), we notice that the artifact which induces the texture to remain on the right side of the tree is reduced but not completely removed in I_{82}. This artifact is due to the occlusion which disturbs the matching process by encouraging a matching with a still visible similar periodic structure instead of determining that the current periodic structure is in fact occluded.

As in Fig. 12.26, the results just after the temporary occlusion are slightly worse with the proposed extended inter-reference frames refinement (Fig. 12.27 (*l*)) compared to the single-reference frame strategy (Fig. 12.27 (*i*)). This proves that temporary occlusions can be accurately handled by *multi-step* estimates only if the motion estimator starts from a reference frame relatively far temporally from the occlusion in order to consider multiple *paths* (and not only one single *path*) to jump the temporary occlusion. For this specific situation, we perceive the limits of the proposed method which consists in applying for each pair $\{I_{ref}, I_n\}$ independently the inter-reference frames refinement. An a-posteriori filtering could allow to bring back more accurate information from frames temporally distant from the temporary occlusion to the frames which is located just after.

Finally, in Fig. 12.27, we observe slight improvements with the extended inter-reference frames refinement for the last frames of the sequence, especially in term of reduction of motion *outliers*. Generally, the accuracy of the refined *to-the-reference* displacement vectors appears to be good despite small holes which occur due to wrongly estimated occluded pixels.

12.5 Conclusion

The purpose of this chapter was to study how multi-reference frames long-term dense motion estimates can be combined to provide very long-term accurate *from-the-reference* and *to-the-reference* displacement vectors. In this context, two main strategies have been proposed. First, we described a multi-reference frames strategy through trajectory quality assessment whose aim is to insert new reference frames each time the trajectories diverge (Section 12.1). Second, in Section 12.3, we proposed a two-reference frames motion refinement which includes especially a robust inter-reference frames motion refinement stage. We saw that this latter approach can be extended to the whole sequence assuming that a long-term dense motion estimator is applied from each frame of the sequence.

The key aspect which plays a major role within the proposed multi-reference frames frameworks deals with the assessment of the intrinsic displacement field quality. In particular, we observed that a robust motion quality assessment can both encourage the insertion of new reference frames and make the distinction between the displacement vectors for which a refinement is required and the displacement vectors which can be used for the refinement task. The inconsistency has proved to be a robust feature to accurately and automatically assess the displacement fields quality.

Experimental results in Section 12.2 and 12.4 have revealed that considering multiple reference frames can significantly increase the length of accurate long-term trajectories and more generally improve both *from-the-reference* and *to-the-reference* displacement vectors. We focused especially on very complex sequences featuring temporary occlusions, strong illumination variations and periodic structures. Without giving perfect results for such complex scenes, the proposed multi-reference frames approaches have shown to give however satisfactory performance.

While concluding and summarizing our contributions in the field of long-term dense motion estimation, the next chapter, Chapter 13, focuses on aspects which must deserve more attention for further research.

Chapter 13

Conclusion and further work

This chapter concludes the Part II dedicated to long-term dense motion estimation. In the following, we briefly summarize the context of our study and our contributions. We finally give some clues and information toward further work in this field of motion estimation.

Starting from the color constancy assumption, different strategies and numerical solutions have been involved to robustify the estimation of *optical flow* vectors (Chapter 7). These vectors can be used to reach long-term requirements through straightforward concatenations or more sophisticated integration strategies (Chapter 8). However, applications which require a very high quality such as video editing tasks need more robust strategies in order to manipulate more reliable dense long-term trajectory/displacement fields.

Therefore, the main issue of this Part II was to introduce alternative approaches to perform a long-term dense motion estimation while limiting the accumulation of estimation errors over time (Chapter 9). We focused especially on how to integrate in a robust manner *multi-step* elementary *optical flow* vectors across video sequences and how to extend the classical color constancy assumption to accurately select long-term displacement fields among a set of candidate displacement fields.

In this context, we proposed new long-term dense motion estimation methods:

1. *multi-step* flow via *graph-cuts* (*MS-GC*) and *multi-step* fusion flow (*MSF*) which consist in both accumulating *multi-step* elementary *optical flow* vectors through *inverse* integration and merging the resulting candidate long-term displacement fields (Chapter 10),

2. *statistical multi-step flow* (*StatFlow*) which is based on a combinatorial integration of *multi-step* elementary *optical flow* vectors followed by a statistical-based displacement vectors selection (Chapter 11).

These methods efficiently combine *multi-step* elementary *optical flow* vectors in order to reach a good compromise between consecutive *optical flow* concatenation which is prone to motion drift and direct matching which is sensitive to ambiguous correspondences (especially in presence of strong illumination changes, periodic structures or large uniform areas). In addition, *multi-step* estimates are exploited in order to be able to jump occluding objects when temporary occlusions occur.

With respect to state-of-the-art methods, the proposed quantitative and qualitative experiments for the proposed methods have shown a clear improvement in terms of accuracy and robustness of the resulting long-term trajectory/displacement fields.

In Chapter 12, we also studied how these approaches can be involved within a multi-reference frames framework in order to repair the tracking failures and to push the motion estimation process as far as possible temporally (i.e. toward longer accurate dense long-term trajectories). Our multi-reference frames strategies have been built in order to correlate reliable pieces of trajectories estimated with respect to different reference frames. A good evaluation of the intrinsic quality of the displacement vectors has shown to be crucial because it guides the insertion of new reference frames and makes the distinction between reliable displacement vectors and motion outliers.

Let us now evoke the aspects which must deserve more attention for further research.

Occlusion and discontinuity management

This Part II has mainly focused of the computation of the displacement fields themselves and considerations about occlusion detection has been confined to a method which consists in both back-projecting *backward* vectors and identifying the pixels which did not receive any vectors.

The detection of occlusion is a crucial task which goes hand in hand with the computation of displacement fields. Therefore, further work must deserve more attention on this issue. Following [XCS+06], an idea to more accurately detect occlusions could consist in explicitly introducing occlusions within the energy functional through an occlusion probability term.

Moreover, a more sophisticated occlusion reasoning could involve the computation of both occluded and occluding objects via a thorough study of motion discontinuities. In this direction, strategies such as the reasoning proposed in [LASL11] about occlusions and disocclusions through depth ordering constraints seems to be promising.

Critical cases

Some critical cases have not been handled explicitly in our work. They deal especially with zooming, transparency and rotational motion, as detailed below.

First, a robust management of zooming can allow to distinguish zooming from occlusions. To avoid this problem, our approach was to consider as reference frames the images at the highest resolution, i.e. the frames for which the area to be tracked is described by more pixels. However, a more subtle approach could allow to explicitly take into account the convergent (for zoom-in) or divergent (for zoom-out) dynamic of motion trajectories.

Second, transparency is a very specific problem we did not consider explicitly. Handling transparency requires a multi-layer representation as well as multiple parametric models to describe the displacement of the multiple surface layers. Such approach, as the one described in [JBJ96], requires an extended spatial regularization method to cope with the multiple local motion estimates.

Third, rotational motion has not been directly taken into account except when parametric motion models were considered as inputs of the proposed long-term dense motion estimators (see the related preliminary experiments for *multi-step flow fusion* in Chapter 10). Nevertheless, to recover satisfactorily such motion (rotational motion in the image plane for instance), we suggest to focus on alternative strategies with a stronger focus on parametric models such as homographies. In this direction, *a priori* information about the nature of the displacement as well as object pre-segmentations can naturally robustify the establishment of motion correspondences, especially between distant frames.

Motion models and sparse trajectories

As suggested for rotational motion, involving rigid and deformable models to recover specific motion types could be judicious to constrain the estimation for some regions. In the same spirit, sparse trajectories estimated with standard tool such as *KLT* [LK81, TK91, ST94] or *SIFT* [Low04] features could also be considered. For both cases (motion models or sparse trajectories), the idea is to take the corresponding (dense or sparse) motion fields as inputs of our dense long-term motion estimators. These additional inputs could provide *a priori* information to the motion estimation process and therefore could contribute to the improvement of long-term displacement fields.

Gain processing

In this Part II, we saw several times that performing motion estimation on real content may find some issues in efficiently working in presence of illumination changes. In Chapter 12, we implicitly tried to handle such illumination changes through a two-reference frames strategy. Indeed, by establishing accurate inter-reference motion correspondences even with strong color variations, the idea was to combine trajectories starting from both reference frames in order to avoid the motion drift due to illumination changes.

However, it seems that color variations could be more explicitly modelled in order to limit their bad influence on the computation of dense long-term trajectories. A first step in this direction could consist in introducing a color or luminance gain factor to locally quantify the differences in terms of illumination changes. Such gain measures can be used to obtain gain-compensated matching costs which would allow to assess an intrinsic vector quality while being less sensitive to illumination changes than classical matching cost.

Another idea could consist in spatially regularizing the long-term displacement fields in term of gain similarities as done in terms of motion similarities within classical energy functionals. Moreover, a joint estimation of both gain factors and displacement fields (in $x-$ and $y-$ directions) may be more efficient than separated estimations.

Semi-automatic processing

The proposed approaches related to long-term dense motion estimation are fully automatic. However, whatever the methods used (including ours), it is not always possible to guarantee exact long-term dense displacement/trajectory fields, especially when dealing with complex scenes. That is why it can be interesting in some cases to consider a semi-automatic approach to improve automatic motion estimation by incorporating information provided by an operator within the automatic process.

In the following, we discuss the topic of semi-automatic processing by approaching the problem from two different angles.

First, in a general point of view, we can talk about user-assisted motion estimation where the user interacts directly with motion fields in order to improve image correspondences *a posteriori* as done in [KRLM11, RHK+12]. For this task, the results of an initial automatic motion estimation can be displayed to the operator in order to be evaluated and manually corrected if necessary. The user actions may concern the modification of point correspondences as well as pixel occlusion labels. Moreover, an automatic evaluation of motion quality can guide the operator in the examination of the sequence of results. Instead of directly modifying motion

correspondences themselves, a more intuitive way to interact with motion consists in assigning the motion of a reference correct area with a similar motion to another area for which the motion fields have been badly estimated.

Second, in a more applicative and *a priori* point of view, considering interactions between the operator actions and the automatic processing can allow to meet the objectives for a given application. In this context, we suggest in particular two possible user interactions:

1. The operator can provide bounding boxes for each key frame in order to track one or several specific objects. This action can be very useful in the context of semi-automatic texture insertion for instance. Indeed, these bounding boxes can constrain motion estimation and therefore limit or avoid motion outliers. In this direction, one can imagine to resort to automatic object tracking algorithms in order to propagate these bounding boxes to the whole sequence and therefore restrict the trajectory estimation to a specific spatio-temporal motion tube. Nevertheless, we must ensure that the automatic tracking of the bounding boxes is done without estimation errors.

2. Other simple interaction tools such as paint-brushes can help the operator to roughly segment the objects to be tracked. In this case, a strong spatial regularization could be applied on motion estimation for areas belonging to the same object. However, it is crucial to provide to the motion estimation process user strokes which do not cross any motion discontinuities.

Representation

For applications such as video editing where image modifications are propagated along the video sequence using the estimated long-term dense displacement vectors, we adopted the approach which consists in manually selecting the frames which are the most representative of the object to be modified. These particular frames on which we perfom the editing tasks were referred to as reference frames. In case of zoom-in for instance, the idea was to insert image modifications in the frame with the highest resolution which makes the area under consideration described by the widest possible set of pixels (see especially the logo insertion and propagation results for the *Amelia-Retro* sequence in Chapter 10).

With our long-term dense displacement fields, we can also easily adopt a mosaic representation and therefore create mosaics to perform video editing tasks, as done in [RAKRF08]. In this context, the mosaic representation can be useful since objects are agglomerated in a compact way. Editing operations can then be re-composited from the mosaic to the original sequence.

In case of self-occlusion, the editing operations can be easily done with a mosaic representation since the object under consideration is outspread within the mosaic. Nevertheless, relying only on reference frames (instead of mosaics) to perform video editing tasks is still possible in presence of self-occlusions. Indeed, if the object on which we aim at bringing image modifications is not totally visible in the selected reference frame, it is conceivable to select a second reference frame for which the parts of the object occluded in the first reference frame are visible and then to continue the modification of the sequence content through this second reference frame.

More generally, by applying recursively this reference frames selection approach to describe totally a given object, we can identify the smallest set of frames that contain alone all the visible regions of the object under consideration along the shot. Of course, a point which has not been

treated and that would deserve further investigation is the automatic selection of these reference frames. The previously described recursive approach combined to an automatic reference frames selection could allow to describe very compactly the video content and its associated long-term displacement.

By repeating this procedure for each area or object belonging to the video sequence, one could expect to reach the requirements of [RT13] whose goal is to guarantee a complete coverage of all the visible points in all the frames. The underlying idea is to assign to each pixel of the whole video shot a single trajectory running through the sequence and starting from the most appropriate frame given that each trajectory is allowed to start in any frame and must be able to deal with temporary occlusions. How to compactly represent the totality of the video content with associated long-term motion behavior while taking into account local and global variations of illuminations is however an open challenge.

Bibliography Part II

[ADPS02] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions detection. In *European Conference on Computer Vision*, pages 721–735. Springer, 2002.

[BA96] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[Bar04] A. Barjatya. Block matching algorithms for motion estimation. *IEEE Transactions Evolution Computation*, 8(3):225–239, 2004.

[BBPW04] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision*, pages 25–36, 2004.

