Schémas de suivi d'objets vidéo dans une séquence animée : application à l'interpolation d'images intermédiaires.

Laurent Bonnaud

N° d'ordre: 2055

THÈSE

Présentée devant

l'Université de Rennes 1
pour obtenir

le grade de : Docteur de l'Université de Rennes 1
Mention Informatique
par

Laurent Bonnaud

Équipe d'accueil : Temics (Irisa, Rennes)
École Doctorale : Informatique, Traitement du Signal et Télécommunications
Composante universitaire : Institut de Formation Supérieure en Informatique et Communication (Ifsic)

Titre de la thèse :

Schémas de suivi d'objets vidéo dans une séquence animée : application à l'interpolation d'images intermédiaires.

Soutenue le 20 Octobre 1998 devant la commission d'examen composée de :

M. :	Jean-Pierre	Banâtre	
MM. :	Fabrice	Heitz	Rapporteurs
	Rachid	Deriche	
MM. :	Dominique	Barba	Examinateurs
	Janusz	Konrad	
	Henri	Sanson	
	Claude	Labit	

Ceci n'est pas une citation.

Remerciements

Je remercie Jean-Pierre Banâtre, Professeur, IRISA, qui me fait l'honneur de présider ce jury.

Je remercie Fabrice Heitz, Professeur, ENSPS, et Rachid Deriche, Directeur de recherches, INRIA, d'avoir bien voulu accepter la charge de rapporteur.

Je remercie Dominique Barba, Professeur, IRESTE, Janusz Konrad, Associate Professor, INRS-Telecom, Montréal, et Henri Sanson, Ingénieur CNET/CCETT, d'avoir bien voulu juger ce travail. Je remercie aussi Janusz Konrad de m'avoir accueilli dans son laboratoire à l'INRS, pour un séjour à Montréal très constructif.

Je remercie enfin Claude Labit, pour m'avoir proposé un sujet passionnant et accueilli dans son équipe, pour m'avoir encadré avec un intérêt constant et une grande compétence et pour m'avoir encouragé et fait confiance pendant toutes ces années.

Je remercie aussi mes parents pour leur soutien apporté pendant toute cette thèse.

Merci enfin à tous les gars et les filles de l'Irisa pour leur bonne humeur et la chaleureuse ambiance qu'ils contribuent à créer.

Table des matières

Liste des figures

Liste des tableaux

Glossaire, définitions et notations

AAF	Ajustement affine des frontières.
ALS	Ajustement local des sommets.
CELP	*Code Excited Linear Prediction.*
DPCM	*Delta Pulse Code Modulation.*
DFD	*Displaced Frame Difference.*
ECMA	Équation de Contrainte du Mouvement Apparent.
ECM	Éléments de Contours en Mouvement.
EQM	Erreur Quadratique Moyenne.
GOP	*Group Of Frames.*
HCF	*Highest Confidence First.* Algorithme de relaxation déterministe.
ICM	*Iterated Conditional Modes.* Algorithme de relaxation déterministe.
ICBM	Interpolation par Compensation Bidirectionnelle de Mouvement.
IPBS	Interpolation par Prédiction Bidirectionnelle de Segmentation.
IPM	Interpolation par Prédiction du Mouvement (interpolation pure).
LPE	Ligne de Partage de Eaux.
MDL	*Minimum Description Length.*
MPEG	Motion Picture Expert Group. MPEG1, MPEG2 et MPEG4 sont des normes de compression pour la vidéo numérique.
PSNR	*Peak Signal to Noise Ratio.* Défini à partir de l'EQM par la formule : $$\text{PSNR} = -10 \log_{10} \frac{\text{EQM}}{255^2}$$
RCE	Représentation par Carte d'Étiquettes (voir la section 1.3.2). Il s'agit d'une représentation possible pour une partition d'une image, utilisée dans le cadre d'une modélisation du mouvement par régions.
RCF	Représentation par Contours Fermés (voir la section 1.3.4).
RFO	Représentation par Frontières Ouvertes (voir le chapitre 2).
VOP	*Video Object Plane.* Objet vidéo tel qu'il est défini dans la norme MPEG4.

recouvrement	Le fait qu'une partie de la texture d'un objet disparaît derrière un autre objet à cause de leur profondeur et de leur mouvement relatifs
découvrement	Idem sauf qu'une partie cachée de la texture de l'objet apparaît
occultation	Terme générique qui désigne soit un recouvrement, soit un découvrement entre objets vidéo
interpolation pure	Création d'images non observées au sein d'une séquence par interpolation temporelle
codage interpolatif	Schéma de compression utilisant l'interpolation temporelle d'images dans une séquence.
I-*frame*	(ou image de type I)
P-*frame*	(ou image de type I)
B-*frame*	(ou image de type I)

Θ	Descripteur de mouvement. Désigne à la fois la transformation plane permettant d'effectuer la compensation de mouvement et le vecteur des paramètres de cette transformation.
$\Theta^{\pm R}_{t_1 \to t_2}$	Descripteur de mouvement de la région R, de l'image I_{t_1} vers l'image I_{t_2}, dans le sens des t croissants ($^+$) ou décroissants ($^-$) selon si $t_1 < t_2$ ou si $t_2 < t_1$.
$\widehat{\Theta}^{+R,t\|t-1}_{t-1 \to t}$	Prédiction du descripteur de mouvement de la région R (estimation a priori donnée par un filtre de Kalman pour l'instant t, connaissant les observations à l'instant $t-1$).
$\widehat{\Theta}^{+R,t\|t}_{t-1 \to t}$	Estimation du descripteur de mouvement de la région R (résultat de l'estimateur de mouvement ou estimation a posteriori donnée par un filtre de Kalman pour l'instant t, connaissant les observations à l'instant t).
$R_1 \prec R_2$	La région R_1 «est en dessous de» la région R_2.
n_a	Nombre d'arcs dans un graphe ou de frontières ouvertes dans notre représentation d'une segmentation
n_s	Nombre de sommets dans un graphe
n_{ptm}	Nombre de points multiples dans notre représentation
n_{fr}	Nombre de frontières dans notre représentation
n_{somm}	Nombre de sommets dans l'approximation polygonale des frontières
n_{segm}	Nombre de segments dans l'approximation polygonale des frontières

Introduction générale

Contexte de l'étude

Le domaine d'étude de cette thèse est le traitement de séquences d'images numériques. Nous nous intéressons en particulier au problème de la segmentation automatique des images en objets vidéo. Un aspect important de notre travail concernera par conséquent l'étude du mouvement dans les images. De plus, nous cherchons à relier ce problème à celui de l'interpolation temporelle, visant à générer des images intermédiaires au sein d'une séquence, en tenant compte du mouvement des objets.

Notre thèse a pour contexte applicatif les domaines du multimédia et des télécommunications dans leurs aspects liés à la vidéo numérique. Les applications visées concernent le stockage, la transmission et la restitution de séquences d'images sur des réseaux numériques par paquets, comme Internet et les réseaux ATM. En effet, l'interpolation est un outil puissant qui peut permettre d'améliorer les techniques existantes. Notre étude se positionne aussi en complément des normes multimédia MPEG4 et MPEG7, qui permettent respectivement l'édition de séquences au niveau des objets vidéo et leur indexation. Dans ce cadre, il est souhaitable de disposer d'algorithmes de segmentation automatique d'images, sur des critères d'homogénéité du mouvement des objets.

Applications principales, motivations

Plus précisément, les applications visées pour l'interpolation sont :

- une meilleure compression, visant à réduire le débit d'information nécessaire à la transmission ou au stockage de vidéo,
- la conversion entre standards vidéo de fréquences d'affichage différentes,
- la transmission multipoints vers des terminaux ayant des fréquences de rafraîchissement différentes,
- l'interpolation d'images manquantes si des informations ont été perdues à la suite d'une perte de paquets lors de la transmission.

Quant à la segmentation, les applications potentielles sont très nombreuses, mais on peut mentionner la manipulation d'objets dans une séquence et l'indexation vidéo, déjà citées.

Objectifs

Nos objectifs se divisent en deux ensembles, concernant la segmentation et l'interpolation, mais qui sont fortement reliés par l'imbrication des deux problèmes.

D'une part il s'agira de réfléchir aux modélisations du mouvement les plus adaptées aux applications visées, de définir une représentation adaptée de la segmentation en objets vidéo et de concevoir un algorithme de suivi temporel de segmentation, opérant sur cette représentation.

D'autre part, il nous faudra développer une technique d'interpolation basée sur les objets vidéo segmentés précédemment, pouvant tirer parti de la représentation particulière adoptée et démontrer ses avantages par rapport aux techniques de compression sans interpolation.

Difficultés

Les difficultés de ce travail sont grandes puisque le problème de la segmentation automatique, si l'on veut des objets vidéo ayant une signification sémantique, est un authentique problème d'intelligence artificielle, pouvant même dépendre de la subjectivité de l'observateur. Plus modestement, nous nous contenterons d'objets définis par des critères objectifs (par exemple, luminance constante ou mouvement homogène) suffisants pour nos applications, en espérant approcher, dans les cas favorables, le niveau sémantique. Ce problème est extrêmement difficile puisque de nombreuses études y ont déjà été consacrées et que d'autres travaux semblent encore nécessaires pendant des années. De plus, vu les applications visées, nous ne pouvons pas faire d'hypothèses classiques trop limitatives, comme la fixité de la caméra, la connaissance préalable du fond, ou encore des hypothèses sur la nature des objets observés.

Une autre difficulté importante est que nous ne pouvons pas nous contenter de segmenter et de suivre un seul objet, comme cela est parfois fait pour certaines applications. Nous avons besoin d'une représentation exhaustive, prenant en compte tous les objets présents dans la séquence étudiée. Il faut donc traiter le problème des occultations entre objets. De plus, l'utilisation de la segmentation à des fins d'interpolation nécessite une grande stabilité temporelle de celle-ci, et une bonne précision de la localisation des contours des objets.

Structuration de l'étude et plan du document

Notre travail comporte deux grands axes : la segmentation et l'interpolation. Un chapitre propre est consacré à chacun d'entre eux, et d'autres chapitres préliminaires tentent de faire le lien entre ces deux problèmes.

Un premier chapitre consiste en une étude bibliographique qui va nous permettre d'examiner les différentes modélisations possibles du mouvement dans les images. En nous fondant sur les contraintes imposées par notre application, nous choisirons la modélisation qui nous paraît la plus adaptée, à savoir une partition de l'image en régions, le mouvement de

chaque région étant décrit par un modèle paramétrique de mouvement. Nous examinerons aussi les représentations de segmentation qui ont été utilisées, et les algorithmes de suivi temporel associés, et nous soulignerons les inconvénients qu'elles présentent à la fois pour le suivi et pour notre application, ce qui nous amènera à définir notre propre représentation dans le chapitre suivant.

Dans le deuxième chapitre, nous définirons la représentation qui nous paraît la plus adaptée à un suivi temporel de segmentation et au problème de l'interpolation. Elle sera fondée principalement sur les frontières ouvertes séparant les objets de la scène étudiée. Nous nous intéresserons aussi au problème de l'initialisation de cette représentation dans la première image de la séquence traitée, puis nous montrerons comment effectuer quelques manipulations de base sur cette représentation.

Le troisième chapitre est celui qui est principalement consacré au problème du suivi temporel. Nous commençons par y exposer les hypothèses que nous faisons sur la séquence pour un fonctionnement correct. Ensuite, nous détaillons l'algorithme de suivi dont le mécanisme de fonctionnement est du type prédiction/ajustement. Il opère principalement sur les frontières de la représentation, mais utilise aussi le mouvement de l'intérieur des objets.

Le problème de l'interpolation est traité dans le quatrième et dernier chapitre. Nous y présentons la technique de base qui est la compensation de mouvement bidirectionnelle. Nous montrons comment l'améliorer si l'on connaît une segmentation des images pour traiter le cas des zones découvertes et recouvertes par les occultations entre objets. Nous examinons aussi l'intégration de notre technique d'interpolation dans un schéma de codage basé régions en montrant les coûts de codage nécessaires aux différents éléments de notre représentation. Enfin, nous présentons plusieurs utilisations possibles de l'interpolation par le moyen de modes interpolatifs rentrant dans le cadre d'un codage par niveaux hiérarchiques.

Contributions

Dans l'ensemble de ces chapitres, des problèmes de circularité se présentent :

- la représentation est définie en fonction des problèmes du suivi temporel et de l'interpolation
- l'algorithme de suivi possède lui aussi certaines caractéristiques dictées par son utilisation pour l'interpolation
- l'interpolation utilise le résultat du suivi et certaines propriétés de la représentation

Ces dépendances comportant un cycle, il est impossible d'ordonner les chapitres de sorte que toutes les explications nécessaires à un chapitre sont données avant. Nous avons donc été obligés de faire quelques anticipations dans les trois premiers chapitres, mais nous pensons avoir choisi l'ordre minimisant ces explications anticipées.

Séquences expérimentales

Nous présentons ici les séquences qui nous ont servi pour tester nos différents algorithmes. Elles sont au nombre de quatre :

- séquence *«Miss America»* : 360×288 (format CIF), en niveaux de gris
- séquence *«Interview»* : 674×536 (format TV), en niveaux de gris
- séquence *«Flower Garden»* : 352×240, en niveaux de gris
- séquence *«Tennis»* : 360×240, en niveaux de gris

Pour avoir une idée du contenu de ces séquences et des mouvements qu'elles contiennent, on pourra se reporter aux figures 3.29 à 3.33, qui en montrent un échantillonnage régulier.

Chapitre 1

Modélisations et représentations du mouvement : un état de l'art

Introduction

Dans le domaine du codage de séquences vidéo [Tziritas et Labit 94a], de très nombreux algorithmes de compression ont été proposés. Mais la plupart des algorithmes, dès les origines [Jain et Jain 81] et jusqu'aux standards MPEG2 [Le Gall 91, Le Gall 92] et MPEG4 [Pereira 96] prennent tous en compte le mouvement observé dans les images. En effet, dans la plupart des séquences, il existe une forte corrélation temporelle du signal le long des trajectoires des pixels. Il est possible de l'exploiter à des fins de réduction de débit, en plus de la corrélation spatiale.

Mis à part la compression, la plupart des traitements sur les séquences d'images nécessitent une compensation de mouvement. Citons notamment le filtrage [Dubois 92], la restauration [Buisson et al. 97], le désentrelacement [Depommier et Dubois 92] [De Haan et Bellers 98]. En ce qui concerne l'application qui nous intéresse le plus, à savoir l'interpolation temporelle, la prise en compte du mouvement est une constante [Konrad 88] [Cafforio et al. 90] [Bergeron et Dubois 90] [Tziritas et Labit 94b].

Le mouvement observé peut être causé par le mouvement relatif des objets dans la scène observée ou encore par le mouvement de la caméra. Pour une application de codage on peut se contenter d'une description bidimensionnelle du mouvement, sans avoir à considérer le vrai mouvement tridimensionnel. On parle alors du «mouvement apparent» ou «mouvement 2D». C'est le cas dans un compromis débit/distorsion, lorsque le coût de codage d'un mouvement tridimensionnel est trop important (sans parler de la difficulté de son estimation).

On a alors besoin de définir un modèle de mouvement. Un exemple de modèle est le mouvement translationnel par bloc. C'est celui qui est le plus utilisé, notamment dans les normes MPEG. Il faut ensuite identifier ce modèle pour la séquence traitée, ce qui est fait par un algorithme d'estimation de mouvement adapté à la modélisation choisie. Lorsqu'on utilise le mouvement pour faire de la compression, il faut aussi le transmettre du codeur au décodeur, ce qui pose le problème de son coût de codage.

Le choix d'une modélisation n'est pas neutre car les différentes modélisations n'ont pas les mêmes caractéristiques vis à vis de la représentativité, de l'identification, et de la transmission [Nicolas 92]. Lorsque l'on va de la plus simple à la plus complexe [Nicolas et Labit 93], on peut globalement dire que ces 3 paramètres évoluent de la façon suivante :

- Les différentes modélisations ne permettent pas de représenter les mêmes classes de mouvement. Certaines sont plus riches que d'autres et incluent même d'autres modélisations en tant que sous-classe de mouvements.
- Une modélisation plus simple a moins de paramètres qu'une plus complexe. Elle donne plus d'information *a priori* sur le mouvement, ce qui aide à son estimation.
- À cause du nombre plus élevé de paramètres, une modélisation complexe aura un coût de codage plus important.

Il y a donc un compromis à trouver entre une grande représentativité d'une part ou une estimation facilitée et une transmission peu coûteuse d'autre part. En fonction de l'application visée, ces 3 paramètres auront une importance relative différente. Il importe de bien choisir la modélisation en vue de notre application au codage interpolatif. C'est donc à cela nous allons nous attacher dans ce chapitre.

Plan du chapitre

Dans une première section (1.1), nous commençons par un rapide exposé des modélisations possibles. Pour chacune d'entre elles, nous présentons quelques algorithmes d'estimation et les applications auxquelles elle est adaptée.

Ensuite nous regardons ces modélisations à la lumière de notre application au codage interpolatif (section 1.2). Nous détaillons en particulier quatre points importants pour cette application.

Enfin, dans la section 1.3, pour la modélisation choisie, nous nous intéressons aux représentations déjà utilisées et nous soulignons leurs inconvénients, ce qui nous amènera à définir notre propre représentation dans le chapitre suivant (chapitre 2).

1.1 Modélisations et algorithmes associés

Dans cette section, nous nous intéressons à la modélisation du mouvement apparent entre deux images I_{t_1} et I_{t_2} observées respectivement aux instants t_1 et t_2, avec $t_1 < t_2$. Il s'agit en général d'images consécutives dans la séquence, mais ces modèles peuvent en fait décrire le mouvement entre tout couple d'images. Cette remarque aura son importance dans le chapitre 4 sur l'interpolation temporelle.

Comme nous travaillons en temps discret, nous nous intéressons aux modèles de **déplacement**, par opposition aux modèles de vitesse. D'un point de vue mathématique, un modèle de déplacement est une transformation plane appliquée au plan image. Un modèle de vitesse est un champ de vecteurs vitesse défini sur le plan image. Il y a donc identité des objets mathématiques. Les deux sont souvent amalgamés car, dans le cas des modèles

linéaires, il y a de plus égalité entre les deux. Par contre, pour des modèles non linéaires, des différences apparaissent (voir par exemple le modèle homographique, annexe B). Nous noterons p un point représentant un pixel dans le plan image et $\vec{d}(p)$ son déplacement.

Notons que le rôle des deux images est dissymétrique. On considère que ce sont les pixels de I_{t_2} dont on cherche les correspondants dans I_{t_1}. C'est le sens du mouvement qu'il faut utiliser dans le contexte du codage par compensation de mouvement, car le mouvement permet alors de prédire I_{t_2} à partir de I_{t_1} (schéma de codage du type hybride I [Woods 91]). Le déplacement correspondant est alors noté $\vec{d}_{t_2 \to t_1}(p)$ et la formule de reconstruction est $\widehat{I_{t_2}}(p) = I_{t_1}(p + \vec{d}_{t_2 \to t_1}(p))$. Comme le mouvement d'un pixel n'est en général pas un nombre entier de pixels, il est nécessaire de réaliser une interpolation spatiale dans I_{t_1} (interpolation bilinéaire ou bicubique [Keys 81] par exemple).

Dans la typologie que nous avons adoptée, un modèle est défini par la donnée conjointe de deux éléments :

- une entité géométrique qui définit la forme de supports pour le mouvement,
- une entité cinématique qui définit le mouvement sur le support précédent.

Quant aux algorithmes d'estimation du mouvement ils font pour la plupart l'hypothèse que le niveau de gris d'un pixel est constant au cours du temps, le long de sa trajectoire [Horn et Schunck 81]. Des modèles plus complets ont été proposés pour s'affranchir de cette hypothèse forte en introduisant en plus un modèle de variation de l'illumination [Nicolas et al. 93]. Ils sont principalement basés sur deux principes :

Équation de contrainte du mouvement apparent (ECMA) Elle découle de l'hypothèse de conservation du niveau de gris d'un pixel au cours de son mouvement [Horn et Schunck 81] et s'écrit :

$$\vec{\nabla} I(p).\vec{d}(p) + \frac{\partial I}{\partial t}(p) = 0$$

En pratique on l'utilise entre les instants t_1 et t_2 en approchant la dérivée partielle par une différence finie :

$$\vec{\nabla} I(p).\vec{d}_{t_2 \to t_1}(p) + (I_{t_1} - I_{t_2})(p) = 0$$

Minimisation de l'erreur de prédiction Le principe de l'algorithme est de minimiser une erreur définie sur le support du mouvement :

$$Err(\vec{d}) = \sum_{p \in support} \rho[I_{t_2}(p) - I_{t_1}(p + \vec{d}_{t_2 \to t_1}(p))]$$

avec ρ une fonction positive, symétrique et en général croissante. La fonction ρ est souvent choisie comme étant la norme l_2. Dans ce cas particulier, on parle d'«Erreur Quadratique Moyenne» (EQM) et l'erreur peut être considérée comme une énergie. D'autres choix possibles sont la norme l_1, la norme l_∞ ou une fonction robuste [Odobez 94]. La minimisation est effectuée sur le champ de déplacements pour obtenir le champ estimé $\widehat{\vec{d}}$:

$$\widehat{\vec{d}} = \arg\min_{\vec{d}} Err(\vec{d})$$

Selon les modèles, ces équations sont insuffisantes et il faut introduire des termes ou des contraintes supplémentaires.

1.1.1 Champ dense

Dans ce modèle, le mouvement est modélisé par un champ de vecteurs quelconque. L'entité géométrique est donc le pixel p et l'entité cinématique est le vecteur $\vec{d}_{t_2 \to t_1}(p)$. Ce vecteur peut avoir des composantes réelles ou discrètes (entières ou quantifiées), selon l'algorithme d'estimation.

En pratique on n'obtient pas un champ satisfaisant avec ce modèle tel quel. En effet, si l'on utilise l'EQM, cela revient à rechercher le meilleur vecteur pixel par pixel, et on obtient en général un champ très bruité, voire complètement aléatoire. Si l'on utilise l'ECMA, elle ne donne que la projection de chaque vecteur sur la direction du gradient spatial, indétermination qui est appelée «problème de l'ouverture» [Horn et Schunck 81].

Dans les deux cas, il est donc nécessaire de régulariser le champ [Horn et Schunck 81] [Nagel et Enkelmann 81]. Les premiers auteurs proposent par exemple de rajouter un terme à la fonction d'énergie à minimiser, qui pénalise des différences importantes entre les vecteurs de déplacement voisins.

Une autre technique de régularisation spatiale est basée sur les champs de Markov [Besag 74, Besag 86, Geman et Geman 84, Azencott 87, Derin et Elliot 87]. Dans ce cadre, le vecteur de mouvement de chaque pixel est considéré comme une variable aléatoire cachée qu'il faut estimer grâce à des observations indirectes. Cette technique a beaucoup été appliquée au problème de l'estimation d'un champ dense de mouvement [Konrad et Dubois 88a] [Konrad 89] [Konrad et Dubois 92]. Chaque pixel est un site relié aux pixels proches appartenant à un certain voisinage (en général un 4- ou 8-voisinage). Le champ de Markov utilise la probabilité conjointe *a priori* de l'occurence de deux vecteurs voisins. Il sert à la régularisation statistique de la solution. Par ailleurs, pour chaque pixel, il faut pouvoir calculer la probabilité que son vecteur déplacement soit correct, grâce à une observation locale, en général l'ECMA. Il s'agit d'une probabilité *a posteriori*, connaissant ce vecteur.

On se ramène alors à la minimisation d'une fonction d'énergie globale définie sur autant de variables que de pixels. Elle se compose de deux termes :

- une énergie d'attache aux données qui exprime l'adéquation de chaque vecteur à l'observation locale (ECMA)
- une énergie de régularisation qui favorise les situations où deux pixels voisins ont des vecteurs similaires

Ce type de régularisation est assez général et la régularisation proposée dans [Horn et Schunck 81] en est un cas particulier. D'autres travaux vont plus loin et prennent en compte les discontinuités qui peuvent intervenir dans un champ de déplacement (contours en mouvement). Pour cela ils utilisent des «étiquettes de discontinuité» qui annulent le terme de régularisation pour les vecteurs de part et d'autre du contour en mouvement [Heitz et Bouthemy 90a] [Heitz et Bouthemy 90b] [Heitz et Bouthemy 93].

Un inconvénient de ces techniques est leur coût de calcul très important lié au nombre très élevé de variables à traiter et aux algorithmes de minimisation itératifs à utiliser. Un

palliatif à cette difficulté sont les modèles de Markov multi-résolutions ou multi-échelles [Konrad et Dubois 88b] [Perez 93].

Des algorithmes d'estimation plus simples sont les algorithmes récursifs (par opposition aux algorithmes itératifs). Dans ces techniques, l'estimation est réalisée en effectuant un certain parcours de l'image. Le mouvement de chaque pixel est alors estimé à partir des mouvements des pixels déjà parcourus. Le parcours le plus usuel est un simple balayage de l'image dans un sens puis un deuxième balayage dans l'autre sens. Les premiers représentants de ces techniques dites «*pel-recursives*» sont [Netravali et Robbins 79] et [Walker et Rao 84]. D'autres travaux utilisent le filtre de Kalman [Rougée et al. 88] [Tziritas 90] [Tziritas et Labit 94c].

1.1.2 Partition fixe régulière

Dans ce modèle, l'image est découpée en blocs carrés (ou rectangulaires) ou plus généralement en une partition régulière et fixe. Les pixels d'un bloc ont alors tous le même vecteur de mouvement (certains travaux sortent de cette limitation en utilisant des modèles de mouvement plus complexes, comme ceux de la section 1.1.5). Par rapport au modèle précédent, on peut dire qu'il s'agit d'un champ de vecteurs constant par blocs. C'est le modèle choisi dans les normes MPEG1 et MPEG2 [Le Gall 91] [Le Gall 92], avec des blocs carrés de taille 16×16 pixels, et des vecteurs de translation ayant des coordonnées entières ou demi-entières et une amplitude maximale de 32 pixels. La norme MPEG4 autorise en plus des blocs de taille 8×8 pixels [Ebrahimi 97].

Les algorithmes d'estimation associés sont les algorithmes de *block-matching*. Ils sont basés sur la minimisation de l'erreur de prédiction de I_{t_2} par I_{t_1} (en général l'EQM). Dans le cas de vecteurs de mouvement discrétisés, l'algorithme le plus simple est celui de recherche exhaustive : on calcule toutes les erreurs correspondant à tous les vecteurs et on retient celui qui produit l'erreur la plus faible. Cette recherche est assez coûteuse si bien que des accélérations ont été envisagées : recherche de la composante horizontale du vecteur de mouvement puis de la composante verticale, recherche multi-échelle, ... Ces méthodes sont plus rapides puisqu'elles ne considèrent qu'un sous-ensemble des vecteurs possibles, mais donnent des vecteurs sous-optimaux.

Le choix de cette modélisation a été fait dans les normes MPEG, car l'accent a été mis sur la relative simplicité de l'estimation du mouvement et son implantation sur des processeurs spécialisés.

1.1.3 Maillages actifs

Cette modélisation s'inspire de la méthode très générale des éléments finis (MEF) qui sert à résoudre des équations aux dérivées partielles définies sur un domaine continu. La MEF repose sur la discrétisation de ce domaine. Dans notre cas, le champ inconnu est le champ des vecteurs déplacement défini sur un rectangle du plan image. Pour cela, on découpe le domaine de définition du champ en mailles de formes quelconques, comme des triangles ou des tétraèdres, des quadrilatères ou des pavés, ... Les côtés ou les surfaces des mailles peuvent même être de degré supérieur à 2. Ensuite, sur chaque maille, on approche

le champ par un modèle polynomial : modèle constant, linéaire ou de degré supérieur. L'ensemble des mailles et des modèles forment une base d'un espace vectoriel de dimension finie. Cet espace vectoriel approche de plus en plus finement l'espace original lorsque l'on prend des mailles de plus en plus petites. Enfin, les équations aux dérivées partielles sont projetées sur cette base, ce qui donne un système d'équations algébriques, le plus souvent linéaire, qu'il faut ensuite résoudre.

Un des problème qui se pose dans ce modèle est celui de la régularité spatiale du champ discrétisé aux limites des mailles. On peut simplement ignorer le problème comme dans MPEG, où le champ est discontinu aux limites des blocs. Ou alors on peut choisir la forme des mailles et le modèle d'approximation de sorte à assurer une certaine régularité du champ. Dans le cas d'un champ de mouvement, il semble que la simple continuité soit suffisante. Satisfaisant cette contrainte, deux modèles sont les plus utilisés :

- le modèle à mailles triangulaires, avec un modèle de mouvement linéaire (on dit aussi «affine») dans chaque maille [Wang et Lee 93] [Altunbasak et al. 95] [Dudon 96] [Toklu et al. 97b]
- le modèle à mailles quadrilatères, avec un modèle de mouvement bilinéaire dans chaque maille. Ce modèle est aussi appelé *Control Grid Interpolation* [Sullivan et Baker 91].

Ces deux modèles sont en fait assez contraints : si l'on donne les vecteurs de mouvement aux sommets du maillage, il n'y a plus de degré de liberté pour faire varier le champ à l'intérieur de la maille. En effet, une transformation affine (6 paramètres indépendants), est complètement déterminée par la donnée de 3 points et de leurs transformés (6 paramètres aussi).

Les méthodes d'estimation étant trop nombreuses pour être détaillées ici, nous n'en mentionnerons donc que deux. Dans [Dudon 96], cette modélisation est appliquée sur toute l'image. La grille et les mouvements sont estimés de façon successive. Les sommets des triangles sont déterminés selon des critères de gradient spatial ou spatio-temporel connus à la fois du codeur et du décodeur. Pour l'estimation du mouvement, trois approches ont été testées : la mise en correspondance, un algorithme génétique et une méthode différentielle, et c'est la première qui a été retenue. Dans [Altunbasak et Tekalp 96] et [Toklu et al. 96] le maillage est appliqué sur un objet unique, qu'il faut déjà avoir segmenté par une autre méthode.

1.1.4 Points singuliers

Ce modèle consiste en pixels épars auxquels sont associés des vecteurs de mouvement. Le champ dense de mouvement s'obtient par interpolation des vecteurs. Ce modèle dépend donc en grande partie de la méthode d'interpolation choisie.

Si l'on utilise une triangulation et que l'on réalise une interpolation linéaire dans chaque triangle avec une pondération barycentrique, on retombe sur le modèle précédent des mailles triangulaires avec mouvement affine. Dans une application au codage, il importe que la triangulation soit connue de façon implicite du codeur et du décodeur, pour éviter sa transmission. Un choix fréquent est celui de la triangulation de Delaunay [Davoine 95].

Par contre, [Le Floch 97] a choisi un autre mode d'interpolation où les points les plus proches sont utilisés, sans être obligatoirement au nombre de 3. La pondération se fait alors avec des coefficients dépendant, de façon non linéaire, de la distance entre le pixel interpolé et les points de référence.

Dans [Maurizot 97], le sujet d'étude est les écoulements fluides. Dans ce domaine particulier, les points singuliers sont ceux où le champ de mouvement présente une singularité (comme par exemple le centre d'un tourbillon). Le mouvement de ces points et alors estimé sur une fenêtre dont la taille est choisie itérativement de façon optimale. La spécificité du domaine fait que l'on peut non seulement estimer la translation du point, mais aussi les composantes locales du champ (divergence, rotationel). Il est alors possible d'interpoler un champ dense à partir de ces informations plus riches.

Dans [Le Floch 97] l'estimation se fait sur des vecteurs quantifiés. On part d'un champ nul pour lequel on calcule l'erreur de reconstruction. Ensuite on calcule, pour chaque pixel et pour chaque vecteur possible, la diminution d'erreur qu'apporterait l'ajout de ce vecteur en ce pixel. On retient alors le couple (pixel, vecteur) qui apporte la plus grande diminution d'erreur. Ce processus est itéré jusqu'à ce que l'on atteigne le nombre de points que l'on s'est donné (c'est-à-dire un certain taux de compression) ou une qualité de reconstruction donnée.

1.1.5 Régions et mouvements paramétriques

Dans ce modèle, le plan image est partitionné en régions connexes de forme quelconque qui constituent les entités géométriques du modèle. La partie cinématique consiste en un modèle de mouvement permettant de calculer le vecteur mouvement de chaque pixel contenu dans la région. Ce modèle peut être vu comme un cas plus général des modèles par blocs et par éléments finis. En théorie il généralise aussi le modèle de champ dense, mais en pratique des régions de 1 pixel n'ont aucun intérêt.