[BDRB07] O. Brouard, F. Delannay, V. Ricordel, and D. Barba. Fast long-term motion estimation for high definition video sequences based on spatio-temporal tubes and using the nelder-mead simplex algorithm. In *Picture Coding Symposium*, 2007.

[BF06] A. Buchanan and A. Fitzgibbon. Interactive feature tracking using KD Trees and dynamic programming. In *IEEE International Conference on Computer Vision Pattern Recognition*. IEEE Computer Society, 2006.

[BHB00] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *IEEE International Conference on Computer Vision Pattern Recognition*, volume 2, pages 690–696. IEEE, 2000.

[BJ96] M.J. Black and A.D. Jepson. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):972–986, 1996.

[BM01] A.C. Berg and J. Malik. Geometric blur for template matching. In *IEEE International Conference on Computer Vision Pattern Recognition*, volume 1, pages I–607. IEEE, 2001.

[BM10] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, pages 282–295. Springer, 2010.

[BM11] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[BSFG09] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman. Patchmatch: a randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics*, volume 28, page 24. ACM, 2009.

[BTVG06] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006.

[But08] J.C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2008.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[CCR+12] T. Crivelli, P.-H. Conze, P. Robert, M. Fradet, and P. Pérez. Multi-step flow fusion: Towards accurate and dense correspondences in long video shots. In *British Machine Vision Conference*, 2012.

[CCRM13a] P.-H. Conze, T. Crivelli, P. Robert, and L. Morin. Dense motion estimation between distant frames: Combinatorial multi-step integration and statistical selection. In *IEEE International Conference on Image Processing*, 2013.

[CCRM13b] P.-H. Conze, T. Crivelli, P. Robert, and L. Morin. Estimation de mouvement dense entre images distantes: Intégration combinatoire multi-steps et sélection statistique. In *GRETSI Symposium on Signal and Image Processing*, 2013.

[CCRM14] P.-H. Conze, T. Crivelli, P. Robert, and L. Morin. Dense long-term motion estimation via statistical multi-step flow. In *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2014.

[CCRP12] T. Crivelli, P.-H. Conze, P. Robert, and P. Pérez. From optical flow to dense long term correspondences. In *IEEE International Conference on Image Processing*, 2012.

[CFC+14] T. Crivelli, M. Fradet, P.-H. Conze, P. Robert, and P. Pérez. Robust optical flow integration. *Transactions on Image Processing*, 2014.

[CLD11] Xun Cao, Zheng Li, and Qionghai Dai. Semi-automatic 2D-to-3D conversion using disparity propagation. *IEEE Transactions on Broadcasting*, 57(2):491–499, 2011.

[CMP02] T. Corpetti, É. Mémin, and P. Pérez. Dense estimation of fluid flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):365–380, 2002.

[CMPP09] M. Cagnazzo, T. Maugey, and B. Pesquet-Popescu. A differential motion estimation method for image interpolation in distributed video coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1861–1864. IEEE, 2009.

[CRM00] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE International Conference on Computer Vision Pattern Recognition*, volume 2, pages 142–149. IEEE, 2000.

[DMC10] C. Dehais, G. Morin, and V. Charvillat. From rendering to tracking point-based 3D models. *Image and Vision Computing*, 28(9):1386–1395, 2010.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[EW02] G. Egnal and R.P. Wildes. Detecting binocular half-occlusions: empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1127–1133, 2002.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FPR08a] M. Fradet, P. Pérez, and P. Robert. Semi-automatic motion segmentation with motion layer mosaics. *European Conference on Computer Vision*, pages 210–223, 2008.

[FPR08b] M. Fradet, P. Pérez, and P. Robert. Time-sequential extraction of motion layers. In *IEEE International Conference on Image Processing*, pages 3224–3227, 2008.

[FSBC11] G. Facciolo, R. Sadek, A. Bugeau, and V. Caselles. Temporally consistent gradient domain video editing. In *Energy minimization methods in computer vision and pattern recognition*, pages 59–73. Springer, 2011.

[GKT+] M. Granados, K.I. Kim, J. Tompkin, J. Kautz, and C. Theobalt. MPI-S1. http://www.mpi-inf.mpg.de/~granados/projects/vidbginp/index.html.

[GRA11a] R. Garg, A. Roussos, and L. Agapito. Flag dataset. http://www.eecs.qmul.ac.uk/~lourdes/subspace_flow, 2011.

[GRA11b] R. Garg, A. Roussos, and L. Agapito. Robust trajectory-space TV-L1 optical flow for non-rigid sequences. *Energy Minimazation Methods in Computer Vision and Pattern Recognition*, pages 300–314, 2011.

[GRA13] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 2013.

[HB93] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, 1993.

[HMAB11] A. Humayun, O. Mac Aodha, and G.J. Brostow. Learning to find occlusion regions. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 2161–2168. IEEE, 2011.

[HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981.

[Hub73] P.J. Huber. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799–821, 1973.

[IA00] M. Irani and P. Anandan. About direct methods. *Vision Algorithms: Theory and Practice*, pages 267–277, 2000.

[Ira99] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *IEEE International Conference on Computer Vision*, volume 1, pages 626–633. IEEE, 1999.

[Ira02] M. Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, 2002.

[JB93] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 760–761. IEEE, 1993.

[JBJ96] S.X. Ju, M.J. Black, and A.D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 307–314. IEEE, 1996.

[KA11] S. Korman and S. Avidan. Coherency sensitive hashing. In *IEEE International Conference on Computer Vision*, pages 1607–1614. IEEE, 2011.

[KMM10] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *International Conference on Pattern Recognition*, pages 2756–2759. IEEE, 2010.

[KRLM11] F. Klose, K. Ruhl, C. Lipski, and M. Magnor. Flowlab: an interactive tool for editing dense image correspondences. In *Conference on Visual Media Production*, pages 59–66. IEEE, 2011.

[KZ04] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[LASL11] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 3369–3376. IEEE, 2011.

[LK81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, pages 674–679. Citeseer, 1981.

[LK10] J. Luo and E.E. Konofagou. A fast normalized cross-correlation calculation method for motion estimation. *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 57(6):1347–1357, 2010.

[LMSR08] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 1–8. IEEE, 2008.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LRR08] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 1–8. IEEE, 2008.

[LRRB10] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for Markov Random Field optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1392–1405, 2010.

[LTF+05] C. Liu, A. Torralba, W.T. Freeman, F. Durand, and E.H. Adelson. Motion magnification. In *ACM Transactions on Graphics*, volume 24, pages 519–526. ACM, 2005.

[LYT+08] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman. SIFT Flow: dense correspondence across different scenes. In *European Conference on Computer Vision*, pages 28–42. Springer, 2008.

[LYT11] C. Liu, J. Yuen, and A. Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011.

[MAB08] R. Mégret, J.-B. Authesserre, and Y. Berthoumieu. The bi-directional framework for unifying parametric image alignment approaches. In *European Conference on Computer Vision*, pages 400–411. Springer, 2008.

[MABP10] O. Mac Aodha, G.J. Brostow, and M. Pollefeys. Segmenting video into classes of algorithm-suitability. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 1054–1061. IEEE, 2010.

[MB87] D.W. Murray and B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):220–228, 1987.

[MP02] E. Memin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, 2002.

[MS04] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3):201–218, 2004.

[Nag90] H.-H. Nagel. Extending the oriented smoothness constraint into the temporal domain and the estimation of derivatives of optical flow. *European Conference on Computer Vision*, pages 139–148, 1990.

[NLD11] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision*, pages 2320–2327. IEEE, 2011.

[NLMLCB09] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba. Considering temporal variations of spatial visual distortions in video quality assessment. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):253–265, 2009.

[OB95] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of visual communication and image representation*, 6(4):348–365, 1995.

[OVCP13] A. Ozerov, J.-R. Vigouroux, L. Chevallier, and P. Pérez. On evaluating face tracks in movies. In *IEEE International Conference on Image Processing*, 2013.

[PB12] D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 97(1):54–70, 2012.

[PHVG02] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages 661–675. Springer, 2002.

[RAKRF08] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: a new representation for video editing. In *ACM SIGGRAPH*, pages 1–11. ACM, 2008.

[RHK+12] K. Ruhl, B. Hell, F. Klose, C. Lipski, S. Petersen, and M. Magnor. Improving dense image correspondence estimation with interactive user guidance. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1129–1132. ACM, 2012.

[RLF12] M. Rubinstein, C. Liu, and W. T. Freeman. Towards longer long-range motion trajectories. In *British Machine Vision Conference*, 2012.

[RT13] Susanna Ricco and Carlo Tomasi. Video motion for every visible point. In *IEEE International Conference on Computer Vision*, 2013.

[RTDC12] P. Robert, C. Thébault, V. Drazic, and P.-H. Conze. Disparity-compensated view synthesis for s3D content correction. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.

[RWDF13] M. Rubinstein, N. Wadhwa, F. Durand, and W. T. Freeman. Revealing invisible changes in the world. *Science*, 339(6119):519–519, 2013.

[Sam90] H. Samet. *The design and analysis of spatial data structures*, volume 199. Addison-Wesley Reading, MA, 1990.

[SBK10] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. *European Conference on Computer Vision*, pages 438–451, 2010.

[SEASM09] V. Scholz, S. El-Abed, H.-P. Seidel, and M. Magnor. Editing object behaviour in video sequences. In *Computer Graphics Forum*, volume 28, pages 1632–1643. Wiley Online Library, 2009.

[SFAC13] R. Sadek, G. Facciolo, P. Arias, and V. Caselles. A variational model for gradient-based video editing. *International Journal of Computer Vision*, pages 1–36, 2013.

[SN11] R. Shah and P.J. Narayanan. Trajectory-based video object manipulation. In *IEEE International Conference on Multimedia & Expo*, pages 1–4. IEEE, 2011.

[SN13] R. Shah and P.J. Narayanan. Interactive video manipulation using object trajectories and scene backgrounds. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013.

[SPC09] F. Steinbrucker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *IEEE International Conference on Computer Vision*, pages 1609–1614. IEEE, 2009.

[SRB10] D. Sun, S. Roth, and M.J. Black. Secrets of optical flow estimation and their principles. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 2432–2439, 2010.

[SS96] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1199–1210, 1996.

[SS07] A. Salgado and J. Sánchez. Temporal constraints in large optical flow estimation. In *Computer Aided Systems Theory*, pages 709–716. Springer, 2007.

[ST94] J. Shi and C. Tomasi. Good features to track. In *IEEE International Conference on Computer Vision Pattern Recognition*, pages 593–600. IEEE, 1994.

[ST06] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. In *IEEE International Conference on Computer Vision Pattern Recognition*, volume 2, pages 2195–2202. IEEE, 2006.

[ST08] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *International Journal of Computer Vision*, 80(1):72–91, 2008.

[TB02] L. Torresani and C. Bregler. Space-time tracking. In *European Conference on Computer Vision*, pages 801–812. Springer, 2002.

[TK91] C. Tomasi and T. Kanade. *Detection and tracking of point features.* School of Computer Science, Carnegie Mellon University, 1991.

[TK92] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *IEEE International Conference on Computer Vision*, pages 839–846. IEEE, 1998.

[TV07] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE International Conference on Computer Vision Pattern Recognition.* IEEE, 2007.

[VBVZ11] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer. Modeling temporal coherence for optical flow. In *IEEE International Conference on Computer Vision*, 2011.

[WB04] J. Wills and S. Belongie. A feature-based approach for determining dense long range correspondences. *European Conference on Computer Vision*, pages 170–182, 2004.

[WDAC06] J. Wang, S.M. Drucker, M. Agrawala, and M.F. Cohen. The cartoon animation filter. In *ACM Transactions on Graphics*, volume 25, pages 1169–1173. ACM, 2006.

[WKSL11] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. *IEEE International Conference on Computer Vision Pattern Recognition*, pages 3169–3176, 2011.

[WKSL13] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, pages 1–20, 2013.

[WLB04] Z. Wang, L. Lu, and A.C. Bovik. Video quality assessment based on structural distortion measurement. *Signal processing: Image communication*, 19(2):121–132, 2004.

[WPZ+09] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for TV-L1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45. Springer, 2009.

[WRS+12] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics*, 31(4):65, 2012.

[WS01] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, 2001.

[WTP+09] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. *British Machine Vision Conference*, 2009.

[XCS+06] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. *European Conference on Computer Vision*, pages 211–224, 2006.

[XJM10] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. In *IEEE International Conference on Computer Vision Pattern Recognition*, 2010.

[XJM12] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1744–1757, 2012.

[YYGN96] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo. Weighted median filters: a tutorial. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(3):157–192, 1996.

[ZDJ+10] G. Zhang, Z. Dong, J. Jia, T.T. Wong, and H. Bao. Efficient non-consecutive feature tracking for structure-from-motion. *European Conference on Computer Vision*, pages 422–435, 2010.

[ZPB07] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. *Pattern Recognition*, pages 214–223, 2007.

Part III

Application to joint stereo and motion processing

Chapter 14

VSQA and long-term dense motion estimation for disparity correction

Up to now, we have described separately the fields of disparity estimation, view synthesis and view synthesis quality assessment (Part I) and the research works concerning long-term dense motion estimation (Part II). However, they may be combined when dealing with stereoscopic video sequence. Indeed, in the literature, many works jointly consider disparity estimation (optionally followed by view synthesis) and motion estimation or quality assessment and motion estimation (for spatio-temporal quality assessment metrics especially).