Les modèles de mouvement 2D utilisés sont assez nombreux et sont détaillés dans l'annexe B. Citons pour mémoire les modèles paramétriques [Adiv 85] translationnel, linéaire simplifié, affine, homographique et quadratique [Nicolas 92]. Le fait de définir le champ de vecteurs par une approximation par morceaux introduit des discontinuités du mouvement aux frontières des régions, qui peuvent être artificielles. Il faut cependant nuancer cette observation. Dans le cas de frontières entre objets séparés par un contour en mouvement, la discontinuité est même souhaitable, si la partition est correctement ajustée sur ces contours. Le cas de frontières intérieures à un objet se rencontre lorsque le mouvement d'un objet est trop complexe pour être décrit par le modèle choisi. Dans ce cas, si les mouvements de part et d'autre sont bien estimés, la discontinuité a une forte probabilité d'être faible. De plus les éventuels artefacts ont moins de chances de se voir qu'avec des blocs, car ils ne sont pas formés de lignes horizontales ou verticales.

L'estimation d'un tel modèle est assez difficile puisqu'elle pose un problème «de la poule et de l'œuf». Pour estimer le mouvement, il est nécessaire de connaître les supports d'estimation, c'est-à-dire les régions. Et pour pouvoir ajuster précisément les contours des régions, il est nécessaire de connaître leur mouvement. Pour sortir de ce problème circulaire,

3 approches ont été employées :

- On peut partir d'une segmentation spatiale, par exemple une segmentation hiérarchique [Healey 93], une segmentation morphologique [Salembier et Pardas 94], ou une segmentation selon le critère du MDL [Zheng et Blostein 95]. Puis on estime le mouvement de ces régions spatiales pour les fusionner en régions spatio-temporelles homogènes au sens du mouvement. C'est aussi l'approche des méthodes de suivi par contours actifs, où le contour actif est initialisé dans la première image sur des informations spatiales (gradients), puis suivi dans le temps grâce au mouvement estimé [Ueda et Mase 92] [Terzopoulos et Szeliski 92] [Bascle et al. 94].

- On peut partir d'un champ dense de déplacements, puis le segmenter en estimant des modèles de mouvement par la transformée de Hough [Adiv 85] [Kruse 96]. Une autre variante consiste à réaliser une approximation affine du champ sur des blocs puis à fusionner ces blocs selon un test de vraisemblance [Bouthemy et Santillana Rivero 87] ou un algorithme de classification [Adelson et Wang 94].

- La troisième approche consiste à réaliser une estimation conjointe de la segmentation et des mouvements. Pour cela, ces deux éléments sont pris en compte dans une fonction d'énergie globale qu'il faut minimiser [Murray et Buxton 87]. La plupart des travaux dans ce domaine partent d'une modélisation markovienne semblable à celle vue dans la section 1.1.1 [Lalande 90] [Lalande et Bouthemy 90] [Bouthemy et Lalande 90] [François 91] [Bouthemy et François 93] [Stiller 93] [Chang et al. 94] [Odobez 94] [Garcia-Garduño 96].

1.1.6 Modèles déformables *ad hoc* 2D et 3D

L'utilisation de ces modèles suppose que l'on se fixe un domaine d'application bien particulier, de sorte à connaître à l'avance les types d'objets vidéo que l'on va y rencontrer. Si l'on prend par exemple la visiophonie, on peut faire l'hypothèse que l'on va rencontrer une ou plusieurs personnes, cadrées d'une certaine façon. On peut alors utiliser un modèle de tête [Aizawa et al. 89] [Yulie et al. 89] [Samal et Iyengar 92] [Bala et al. 97] [Wollborn et al. 97] [Kampmann 97], de tête et épaules, de buste, ... Ces modèles peuvent être tridimensionnels [Musmann et al. 89] [Yin et Basu 97] mais ils sont le plus souvent bidimensionnels. Ils ressemblent alors fortement aux modèles d'éléments finis, avec en plus un fort *a priori* sur la forme. La mise en correspondance entre le modèle et l'objet observé dans les images est effectuée par la succession d'une déformation globale (modèle de transformation comme dans la section 1.1.5) et de déformations locales.

L'*a priori* sur la forme est utilisé avec profit dans l'estimation du modèle. Dans [Cootes et al. 95] est exposée une technique générale qui peut s'appliquer à différents modèles déformables. Par exemple, [Kervrann et Heitz 94] [Kervrann 95] traite les cas des modèles de mains et de lèvres. Une analyse en composantes principales (ACP) est effectuée sur une base d'apprentissage de formes. La moyenne de la base d'apprentissage donne un modèle prototype moyen. De plus, l'ACP permet deux choses. D'une part l'estimation la vraisemblance d'une occurrence déformée du modèle. D'autre part la définition des modes

de déformations du modèle, qui sont les vecteurs propres de la matrice de covariance des formes de la base d'apprentissage. Ensuite un suivi temporel peut être réalisé grâce à une évolution des paramètres de ce modèle au cours du temps (position, échelle, rotation, modes de déformation).

1.1.7 Combinaisons possibles de ces modèles

Une première façon de combiner ces modèles est d'utiliser le résultat de l'estimation d'un modèle pour initialiser l'estimation d'un modèle plus complexe. Par exemple on peut commencer par effectuer une mise en correspondance de blocs (*block-matching*) pour initialiser un champ dense, qui lui-même pourra servir de base pour initialiser des régions et leurs mouvements paramétriques.

Une deuxième façon consiste à hybrider spatialement ces modèles. Par exemple, dans MPEG4, les VOP peuvent être vus comme des régions au sein desquelles le mouvement est décrit par blocs comme dans MPEG2. Dans des travaux comme [Altunbasak et Tekalp 96] on s'intéresse à une région dans laquelle le mouvement est décrit par un maillage actif. On peut même imaginer un modèle basé sur un maillage actif, mais où le mouvement dans chaque maille serait indépendant du mouvement des sommets du maillage. On pourrait alors choisir l'un des modèles de mouvement de la section 1.1.5, avec comme inconvénient un coût de codage plus important.

Dans ces exemples, on voit en fait que le modèle par régions est le plus riche dans la mesure où il englobe les autres. Une voie de recherche intéressante serait de mettre en œuvre un modèle unificateur dans lequel le mouvement de chaque région pourrait être décrit soit par un maillage, soit par des points singuliers, soit par un modèle paramétrique, soit par un modèle déformable *ad hoc*.

1.2 Discussion

Dans cette section, nous examinons les modélisations décrites précédemment avec un angle de vue lié à notre application à l'interpolation temporelle.

1.2.1 Richesse des modélisations

La richesse de modélisation est une caractéristique importante pour notre application car nous voulons pouvoir traiter toute séquence, quel que soit le mouvement des objets.

De ce point de vue, le modèle de champ dense est le plus riche car il permet de représenter un mouvement quelconque. La contrepartie de cette richesse est la difficulté de son estimation. La régularisation spatiale, indispensable pour l'estimation peut être vue comme une façon de réduire *a priori* le nombre de champs admissibles. Inversement, des modèles plus restreints offrent une connaissance *a priori* qui aide l'estimation d'autant plus que cet *a priori* est fort. Mais dans le cas extrême des modèles déformables *ad hoc*, les hypothèses sont trop restrictives pour notre application dans laquelle nous souhaitons pouvoir traiter des séquences quelconques. Ils ne pourraient donc être envisagés que dans le cadre d'un modèle hybride (section 1.1.7).

D'après ces remarques, le modèle le plus adapté semble celui des régions, avec un mouvement décrit par maillage, par des points singuliers ou par un modèle paramétrique. En effet, il s'agit d'un modèle générique et même dans le cas des modèles paramétriques de mouvement, on peut toujours approcher un champ dense avec la précision souhaitée, quitte à réduire la taille des régions.

1.2.2 Codage du mouvement

Le coût de codage d'un modèle de mouvement doit être pris en compte pour notre application au codage interpolatif. C'est particulièrement vrai dans le cas du codage à bas débit, où le coût de l'information de mouvement augmente sa part relative par rapport au coût de l'information de texture (codage de l'image d'erreur). Le coût du mouvement se décompose en la somme de deux termes : le coût de la partie géométrique du modèle et le coût de la partie cinématique.

C'est surtout le modèle du champ dense qui est handicapé dans cette comparaison. En effet, le coût de la partie géométrique a beau être nul, il y a un très grand nombre de vecteurs de déplacement à transmettre. Une façon d'annuler ce coût est d'employer une méthode d'estimation pel-récursive [Netravali et Robbins 79] [Walker et Rao 84]. La transmission du champ est alors inutile car il est estimé en parallèle par le codeur et le décodeur. La contrepartie est une qualité moindre du champ obtenu, et un plus grand coût de codage de l'erreur de prédiction.

Quelques méthodes ont tenté de réduire le coût du champ. Dans [Nguyen 90] les composantes horizontales et verticales du champ subissent une transformation par DCT puis sont quantifiées et encodées, d'une façon similaire aux images de niveaux de gris. Dans [Nguyen et Dubois 90] deux méthodes sont employées : une découpe du champ en quadtree, et un codage par l'algorithme CELP bidimensionnel (codage DPCM suivi d'une quantification vectorielle). Mais les résultats obtenus sont de l'ordre de 0,1 bit/pixel, ce qui reste assez important pour un codage à bas débit.

Quant au modèle par régions, c'est le modèle de mouvement paramétrique qui semble le moins coûteux à transmettre. En effet, dans un modèle par maillage à l'intérieur d'une région comme dans [Altunbasak et Tekalp 96] [Toklu et al. 97a], le maillage est contraint à épouser la forme de la région, ce qui génère un nombre important de points qui ne sont pas forcément utiles à une description précise du mouvement.

1.2.3 Discontinuités spatiales du mouvement

Le champ de mouvement réel comporte souvent des discontinuités. Celles-ci sont dues aux discontinuités de profondeur dans la scène, combinées à un mouvement de la caméra ou à un mouvement relatif de deux objets. Elles sont aussi liées aux zones découvertes et recouvertes, qu'il est important de traiter dans un algorithme d'interpolation temporelle. Une modélisation adaptée à l'interpolation doit donc être capable de les représenter.

Les figures 1.1 et 1.2 montrent les résultats que l'on obtient sur nos séquences de test avec la modélisation du mouvement par blocs. On peut y voir l'un des défauts bien connus de ces méthodes, à savoir les effets de blocs. Ils sont causés par l'inadéquation de la partition

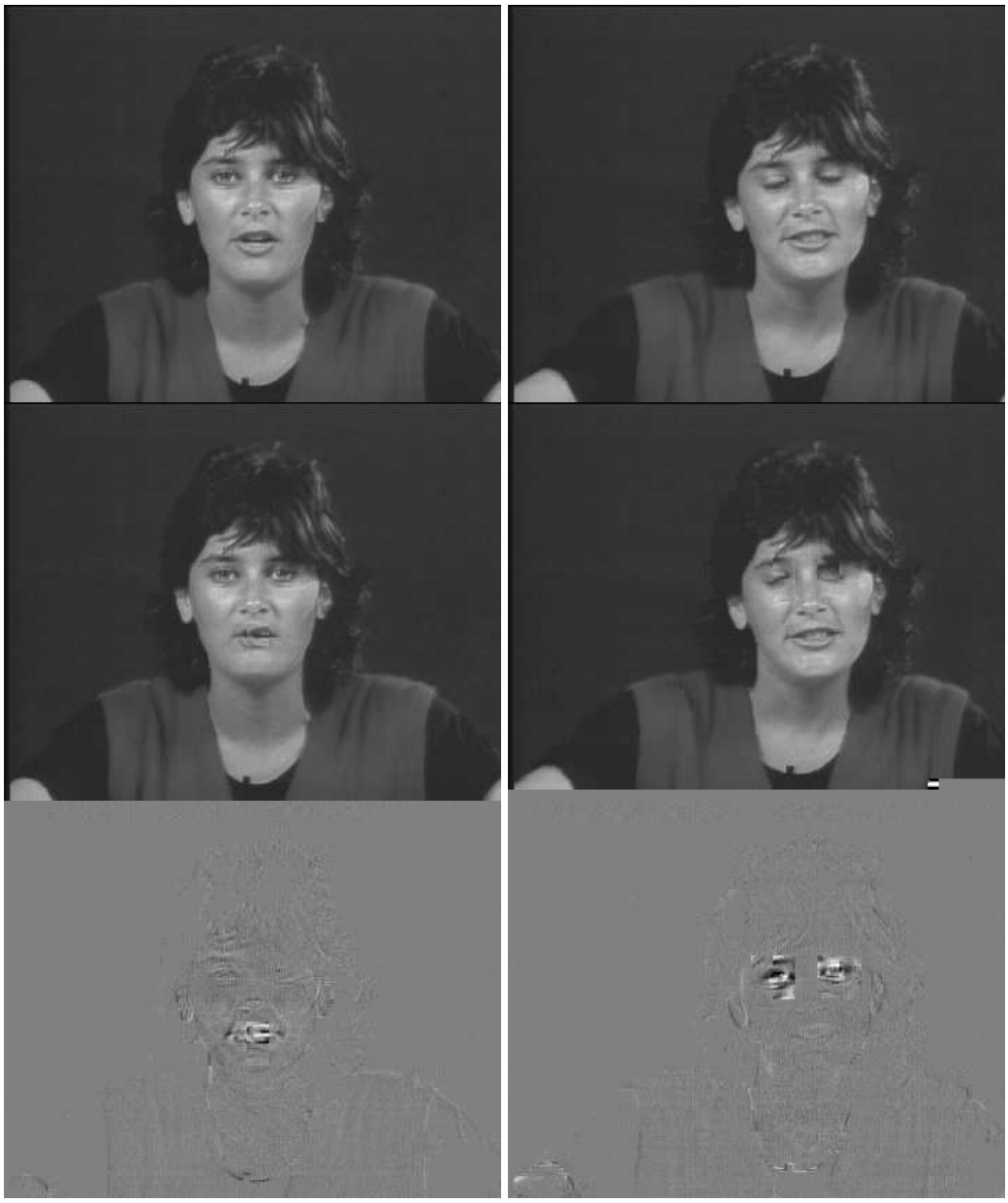


FIG. 1.1 – *Séquence* «Miss America»*, images 30 et 48. Modélisation du mouvement par blocs (16×16 pixels, mouvement de translation). À chaque instant : image originale, image compensée en mouvement, image d'erreur. L'image d'erreur est amplifiée d'un facteur 2 et recentrée autour du niveau de gris 128. Les erreurs les plus visibles sont respectivement localisées sur la bouche et les yeux.*

FIG. 1.2 – *Séquence* «Interview»*, image 19. Modélisation du mouvement par blocs (16×16 pixels, mouvement de translation). Image originale, image compensée en mouvement, image d'erreur. L'image d'erreur est amplifiée d'un facteur 2 et recentrée autour du niveau de gris 128. L'erreur la plus visible est à l'intersection de l'épaule et des carreaux, où une ligne horizontale a été dupliquée.*

fixe et régulière aux objets en mouvement de l'image. Ils sont visuellement gênants car ils forment des lignes horizontales ou verticales que l'œil perçoit mieux que d'autres types d'artefacts. De plus, les vraies discontinuités ne peuvent être représentées.

Quant aux modélisations par maillage ou par points singuliers, elles produisent un champ continu. Pour représenter un champ discontinu, il faut alors utiliser un artifice :

- Dans le cas du maillage plaqué sur toute l'image [Dudon 96], il faudrait introduire des mailles infiniment fines dont les frontières seraient de part et d'autre des contours en mouvement. Mais en pratique, les points du maillage sont placés sur les contours d'occultation. Cela a une influence sur les mailles de la région située en arrière plan : dans les zones recouvertes elles se contractent artificiellement et dans les zones découvertes elles se dilatent artificiellement. À cause du lien direct entre la position des mailles et le champ de mouvement, cet effet a aussi une influence préjudiciable sur le champ des déplacements à l'intérieur de ces mailles. Dans [Hsu et Liu 97] une technique est proposée pour découper en deux parties les mailles traversées par un contour en mouvement, mais cette solution n'est pas entièrement satisfaisante car elle se prête mal à un suivi temporel.

- Dans le cas des points singuliers [Le Floch 97], le même inconvénient apparaît : pour représenter les discontinuités, il est nécessaire d'accumuler un grand nombre de points de part et d'autre des contours en mouvement, ce qui se traduit par un coût de codage plus important.

Là encore, le modèle par régions apparaît satisfaisant car les discontinuités y apparaissent naturellement.

1.2.4 Continuité temporelle du mouvement

Toutes les modélisations présentées peuvent se prêter à une interpolation temporelle à court terme. Il s'agit de recréer une ou plusieurs images entre les deux images I_{t_1} et I_{t_2} ayant servi à estimer le mouvement. Les modélisations présentées y suffisent dans la limite des problèmes d'occultations (voir la sous-section 1.2.3). Mais pour des raisons expliquées dans la section 1.3.1 et dans le chapitre 3, nous souhaitons pouvoir effectuer un suivi et une interpolation temporelle à long terme du mouvement. Pour la suite de cette discussion, nous supposerons que le mouvement a été estimé de I_{t_1} vers I_{t_2}, ce qui est plus naturel lorsqu'il s'agit de suivi temporel et plus seulement de codage par compensation de mouvement.

Le suivi temporel impose le choix d'un modèle de mouvement tel que l'entité géométrique qui le compose puisse être suivie dans le temps c'est-à-dire identifiée comme un même objet tout au long d'une séquence d'images. Les modèles de champ dense et de blocs sont donc exclus car :

- un pixel de I_{t_1} est déplacé vers un point de coordonnées réelles dans I_{t_2}. Si l'on prenait des déplacements entiers, il y aurait des problèmes de pixels de I_{t_1} rentrant en collision dans I_{t_2} et des pixels de I_{t_2} sans antécédent dans I_{t_1}. Le seul moyen d'avoir une bijection entre I_{t_1} et I_{t_2} est de modéliser le mouvement avec une pure

translation sur toute l'image, comme cela est fait dans [Taubman et Zakhor 94], mais c'est trop restrictif dans notre cas.

- un bloc de I_{t_1} est déplacé vers un autre bloc de I_{t_2}, mais celui-ci n'est en général plus positionné sur la grille des blocs, ce qui interdit de le suivre dans une troisième image.

Quant aux autres modèles, ils sont parfaitement aptes au suivi temporel, par des modifications de la géométrie des supports et par une évolution des paramètres cinématiques.

1.3 Représentations associées à la modélisation par régions et suivi temporel

D'après la discussion précédente, il apparaît que la modélisation du mouvement par régions est la plus adaptée à notre application. Il convient maintenant de choisir une représentation de la partition en régions. En effet, plusieurs représentations sont possibles : la représentation par carte d'étiquettes, les représentations par contours fermés, ... Nous montrons ici les inconvénients de ces représentations pour le suivi temporel et pour une application à l'interpolation, ce qui nous amènera à définir notre propre représentation, mieux adaptée, dans le chapitre 2. Nous y ferons alors une comparaison plus détaillée entre représentations.

En fait, ce n'est pas tant la représentation elle-même qui nous intéresse que les algorithmes qui opèrent sur elle. Son choix va donc être guidé par les algorithmes de suivi temporel qui peuvent s'appliquer à telle ou telle représentation. Dans les sous-sections qui suivent, nous regardons plus en détails quelques algorithmes de suivi temporel qui s'adaptent à la modélisation par région, et des algorithmes spécifiques à ces représentations. Nous les examinons à la lumière de notre application, qui nous impose des contraintes de stabilité temporelle, de précision de la partition et de prise en compte des occultations.

1.3.1 Suivi temporel

Notons dès à présent la différence essentielle entre le suivi temporel d'un objet unique, généralement en avant-plan, et le suivi de n objets multiples formant une partition de l'image. Le second problème est plus difficile que les n problèmes indépendants puisqu'il faut tenir compte des occultations entre objets.

Des algorithmes de suivi temporel sont souvent employés pour suivre dans une séquence d'images des primitives simples comme des segments [Deriche et Faugeras 90] et *a fortiori* pour des primitives plus complexes, en particulier dans le cas présent, des régions de mouvement homogène paramétrique. En effet ils présentent *a priori* de nombreux avantages par rapport à des estimations effectuées indépendamment sur chaque image :

Gain en temps de calcul : Il est plus rapide de partir d'une première estimation de la partition et des mouvements, tirée de l'image précédente, que de ne partir de rien, puisqu'il suffit de la réajuster.

Stabilité temporelle : Les régions n'étant pas réinitialisées à chaque image, il y a des chances (plus ou moins grandes selon la représentation et l'algorithme choisi) de les retrouver d'une image sur l'autre.

Enrichissement du modèle : Le fait d'identifier une région comme un même objet dans la séquence, si l'algorithme y parvient, permet de définir un modèle que l'on peut qualifier de «modèle d'objet vidéo». Dans ce modèle, l'entité géométrique est un «**tube**» dans l'espace formé par les plans image et l'axe du temps, composé de la région considérée à ses différents instants. Quant à l'entité cinématique, elle devient une **trajectoire** dans ce même espace.

La base de la plupart des algorithmes de suivi est un mécanisme de prédiction puis d'ajustement. On utilise le mouvement passé d'une région pour prédire son mouvement futur, ainsi que sa position dans l'image à traiter. Cette prédiction peut être réalisée par une simple utilisation de son mouvement à l'instant précédent. Une prédiction plus complexe consiste à employer un filtre de Kalman, incluant un modèle d'évolution du mouvement à vitesse ou à accélération constante [Meyer et Bouthemy 92] [Meyer 93] [Meyer et Bouthemy 94] [Bascle et al. 94]. Une généralisation du filtre de Kalman est présentée dans [Isard et Blake 98a] et [Isard et Blake 98b]. Il s'agit de l'algorithme dit de «condensation» qui prend en compte des densités de probabilité multi-modales (et plus seulement gaussiennes), ce qui permet de suivre des contours multiples et de suivre simultanément plusieurs hypothèses de prédiction.

L'ajustement dépend, lui, de la représentation choisie. Nous allons donc maintenant considérer les différentes représentations annoncées et les algorithmes d'ajustement qui leur sont associés.

1.3.2 Représentation par cartes d'étiquettes

Cette représentation est la plus simple qui soit : à chaque pixel de l'image est associée une étiquette qui est le numéro de la région à laquelle appartient le pixel. C'est la plus employée dans de très nombreux travaux en raison de la simplicité de sa mise en œuvre [Lalande et Bouthemy 90] [Bouthemy et Lalande 90] [Diehl 91] [Black 92] [Odobez 94] [Odobez et Bouthemy 95] [Nzomigni 95] [Garcia-Garduño 96].

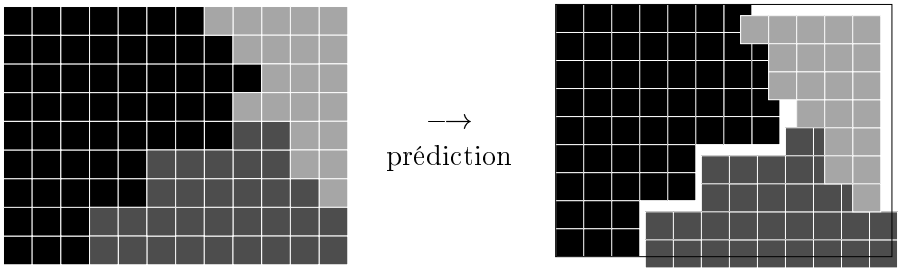


FIG. 1.3 – *Suivi temporel sur une représentation par carte d'étiquettes.*

Un premier inconvénient apparaît pour la prédiction, illustrée dans la figure 1.3. La carte d'étiquette est prédite par compensation de mouvement : on applique le mouvement

prédit de chaque région à son masque. On voit que le résultat est une carte dans laquelle des régions se chevauchent et des zones non affectées existent. Ces problèmes peuvent cependant se résoudre si l'ordre de superposition des régions est connu.

Dans les travaux cités, l'ajustement de la partition prédite est le plus souvent effectué selon un modèle markovien par la minimisation d'une fonction d'énergie définie sur les étiquettes (voir la sous-section 1.1.5). Nous allons en particulier considérer la technique utilisée dans [Odobez 94] et commenter les résultats obtenus. Les figures 1.4 et 1.5 montrent la segmentation de nos séquences de test.

FIG. 1.4 – *Séquence* «Miss America»*, images 1, 32, 33, 56 et 62. Représentation du mouvement par carte d'étiquettes. Segmentation basée mouvement obtenue par une technique markovienne [Odobez 94].*

Dans la séquence *«Miss America»*, qui est assez difficile à traiter correctement, on constate que la méthode ne conduit pas à des résultats toujours satisfaisants (voir la figure 1.4). Les images présentées sont celles où suffisamment de mouvement est présent pour obtenir une région autre que le fond. Dans les premières images, le mouvement est suffisant pour que l'algorithme détecte quelques régions. Mais dans le reste des images, malgré le suivi temporel, le mouvement est trop faible pour que l'algorithme trouve des objets en mouvement. Quand le mouvement redevient significatif (images 32 et 33), les régions observées n'ont plus rien à voir avec celles du début de la séquence. On observe même un changement de topologie important de la partition entre ces deux images pourtant consécutives.

Dans la séquence *«Interview»*, les résultats sont nettement meilleurs, mais on observe cependant deux défauts principaux : (voir la figure 1.5)

Imprécision des frontières : Ceci se produit pour les frontières entre un objet en avant plan et un fond peu texturé. Le terme de régularisation spatiale de la carte d'éti-

FIG. 1.5 – *Séquence* «Interview»*, images 11, 12, 14 et 15. Représentation du mouvement par cartes d'étiquettes. Segmentation basée mouvement obtenue par une technique markovienne [Odobez 94].*

quettes tend alors à étendre artificiellement la région d'avant plan au détriment des régions du fond. Ce phénomène peut être observé par exemple tout autour des cheveux de la personne.

Instabilité temporelle : Lorsqu'une frontière ne correspond pas à un contour en mouvement, mais plutôt à un changement progressif du mouvement au sein d'un même objet, il n'y a pas de raison pour qu'elle reste en place d'une image sur l'autre si l'on ne rajoute pas dans la fonction d'énergie un terme de régularisation temporelle. Dans [Odobez 94], la régularisation temporelle est assurée par un terme d'énergie qui tend à favoriser la similitude entre la carte courante et la carte précédente, compensée en mouvement. On pourrait augmenter le poids de ce terme de régularisation temporelle, mais ce serait au détriment de la précision des frontières. Ce phénomène peut être observé entre les images 11 et 12 en haut des cheveux et sur la frontière horizontale située au milieu de la salopette, qui passe de bas en haut d'un replis sombre du vêtement. De même, entre les images 14 et 15, la région entourant la mèche de cheveux est coupée en deux, et la région de la main est étendue à une partie de l'avant bras.

Des résultats similaires ont été observés avec trois algorithmes, disponibles pour nos tests, utilisant la représentation par carte d'étiquettes [Odobez 94] [Nzomigni 95] [Garcia-Garduño 96]. Ceci tend à nous faire penser que ces défauts sont plus dus à la représentation elle-même qu'aux algorithmes. Peut-être que le pixel, ou le bloc de pixels dans le cas des modèles hiérarchiques, n'est pas une entité d'assez haut niveau. Par opposition, un contour entre objets se situe à un niveau de description plus élevé et semble mieux adapté au problème de la segmentation.

1.3.3 Contour actif

Les contours actifs ont été utilisés à des fins de segmentation spatiale d'un objet [Kass et al. 87] [Kass et al. 88] [Berger 91], mais aussi pour effectuer un suivi temporel [Terzopoulos et Szeliski 92] [Ueda et Mase 92] [Blake et al. 93] [Leymarie et Levine 93] [Berger 93] [Bascle 94] [Bascle et al. 94].

Le principe consiste à définir une courbe paramétrique fermée qui représentera le contour de l'objet à segmenter. Il peut s'agir d'un polygone, d'une B-spline, ... Ensuite, on définit une fonction d'énergie sur ce contour. Elle se compose d'un terme d'attache aux données qui, dans le cas le plus simple, est l'opposé de l'intégrale du gradient spatial de l'image le long de la courbe. Viennent ensuite des termes de régularisation comme par exemple l'intégrale de la valeur absolue de la courbure. On peut aussi rajouter un terme qui incite la courbe à se raccourcir, défini comme l'intégrale de la norme de la dérivée de la courbe. Dans ce dernier cas, on initialise la courbe près des bords de l'image, et elle vient se coller sur les contours d'un objet qui était initialement à l'intérieur. L'approche opposée a été proposée dans [Cohen 91] et [Cohen et Cohen 93] où un terme d'énergie fait «gonfler» la courbe comme un ballon.

Quant au suivi temporel, dans [Bascle et al. 94], il est effectué par une prédiction par compensation de mouvement, grâce au mouvement estimé à l'intérieur de la région. L'ajustement est décomposé en deux étapes : un ajustement par une transformation affine, et une déformation libre selon le même algorithme que dans le cas spatial.

Ces techniques ont deux avantages principaux par rapport aux algorithmes de suivi d'une carte d'étiquettes, mais un inconvénient supplémentaire :

Précision de la localisation des frontières : Elle est potentiellement très bonne puisque l'ajustement se fait sur des informations de gradient spatial.

Stabilité temporelle : Elle peut aussi être assurée puisque l'on peut faire en sorte que le polygone ou la courbe spline qui entoure l'objet garde ses sommets ou ses points de contrôle stables.

Prise en compte d'objets multiples : Par contre, ces techniques s'adaptent difficilement à un suivi temporel d'un objets multiples engendrant des occultations ou des découvrements. Elles sont généralement appliquées à un objet unique en avant plan. En effet, comme on peut le voir sur la figure 1.6, une région qui n'est pas en avant plan peut subir une variation trop grande de sa forme pour que le contour actif puisse la suivre. Ces techniques ont été étendues au cas multi-objets par les techniques de *level set* dans le cas spatial [Caselles et al. 95], puis dans le cas d'objets en mouvement pour le problème de la détection du mouvement [Paragios et Deriche 97] et pour le

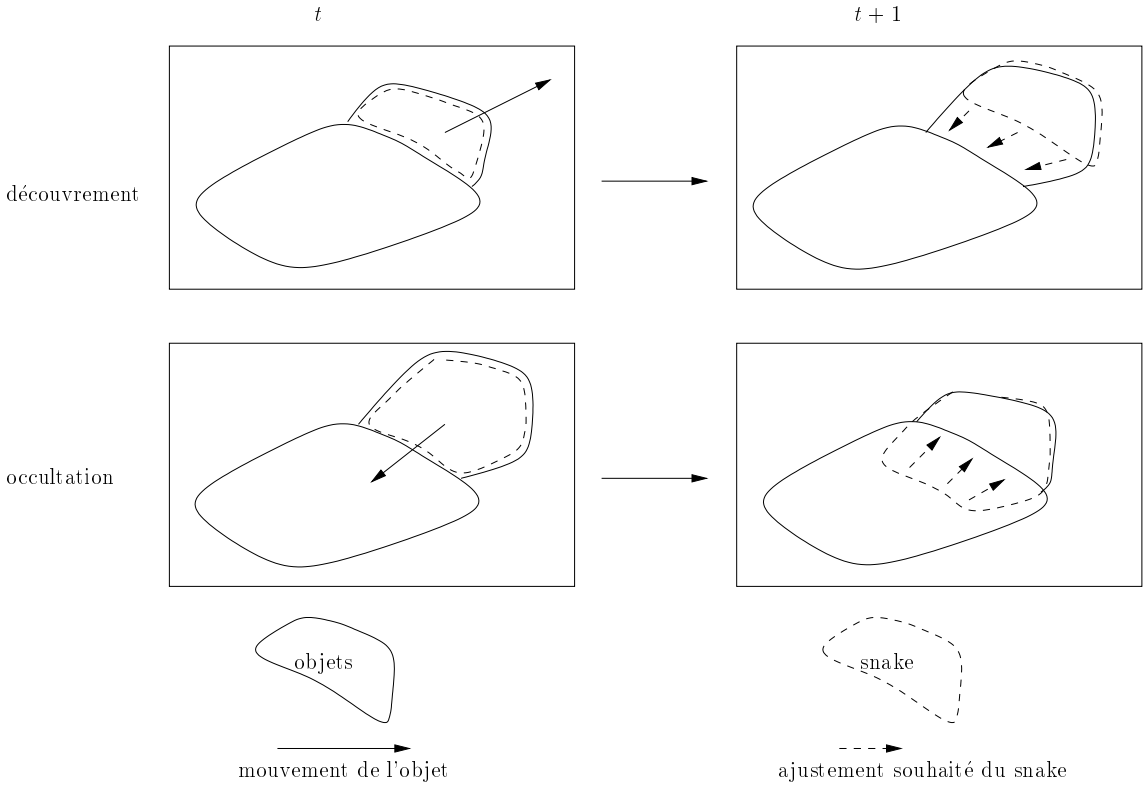


FIG. 1.6 – *Suivi temporel par contour actif fermé : recouvrement et découvrement. L'ajustement doit être d'une amplitude très importante et risque d'être perturbé par les gradients spatiaux présents dans la zone de découvrement ou de recouvrement.*

problème du suivi [Paragios et Deriche 98]. Mais on ne peut suivre que quelques objets bien distincts, n'étant pas en contact les uns avec les autres. Les objets rentrant en contact fusionnent naturellement et l'algorithme ne garde pas trace des anciennes régions.