In the following, we focus on simultaneous motion and disparity processing in a sequence of stereo images. Toward this goal, we will start by briefly describing the related works. In particular, three main topics have been identified in the literature: 1) joint motion and disparity estimation, 2) temporally consistent disparity map estimation, 3) 2D-to-3D conversion through disparity propagation. Related papers are described in Section 14.1.

Section 14.2 studies how long-term dense motion estimation can be involved to refine disparity maps which have been initially estimated in a stereoscopic sequence. In particular, we suggest to extend the classical approach which consists in being restricted to a consecutive quadruplet of images (i.e. two consecutive stereo frames) by involving long-term displacement or trajectory fields to both propagate accurate disparity information from distant frames and promote temporal consistency.

In this context, our main contribution consists in performing disparity correction by combining view synthesis (Part I), view synthesis quality assessment (Part I) as well as long-term dense motion estimation (Part II). More precisely, we propose a new disparity correction framework which aims at: 1) identifying disparity estimation failures through view synthesis and view synthesis quality assessment, 2) correcting these wrong estimations by bringing back more accurate disparity information via long-term dense displacement fields.

After having provided a brief review of existing works in joint stereo and motion analysis (Section 14.1) and a precise description of the proposed disparity correction framework (Section 14.2), Section 14.3 gives early experimental results obtained on a real stereoscopic sequence. Finally, Section 14.4 concludes this chapter and gives perspectives towards long-term joint disparity and motion analysis.

14.1 Review of joint stereo and motion analysis

Let us briefly describe existing work related to joint stereo and motion analysis. Three main topics are reviewed in the following: joint motion and disparity estimation (Section 14.1.1), temporally consistent disparity map estimation (Section 14.1.2), 2D-to-3D conversion through disparity propagation (Section 14.1.3). Finally, Section 14.1.4 concludes this state-of-the-art section.

14.1.1 Joint motion and disparity estimation

Recovering disparity and motion information has been deeply studied [YNLS05, HD07, MPPC09, DMPP10, WBV+11], especially for applications such as 3D tracking [MDC07, SWWES10], stereo or multiview video coding [RSK+12], 3D scene interpretation or for 3D television.

For a given sequence of stereo images, simultaneous motion and disparity estimation is based on the *stereo-motion consistency* constraint which relates two displacement fields and two disparity fields for each two consecutives stereo frames. This constraint, illustrated in Fig. 14.1, allows to compute disparity for each stereo pair of frames based on the estimated left and right elementary *optical flow* fields and the disparity field estimated for the previous stereo pair of frames.

In [MPPC09, DMPP10], joint disparity and motion estimation is performed through a variational optimization method which estimates left ($[u_l, v_l]$) and right ($[u_r, v_r]$) motion vectors as well as disparity at time $n+1$ ($d_{n+1}^{l/r}$) assuming that disparity at time n ($d_n^{l/r}$) has already been computed and therefore is known.

Considering two rectified consecutive stereo frames ($\{I_n^l, I_n^r\}$ and $\{I_{n+1}^l, I_{n+1}^r\}$) for instants n and $n+1$, the joint disparity/motion estimation of [MPPC09, DMPP10] is performed through the minimization of Eq. 14.1.

$$\begin{aligned} E(u_l, v_l, u_r, v_r) = & \sum_{(x,y)} [I_n^l(x,y) - I_{n+1}^l(x + u_l(x,y), y + v_l(x,y))]^2 \\ + & \sum_{(x,y)} [I_n^r(x + d_n^{l/r}(x,y), y) - I_{n+1}^r(x + d_n^{l/r}(x,y) + u_r(x,y), y + v_r(x,y))]^2 \\ + & \sum_{(x,y)} [I_{n+1}^l(x + u_l(x,y), y + v_l(x,y)) - I_{n+1}^r(x + d_n^{l/r}(x,y) + u_r(x,y), y + v_r(x,y))]^2 \end{aligned} \quad (14.1)$$

This energy is based on the conservation of the intensity values between the four following points which in fact correspond to the projections of the same physical point (Fig. 14.1):

- (x, y) in I_n^l,
- $(x + d_n^{l/r}(x,y), y)$ in I_n^r,
- $(x + d_n^{l/r}(x,y) + u_r(x,y), y + v_r(x,y))$ in I_{n+1}^r,
- $(x + u_l(x,y), y + v_l(x,y))$ in I_{n+1}^l.

The *stereo-motion consistency* constraint also brings the two relationships written in Eq. 14.2. According to the first one, the disparity field at instant $n+1$ can be easily recovered using

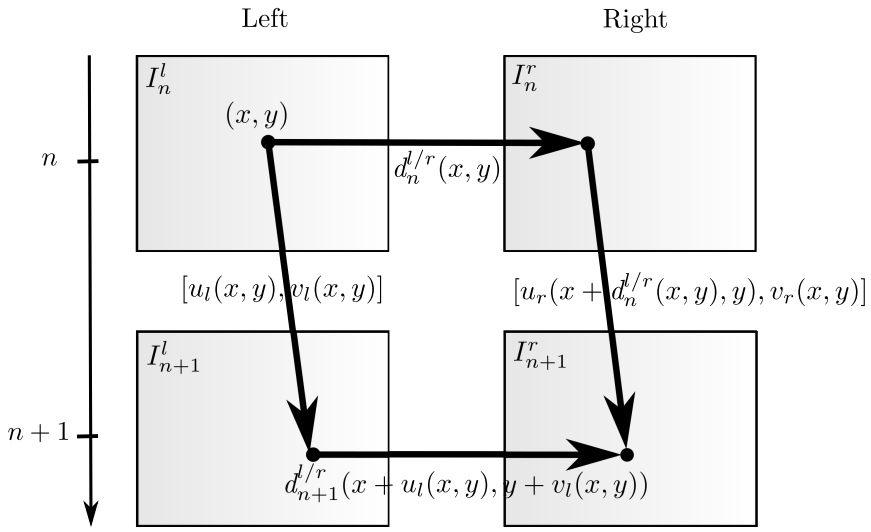


Figure 14.1: The *stereo-motion consistency* constraint involved for joint disparity/motion estimation

$\{u_r, d_n^{l/r}, u_l\}$ (Fig. 14.1). The second relationship indicates that the vertical component of the right motion vector (v_r) can be deduced directly from the vertical component of the left motion vector (v_l).

$$\begin{cases} d_{n+1}^{l/r}(x + u_l(x, y), y + v_l(x, y)) = u_r(x + d_n^{l/r}(x, y), y) + d_n^{l/r}(x, y) - u_l(x, y) \\ v_r(x + d_n^{l/r}(x, y), y) = v_l(x, y) \end{cases} \tag{14.2}$$

Finally, Eq. 14.1 and Eq. 14.2 allow a joint motion/disparity estimation which sequentially processes consecutive quadruplets of images (i.e. consecutive stereo frames) across a stereoscopic sequence.

14.1.2 Temporally consistent disparity map estimation

In the same spirit, many works such as [DSMNP01, Gon06, LH10, HRBG12] have focused on temporally consistent disparity map estimation. Compared to Section 14.1.1, these papers do not focus on joint disparity and motion estimation but incorporate motion information into the disparity estimation process. The underlying idea is to propose alternative approaches to replace temporally independent disparity estimation which often results in an unpleasing flickering.

To reduce the flicker, [Gon06, LH10] incorporate motion information within the disparity estimation process (via disparity prediction for [Gon06] and through a temporal weighting function for [LH10]) in order to refer to the previously estimated disparity map and therefore to enforce temporal consistency.

[DSMNP01] proposes to temporally filter dense disparity estimates by analyzing the temporal evolution of disparity values for a certain number of consecutive frames, using this analysis to identify reliable and unreliable disparity values and constraining the disparity to be consistent over time. [HRBG12] presents a more sophisticated framework via spatio-temporal filtering where disparity changes are aligned with spatio-temporal edges of the video sequence.

14.1.3 2D-to-3D conversion through disparity propagation

Disparity and motion information can also be jointly processed within automatic [PXZ+10] or semi-automatic [VB07, CLD11] 2D-to-3D conversion frameworks. The purpose of these frameworks is to estimate 3D information from a monocular sequence which leads to a process for 3D content creation at a lower cost and with less time compared to 3D content generation [CLD11].

For instance, in the field of semi-automatic 2D-to-3D conversion, *Cao* et al. propose in [CLD11] to assign initial disparity maps to key frames and then to propagate this disparity information to the whole sequence taking into account both color similarity and motion information. The propagation is performed bi-directionally to generate the disparity maps of non-key frames between two key frames through displacement vectors computed with respect to both key frames. The propagation is coupled with a shifted bilateral filtering algorithm which allows to estimate unknown disparity vectors for non-key frames.

14.1.4 Conclusion

Starting from the *stereo-motion consistency* constraint, state-of-the-art methods are generally restricted to quadruplets of images which are processed sequentially across the stereoscopic sequence. As suggested in [CLD11], it is more interesting to involve long-term displacement fields (i.e. between distant frames) to both perform a long-term disparity propagation from key frames and promote temporal consistency.

Starting from this review of existing methods in joint disparity and motion analysis, we propose to study in Section 14.2 the disparity correction task through view synthesis, view synthesis quality assessment and long-term dense motion estimation.

14.2 The proposed disparity correction framework

Toward the goal of refining the disparity estimation for a given binocular sequence, we propose in this section to study a framework which includes disparity estimation, view synthesis, view synthesis quality assessment (Part I) as well as long-term dense motion estimation (Part II). In this context, we assume that the binocular sequence includes $N+1$ RGB left images $\{I_n^l\}_{n\in[\![0,\ldots,N]\!]}$ and $N+1$ RGB right images $\{I_n^r\}_{n\in[\![0,\ldots,N]\!]}$.

Our goal is to perform the correction of left to right (resp. right to left) disparity maps which have been initially computed sequentially between each left frame I_n^l (resp. right frame I_n^r) and its corresponding frame in the right view I_n^r (resp. left frame I_n^l). The disparity maps link each grid point $\boldsymbol{x}_n^l$ (resp. $\boldsymbol{x}_n^r$) of I_n^l (resp. I_n^r) to a position in I_n^r (resp. I_n^l): $\boldsymbol{x}_n^l + d_n^{l/r}$ (resp. $\boldsymbol{x}_n^r + d_n^{r/l}$). To correct these disparity maps, our framework has been built in order to identify the disparity vectors which have been wrongly computed for each pair $\{I_n^l, I_n^r\}$ through view synthesis and image quality assessment. Then, we propose to use previously estimated long-term dense displacement fields to bring back accurate disparity estimates from disparity maps computed for other pairs $\{I_m^l, I_m^r\}$ with $m \neq n$. These new disparity estimates are used to replace the wrongly computed ones at instant n.

The proposed disparity correction framework includes two main stages:

1. Identification of the wrongly estimated disparity vectors through view synthesis and view synthesis quality assessment: Section 14.2.1
2. Disparity correction using long-term dense displacement fields used to bring back more accurate disparity proposals: Section 14.2.2

Each of these two main stages is made of several steps which are detailed below. This description is dedicated to the correction of left to right disparity maps. Note that an exactly similar processing can correct the right to left disparity maps.

14.2.1 Identification of the wrongly estimated disparity vectors

To identify the wrongly estimated disparity vectors for each pair of frames $\{I_n^l, I_n^r\}$, we propose the following steps:

1. Initial disparity estimation between each pair of frames $\{I_n^l, I_n^r\}$ performed by any disparity estimator ([RTDC12] for instance),
2. View synthesis in order to reconstruct $\forall n \in [\![0,\ldots,N]\!]$ I_n^r using the left view I_n^l through the disparity field $d_n^{l/r}$ estimated during step 1. The reconstructed right image obtained through this extrapolation procedure is referred to as $\tilde{I}_n^r$ in what follows,
3. View synthesis quality assessment using the *VSQA* metric (described in Section 6.1) which compares the reconstructed image $\tilde{I}_n^r$ and the source image I_n^r. It provides:
   (a) the localization of the view synthesis artifacts in $\tilde{I}_n^r$. The idea here is to identify the pixels $\tilde{I}_n^r$ for which the reconstruction has not been accurately performed. More precisely, this step outputs in a binary mask which indicates the pixels of $\tilde{I}_n^r$ whose *VSQA* quality value is below (resp. above) a threshold if the metric on which *VSQA* relies is a similarity metric (resp. a distortion metric). These pixels are considered

as inaccurate in terms of reconstruction. The threshold involved for this task is the same as the one used for the computation of the *VSQA* score (Section 6.1.5). It is described in Eq. 6.10.

(b) a global quality score (the *VSQA* score described in Section 6.1.5) which globally evaluates the quality of the view synthesis, i.e. the quality of the construction of $\tilde{I}_n^r$ for each instant n,

4. Identification of the disparity vectors of $d_n^{l/r}$ which have caused the view synthesis artifacts detected in step 3(a). In practice, for each wrongly reconstructed pixel $\boldsymbol{x}_n^r$ of $\tilde{I}_n^r$, we identify the disparity vectors coming from grid points of the left view I_n^l which end in the close neighbourhood of $\boldsymbol{x}_n^r$ (resp. $\boldsymbol{x}_n^l$). We consider that they have been used to extrapolate the color components of $\boldsymbol{x}_n^r$ in $\tilde{I}_n^r$ during step 2. Therefore, the identified disparity vectors need to be corrected.

Through the four previously described steps, we have been able to identify the wrongly estimated disparity vectors between each pair of frames $\{I_n^l, I_n^r\}$. The goal of the second main stage (Section 14.2.2) is now to refine these disparity vectors using long-term dense displacement fields which allow to bring back more accurate disparity proposals from other pairs of frames within the binocular sequence.