1.3.4 Représentation par contours fermés

Dans le but de résoudre les difficultés posées par les contours actifs, une autre représentation de la partition en régions a été définie dans [Wu et al. 95] [Wu 95] [Benois-Pineau et al. 96]. Il s'agit d'une représentation dans laquelle chaque région est définie par un polygone fermé.

L'algorithme de suivi temporel associé se décompose en 4 étapes ([Wu 95], paragraphe 3.3.2.2) (voir la figure 1.7) :

1. un premier ajustement des paramètres de mouvement des régions par une estimation de mouvement
2. un ajustement des frontières, au niveau des polygones, par une technique de polygones ajustables [Delagnes et al. 95]
3. un traitement, au niveau des pixels, des zones de recouvrement et de découvrement (occultations)
4. la restitution de la connexité de la segmentation et la réestimation des paramètres de mouvement

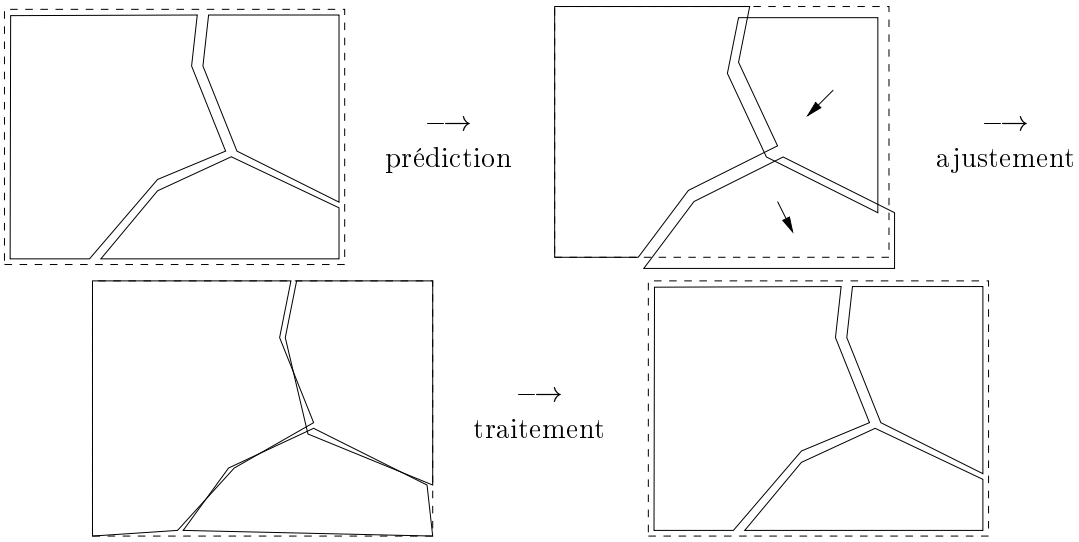
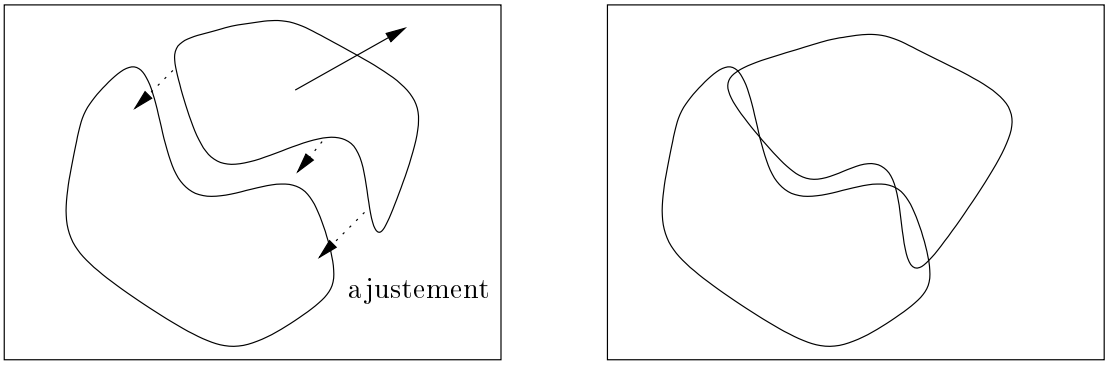


FIG. 1.7 – *Suivi temporel sur une représentation par contours fermés.*

Cette représentation, associée à cet algorithme, marche assez bien, mais présente tout de même quelques inconvénients :

- La représentation est redondante dans la mesure où chaque frontière entre deux régions apparaît deux fois dans la représentation, une fois pour chaque région de part et d'autre de cette frontière. Ceci conduit à un double ajustement : chaque frontière donc est ajustée deux fois, comme appartenant à deux régions.



après prédiction
zone découverte

après ajustement
zones d'intersection et de non-recouvrement

FIG. 1.8 – *Ajustement sur des contours fermés. Sur cette figure, on suppose que l'ajustement s'est correctement effectué pour les deux côtés de la frontière séparant les deux régions. Dans ce cas, une zone de découvrement devient un ensemble de zones d'intersection et de non-recouvrement.*

- Après ajustement des frontières (étape 2), les polygones obtenus ne forment pas une partition de l'image. Appelons «zone d'intersection» une intersection entre polygones et «zone de non-recouvrement» une zone n'appartenant à aucun polygone. Comme le montre la figure 1.8, une zone d'intersection n'est pas forcément liée à un recouvrement et une zone de non-recouvrement n'est pas forcément liée à un découvrement. Ainsi l'étape 3 porte mal son nom et devrait en fait s'appeler «traitement des zones d'intersection et de non-recouvrement». L'inconvénient est que l'ajustement de l'étape 2 risque de multiplier artificiellement le nombre de ces zones. De plus comme elles ne sont plus liées aux occultations, il est délicat de se servir des informations de mouvement pour les traiter.

- Dans l'étape 3, cet algorithme traite séparément les cas d'intersections et de non recouvrement. Cela n'est pas gênant dans [Wu 95] car il est utilisé dans le cadre d'un codage causal par compensation de mouvement. Mais pour notre étude appliquée à l'interpolation, nous souhaitons avoir un traitement symétrique des deux cas. En effet, par un retournement de l'axe du temps les découvrements deviennent des recouvrements et inversement. Nous verrons dans la section 3.5 que le critère nous servant au traitement des occultations est symétrique vis à vis d'une inversion du temps.

- Le traitement des zones non recouvertes est effectué après la prédiction, au niveau du pixel et à chaque image. Sur ces trois points l'algorithme de suivi temporel que nous détaillerons dans le chapitre 3 apportera une amélioration, mais il est trop tôt pour effectuer la comparaison, qui sera donc faite dans la section 3.6.4.

1.4 Synthèse

Après avoir examiné la plupart des modélisations du mouvement dans une séquence d'image, il apparaît que de nombreux modèles sont insuffisants pour les applications que nous nous sommes fixées :

- les champs denses car ils se prêtent difficilement à un codage efficace et que la continuité temporelle est problématique
- les modélisations par blocs car elles introduisent des discontinuités spatiales artificielles et ne permettent pas la continuité temporelle
- les maillages car, utilisés seuls, ils ne permettent pas de traiter les discontinuités spatiales de façon satisfaisante
- les modèles déformables *ad hoc* car ils ne sont pas assez généraux

Notre choix c'est donc porté sur une modélisation du mouvement par régions, avec un mouvement décrit par un modèle paramétrique à l'intérieur de chaque région.

Nous avons ensuite examiné la question de comment représenter la partition de l'image en régions. Il nous a semblé que la représentation par une carte d'étiquettes, déjà largement explorée, pouvait trouver une alternative intéressante dans des représentations par les contours des régions. Les algorithmes de suivi temporel associés sont alors basés sur des contours actifs multi-objets.

Mais la représentation par contours fermés, utilisée dans [Wu 95], nous a paru perfectible à cause de sa redondance et du traitement particulier des occultations qu'elle oblige à effectuer. Nous allons donc par la suite nous attacher à la conception d'une représentation adaptée pour le suivi temporel et pouvant satisfaire les contraintes imposées par son application au problème de l'interpolation temporelle.

Chapitre 2

Représentation d'une segmentation par les frontières des régions

Introduction

Nous avons vu dans la partie bibliographique précédente que dans le cadre d'une modélisation du mouvement par régions, les représentations habituelles, basées soit sur une carte de d'étiquettes, soit sur des contours fermés, ont toutes des défauts difficilement contournables. C'est pourquoi nous nous sommes orientés vers une autre représentation que nous présentons dans ce chapitre.

Nous avons choisi une représentation basée sur des contours car ce type de représentation offre une précision de la découpe en régions, indispensable pour notre application à l'interpolation. Mais nous voulions éviter les deux inconvénients principaux des représentations par contours fermés qui sont

- la redondance de la représentation et donc des calculs inutiles dans les algorithmes qui la manipulent,
- le traitement non satisfaisant des recouvrements et découvrements entre objets.

L'idée de départ est donc d'éviter de dupliquer les contours. Ainsi nous obtenons un gain en complexité de stockage. De plus, nous verrons dans le chapitre 3 que nous obtiendrons un gain en complexité opératoire. Pour cela, au lieu de considérer des contours entourant les régions, nous considérons des **frontières** séparant deux régions. Il s'agit alors de courbes reliant des points particuliers que nous appellerons «**points multiples**» car ils appartiennent au voisinage de plusieurs régions. Cette idée de base est développée tout au long de ce chapitre, dans quatre sections.

Plan du chapitre

Dans la section 2.1, nous commençons par présenter et définir la notion de graphe d'arrangement qui nous a servi de base pour notre représentation. Puis nous montrons

l'adaptation que nous en avons faite pour notre usage à des fins de segmentation spatio-temporelle.

Ensuite, dans la section 2.2, nous nous intéressons à l'initialisation de notre segmentation dans la première image de la séquence traitée. Nous y faisons une brève étude bibliographique et expérimentale pour comparer quelques méthodes de segmentation spatiale.

Dans la section 2.3, nous décrivons les étapes qui partent d'une segmentation spatiale initiale représentée par une carte d'étiquettes pour arriver à notre structure de représentation.

Enfin, dans la section 2.4, nous montrons quelques opérations de base que l'on peut réaliser sur cette représentation. Chaque opération est, selon le cas, soit plus simple, soit plus compliquée que dans le cas d'une représentation par carte d'étiquettes.

2.1 Définition d'une structure de représentation

La représentation de segmentation spatio-temporelle que nous avons décidé d'adopter est basée sur un graphe. Les graphes [Gondran et Minoux 86] [Aho et al. 87] ont déjà été beaucoup utilisés en traitement d'images, mais leur utilisation dépend de la modélisation choisie.

Dans des modélisations d'images basées sur les pixels, un graphe peut servir à modéliser les relations de voisinage entre pixels. Les sommets du graphe sont alors les pixels et chaque pixel est relié à ses 4 ou 8 voisins selon le type de voisinage considéré. Le graphe résultant est alors une grille régulière. Certains travaux [Wu et Leahy 93], [Kropatsch 94, Kropatsch 95], [Shi et Malik 97, Shi et Malik 98] utilisent ce graphe pour guider un processus d'agrégation optimale de pixels en vue d'une segmentation. C'est aussi le cas dans les modélisations markoviennes où chaque sommet (aussi appelé «site») est associé à une variable cachée que l'on essaye d'estimer en chaque pixel [Heitz et Bouthemy 90a, Heitz et Bouthemy 90b]. Un arc dans ce graphe traduit alors une dépendance statistique entre ces variables cachées. Dans le cas des modèles markoviens multiéchelles [Perez 93], des arcs supplémentaires peuvent relier des sites placés sur deux niveaux consécutifs de la pyramide d'estimation.

Dans les modélisations à base de régions, c'est le graphe d'adjacence qui est le plus utilisé (voir par exemple [Gu et Kunt 95]). Ses sommets sont les régions de la partition de l'image et ses arcs, non orientés, traduisent les relations d'adjacence entre régions. Ce graphe peut alors servir pour guider un processus de fusion entre régions. Il s'agit alors d'un graphe valué tel que la valeur de chaque arc est le coût associé à la fusion des deux régions [Pateux et Labit 98].

Dans notre cas, nous utilisons aussi une modélisation à base de régions, mais nous utilisons les graphes de façon différente, puisque ce sont les contours des régions qui nous intéressent, aussi bien que les régions elles-mêmes. Nous allons donc maintenant définir la représentation de segmentation que nous utilisons. Elle est aussi basée sur la notion de graphe, mais le point de départ est un graphe particulier, appelé «arrangement». Nous allons donc commencer par définir la notion de graphe d'arrangement, puis la version particulière que nous utilisons.

2.1.1 Définition d'un graphe d'arrangement

La structure d'arrangement est surtout utilisée dans le domaine de la géométrie algorithmique [de Berg et al. 97], en relation avec le problème du calcul de des intersections d'un ensemble de segments [Bentley et Ottman 79]. Nous la décrivons dans cette sous-section, et dans la suivante, nous montrerons comment nous l'avons reprise et adaptée à nos besoins.

Un graphe G est défini par un couple $G = (X, \Gamma)$ avec X un ensemble quelconque (ensemble des sommets) et Γ une partie de $X \times X$ (ensemble des arcs). Appelons n_s le nombre de sommets et n_a le nombre d'arcs (éventuellement infinis). Le nombre d'arcs incidents ou sortant d'un sommet x est appelé son degré. Nous l'appellerons aussi multiplicité et le noterons $mult(x) = card(\Gamma(x)) + card(\Gamma^{-1}(x))$.

Un arrangement A est défini par un triplet $A = (G, P, C)$ avec G un graphe non orienté, P une application de X dans un espace vectoriel normé E qui à chaque sommet associe sa position, et C une application de Γ dans un ensemble $\mathcal{C}$ de courbes paramétrées continues. Cet ensemble est par exemple un ensemble de courbes linéaires (segments) ou d'arcs de quadriques. On impose les conditions suivantes :

- les courbes doivent relier les sommets :

$$C : \left| \begin{array}{l} \Gamma \longrightarrow \mathcal{C} \\ (x, y) \longmapsto c \end{array} \right. \quad \text{avec} \quad c : \left| \begin{array}{l}]0, 1[\longrightarrow E \\ s \longmapsto c(s) \text{ tel que } c(0^+) = x \text{ et } c(1^-) = y \end{array} \right.$$

- les courbes doivent être invectives (pas d'auto-intersection) et ne doivent pas s'intersecter 2 à 2 :

$$\bigcap_{\gamma \in \Gamma} C(\gamma)(]0, 1[) = \emptyset$$

Si les courbes sont autre chose que des segments, on peut éventuellement avoir plusieurs courbes reliant 2 sommets donnés. G n'est plus alors un graphe mais un **multigraphe**.

Parmi ces arrangements, on s'intéresse plus particulièrement à un sous ensemble appelé **arrangements de type fini** et qui vérifient les deux conditions supplémentaires suivantes :

- chaque sommet a un nombre fini d'arcs incidents,
- les sommets sont isolés, c'est-à-dire l'ensemble des positions des sommets $P(X)$ n'a pas de valeur d'adhérence, ou encore pour tout compact K, $K \cap P(X)$ est un ensemble fini.

En fait, si le graphe est fini, c'est-à-dire si $n_s < \infty$, alors $n_a < \infty$ et l'arrangement est forcément de type fini. De plus, en pratique, pour représenter des données à support compact (image), on s'intéresse à des arrangements contenus dans une partie bornée de E, avec des courbes c de longueur finie.