14.2.2 Disparity correction

For the disparity correction task, four steps are performed for each instant n:

1. We consider the VSQA scores computed in step 3(b) (Section 14.2.1) as a function of time. The resulting curve ($VSQA(n)$) indicates the temporal behavior of the overall view synthesis quality (the lower the *VSQA* score, the better the view synthesis quality). This curve is then used to select two reference pairs of frames for which the view synthesis has been performed with a relatively good quality. We have in mind the fact that the corresponding disparity maps will be used to refine the other ones within the binocular sequence. In practice, the choice of these two reference pairs of frames translates in identifying the instants for which the *VSQA* score is minimal. In the following, these two reference pairs of frames are referred to as $\{I_{ref_0}^l, I_{ref_0}^r\}$ and $\{I_{ref_1}^l, I_{ref_1}^r\}$. The following steps aim at correcting the disparity maps inside the interval $[\![ref_0, \ldots, ref_1]\!]$.

2. A long-term dense motion estimation (such as *StatFlow* described in Section 11.3) is performed with respect to both left reference frames $I_{ref_0}^l$ and $I_{ref_1}^l$. This translates in the computation of the following *to-the-reference* displacement fields $\forall n \in [\![ref_0+1, \ldots, ref_1-1]\!]$:

   - $\boldsymbol{d}_{n,ref_0}^l$ with respect to $I_{ref_0}^l$,
   - $\boldsymbol{d}_{n,ref_1}^l$ with respect to $I_{ref_1}^l$.

3. For each disparity vector $d_n^{l/r}$ starting from a grid point $\boldsymbol{x}_n^l$ of I_n^l and identified as inaccurate in step 4 (Section 14.2.1), we suggest to involve $\boldsymbol{d}_{n,ref_0}^l(\boldsymbol{x}_n^l)$ and $\boldsymbol{d}_{n,ref_1}^l(\boldsymbol{x}_n^l)$ to bring back accurate disparity vectors from $I_{ref_0}^l$ and $I_{ref_1}^l$ and finally propose an alternative disparity value to replace $d_n^{l/r}$. This procedure is illustrated in Fig. 14.2. The resulting new disparity vectors $\bar{d}_n^{l/r}$ are computed through linear interpolation, as shown in Eq. 14.3.

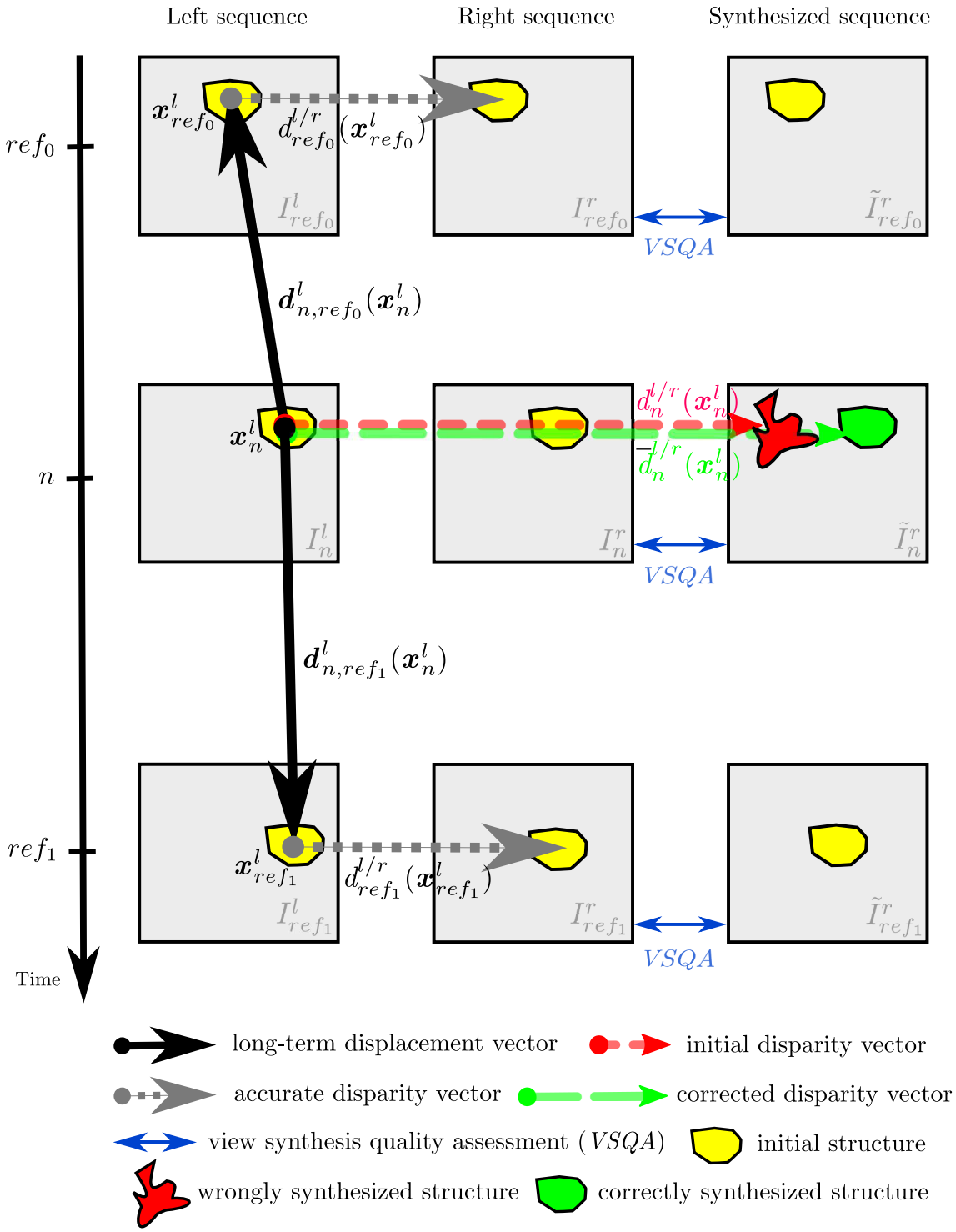


Figure 14.2: We propose a disparity correction framework which includes two main stages: 1) identification of the wrongly estimated disparity vectors through view synthesis and view synthesis quality assessment (Section 14.2.1), 2) disparity correction using long-term dense displacement fields used to bring back more accurate disparity proposals (Section 14.2.2) from two reference frames (in this case $I^l_{ref_0}$ and $I^l_{ref_1}$).

$$\overline{d}_n^{l/r}(\boldsymbol{x}_n^l) = w_{ref_0}(n) \times \tilde{d}_{ref_0}^{l/r}(\boldsymbol{x}_{ref_0}^l) + w_{ref_1}(n) \times \tilde{d}_{ref_1}^{l/r}(\boldsymbol{x}_{ref_1}^l) \tag{14.3}$$

where $\tilde{.}$ denotes an interpolated disparity. $\boldsymbol{x}_{ref_0}^l$ and $\boldsymbol{x}_{ref_1}^l$ are the corresponding points to $\boldsymbol{x}_n^l \in I_n^l$ respectively in $I_{ref_0}^l$ and $I_{ref_1}^l$ (Fig. 14.2). They are defined as follows:

$$\boldsymbol{x}_{ref_0}^l = \boldsymbol{x}_n^l + \boldsymbol{d}_{n,ref_0}^l(\boldsymbol{x}_n^l) \tag{14.4}$$

$$\boldsymbol{x}_{ref_1}^l = \boldsymbol{x}_n^l + \boldsymbol{d}_{n,ref_1}^l(\boldsymbol{x}_n^l) \tag{14.5}$$

The disparity values in Eq. 14.3 are weighted with respect to the temporal distances to the reference frames $I_{ref_0}^l$ and $I_{ref_1}^l$. The respective weights are defined in Eq. 14.6 and Eq. 14.7 assuming that $w_{ref_0}(n) + w_{ref_1}(n) = 1 \ \forall n \in [\![ref_0 + 1, \ldots, ref_1 - 1]\!]$.

$$w_{ref_0}(n) = \frac{n - ref_0}{ref_1 - ref_0} \tag{14.6}$$

$$w_{ref_1}(n) = \frac{ref_1 - n}{ref_1 - ref_0} \tag{14.7}$$

This disparity correction procedure is performed only if the displacement vectors $\boldsymbol{d}_{n,ref_0}^l$ and $\boldsymbol{d}_{n,ref_1}^l$ and the disparity vectors $\tilde{d}_{ref_0}^{l/r}$ and $\tilde{d}_{ref_1}^{l/r}$ are intrinsically consistent respectively according to the *from/to-the-reference* inconsistency values (Section 10.2.3) and the *left-right* inconsistency values [RTDC12]. These conditions ensure a good estimation quality for each one of the involved vectors (displacement or disparity) and finally limit the possibility of a wrong disparity correction. Indeed, this ensures that the new disparity proposals are brought back from accurate positions and that they have been initially accurately computed themselves.

4. Once all the wrongly estimated disparity vectors have been corrected, we can perform again both view synthesis and view synthesis quality assessment. If it shows that the view synthesis artifacts have been removed, it means that the disparity correction has been performed accurately. Fig. 14.2 illustrates the fact that the correction of the disparity vectors allows a better view synthesis of the initial yellow structure (compare red and green structure in $\tilde{I}_n^r$). To objectively evaluate how disparity maps have been corrected, we can refer again to both *VSQA* distortion maps and *VSQA* scores.

We suggest to apply this disparity correction framework for small temporal segments which means that the two reference pairs of frames must not be too far from each other. Indeed, our processing relies on a linear interpolation of the disparity vectors which is valid only for short pieces of the sequence since one can expect moderate disparity variations between two close frames. However, disparity correction can be applied several times by cutting the sequence into small temporal segments according to the temporal evolution of the *VSQA* scores across the sequence.

14.3 Experimental evaluation of the proposed disparity correction framework

This section presents a very early evaluation of the disparity correction framework described in Section 14.2. The experiments focus on two temporal sections of the *Book-Arrival* binocular sequence (Fig. 14.3): between $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$ and between $\{I^l_{55}, I^r_{55}\}$ and $\{I^l_{97}, I^r_{97}\}$. As shown in Fig. 14.3, the *Book-Arrival* sequence contains large uniform areas as well as thin structures, large motion with fixed camera.

Protocol

Whatever the temporal section under consideration, an initial disparity estimation is performed between each pair of frames $\{I^l_n, I^r_n\}$ by the disparity estimator of [RTDC12] described in Section 3.2 (Chapter 3). The experiment consists in demonstrating that the resulting initial disparity maps can be improved by the disparity correction framework proposed in Section 14.2.

The study of view synthesis quality across the sequence through *VSQA* scores (blue curve of Fig. 14.6) has first of all led to the selection of the reference pairs of frames $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$. Therefore, we focus on the temporal section located between $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$. $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$ have been selected as reference pairs of frames because the corresponding instants are characterized by an accurate view synthesis in terms of reconstruction quality with respect to other frames.

After the reference pairs selection, long-term motion estimations have been run starting from I^l_{69} in the *forward* direction and from I^l_{84} in the *backward* direction using the *Statistical Multi-Step Flow* approach (*StatFlow*) proposed in Chapter 11 (Section 11.3).

Through the procedure described in Section 14.2, we are able to identify the wrongly estimated disparity vectors within each disparity map $d^{l/r}_n$ with $n \in [\![70, \ldots, 83]\!]$. The *to-the-reference* long-term displacement vectors computed for the left sequence are used to correct the identified inaccurate disparity vectors by bringing back more accurate disparity information from $d^{l/r}_{69}$ and $d^{l/r}_{84}$ respectively defined with respect to I^l_{69} and I^l_{84}.