On parle d'un arrangement plan quand E est le plan ($\dim_E = 2$). Le graphe sous-jacent est alors forcément planaire. Dans ce cadre, on définit l'ensemble $\mathcal{R}$ des **régions** où chaque

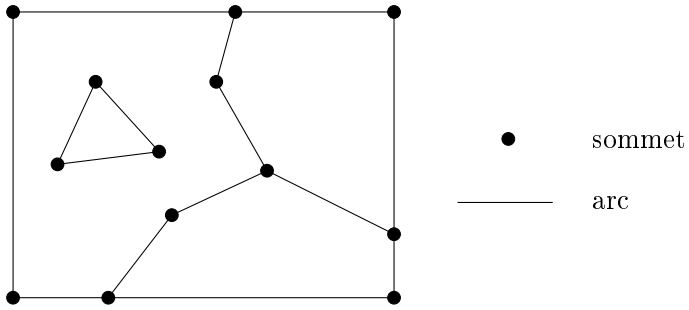


FIG. 2.1 – *Un exemple d'arrangement.*

région R est l'une des composantes connexes du complémentaire du graphe d'arrangement :

$$\bigcup_{R\in\mathcal{R}} R = \complement_E \left(P(X) \cup \bigcup_{\gamma\in\Gamma} C(\gamma)(]0,1[) \right)$$

Inversement, quand on dispose d'une partition du plan en régions connexes, on peut se poser le problème de quel arrangement peut la décrire. En fait, si elle existe, cette représentation par arrangement n'est pas unique. En effet, on peut toujours remplacer une courbe c_0 par deux courbes c_1 et c_2 mises bout à bout et reliées par un sommet x de degré 2 tel que $P(x) = c_0(k)$ avec $0 < k < 1$:

$$\left\{ \begin{array}{lcl} c_1(0) & = & c_0(0) \\ c_1(s) & = & c_0(ks) \\ c_1(1) & = & c_0(k) \end{array} \right. \quad \text{et} \quad \left\{ \begin{array}{lcl} c_2(0) & = & c_0(k) \\ c_2(s) & = & c_0(k+(1-k)s) \\ c_2(1) & = & c_0(1) \end{array} \right.$$

Les courbes reliant les sommets peuvent être quelconques (polygones, arcs de quadriques, splines, ...), mais dans la pratique on se limite souvent à des segments. C'est ce qui explique que l'on garde dans le graphe des sommets de degré 2.

2.1.2 Points multiples et frontières

À cause du problème d'unicité évoqué dans la sous-section précédente, la représentation par arrangement n'est pas entièrement satisfaisante. De plus, les arrangements contenant des sommets isolés (sommets de degré 0) ou des courbes sans prolongement (reliées à un sommet de degré 1) ne sont pas pertinents dans notre cas. Enfin, si l'on veut stocker une information commune à tous les arcs séparant 2 régions, cette information sera dupliquée dans chaque arc. D'où l'idée de se restreindre à une classe d'arrangements n'ayant pas de sommets de degré strictement inférieur à 3 (voir la figure 2.2 et la comparer avec la figure 2.1).

La représentation que nous avons finalement adoptée, est donc définie ainsi :

- G est un multigraphe.

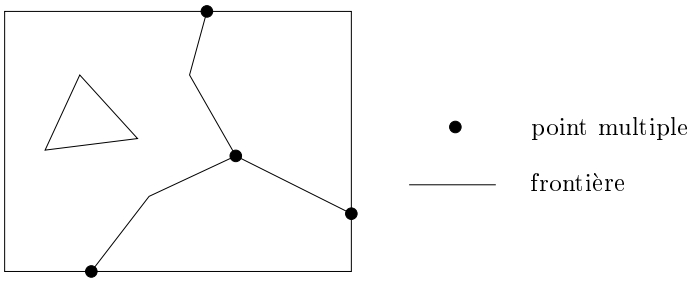


FIG. 2.2 – *Un arrangement réduit.*

- Les sommets sont de degré au moins 3 et sont appelés **points multiples**. Un sommet de degré $n \geq 3$ est donc relié à n frontières. Il appartient aussi en général au voisinage de n régions, mais il peut y en avoir moins, comme le montre la figure 2.3.

- Les arcs reliant les sommets sont associés à des courbes polygonales appelées **frontières ouvertes**. De sorte à avoir unicité de la représentation, on impose un paramétrage uniforme le long de ces polygones. Certaines frontières sont en forme de boucle isolée comme le montre la figure 2.2 (elle ne sont pas à confondre avec les boucles reliées à un point multiple (voir la figure 2.3)). Dans l'arrangement, elles ne relient que des sommets de degré 2. Pour compenser l'absence des sommets de degré 2 on définit ces frontières à part, comme des **frontières fermées**.

- Comme nous le verrons plus tard, pour faciliter certaines opérations, nous avons besoin que chaque arc soit orienté. Ceci permet de savoir quelle région est à gauche ou à droite, quand on parcourt l'arc dans le sens de son orientation.

À partir de maintenant, nous appellerons n_{ptm} le nombre de points multiples (anciennement n_s). Le mot «sommet» ne désignera plus les sommets du graphe, mais les sommets de l'approximation polygonale et leur nombre sera noté n_{somm}. Le nombre de frontières, comprenant les n_a frontières ouvertes plus les frontières fermées, sera noté n_{fr}.

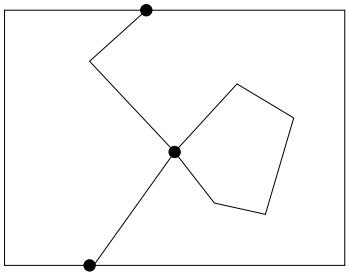


FIG. 2.3 – *Un arrangement particulier : exemple où le degré d'un point multiple est différent du nombre de régions voisines. Cette figure illustre aussi le fait qu'une boucle peut être reliée par un point de degré ≥ 4 au reste de la structure.*

Dans le cas où certaines régions comportent des «trous», la segmentation comporte des frontières fermées ou des ensembles disjoints de contours ouverts, inclus dans ces régions (voir la figure 2.2). Cette représentation qui est «plate» ne décrit donc pas complètement la situation. En plus de la structure d'arrangement, nous mémorisons donc un arbre représentant la relation suivante entre régions : $R_i \sqsubset R_j$ ssi R_j «entoure» R_i. Il s'agit de l'**arbre d'homotopie** [Coster et Chermant 85]. La racine de cet arbre est une région spéciale appelée «bord» qui est en fait l'extérieur de l'image. L'ensemble formé par le graphe G et cet arbre est ce que l'on peut appeler la «topologie» d'une segmentation. Un avantage de cette représentation est qu'elle rend explicite ces relations géométriques qui étaient implicites dans le graphe d'arrangement.

2.1.3 Contours et régions

Comme cela a déjà été dit, les **régions** sont définies comme les composantes connexes du complémentaire du graphe d'arrangement.

Un **contour** d'une région R peut se définir ainsi : il s'agit d'un n-uplet de frontières $(c_1, \ldots, c_n)$ telles que

$$\forall i \in [1..n], \begin{cases} c_i \text{ borde } R \\ c_i(1) = c_{i+1}(0) \quad (\text{et } c_n(1) = c_0(0)) \end{cases}$$

Notons qu'une région peut avoir plusieurs contours dans le cas où elle comporte des «trous» : un contour extérieur et des contours intérieurs.

Dans cette représentation, contours et régions sont donc des objets annexes, dérivés des objets principaux que sont les points multiples et les frontières. C'est une différence importante avec la représentation adoptée dans [Wu 95] où régions et contours sont les objets principaux.

2.1.4 Conversions entre représentations

Revenons maintenant sur les 3 représentations auxquelles peut donner lieu la modélisation du mouvement par régions : représentations par carte d'étiquettes (RCE), par contours fermés (RCF) et par frontières ouvertes (RFO). Comme il est parfois nécessaire de passer d'une représentation à l'autre (voir ci-dessous), le problème de la conversion entre ces 3 représentations est donc posé. Parmi les 6 conversions possibles, nous allons en examiner 3 en particulier, car elles seront utilisées à un moment ou à un autre de notre travail. Ces conversions peuvent être réalisées avec ou sans pertes, soit du fait de leur nature même, soit par le choix de l'algorithme utilisé ; ce sera indiqué à chaque description.

RFO $\longrightarrow$ RCF Cette conversion est utile par exemple pour tester si un point appartient à une région. Il suffit alors de tester si le point appartient au polygone fermé entourant la région. Cette conversion sert aussi de première étape pour une conversion vers une RCE. Elle peut se faire, région par région, grâce à un algorithme de parcours du contour de chaque région en utilisant la règle «de la main gauche» (voir la figure 2.4).

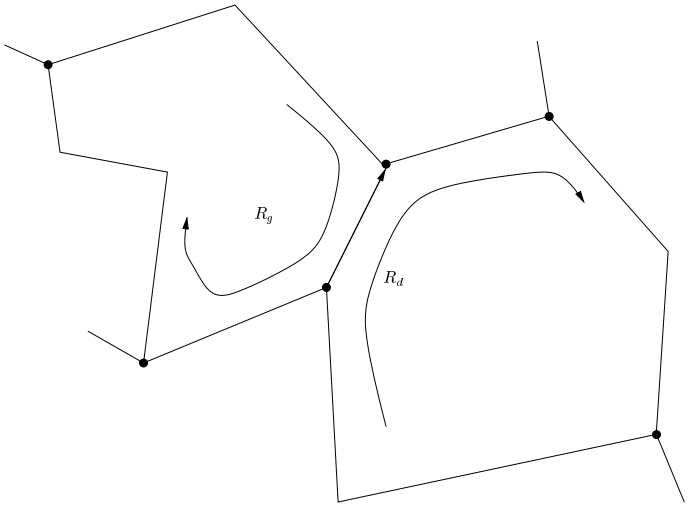


FIG. 2.4 – *Orientation et parcours du graphe de représentation. Les étiquettes de régions R_g et R_d sont attribuées aux 2 régions adjacentes en fonction de l'orientation indiquée sur la figure, qui est choisie arbitrairement. Le sens de parcours choisi est celui «de la main gauche» : en suivant le contour de la région R_g on parcourt la frontière dans le sens inverse de son orientation.*

Pour cela, on munit chaque frontière de 2 marques, chacune correspondant à un sens de parcours. On part d'une frontière quelconque que l'on parcourt dans un sens quelconque. En arrivant sur le point multiple situé à l'extrémité de cette frontière, on choisit, parmi les autres frontières, celle qui forme avec elle l'angle le plus petit (précisons que l'on choisit une orientation directe pour le plan). On marque la frontière choisie pour le sens de parcours effectué, et on répète cette étape jusqu'à revenir sur la frontière de départ. Tant que toutes les frontières ne sont pas marquées deux fois, on réitère le processus.

En pratique, on munit chaque frontière (c'est-à-dire chaque arc du graphe) d'une orientation. Les marques peuvent alors être remplacées par les indices des régions gauche et droite (voir la figure 2.4). Ces indices sont alors conservés car ils facilitent un nouveau parcours. En effet, en arrivant sur un point multiple, au lieu de recalculer des angles entre frontières, il suffit alors de regarder quelle(s) frontière(s) porte(nt) le même indice de région, et ce du même côté gauche ou droit.

Cette conversion est sans pertes et réversible, bien que nous n'utilisons pas cette dernière propriété. Sa complexité est dans $O(n_a)$.

RFO⟶ RCE Cette conversion est utile pour l'estimation du mouvement des régions. En effet, les estimateurs de mouvement classiques sont fondés sur la minimisation d'une fonction d'énergie, définie comme une somme sur les pixels de la région. Ils ont donc besoin de la liste des pixels contenus dans une région. Or l'un des moyens les plus efficaces d'obtenir cette liste est de passer par une RCE. Elle s'obtient tout simplement par un algorithme de remplissage (ou de coloriage) appliqué à chaque région.

Pour cela, nous avons utilisé l'algorithme décrit dans [Pateux et Labit 97]. Cependant, un problème se pose dans le cas où 2 segments forment un angle très aigu (voir [Pateux et Labit 97], figure 15a). Des pixels isolés peuvent se trouver dans cet angle aigu et donc former des régions qui sont en fait des artefacts. Pour résoudre le problème, ces petites régions sont détectées puis les pixels réaffectés à leur vraie région, grâce à un test exact d'inclusion dans le contour polygonal de la région. Toutefois, ce problème n'est pas vraiment pénalisant dans la mesure où quelques pixels manquant dans une région ne risquent pas de biaiser l'estimation de son mouvement.

Cette conversion est avec perte. En effet, la conversion inverse utilise une approximation polygonale. Donc si l'on repasse de la carte d'étiquettes à une approximation polygonale de frontière, on n'est pas certain de retrouver le même polygone. En théorie, sa complexité est dans $O(n_{fr} + n_{pixels})$, avec n_{pixels} le nombre de pixels contenus dans l'image. En pratique la complexité est dans $O(n_{pixels})$, car on a toujours $n_{fr} \leq 2 * n_{pixels}$ (l'égalité correspondant au cas extrême où il y a n_{pixels} régions de taille 1 pixel).

RCE$\longrightarrow$ RFO Cette conversion est utile dans la phase initiale de l'algorithme de suivi de segmentation. Nous faisons cette conversion une fois pour toutes dans la première image de la séquence pour initialiser notre représentation à partir de la RCE d'une segmentation spatiale, obtenue par l'une des méthodes de la section 2.2. Cette conversion n'est pas détaillée ici, mais le sera dans la section 2.3.

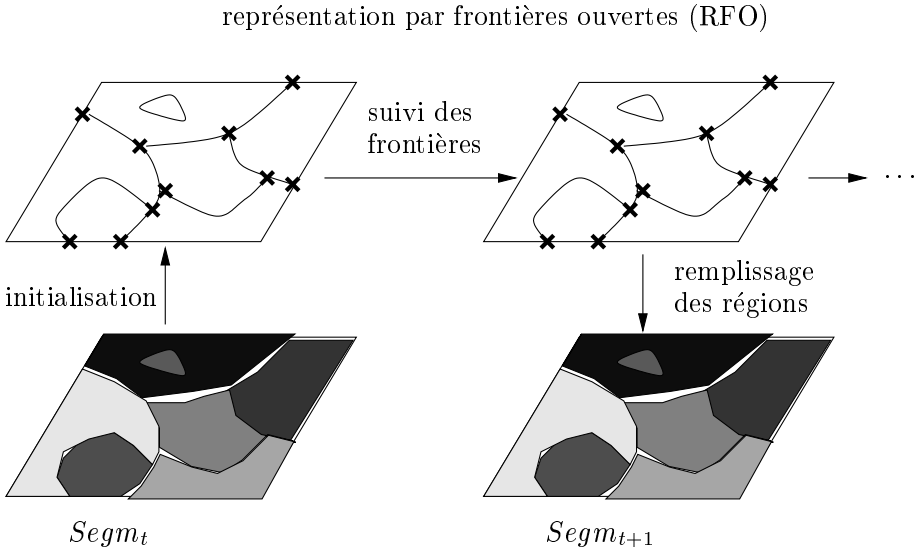


FIG. 2.5 – *Suivi temporel dans le cadre d'une* RFO.

La RFO est la représentation que nous allons utiliser pour le suivi temporel. Elle est plus adaptée pour cette tâche puisque nous l'avons définie dans ce but. Cependant nous

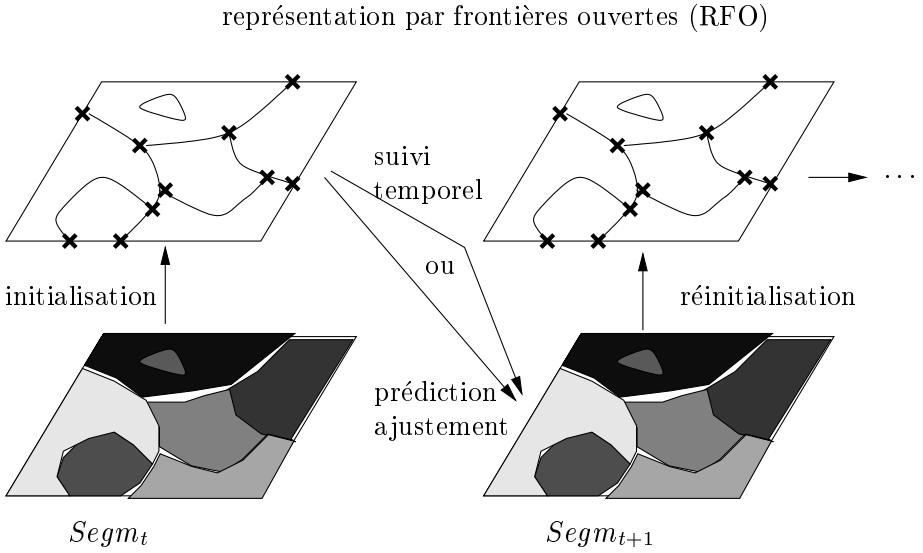


FIG. 2.6 – *Suivi temporel dans le cadre d'une utilisation conjointe* RFO–RCE.

avons besoin à deux reprises d'une représentation RCE (voir la figure 2.5) :

- l'initialisation de la RFO se fait à partir d'une RCE car la plupart des algorithmes de segmentation spatiale opèrent sur cette dernière (voir la section 2.2),
- l'estimation du mouvement des régions se fait sur une RCE (voir la section 3.4).