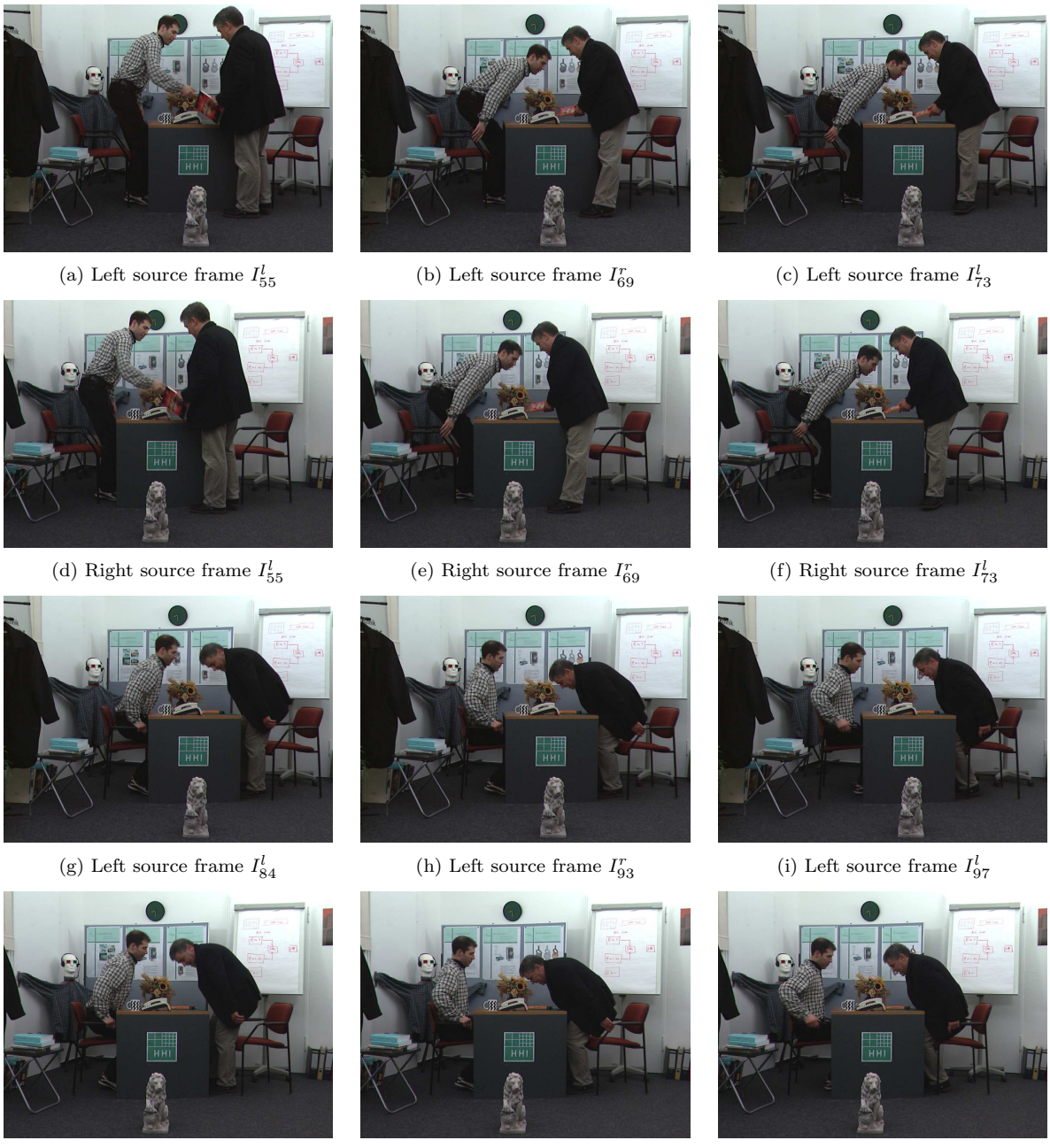
As mentioned in Section 14.2.2 (step 4), we refer in this experimental section to both *VSQA* scores and view synthesis results to assess how the disparity maps have been corrected using our framework. Therefore, we compare the results obtained before and after disparity correction. Such comparisons are provided in Fig. 14.4 (*VSQA* scores) and Fig. 14.5 (view synthesis results).

Results through *VSQA* scores between $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$

Fig. 14.4 shows that the proposed disparity correction framework allows to reduce the number of erroneous pixels (information given by the *VSQA* score) for all pairs except for $\{I^l_{80}, I^r_{80}\}$. For the pair $\{I^l_{81}, I^r_{81}\}$ for instance, 1679 pixels of $\tilde{I}^r_{80}$ have been refined in terms of view synthesis quality according to our *VSQA* quality metric. This corresponds to 6.88% of the total number of erroneous pixels.

Results through view synthesis between $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$

Fig. 14.5 presents the correction of the disparity map $d^{l/r}_{73}$ initially computed between I^l_{73} (Fig. 14.5 (a)) and I^r_{73} (Fig. 14.5 (b)). Fig. 14.5 (c) and (d) respectively show the synthesized right view obtained using the initial disparity estimation and the thresholded *VSQA* mask



(a) Left source frame I^l_{55} (b) Left source frame I^r_{69} (c) Left source frame I^l_{73}

(d) Right source frame I^l_{55} (e) Right source frame I^r_{69} (f) Right source frame I^l_{73}

(g) Left source frame I^l_{84} (h) Left source frame I^r_{93} (i) Left source frame I^l_{97}

(k) Right source frame I^l_{84} (l) Right source frame I^r_{93} (m) Right source frame I^l_{97}

Figure 14.3: Source frames of the *Book-Arrival* binocular sequence.

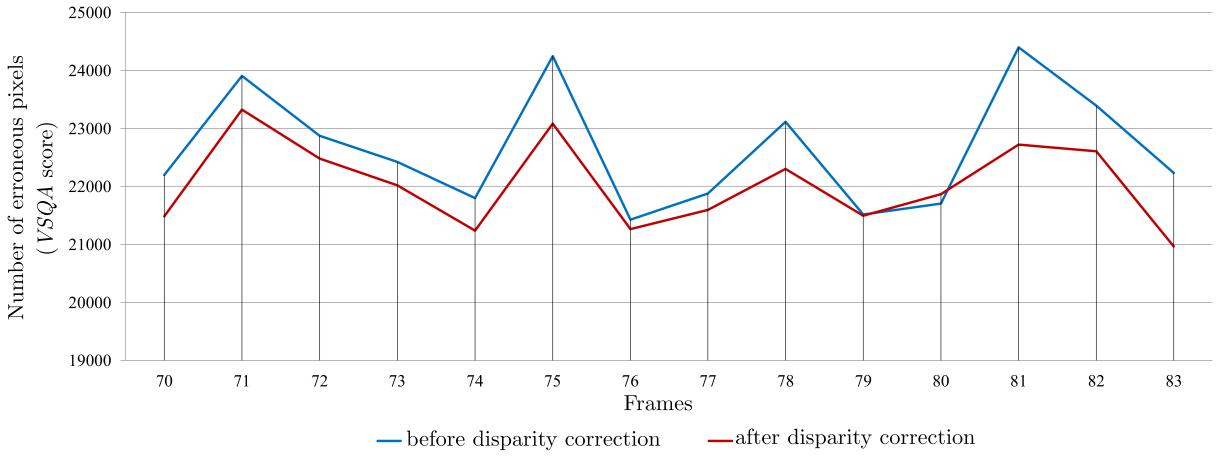
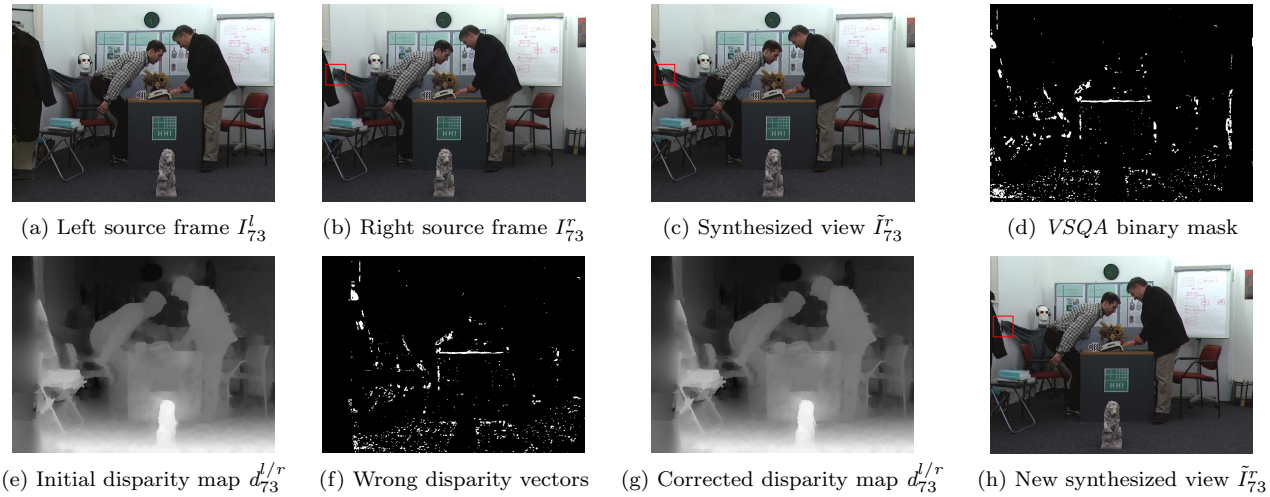


Figure 14.4: Comparison between the *VSQA* curves (built using *VSQA* scores) obtained before and after disparity correction (*Book-Arrival* binocular sequence). The two reference pairs of frames used for this experiment are as follows: $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$.



(a) Left source frame I^l_{73} (b) Right source frame I^r_{73} (c) Synthesized view $\tilde{I}^r_{73}$ (d) *VSQA* binary mask

(e) Initial disparity map $d^{l/r}_{73}$ (f) Wrong disparity vectors (g) Corrected disparity map $d^{l/r}_{73}$ (h) New synthesized view $\tilde{I}^r_{73}$

(i) Zoom on I^r_{73} (j) Zoom on $\tilde{I}^r_{73}$ **before disparity correction** (k) Zoom on $\tilde{I}^r_{73}$ **after disparity correction**

Figure 14.5: Accuracy of the proposed disparity correction framework (Section 14.2) assessed through view synthesis.

Figure 14.6: Comparison between the *VSQA* curves (built using *VSQA* scores) obtained before and after disparity correction (*Book-Arrival* binocular sequence). The two reference pairs of frames used for this experiment are as follows: $\{I^l_{55}, I^r_{55}\}$ and $\{I^l_{97}, I^r_{97}\}$.

coming from the comparison of the synthesized right image ($\tilde{I}^l_{73}$, Fig. 14.5 (c)) with the original right image (I^l_{73}, Fig. 14.5 (b)). This *VSQA* mask indicates in white the wrongly synthesized areas according to the *VSQA* quality metric (with respect to the right view). The disparity vectors of $d^{l/r}_{73}$ which have caused these view synthesis artifacts are identified in the left view (white pixels of the binary mask, Fig. 14.5 (f)) and corrected using the initial disparity maps estimated for the reference pairs of frames: $d^{l/r}_{69}$ and $d^{l/r}_{84}$.

To assess how the disparity maps have been corrected, we can compare Fig. 14.5 (e) and (g) which provide the disparity map $d^{l/r}_{73}$ respectively before and after correction and Fig. 14.5 (c) and (h) which give the synthesized view respectively before and after correction. We focus especially on a small area located within the coat hooked on the coat rack (Fig. 14.5 (i), red box in Fig. 14.5 (b)). By comparing the synthesized patches Fig. 14.5 (j) (before disparity correction) and (k) (after disparity correction), we notice that the reconstruction of the area is clearly better after having applied the disparity correction. Although not totally perfect, the discontinuity between the coat and the background is more accurately delimited thanks to more accurate disparity vectors coming from $d^{l/r}_{69}$ and $d^{l/r}_{84}$.

Results through *VSQA* scores between $\{I^l_{55}, I^r_{55}\}$ and $\{I^l_{97}, I^r_{97}\}$

In the same spirit of Fig. 14.4, Fig. 14.6 displays the *VSQA* curves before and after disparity correction for the temporal segment located between $\{I^l_{55}, I^r_{55}\}$ and $\{I^l_{97}, I^r_{97}\}$ ($\{I^l_{55}, I^r_{55}\}$ and $\{I^l_{97}, I^r_{97}\}$ have been selected as reference pairs of frames). We can draw the same conclusions with this larger segment: the disparity correction framework is able to reduce the number of pixels which have been wrongly synthesized. For the pairs $\{I^l_{67}, I^r_{67}\}$, the number of erroneous pixels has decreased of 1888 after having applied the proposed disparity correction which corresponds to 8.47% with respect to the total number of erroneous pixels.

Discussion

This early evaluation of the disparity correction framework proves in both cases (between $\{I^l_{69}, I^r_{69}\}$ and $\{I^l_{84}, I^r_{84}\}$ and between $\{I^l_{55}, I^r_{55}\}$ and $\{I^l_{97}, I^r_{97}\}$) a slight improvement of the view synthesis quality which denotes an accurate correction of the disparity maps. However, we notice that a significant number of erroneous pixels remains after disparity correction. This reveals the main disadvantage of the proposed automatic framework, outlined as follows: how to ensure that a given distorted area in $\tilde{I}^r_n$ with $n \in [\![ref_0 + 1, \ldots, ref_1 - 1]\!]$ has been accurately synthesized for the two reference pair of frames from which we aim at bringing back more consistent disparity information?

To improve the results and therefore to increase the number of corrected pixels, we propose two ideas which would deserve further investigation:

1. The first point deals with the reference selection process and suggest to select the best reference pairs of frames for each distorted area to be corrected. Through this approach, the idea would be to bring back more accurate disparity information from multiple reference pairs of frames instead of relying on only one single reference pair of frames. Thus, instead of focusing on a global quality as done with the *VSQA* score, such method would require a study of how evolves temporally the quality of each distorted area in order to finally identify the pair of frames that best suits for the disparity correction task (i.e. the pair of frames for which the synthesis has been performed with the best possible quality). Although more computationally expensive since it requires additional long-term dense motion estimations (two more for each inserted reference pair of frames), this method could allow a more efficient correction of the wrongly computed disparity vectors.

2. The second idea consists in extending the proposed framework to a semi-automatic approach. Instead of temporally propagating already computed disparity vectors considered as consistent to correct wrongly estimated disparity vectors, we could in the same spirit propagate disparity information that would have been manually corrected by an operator. The *VSQA* metric can of course guide the operator in the examination of the sequence of results.

14.4 Conclusion and perspectives

Motion and disparity information are rarely involved together for solving computer vision tasks. When simultaneously considered, joint stereo and motion processing are generally restricted to quadruplets of images which are processed sequentially across the binocular sequence. Following the approach proposed in [CLD11] where long-term displacement fields are involved in the context of 2D-to-3D conversion, we proposed in this chapter a new disparity correction framework which includes disparity estimation, view synthesis, view synthesis quality assessment (Part I) as well as long-term dense motion estimation (Part II).

For each pair of left and right frames belonging to a given binocular sequence, our disparity correction framework is able to classify the estimated disparity vectors between accurate and inaccurate vectors. This identification stage precedes a correction stage where the wrongly estimated disparity vectors are corrected by bringing back more accurate disparity vectors from reference pairs of frames using *to-the-reference* long-term displacement vectors. The reference pairs of frames can be selected by studying the temporal evolution of the view synthesis quality using a image quality metric such as *VSQA* (Chapter 6).

The early experimental results have shown through *VSQA* scores and view synthesis results that our approach can improve disparity maps. Nevertheless, it appears that is very important to rely on reference pairs of frames for which the disparity maps have been accurately estimated since this information is used to correct to whole sequence. Unfortunately, our experiments have shown that a significant number of wrongly synthesized pixels still remains erroneous after disparity correction. As suggested in Section 14.3, the results could be improved by considering one of the two following methods:

1. identify reference pairs of frames per wrongly synthesized area instead of relying only on a single reference pair of frames for the whole correction process,

2. turn the automatic framework into a semi-automatic approach in order to temporally propagate manual user corrections. In the same spirit than the semi-automatic 2D-to-3D conversion framework of [CLD11], the operator could through this suggested semi-automatic approach manually correct the disparity maps (or the corresponding view synthesis artifacts visible in the synthesized views) estimated between the reference pairs of frames. These corrections could be then automatically propagated across the sequence through the previously computed long-term dense displacement fields.

Another point would deserve further investigation. In Section 14.2, the idea was to use *to-the-reference* displacement vectors to bring back accurate disparity vectors from reference pairs of frames. An alternative could consist in involving *from-the-reference* displacement vectors (and therefore trajectories) in order to study how the disparity estimates temporally evolve across the sequence. In this context, two directions can be followed:

1. We could imagine to detect wrongly estimated disparity vectors by checking the temporal evolution of the disparity values with respect to a temporal disparity range. Sudden changes in terms of disparity values could alert the automatic processing or an operator (in a semi-automatic context) to a probable estimation error.

2. To further exploit the long-term trajectories, we could both build a model to describe the temporal behavior of the disparity estimates and refine the disparity vectors by constraining them to lie near the resulting model. As done in [GRA13] for displacement vectors,

the model could consist in a low-dimensional subspace built from accurately estimated disparity vectors.

Additionally to the disparity correction procedure, a post-processing step can be considered to perform a further refinement of the disparity maps following a multilateral spatio-temporal filtering-based approach. The idea is to spatio-temporally diffuse the corrections while strengthening the consistency along trajectories. Toward this goal, we can extend the multilateral spatio-temporal filtering presented in Chapter 10 (Section 10.2.3) and dedicated to the refinement of long-term displacement fields to refine again the disparity maps. The trajectory similarity feature involved within the proposed multilateral spatio-temporal filtering could in particular have a great impact on the disparity refinement quality.

Finally, similarly to the proposed disparity correction method using long-term motion estimates, it is conceivable for stereoscopic video setups to consider a long-term displacement correction process using pre-computed disparity correspondences to give additional constraints. Compared to single long-term dense motion estimators, this can help to resolve some matching ambiguities such as the ones which may occur in case of non-rigid deformations, large motion, zooming, large poorly textured areas, transparency, occlusions or illumination changes.

More generally, we can think directly about a long-term joint disparity and motion estimator whose goal would be to alternately estimate disparity from long-term displacement and long-term displacement from disparity. In this direction, the idea would consist in extending into long-term methods classical joint disparity and motion estimators which usually process quadruplets of images in a sequential way.

Bibliography Part III

[CLD11] X. Cao, Z. Li, and Q. Dai. Semi-automatic 2D-to-3D conversion using disparity propagation. *IEEE Transactions on Broadcasting*, 57(2):491–499, 2011.

[DMPP10] I. Daribo, W. Miled, and B. Pesquet-Popescu. Joint depth-motion dense estimation for multiview video coding. *Journal of Visual Communication and Image Representation*, 21(5-6):487–497, 2010.

[DSMNP01] L. Di Stefano, S. Mattoccia, G. Neri, and D. Piccinni. Temporal filtering of disparity measurements. In *11th International Conference on Image Analysis and Processing*, pages 145–150. IEEE, 2001.

[Gon06] M. Gong. Enforcing temporal consistency in real-time stereo estimation. *European Conference on Computer Vision*, pages 564–577, 2006.

[GRA13] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 2013.

[HD07] F. Huguet and F. Devernay. A variational method for scene flow estimation from stereo sequences. In *Computer Vision, IEEE International Conference on*, pages 1–7. IEEE, 2007.

[HRBG12] A. Hosni, C. Rhemann, M. Bleyer, and M. Gelautz. Temporally consistent disparity and optical flow via efficient spatio-temporal filtering. In *Advances in Image and Video Technology*, pages 165–177. Springer, 2012.

[LH10] S. Lee and Y. Ho. Temporally consistent depth map estimation using motion estimation for 3DTV. In *International Workshop on Advanced Image Technology*, page 149, 2010.

[MDC07] J. Morat, F. Devernay, and S. Cornou. Tracking with stereo-vision system for low speed following applications. In *Intelligent Vehicles Symposium*, pages 955–961. IEEE, 2007.

[MPPC09] W. Miled, B. Pesquet-Popescu, and W. Cherif. A variational framework for simultaneous motion and disparity estimation in a sequence of stereo images. In *Acoustics, Speech and Signal Processing, IEEE International Conference on*, pages 741–744. IEEE, 2009.

[PXZ+10] L.-M. Po, X. Xu, Y. Zhu, S. Zhang, K.-W. Cheung, and C.-W. Ting. Automatic 2D-to-3D video conversion technique based on depth-from-motion and color segmentation. In *IEEE International Conference on Signal Processing*, pages 1000–1003, 2010.

[RSK+12] S. Ryu, J. Seo, D.H. Kim, J.Y. Lee, H.C. Wey, and K. Sohn. An independent motion and disparity vector prediction method for multiview video coding. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, volume 8288, page 828826, 2012.

[RTDC12] P. Robert, C. Thébault, V. Drazic, and P.-H. Conze. Disparity-compensated view synthesis for s3D content correction. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.

[SWWES10] H. Sun, C. Wang, B. Wang, and N. El-Sheimy. Independently moving object detection and tracking using stereo vision. In *IEEE International Conference on Information and Automation*, pages 1936–1941, 2010.

[VB07] C. Varekamp and B. Barenbrug. Improved depth propagation for 2D to 3D video conversion using key-frames. *Visual Media Production, IET Conference on*, 2007.

[WBV+11] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers. Stereoscopic scene flow computation for 3D motion understanding. *International Journal of Computer Vision*, 95(1):29–51, 2011.

[YNLS05] W. Yang, K. Ngan, J. Lim, and K. Sohn. Joint motion and disparity fields estimation for stereoscopic video sequences. *Signal Processing: Image Communication*, 20(3):265–276, 2005.

Chapter 15

General conclusion

This thesis has been dedicated to view synthesis quality assessment and long-term dense motion estimation. The main issues were first to create a new image quality assessment metric dedicated to the detection of view synthesis artifacts and second, to propose robust strategies to establish long-term dense motion correspondences across video sequences. Both fields have been finally considered together toward high-quality joint disparity and long-term motion processing.

Based on the study of the general principles and state-of-the-art algorithms in each of these fields, we proposed several contributions and perspectives which are summarized and discussed in what follows. Each of the algorithms mentioned below has been implemented in *C++* language and merged within a multi-user software environment of *Technicolor* dedicated to computer vision algorithms.

View synthesis quality assessment

In part I, we investigated the distortions caused by both disparity estimation and view synthesis involved together within *Depth-Image-Based Rendering* (*DIBR*) algorithms whose aim is to synthesize new viewpoints of 3D scenes from available captured views [1]. Based of the finding that these distortions can seriously impair the viewing experience when displaying content through 3D autosteroscopic displays, we analyzed the variety of situations for which the image rendering process does not lead to artifact-free and realistic-looking synthesized views.

Our review of state-of-the-art image quality assessment methods for both monoscopic and stereoscopic content has revealed that attempts to detect view synthesis artifact through traditional metrics or dedicated methods are not numerous and do not work satisfactorily. In this context, our main contribution has been to propose a new full-reference objective image quality assessment metric dedicated to view synthesis quality assessment [2] whose correlation with subjective measurement is much higher than with state-of-the-art metric.

This new metric, called the *View Synthesis Quality Assessment* (*VSQA*) metric, involves three features whose impact in terms of artifact masking is preponderant: the complexity in terms of textures, the diversity of gradient orientations as well as the presence of high contrast. These features are formalized into three corresponding visibility maps which weight an initial quality map obtained by any existing metric. Following such procedure, we are able to both indicate the exact position of view synthesis artifacts within synthesized views and give an information about the overall view synthesis quality through the *VSQA* score.

An overall performance comparison between *SSIM*-based *VSQA* and existing quality metrics has been performed using the *IRCCyN/IVC DIBR* images database provided with correspond-

ing subjective scores. According to the results, *SSIM*-based *VSQA* shows a better ability to assess the perceptible synthesized image quality compared to state-of-the-art methods with a correlation coefficient with respect to subjective measurements which exceeds 61%.

In terms of perspectives, *VSQA* could be more thoroughly investigated by applying the *VSQA* procedure to other existing metrics such as *MS-SSIM* or *NQM*. Moreover, our view synthesis quality assessment approach could be improved by including a specific quality assessment of disoccluded regions (i.e regions occluded in the original views which become visible in the synthesized view) since sophisticated *DIBR* algorithms usually replace the missing image information by meaningful color information either through inpainting or filtering of the synthesized view itself or by pre-processing the disparity maps in a way that no disocclusions appear in the synthesized view.

Another point which would deserve further investigation deals with the extension of *VSQA* into a video quality assessment metric in order to take into account temporal fluctuations of spatial distortions. In this context, how to model the *Human Visual System* (*HVS*) mechanisms of the perception of temporal distortions is a crucial issue since the perception of spatial distortions over time can be significantly modified by their temporal changes.

Long-term dense motion estimation

Part II, the main part of this thesis, has been dedicated to long-term dense motion estimation. Motivated by recent applications such as spatio-temporal scene segmentation, dynamic scene analysis techniques, high quality automatic or semi-automatic video editing, long-term trajectory manipulation or motion magnification tasks, we focused our study toward the goal of establishing dense and long-term correspondences across monoscopic video sequences through the computation of long-term dense displacement/trajectory fields.

State-of-the-art *optical flow* estimators are usually robust to compute dense motion fields between consecutive frames. However, they show strong limitations when applied to distant frames since classical *optical flow* assumptions such as the *color constancy assumption* are usually not valid for non-consecutive frames. This is especially true for complex scenes (non-rigid deformations, large motion, strong occlusions, poorly textured areas, transparency...) or for significant video content changes in time (zooming, illumination changes...). Recent methods have contributed to the purpose of long-term dense motion estimation but the resulting trajectories usually diverge and do not exceed more than about thirty frames. In addition, relying only on motion fields computed between consecutive frames does not allow to recover trajectories after temporary occlusions.

Toward the goal of limiting the motion drift while dealing with temporary occlusions, we proposed several contributions for a robust computation of long-term dense displacement/trajectory fields. Based on the concept of *multi-step* elementary *optical flow* estimation whose aim is to consider *optical flow* fields computed both between consecutive frames and with larger inter-frame distances (to be able to jump temporary occlusions), the proposed strategies focused on a robust estimation of both *from-the-reference* and *to-the-reference* long-term displacement fields defined with respect to a previously selected reference frame. *From-the-reference* displacement fields link the reference frame to the other frames of the sequence (and therefore describe the trajectory of each pixel of the reference frame along the sequence) contrary to *to-the-reference* displacement fields which connect each pixel of each frame of the sequence to locations into the reference frame.

In this context, we first of all proposed two sequential methods, respectively called *Multi-Step flow* via *Graph-Cuts* (*MS-GC*) [3] and *Multi-Step Fusion flow* (*MSF*) [4], which are based on two following main stages: 1) accumulation of *multi-step* elementary *optical flow* vectors through inverse integration [8], 2) optimal merge of the resulting candidate long-term displacement fields. Compared to *MS-GC*, *MSF* performs a combination of bi-directional *multi-step* elementary *optical flow* fields instead of unidirectional as well as a different long-term displacement selection procedure based on the *fusion moves* algorithm (instead of a *Graph Cuts*-based selection for *MS-GC*). In addition, *MSF* includes a promising multilateral spatio-temporal filtering stage which iteratively refines the long-term displacement fields along trajectories.

An alternative approach has been explored based on the construction of multiple *multi-step* motion *paths* across the sequence through combinatorial integration. Given the resulting large set of motion candidates, we apply then a statistical-based selection procedure where a global optimization stage is preceded by a new statistical processing exploiting both the spatial distribution and the intrinsic quality of candidates. The whole combinatorial *multi-step* integration and statistical selection (*CISS*) framework has been first applied to a pair of distant frames [5, 6]. Promising results have motivated its extension to the whole sequence in order to reach a long-term dense motion estimator referred to as *Statistical multi-step Flow* (*StatFlow*) [7]. *StatFlow* relies on an improved and extended version of *CISS* (called *CISS-K*) which is applied independently between the reference frame and each of the subsequent images of the sequence, followed by a new iterative motion refinement (*IMR*) stage which gives a final dense matching while enforcing temporal smoothness.