On peut alors se demander pourquoi choisir la RFO plutôt que la RCE. Mais en définitive,

- l'initialisation n'est faire qu'une seule fois, et par la suite la RCE initiale n'est plus utile,
- l'estimation de mouvement ne nécessite une conversion que dans le sens RFO$\longrightarrow$ RCE mais pas dans le sens inverse, car elle fournit seulement des paramètres de mouvement, sans nécessiter la conversion inverse RCE$\longrightarrow$ RFO.

C'est donc la représentation RFO que nous adopterons comme représentation principale pour le suivi temporel, la RCE ne servant que d'auxiliaire. Ainsi il sera possible d'assurer une cohérence temporelle forte au cours de l'algorithme de suivi, puisque nous conserverons les mêmes objets (points multiples et frontières) d'un bout à l'autre de la séquence. Comme le montre la figure 2.6, une utilisation conjointe des deux représentations RFO et RCE (comme cela est fait dans [Wu 95]) obligerait à effectuer une double conversion, et surtout à réinitialiser la représentation RFO à chaque image. Il faudrait alors effectuer une mise en correspondance des sommets obtenus, ce qui rendrait incertaine la cohérence temporelle.

2.1.5 Quelques invariants de la représentation

Du fait de la structure particulière du graphe que nous utilisons, cette représentation présente un certain nombre d'invariants. À l'issue de sa création (voir la section 2.3) et lors de ses évolutions temporelles (voir chapitre 3), il est utile de vérifier s'ils sont conservés pour s'assurer de la cohérence globale de la structure. [1]

Multiplicité minimale : Chaque point multiple doit avoir au moins 3 arcs incidents ou qui en sont issus :

$$\forall x \in X, mult(x) \geq 3$$

Nombre de frontières/points multiples : La somme des multiplicités des points multiples doit être égale au double du nombre de frontières ouvertes :

$$\sum_{x \in X} mult(x) = 2.n_a$$

Relations d'adjacence : Chaque point multiple pointe sur ses 2 points multiples extrémités et chaque point multiple pointe sur les frontières qui lui sont reliées. Il faut donc vérifier que ces relations sont bien réciproques.

Intersections vides : Les frontières polygonales reliant les points multiples et les boucles ne doivent pas s'intersecter. Cet invariant assure aussi que le graphe de la représentation est bien planaire.

Régions et contours fermés : Lorsque l'on parcourt un contour fermé en suivant le marquage de chaque frontière (étiquettes des régions gauche et droite) il faut s'assurer que l'on revient bien au point de départ.

2.2 Obtention de la carte de segmentation spatiale initiale

Le but d'un algorithme de segmentation spatio-temporelle est de segmenter et de suivre au cours du temps des régions ayant des propriétés visuelles stables dans le temps, comme des contours bien marqués ou bien la couleur, la texture, le mouvement, ... Cela passe par la définition d'un critère de contraste fort aux frontières ou d'un critère d'homogénéité spatiale des régions.

Les deux cas qui nous intéressent ici sont des régions spatiales séparées par des contours et des régions homogènes au sens du mouvement. Dans bien des situations, une segmentation spatiale sera plus fine qu'une segmentation spatio-temporelle au sens du mouvement. Elle aura plus de régions et plus de contours (ou frontières), et les régions homogènes au sens du mouvement seront des regroupement de régions spatiales. En effet :

- La présence d'une discontinuité du champ de mouvement apparent (contour temporel) est en général causée par une discontinuité du champ de profondeur (transition

1. Ceci est fait dans notre mise en œuvre dans un langage à objets (C++) par le moyen d'un invariant de la classe codant notre représentation.

entre 2 objets), accompagnée d'un mouvement relatif des 2 objets ou d'une translation de la caméra. La transition entre les 2 objets se traduit alors par une frontière spatiale (à moins de conditions d'éclairement particulières).

- Inversement si l'on observe un contour spatial, il ne donne pas forcément lieu à un contour temporel. Il peut exister entre 2 objets sans mouvement relatif ou au sein d'un même objet qui a 2 couleurs bien différentes.

Notre algorithme de suivi de segmentation peut en principe fonctionner à un niveau quelconque entre ces deux extrêmes. Mais en pratique, on se placera à un niveau bien choisi de finesse de la segmentation et du nombre de régions. En effet :

- Une segmentation purement spatiale contient en général trop de régions. Ces régions sont trop petites, ce qui ne permet pas une estimation fiable de leur mouvement, à cause du support d'estimation trop réduit. De plus, les frontières sont trop rapprochées, ce qui ne permettrait pas leur suivi temporel satisfaisant avec une technique de contours actifs (voir le chapitre 3). En effet les frontières auraient tendance à se coller, ce qui modifierait la topologie de la segmentation, et serait en contradiction avec notre souhait de cohérence temporelle.

- Une segmentation spatio-temporelle pure comporte seulement les régions qui sont en mouvement à un instant donné. Leur nombre peut varier énormément d'un instant à l'autre (voir la figure 1.4), ce qui n'est pas souhaitable dans notre contexte de suivi long-terme et d'interpolation temporelle. Si deux objets ne sont pas séparés par un contour temporel à un instant donné, il peut être intéressant de les garder dans notre segmentation pour le cas où ils se mettraient en mouvement.

Le niveau de détail qui convient pour notre application est donc une segmentation spatiale assez grossière, ou une segmentation spatio-temporelle légèrement sur-segmentée. Ce dernier point est aussi nécessaire pour la prise en compte de mouvements plus complexes que le modèle affine retenu, comme par exemple les mouvements articulés ou les objets déformables.

La qualité la segmentation initiale est donc cruciale pour le suivi temporel. Le choix de la méthode est donc important et c'est pour cela que nous en avons testé plusieurs. Comme pour les représentations, nous comparons des méthodes basées contours, des méthodes basées pixels/régions, et des méthodes hybrides.

2.2.1 Méthodes utilisant le gradient spatial

Comme notre représentation et l'algorithme de suivi temporel associé (voir chapitre 3) sont basés frontières, il paraît logique d'essayer d'initialiser le suivi avec une technique de segmentation spatiale similaire. Ces méthodes sont parmi les plus classiques [Cocquerez et Philipp 95] et elles sont facilement disponibles grâce au logiciel «Khoros». Elles sont basées sur l'hypothèse que 2 régions sont séparées par un contour de fort contraste. Elles consistent donc dans un premier temps à calculer le gradient $\vec{\nabla} I$ de l'image I à segmenter, par l'un des opérateurs existant : Roberts, Prewitt, Sobel, Canny–Deriche [Canny 83, Deriche 87],

Shen–Castan (algorithme DRF) [Shen et Castan 92], ... La deuxième étape consiste à détecter les maxima de ce gradient. Pour cela, deux possibilités se présentent :

FIG. 2.7 – *Segmentation spatiale utilisant le gradient. Expérience réalisée avec le logiciel «Khoros». Les colonnes montrent dans l'ordre les méthodes DRF, GEF et SDEF. La première ligne illustre un choix de seuils «raisonnables», alors que pour la deuxième ligne les seuils sont très faibles.*

Seuillage du gradient (GEF) (Khoros) Un seuillage simple est en général insuffisant : si le seuil est trop faible, trop de contours subsistent dont beaucoup sont causés par du bruit, et si le seuil est trop élevé, certains objets risquent de ne pas être détectés. Une façon de pallier ce problème est de réaliser un seuillage par hystérésis. Pour cela, on définit 2 seuils S_1 et S_2, puis les pixels p tels que $\vec{\nabla} I(p) < S_2$ sont éliminés, enfin les pixels tels que $\vec{\nabla} I(p) > S_1$ sont reconsidérés, mais seulement s'ils sont connectés à des pixels p' tels que $\vec{\nabla} I(p') > S_2$. Ainsi, seuls restent les contours dont le gradient est entièrement supérieur à S_1 et dont une partie est supérieure à S_2.

Passages par zéro du Laplacien (SDEF) (Khoros) Une autre possibilité est de calculer la dérivée seconde de I dans la direction du gradient et de considérer ses passages par zéro. En théorie cette méthode donne des contours fermés, mais en pratique il faut de même que précédemment effectuer un seuillage par hystérésis qui ne laisse que des contours ouverts.

Un inconvénient de ces méthodes est le choix délicat des seuils. La figure 2.7 montre qu'avec des seuils raisonnables, certains contours importants (bras droit, séparation fond / cheveux) ne sont pas détectés, alors que le visage est légèrement sur-segmenté. Si l'on abaisse les seuils, ces contours commencent seulement à apparaître, mais ne sont pas complets, et le bruit cause trop de faux contours.

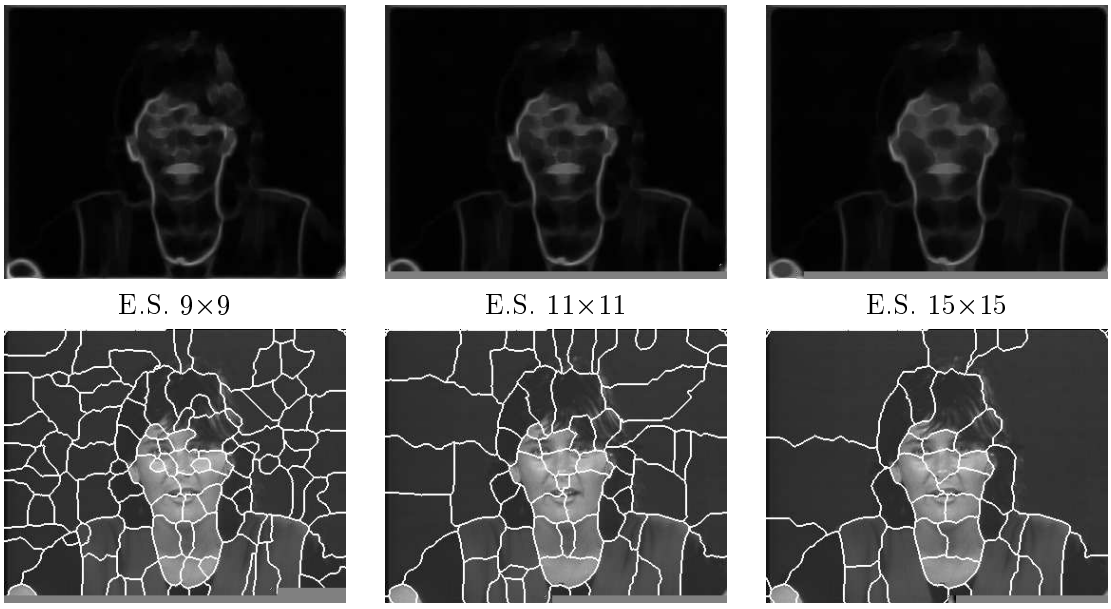


FIG. 2.8 – *Segmentation spatiale utilisant la* LPE. *La première ligne montre les images de gradient filtrées et la deuxième ligne montre l'image segmentée correspondante. Les colonnes correspondent à des tailles croissantes d'éléments structurants utilisées pour le filtrage, qui donnent comme résultats des nombres de régions décroissants.*

Un autre inconvénient de ces méthodes est qu'elles donnent des morceaux de contours qui ne sont pas naturellement fermés. Or notre application nécessite des contours fermés pour obtenir une partition de l'image en régions. Il faudrait alors utiliser des méthodes de fermeture de contours [Moulet et Barba 88] [Cocquerez et Philipp 95] qui sont complexes ou effectuent des fermetures parfois artificielles et bruitées car il faut les appliquer dans des zones de faible contraste. Nous avons donc préféré nous orienter vers des méthodes manipulant directement des régions.

2.2.2 Croissance de régions, morphologie mathématique

Ces techniques sont basées sur la croissance progressive de régions autour de germes initiaux. Les germes sont choisis régulièrement dans l'image, de sorte à avoir des régions de tailles homogènes. Ensuite les régions sont étendues de sorte à respecter le critère d'homogénéité que l'on s'est fixé (luminance, couleur, texture, ...). Un inconvénient de ces techniques est que lorsque 2 régions se rencontrent au cours de leur croissance, la frontière résultante est approximative car elle dépend de la vitesse relative de croissance qui est difficile à régler.

Une solution à cette difficulté est l'adoption d'une approche de coopération contours-régions. On bénéficie alors des avantages des deux approches [Monga 87, Monga 88]. Pour cela, on commence par extraire des contours par l'une des méthodes décrites dans la sec-

tion 2.2.1. On choisit les germes de croissance aux points les plus éloignés de ces contours [Benois et Barba 92a, Benois et Barba 92b] et on contraint la croissance des régions à ne pas aller au-delà des frontières préétablies.

Une autre méthodologie pour traiter le problème de la segmentation, utilise la morphologie mathématique, et en particulier la ligne de partage des eaux (LPE) [Schmitt et Mattioli 94] [Beucher et Meyer 93]. En fait nous allons voir qu'il s'agit d'un cas particulier intéressant d'un algorithme de croissance de régions avec prise en compte des contours. Mais il a la particularité supplémentaire de ne pas nécessiter de seuil pour la définition des contours. Une autre application de la morphologie mathématique à la segmentation est décrite dans [Salembier et Pardas 94].

Le point de départ de la méthode LPE est $I' = F(\vec{\nabla} I)$ le gradient de l'image I auquel on a appliqué un filtre F. Nous verrons qu'un filtre bien adapté est une fermeture morphologique [Haralick et Shapiro 92]. À partir de maintenant, I' est considérée comme une surface tridimensionnelle.

Ensuite on définit la notion de minima dans une image. Un minimum d'une image I' à l'altitude h est un plateau connexe de pixels de valeur h d'où il est impossible d'atteindre un point d'altitude inférieure sans avoir à monter.

Le principe de cette méthode est de considérer l'image I' comme une surface dont chacun des minima a été percé d'un trou. Cette surface est alors immergée progressivement dans un plan d'eau. L'eau passant par les minima troués (et en priorité par ceux de plus faible altitude) va progressivement remplir les différents bassins versants de I'. À chaque endroit où les eaux en provenance de minima distincts se rejoignent, une digue est élevée. À l'issue de cette procédure, chaque minimum est entouré par une digue qui délimite le bassin versant associé. L'ensemble des digues ainsi construites constitue la LPE de I'.

Un algorithme possible pour mettre en œuvre cette méthode considère des seuillages successifs de l'image I' à l'altitude h et calcule les zones d'influence géodésique d'un niveau dans le suivant. Supposons l'immersion accomplie jusqu'au niveau $h-1$. Chacun des bassins versants courants (bassins dont le minimum associé a une altitude $\leq h-1$) est doté d'une étiquette. Pour calculer les zones d'influence géodésique des bassins versants, on va réaliser des dilatations conditionnelles des bassins versants dans l'ensemble des pixels d'altitude h. On étend ainsi progressivement les bassins versants déjà obtenus à l'intérieur des plateaux de pixels à l'altitude h. Les pixels appartenant aux minima d'altitude h n'ayant pas encore été atteints sont dotés d'une nouvelle étiquette.