Instead of relying only on one single reference frame, we also investigated multi-reference frames strategies whose underlying idea was to correlate reliable pieces of trajectories estimated with respect to different reference frames in order to reach very long-term requirements. In this direction, we proposed a first multi-reference frames strategy based on the insertion of new reference frames each time the trajectories diverge. Additionally to this method, we designed a two-reference frames motion refinement framework combining both *forward* and *backward* trajectory fields estimated from two distant reference frames. This latter method includes especially a robust inter-reference frames motion refinement stage which can be extended to the whole sequence assuming that a long-term dense motion estimator has been initially applied starting from each frame of the sequence.

All these single and multi-reference frames strategies have been qualitatively and quantitatively assessed through many experiments including texture insertion and propagation, point tracking, registration and *PSNR* assessment and comparisons with sparse and dense ground-truth trajectories for a wide set of complex scenes. With respect to state-of-the-art methods, the evaluation of each of the proposed approaches has shown a clear improvement in terms of accuracy and robustness of the resulting long-term *from-the-reference* and *to-the-reference* trajectory/displacement fields.

To enhance our single and multi-reference frames strategies toward a increasingly accurate long-term dense motion estimation, several aspects must deserve more attention for further research including:

- an automatic selection of the input set of candidate *steps* depending on the considered video sequence. Through this suggestion, we aim at accurately handling temporary occlusions by providing input elementary *optical flow* fields whose corresponding inter-frame distances would have been appropriately estimated without interaction with the operator,

- a more sophisticated automatic selection of reference frames in order to automatically identify the smallest set of frames that contains all the regions visible along the shot of a given object,

- a robustification of the long-term displacement fields selection task by introducing new displacement field selection criteria such as gradient similarity measures, correlation estimates or gain factors to more accurately handle strong variations of illumination through gain-compensated matching cost or gain-based regularization for instance,

- the introduction of different types of motion information as inputs of our strategies. In particular, we could imagine to simultaneously take into account block matching based estimates, parametric or non-parametric motion fields computed through the estimation of rigid or deformable models, sparse features using tools such as *KLT*, *SIFT* or *SURF*, *optical flow* fields coming from different estimators...

- a robust occlusion and discontinuity management since a relevant dense long-term motion estimation requires a robust occlusion detection in order to identify exactly which areas remain visible and which ones are occluded along the sequence,

- an extended motion candidate construction using both *forward* and *backward multi-step* elementary *optical flow* fields to allow bi-directional motion *paths* or/and *optical flow* fields that join frames that are outside the interval delimited by the pair of frames under consideration,

- a stronger focus of critical cases such as zooming, transparency, or specific motion such as rotational motion,

- semi-automatic interactions instead of fully-automatic processing, in particular through:

    - user-assisted motion estimation for which the operator interacts directly with motion fields in order to improve image correspondences *a posteriori*,
    - an *a priori* approach where the operator provides to the automatic motion estimation process information such as bounding boxes or rough manual segmentation.

In a larger scale, our contributions offer new perspective toward a complete coverage of all the visible points in the whole video sequence. By recursively identifying a set of reference frames for each area or object belonging to the sequence and by applying one of our long-term dense motion estimation strategies starting from these identified reference frames, we can reach a very compact representation of the video content with associated long-term motion behavior.

Application to joint stereo and motion processing

Simultaneously manipulating disparity and long-term dense motion estimates in the context of stereo or multi-view setups offers new perspectives related to joint stereo and long-term motion processing:

- disparity estimation or disparity correction using pre-computed long-term dense displacement fields. Long-term dense displacement vectors can be used to:

    - provide additional constraints (such as displacement or trajectory similarity constraints) as well as new candidates for disparity estimation,

  - accurately propagate estimated or manually corrected disparity vectors to correct wrongly estimated disparity vectors in neighboring frames,

- long-term dense motion estimation using pre-computed disparity fields which can provide, as previously, additional constraints (such as disparity similarity constraints to be combined with already used brightness constancy and inconsistency constraints) as well as new motion candidates,

- joint disparity and long-term motion estimation. By alternatively estimating disparity from long-term displacement and long-term displacement from disparity, we can reach a long-term description of both spatial and temporal dynamics of the 3D scene content.

By analyzing the state-of-the-art of joint stereo and motion analysis, we reported that existing algorithms are generally restricted to quadruplets of images, sequentially processed across the binocular or multiview sequence. We propose to go further in order to adapt each of the previously described perspectives to long-term processing. In terms of applications, we can mention in this context:

- view synthesis artifacts removal which is linked to the perspective of correcting disparity vectors thanks to pre-computed long-term dense displacement fields,

- automatic or semi-automatic stereo video editing which consists, once joint disparity and long-term motion estimation has been performed, in propagating any modification (texture, logo, segmentation labels...) along the binocular or multi-view video sequence using both disparity and long-term displacement fields. This requires in practice an automatic or manual modification in only one frame of one of the available views before automatic propagation to the whole binocular or multi-view video shot.

Among all these perspectives, only an automatic disparity correction using pre-computed long-term dense displacement fields has been thoroughly investigated in Part III. The resulting disparity correction framework has the advantage of involving disparity estimation, view synthesis, view synthesis quality assessment (addressed in Part I) as well as long-term dense motion estimation (studied in Part II). Through this framework, we proposed to perform two main steps:

1. classification of the disparity vectors of each pair of left and right frames belonging to the binocular sequence between accurate and inaccurate disparity vectors. This is performed through both view synthesis and view synthesis quality assessment using the *VSQA* metric,

2. disparity correction stage where the wrongly estimated disparity vectors are corrected by bringing back, via pre-computed *to-the-reference* long-term displacement vectors, more accurate disparity vectors from two reference pairs of frames identified using the *VSQA* global quality score.

Such disparity correction process can be applied to remove view synthesis artifacts. It could be enhanced by:

- identifying reference pairs of frames per wrongly synthesized area,

- turning the automatic framework into a semi-automatic approach in order to temporally propagate manual user disparity corrections,

- involving pre-computed *from-the-reference* long-term displacement vectors to:
    - check the temporal evolution of the disparity values with respect to a temporal disparity range,
    - build a model to describe the temporal behavior of the disparity estimates and refine the disparity vectors by constraining them to lie near the resulting model,
- adding a post-processing step to perform a further refinement of the disparity maps following a multilateral spatio-temporal filtering-based approach.

To conclude this thesis, the establishment of dense correspondences either between slightly different viewpoints or between images captured at different moments in time involves very challenging issues. Both contexts have been respectively investigated through view synthesis quality assessment and dense long-term motion estimation. Open issues still remain despite good performance obtained through the proposed contributions. Application to high-quality joint stereo and long-term motion processing offers new perspectives which would require further investigation.

Publications

[1] P. Robert, C. Thébault, V. Drazic, and **P.-H. Conze**. Disparity-compensated view synthesis for s3D content correction. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.

[2] **P.-H. Conze**, P. Robert, and L. Morin. Objective view synthesis quality assessment. In *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.

[3] T. Crivelli, **P.-H. Conze**, P. Robert, and P. Pérez. From optical flow to dense long term correspondences. In *IEEE International Conference on Image Processing*, 2012.

[4] T. Crivelli, **P.-H. Conze**, P. Robert, M. Fradet, and P. Pérez. Multi-step flow fusion: Towards accurate and dense correspondences in long video shots. In *British Machine Vision Conference*, 2012.

[5] **P.-H. Conze**, T. Crivelli, P. Robert, and L. Morin. Dense motion estimation between distant frames: Combinatorial multi-step integration and statistical selection. In *IEEE International Conference on Image Processing*, 2013.

[6] **P.-H. Conze**, T. Crivelli, P. Robert, and L. Morin. Estimation de mouvement dense entre images distantes: Intégration combinatoire multi-steps et sélection statistique. In *GRETSI Symposium on Signal and Image Processing*, 2013.

[7] **P.-H. Conze**, T. Crivelli, P. Robert, and L. Morin. Dense long-term motion estimation via statistical multi-step flow. In *Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2014.

[8] T. Crivelli, M. Fradet, **P.-H. Conze**, P. Robert, and P. Pérez. Robust optical flow integration. *Transactions on Image Processing*, 2014. *To appear.*

Patents

[9] **P.-H. Conze**, P. Robert, T. Crivelli, and L. Morin. Method and device for motion estimation, *EP 13306076* (European Patent), 2014.

[10] P. Robert, **P.-H. Conze**, and T. Crivelli. Automatic selection of reference frames for video editing, *14305485.6*.

[11] M. Fradet, **P.-H. Conze**, and J. Llach. Method for smoothing homography series. Procédé d'édition d'un plan dans une séquence vidéo, *1354738*.

[12] P. Robert, T. Crivelli, **P.-H. Conze**, M. Fradet, and P. Pérez. Filtering a displacement field between video frames. Filtrage d'un champ de déplacement entre des trames vidéo, *WO2013131819 A1* (International), 2013.

[13] P. Robert, V. Drazic, **P.-H. Conze**, and T. Viellard. Disparity maps in uniform areas, *US20130176300 A1* (U.S. Patent), 2013.

[14] P. Robert, T. Crivelli, and **P.-H. Conze**. Method and device for generating a motion field for a video sequence. Procédé et dispositif permettant de générer un champ de mouvement pour une séquence vidéo, *WO2013107833 A1* (International), 2013.

[15] P. Robert, M. Fradet, and **P.-H. Conze**. Method and apparatus for processing occlusions in motion estimation, *US 20130148730 A1* (U.S. Patent), *EP 2602997 A1* (European Patent), 2013.

Appendix *A*: Description of the disparity estimator of [1][1]

Numerous robust approaches have been proposed to solve the matching ambiguities in disparity estimation. Among them, let us give an overview of the disparity estimation method presented in [1]. This description allows us to provide an example of implementation of the general concepts described in Section 3.2.2 (Chapter 3), inherent to the disparity estimation process. Moreover, the disparity estimator of [1] is involved in many experiments of this thesis.

The disparity estimator proposed in [1] is a stereo algorithm that relies on the four following constraints: minimal correspondence cost through luminance similarity, smoothness constraint, consistency constraint and visibility constraint. In particular, the similarity evaluation is based on the *Normalized Cross-Correlation* (*NCC*) and smoothness constraint is introduced through joint bilateral and trilateral filtering applied on the disparity maps. Both left and right disparity maps are symmetrically estimated under consistency and visibility constraints.

This symmetric stereo algorithm provides for each view a dense disparity map with 1/4 pixel accuracy and an associated occlusion map which indicates which pixels of the current view are occluded in the other view. More precisely, it is made of three main stages:

1. hierarchical block-based estimation,
2. disparity pixel-wise assignment,
3. dense disparity refinement.

Each of them are briefly described in what follows. In particular, the description of the dense disparity refinement (step 3) introduces the concepts of occlusion detection and inconsistency computation, also involved in our contributions in dense long-term motion estimation (Part II).

Hierarchical block-based estimation

This first step combines a classical hierarchical block-based method to deal with large disparity range and a recursive filtering-based regularization to obtain smoother results, especially within textureless regions.

This hierarchical estimation relies on an iterative coarse to fine algorithm that operates on an image pyramid where estimated disparity vectors at a given level are used to constrain a

[1] refers to the list of publications, page 303

more local search for the next finer level. The aggregation of the matching costs is based on *NCC* computed on luminance signal.

Joint bilateral filtering is introduced at each of the levels of the hierarchical block-based estimator and more precisely before transmission of the current disparity map to the next finer level. The filtering encourages the blocks with similar luminance to have similar disparity, as described below:

$$\widehat{d}(\boldsymbol{x}) = \frac{\sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}.d(\boldsymbol{y})}{\sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}} \tag{15.1}$$

where $\boldsymbol{x}$ is the current block, $d(\boldsymbol{x})$ is the disparity value at block $\boldsymbol{x}$, $\boldsymbol{y}$ is a block among the blocks centered around $\boldsymbol{x}$ (i.e. in the neighborhood of $\boldsymbol{x}$, defined as $\mathcal{N}(\boldsymbol{x})$). $w_{\boldsymbol{xy}}$ is the weight assigned to the disparity value of block $\boldsymbol{y}$ which is computed through luminance differences between the blocks $\boldsymbol{x}$ and $\boldsymbol{y}$ and *Euclidean* distance between the centers of the blocks $\boldsymbol{x}$ and $\boldsymbol{y}$. $\widehat{d}$ corresponds to the filtered disparity value. In practice, the blocksize is for this step $N \times N$ with $N = 11$.

Finally, the filtered disparity value $\widehat{d}$ for block $\boldsymbol{x}$ is selected only if its matching cost (referred to as $C(\boldsymbol{x}, \widehat{d}(\boldsymbol{x}))$) is higher than the cost before filtering ($C(\boldsymbol{x}, d(\boldsymbol{x}))$) plus a penalizing weight ϵ:

$$C(\boldsymbol{x}, \widehat{d}(\boldsymbol{x})) > C(\boldsymbol{x}, d(\boldsymbol{x})) + \epsilon \tag{15.2}$$

Before the disparity pixel-wise assignment, the consistency constraint is applied at the finest level of the block-based representation. In practice, this constraint translates in a new bilateral filtering which combines both left and right disparity maps, as follows:

$$\widehat{d}_L(\boldsymbol{x}) = \frac{\sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}.d_L(\boldsymbol{y}) + \sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}.d_R(\boldsymbol{y} - d_L(\boldsymbol{x})}{\sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}} \tag{15.3}$$

where d_L and d_R correspond respectively the left and right disparity vectors.

Disparity pixel-wise assignment

Through this disparity pixel-wise assignment step, the idea is to obtain a dense disparity map from block matching results. For each pixel, the current disparity value plus the four values corresponding to the 4-connected neighboring blocks are candidates. The final disparity assigned to pixel $\boldsymbol{x}$ is the one among the five candidates that provides the minimal cost. For each disparity candidate, the color-weighted cost aggregation is performed as follows:

$$C(\boldsymbol{x}, d(\boldsymbol{x})) = \frac{\sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}.D(\boldsymbol{y}, d(\boldsymbol{x}))}{\sum_{y \in \mathcal{N}(\boldsymbol{x})} w_{\boldsymbol{xy}}} \tag{15.4}$$

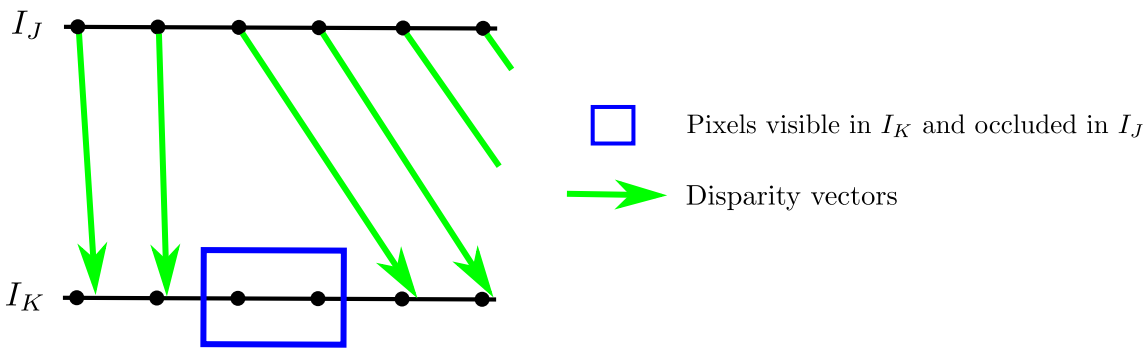


Figure 15.1: Occlusion detection following the *Occlusion Constraint* (*OCC*) described in [EW02] and used within the disparity estimator of [1]. I_K and I_J correspond to the left and right images.

where $\boldsymbol{x}$ is the current pixel, d corresponds to the disparity, $\boldsymbol{y}$ is a pixel of the $N \times N$ window centered on $\boldsymbol{x}$ with $N = 3$. The disparity-compensated difference $D(\boldsymbol{y}, d(\boldsymbol{x}))$ of pixel $\boldsymbol{y}$ with disparity value $d(\boldsymbol{x})$ is defined as follows:

$$D(\boldsymbol{y}, d(\boldsymbol{x})) = \sum_{c \in \{r,g,b\}} \left| I_c^K(\boldsymbol{y}) - I_c^J(\boldsymbol{y} - d(\boldsymbol{x})) \right| \tag{15.5}$$

where I_K and I_J are the left and right images and c corresponds to the 3 RGB color components. In addition, the weight $w_{\boldsymbol{xy}}$ in Eq. 15.4 is computed using the color difference between pixel $\boldsymbol{x}$ and its neighboring pixels $\boldsymbol{y}$.

Finally, this assignment stage is followed by both a 3×3 median filtering and a 21×21 joint trilateral filtering (involving color differences, *Euclidean* distance and disparity-compensated difference D) applied to the whole disparity map.

Dense disparity refinement

The dense disparity refinement first involves a pixel labeling stage which classifies pixels into four categories:

- pixels which are occluded in the other view,
- pixels whose disparity vector points at outside the frame in the other view,
- consistent pixels, i.e. pixels which are visible in the other view and which have a consistent disparity vector,
- inconsistent pixels, i.e. pixels which are visible in the other view and which have a inconsistent disparity vector.

The occlusion detection (first category) is performed following the *Occlusion Constraint* (*OCC*) described in [EW02][2] and illustrated in Fig. 15.1. According to this method, the pixels in view I_K which are occluded in view I_J are detected as follows: considering the disparity map of view I_J and starting from each pixel in I_J, its corresponding point in view I_K is identified via its assigned disparity vector. Then the closest pixel to this point in view I_K is marked as

[2] refers to the bibliography of Part I, page 95

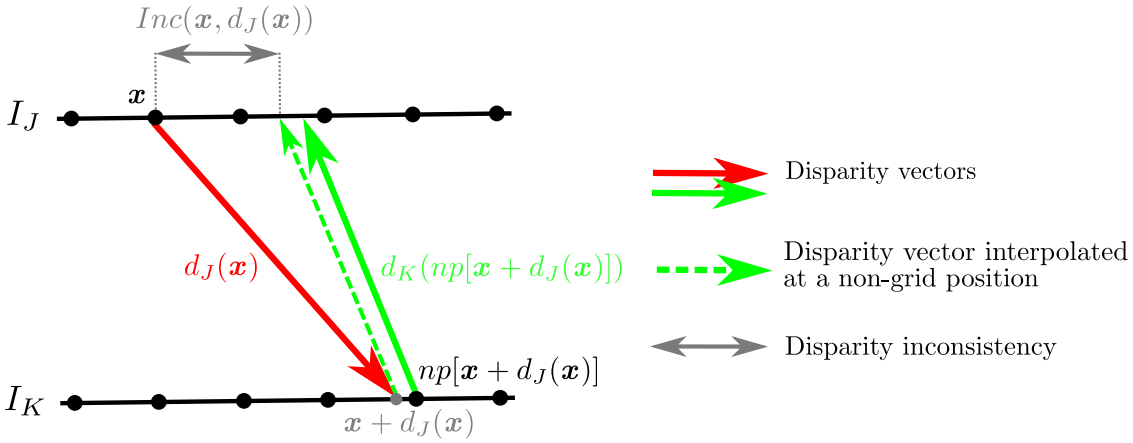


Figure 15.2: Disparity inconsistency computation following the *Left/Right Checking* (*LRC*) described in [EW02] and used within the disparity estimator of [1]. I_K and I_J correspond to the left and right images.

visible. At the end of this visibility detection, the pixels that are not marked are classified as occluded in the other view.

Pixels of I_K which belong to the second category are simply detected by identifying the arrival points of the disparity vectors starting from I_K which are outside I_J.

Disparity inconsistency (which allows to compute the third and four pixel categories) is measured via the comparison of the disparity vector in the current view and its corresponding disparity vector in the other view. This is similar to *Left/Right Checking* (*LRC*) [EW02] except that this is not used to detect occlusions in [1]. Practically, according to Fig. 15.2, for a given pixel $\boldsymbol{x}$ in view I_J, its inconsistency value $Inc(\boldsymbol{x}, d_J(\boldsymbol{x}))$ corresponds to the sum of the disparity vector $d_J(\boldsymbol{x})$ of $\boldsymbol{x}$ and of the disparity vector $d_K(\boldsymbol{u})$ of the pixel $\boldsymbol{u}$ in view I_K that is the closest pixel to the endpoint of $d_J(\boldsymbol{x})$ in I_K with abscissa $\boldsymbol{x} - d_J(\boldsymbol{x})$:

$$Inc(\boldsymbol{x}, d_J(\boldsymbol{x})) = d_J(\boldsymbol{x}) + d_K(np[\boldsymbol{x} - d_J(\boldsymbol{x})]) \tag{15.6}$$

where $\boldsymbol{u} = np[\boldsymbol{x} - d_J(\boldsymbol{x})]$. $np[a]$ is defined as the pixel closest to point a. The symmetrical process is applied to view I_K.

Inconsistency values are simply compared to a threshold ϵ_{Inc} to distinguish between consistent ($Inc(\boldsymbol{x}, d_J(\boldsymbol{x})) \leq \epsilon_{Inc}$) and inconsistent ($Inc(\boldsymbol{x}, d_J(\boldsymbol{x})) > \epsilon_{Inc}$) pixels. In practice, ϵ_{Inc} equals to 1.

The classification task is followed by a joint filtering (in the same spirit as in Eq. 15.3) which is applied only on inconsistent pixels in order to enforce consistency. Both stages are iteratively performed until the classification becomes stable. Finally, occluded regions are filled from neighboring disparity vectors visible in both views and filtered through bilateral filtering and median filtering.

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Analyse de vidéos assistée pour la description d'obejts vidéo et la synthèse de vues

Nom Prénom de l'auteur : CONZE PIERRE-HENRI

Membres du jury :
- Monsieur KERVRANN Charles
- Monsieur Cagnazzo Marco
- Monsieur Charvillat Vincent
- Monsieur Heitz Fabrice
- Monsieur Odobez Jean-Marc
- Monsieur Megret Rémi
- Monsieur Robert Philippe
- Madame MORIN Luce
- Monsieur Bouthemy Patrick

Président du jury : Monsieur Charles KERVRANN

Date de la soutenance : 16 Avril 2014

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 16 Avril 2014

Signature du président de jury

Le Directeur,

M'hamed DRISSI

C. KERVRANN

Résumé

Les nouvelles technologies de la vidéo numérique tendent vers la production, la transmission et la diffusion de contenus de très haute qualité, qu'ils soient monoscopiques ou stéréoscopiques. Ces technologies ont énormément évolué ces dernières années pour faire vivre à l'observateur l'expérience la plus réaliste possible. Pour des raisons artistiques ou techniques liées à l'acquisition et à la transmission du contenu, il est parfois nécessaire de combiner la vidéo acquise à des informations de synthèse tout en veillant à maintenir un rendu photo-réaliste accru. Pour faciliter la tâche des opérateurs de production et post-production, le traitement combiné de contenus capturés et de contenus de synthèse exige de disposer de fonctionnalités automatiques sophistiquées. Parmi celles-ci, nos travaux de recherche ont porté sur l'évaluation de qualité de la synthèse de vues et l'élaboration de stratégies d'estimation de mouvement dense et long-terme.

L'obtention d'images synthétisées de bonne qualité est essentielle pour les écrans 3D auto-stéréoscopiques. En raison d'une mauvaise estimation de disparité ou interpolation, les vues synthétisées générées par DIBR font cependant parfois l'objet d'artéfacts. C'est pourquoi nous avons proposé et validé une nouvelle métrique d'évaluation objective de la qualité visuelle des images obtenues par synthèse de vues.

Tout comme les techniques de segmentation ou d'analyse de scènes dynamiques, l'édition vidéo requiert une estimation dense et long-terme du mouvement pour propager des informations synthétiques à l'ensemble de la séquence. L'état de l'art dans le domaine se limitant quasi-exclusivement à des paires d'images consécutives, nous proposons plusieurs contributions visant à estimer le mouvement dense et long-terme. Ces contributions se fondent sur une manipulation robuste de vecteurs de flot optique de pas variables (multi-steps). Dans ce cadre, une méthode de fusion séquentielle ainsi qu'un filtrage multilatéral spatio-temporel basé trajectoires ont été proposés pour générer des champs de déplacement long-termes robustes aux occultations temporaires. Une méthode alternative basée intégration combinatoire et sélection statistique a également été mise en œuvre. Enfin, des stratégies à images de référence multiples ont été étudiées afin de combiner des trajectoires provenant d'images de référence sélectionnées selon des critères de qualité du mouvement.

Ces différentes contributions ouvrent de larges perspectives, notamment dans le contexte de la coopération stéréo-mouvement pour lequel nous avons abordé les aspects correction de disparité à l'aide de champs de déplacement denses long-termes.

Abstract

Film and consumer electronics industries have known in the last few years huge technological improvements to capture, transmit and display high-quality monoscopic and stereoscopic video content. These improvements aim at providing to the viewer the most realistic viewing experience. Due to artistic intentions or physical limitations to efficiently capture and transmit video contents, it is sometimes necessary to combine simultaneously captured and synthetic data while taking care to maintain a photo-realistic rendering. To efficiently process captured and synthetic content simultaneously, production and post-production operators need to be assisted by sophisticated automatic tools. Among these tools, we thoroughly investigated both view synthesis quality assessment and long-term dense motion estimation issues.

3D autostereoscopic displays rely on the generation of realistic-looking virtual viewpoints through disparity estimation and view interpolation involved together within Depth-Image-Based Rendering (DIBR) algorithms. Despite recent advances, DIBR algorithms do not always provide artifact-free synthesized views and induce new types of artifacts whose impact can be harmful for the observer. Our contribution in this context has been to develop and evaluate a new full-reference objective image quality assessment metric dedicated to view synthesis quality assessment.

Also required by recent applications such as scene segmentation or dynamic scene analysis techniques, long-term dense displacement fields allow to propagate synthetic data to the whole sequence in the context of high quality video editing. However, state-of-the-art optical flow estimators show strong limitations toward long-term requirements since classical optical flow assumptions are not valid for non-consecutive frames. Therefore, we proposed several contributions to long-term dense motion estimation based on multi-step optical flow vectors. First, a sequential fusion approach including a spatio-temporal multilateral filtering has been investigated toward long-term dense correspondences robust to temporary occlusions. Then, an alternative method has been studied based on combinatorial integration and statistical selection. Finally, we proposed multi-reference frames strategies to correlate trajectories estimated with respect to multiple reference frames selected according to motion quality criteria.

Our contributions in both contexts offers new perspectives, especially for joint stereo and motion processing. In this direction, an automatic disparity correction framework using long-term dense displacement fields has been addressed.

N° d'ordre : **14ISAR07 / D14-07**

Institut National des Sciences Appliquées de Rennes
20, Avenue des Buttes de Coëmes • CS 70839 • F-35708 Rennes Cedex
Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96