Cet algorithme a une forte analogie avec les algorithmes de croissance de régions, puisque chaque nouvelle région apparaît quand un nouveau minimum d'altitude h est détecté, puis croît autour de ce germe par dilatations successives. Le nombre de régions obtenues est donc égal au nombre de minima. Si l'on ne filtre pas $\vec{\nabla} I$, on obtient en général un nombre très élevé de régions. Une technique classiquement employée [Meyer et Beucher 90] est celle des marqueurs : les minima sont regroupés en ensembles appelés marqueurs qui servent de points de départ à la croissance des régions. Mais leur détermination automatique est assez problématique. Pour nos expériences, nous avons pris une approche très simple basée sur un filtrage de $\vec{\nabla} I$ par fermeture morphologique. Cette opération a la propriété de boucher les «trous» et les «vallées» étroites dans une image. Nous l'avons donc appliquée à $\vec{\nabla} I$ pour supprimer les minima de trop petite surface.

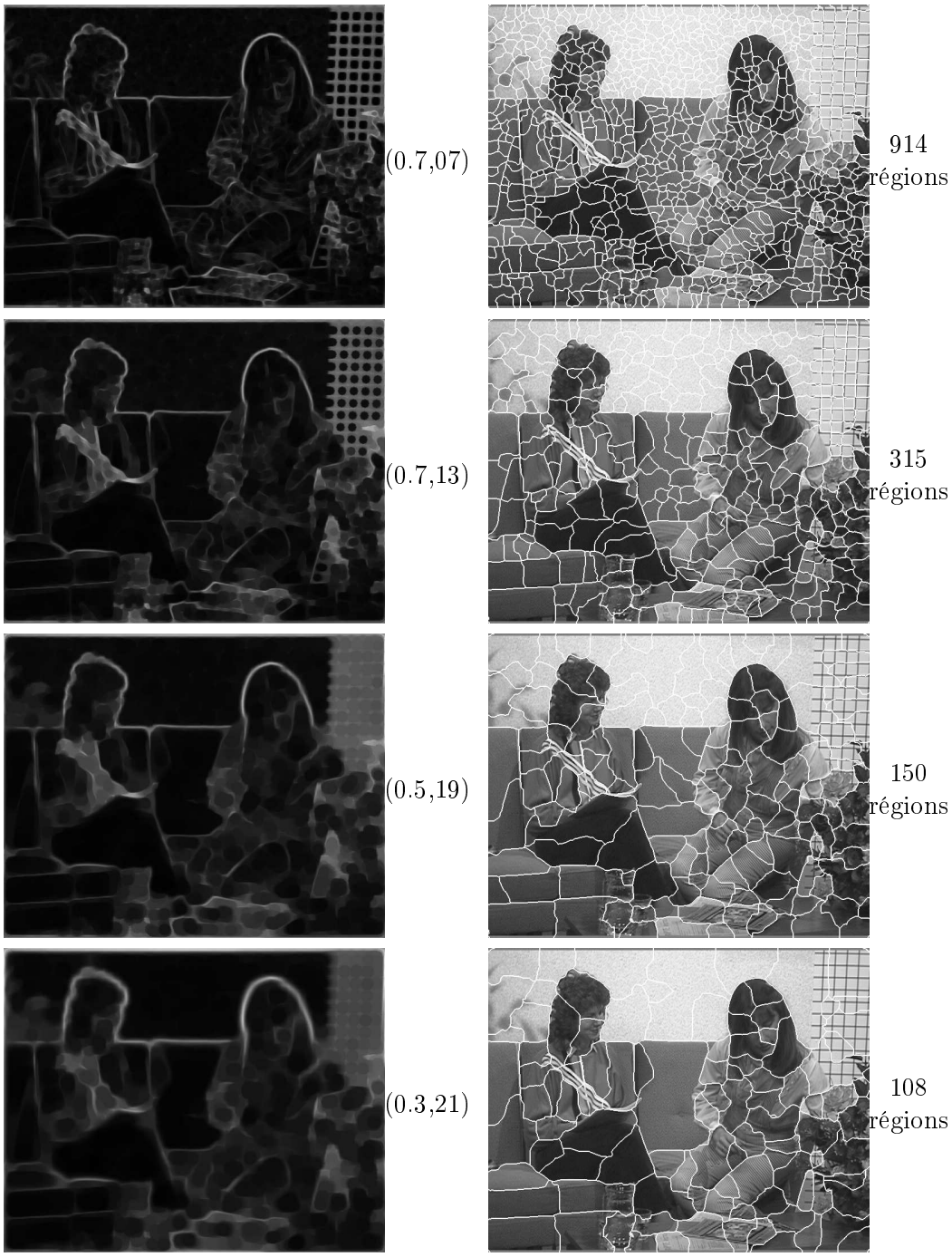


FIG. 2.9 – *Segmentation spatiale utilisant la* LPE. *La première colonne montre les images de gradient filtrées et la deuxième colonne montre l'image segmentée correspondante. Les lignes correspondent à différents réglages des paramètres. Ici nous avons fait varier le couple (paramètre du filtre de Canny-Deriche, taille de l'élément structurant).*

Nous avons réimplanté une mise en œuvre efficace décrite dans [Vincent et Soille 91]. Elle est basée sur une file d'attente, qui permet d'éviter des balayages inutiles de l'image lors des dilatations conditionnelles. Les résultats obtenus sont montrés sur les figures 2.8 et 2.9. Les images montrent à la fois le gradient filtré et le résultat de la segmentation. Nous avons aussi fait varier 2 paramètres de la méthode pour régler le niveau de détail de la segmentation obtenue. Le premier paramètre est celui du filtre de Canny–Deriche utilisé pour calculer $\vec{\nabla} I$ et le deuxième est la taille de l'élément structurant de la fermeture utilisé pour calculer I'. On voit que même en filtrant avec un paramètre «raisonnable», le nombre de régions est très important. Il faut vraiment prendre des paramètres extrêmes pour obtenir un nombre de régions acceptable, mais c'est au détriment de certains détails. Une particularité intéressante de ces résultats est que les régions ont des tailles à peu près similaires et des formes régulières. On verra que ce n'est pas le cas avec les algorithmes qui suivent.

2.2.3 Champs de Markov et critère MDL

Une autre classe d'algorithmes de segmentation spatiale est basée sur les champs de Markov [Besag 74, Besag 86, Geman et Geman 84, Azencott 87, Derin et Elliot 87]. Chaque étiquette de la segmentation est un site relié aux pixels proches appartenant à un certain voisinage (en général un 4- ou 8-voisinage). Pour chaque pixel, on est capable de calculer la probabilité que son étiquette soit la bonne c'est-à-dire qu'il appartienne à telle ou telle région. Il s'agit d'une probabilité *a posteriori* sachant l'étiquette de la région. Pour cela, il faut avoir un modèle statistique de la région. On peut par exemple utiliser un modèle de niveau de gris constant plus un bruit aléatoire, ou un modèle de texture [Kervrann et Heitz 93], etc. Le champ de Markov est l'expression de la probabilité *a priori* que deux étiquettes voisines soient égales ou différentes. Il sert à la régularisation statistique de la solution.

On se ramène alors à la minimisation d'une fonction d'énergie globale définie sur les n_{pixels} variables que sont les étiquettes. Elle se compose de deux termes :

- une énergie d'attache aux données qui exprime l'adéquation de chaque pixel à la région qui lui est attribuée,
- une énergie de régularisation qui favorise les situations où 2 pixels voisins ont la même étiquette.

Nous avons finalement choisi une variante de ces méthodes utilisant le critère du *Minimum Desciption Length* (MDL) [Leclerc 89] [Zheng et Blostein 95]. Dans ce formalisme, les énergies sont remplacées par des coûts de codage. Cela représente un avantage pour notre application au codage interpolatif, dans la mesure où nous partons d'une segmentation initiale dont le coût de codage a déjà été optimisé dans une certaine mesure.

- L'énergie d'attache aux données est remplacée par le coût de codage d'un pixel après prédiction grâce au modèle de sa région.

$C = 24 \longrightarrow 201$ régions $C = 100 \longrightarrow 163$ régions $C = 1000 \longrightarrow 42$ régions

FIG. 2.10 – *Segmentation spatiale utilisant le critère* MDL.

- L'énergie de régularisation est remplacée par le coût de codage des éléments de contour. On choisit en général un coût fixe par élément de contour présent dans la carte d'étiquettes, ce qui revient au même que la régularisation Markovienne dans le cas du 4-voisinage.

Nous avons utilisé l'implantation de [Nzomigni 95] puis [Pateux et Labit 98]. Le modèle de régions est un modèle de texture variant linéairement avec les coordonnées des pixels (cela revient à dire que le gradient spatial doit être homogène). Le coût théorique de codage d'une région C est alors de 24 bits puisqu'il faut 3 paramètres de 8 bits pour coder la texture linéaire. Le coût de codage d'un élément de contour est pris comme égal à 1,3 bit. C'est en effet le coût de codage sans perte généralement observé dans un codage de chaîne de contours avec un contexte de Markov d'ordre 3 ou plus. Par rapport à une pure segmentation au sens du MDL, cet algorithme prend en plus en compte les contours spatiaux en favorisant les éléments de contours situés sur des zones de fort gradient spatial, par une pondération de leur coût de codage.

On peut voir les résultats de cet algorithme dans la figure 2.10. Nous les montrons avec une visualisation en 5 niveaux de gris qui permet de voir tous les détails et les défauts, mieux qu'un plaquage sur l'image originale. Elle est obtenue par un coloriage du graphe d'adjacence des régions, de sorte que deux régions voisines n'ont pas la même couleur. Normalement, le graphe étant planaire, 4 couleurs suffiraient pour le coloriage, mais il existe un algorithme rapide permettant d'obtenir un coloriage en 5 couleurs, basé sur les chaînes de Kleene [Mehlhorn et Näher 89].

On voit qu'avec le coût par défaut $C = 24$, on obtient une forte sur-segmentation. Sur cette séquence, nous avons donc utilisé un coût nettement supérieur pour réduire le nombre de régions. Pour la séquence Interview (voir la figure 2.22), nous avons utilisé en plus un algorithme de fusion des régions selon un critère spatio-temporel d'homogénéité du mouvement, décrit dans [Pateux et Labit 98].

Nous avons trouvé que cet algorithme donnait des résultats globalement satisfaisants, mis à part quelques petits défauts qui peuvent être traités par filtrage *a posteriori*. Nous ne commentons pas ici ces défauts, car ils seront décrits en même temps que le filtrage dans la sous-section suivante (2.3.1).

2.3 Extraction de la structure à partir de la carte initiale

Cette section décrit toutes les étapes intermédiaires nécessaires à la création de notre structure de représentation à partir d'une carte d'étiquettes fournie par la méthode de segmentation spatiale au sens du MDL, décrite dans la section 2.2.3.

Elle se décompose en sous-sections portant sur le filtrage de la carte initiale, l'extraction des points multiples et des éléments de contour, l'approximation polygonale des frontières, la reconstitution des contours fermés, la simplification et l'ajustement des frontières.

2.3.1 Filtrage de la carte initiale

Une première étape de filtrage est nécessaire pour éliminer certains défauts de la carte initiale. Ces défauts sont principalement au nombre de trois :

Excroissances/Invaginations : Dans la carte de segmentation, il s'agit d'une région comportant un «golfe» relié au reste de la région par un chenal étroit. Cela forme un «cap» dans la région complémentaire. Pour les éliminer, il faut boucher le chenal et faire en sorte que le golfe restant soit considéré comme une petite région et fusionné avec l'autre région.

Filets : Il s'agit d'une fine bande d'un ou deux pixels de largeur. On peut les voir apparaître sur le bord des images, ou comme artefact de l'algorithme de segmentation.

Formes inadaptées au suivi temporel : Certaines régions ont des formes trop irrégulières pour être suivies correctement. C'est en particulier le cas d'une région dont la forme est telle que ses frontières sont trop rapprochées : le suivi temporel aurait pour effet de les faire se toucher, ce qui changerait la topologie de la segmentation de façon artificielle. Il s'agit de régions comportant des resserrements en leur milieu qu'il faut donc séparer en deux régions en coupant l'«isthme» qui les relie.

Le filtrage se décompose en 3 étapes : filtrage par bloc, élimination des petites régions et filtrage majoritaire, les 2 premières étant en fait des étapes préparatoires pour le traitement principal qui est le filtrage majoritaire.

2.3.1.1 Filtrage par bloc

Certaines cartes de segmentation initiale présentent des pixels isolés, correspondant à des régions de taille 1 pixel. Ces régions n'ont évidemment aucune signification et sont dues à des artefacts de l'algorithme de segmentation. Il importe donc de les supprimer. Mais il ne s'agit pas de réaffecter ce pixel aléatoirement à l'une des régions adjacentes, ce qui aurait pour effet de créer des régions aux formes non contrôlées. Pour cela nous réalisons donc un filtrage par blocs de taille 2×2 pixels. Pour chaque pixel de l'image, on regarde s'il est possible de créer un bloc de 2×2 pixels ayant la même étiquette. Si c'est possible on change l'étiquette du pixel de sorte à créer ce bloc. Ainsi, si un pixel est isolé et qu'il est entouré par des régions bien constituées, il sera affecté à une région assurant une forme régulière. Ce type de filtrage s'apparente aux filtres de voisinages utilisés en morphologie

mathématique, mais il est étendu du cas binaire au cas multi-étiquettes. Sa complexité est dans $O(n_{pixels})$.

Ce premier filtrage sert aussi pour améliorer le résultat du filtrage suivant. Par exemple, imaginons le cas d'un filet comportant une excroissance de 1 pixel. Si l'on applique directement le filtre majoritaire censé éliminer les filets, ce pixel va perturber le filtrage et à cet endroit il restera une petite région.

Le résultat de ce filtrage peut être vu sur la figure 2.11. Nous avons utilisé la visualisation en 5 niveaux de gris pour bien voir les détails qui ont changé par rapport à la carte d'étiquettes initiale.

FIG. 2.11 – *Résultat du filtrage par bloc. À gauche est montré le résultat du filtrage et à droite le résultat de l'élimination des petites régions. Ici la taille minimale d'une région est de 70 pixels.*

2.3.1.2 Élimination des petites régions

Les segmentations spatiales initiales obtenues par les algorithmes cités ont souvent des régions de petite taille. De plus le filtrage décrit dans la sous-section 2.3.1.3, peut avoir pour effet de créer artificiellement des petites régions (en séparant une région d'un petit isthme, par exemple). Or il n'est pas souhaitable de conserver ces régions pour l'étape de suivi temporel, car on ne peut pas réaliser d'estimation de mouvement fiable sur leur support trop réduit. Il importe donc de les supprimer.

Pour cela, l'idéal serait de les fusionner au sens du mouvement, comme dans la section 2.2.3. Mais justement, nous avons vu qu'un mouvement estimé sur une si petite région peut difficilement être considéré comme fiable. Pour qu'il soit exploitable, il faudrait un mécanisme complexe de relaxations, comme cela est fait dans le processus de fusions de régions de [Pateux et Labit 98]. Or ceci est trop complexe pour une simple étape intermédiaire de filtrage. Nous avons donc décidé d'utiliser le même critère spatial simple que dans la section 2.2.3, à savoir la moyenne spatiale des niveaux de gris.

La complexité de cette opération est dans $O(n_{pixels})$ car il faut réaliser un étiquetage en composantes connexes de la carte d'étiquettes, ce qui impose un balayage de toute l'image,

et le traitement individuel des petites régions est absorbé dans ce calcul.

2.3.1.3 Filtrage majoritaire

L'idée de ce filtre est d'obtenir, sur une carte d'étiquettes, un résultat similaire au filtre médian sur une image en niveaux de gris. Mais on ne peut pas appliquer de façon rigoureuse un filtre médian sur une carte d'étiquettes car les étiquettes ne sont pas ordonnées (même si leur représentation informatique peut l'être). La seule opération disponible sur les étiquettes est le test d'égalité, qui suffit pour réaliser un filtre majoritaire.

Le filtre majoritaire est défini en morphologie mathématique pour une image binaire. Pour notre application, il faut utiliser son extension au cas d'une carte d'étiquettes. Nous parcourons donc tous les pixels p de l'image et nous considérons une fenêtre de taille $k \times k$ pixels autour de p. Nous comptons les occurrences de chaque étiquette de région dans la fenêtre et nous remplaçons l'étiquette de p par l'étiquette majoritaire dans la fenêtre.

Cette opération peut être assez complexe si l'on augmente la taille de la fenêtre. En effet, pour chaque pixel, il faut créer et maintenir un ensemble de k^2 étiquettes, que l'on garde trié pour des raisons d'efficacité, ce qui nécessite dans le pire les cas $k^2 \log k^2$ opérations. Au total, cela représente une complexité dans $O(n_{pixels}.k^2 \log k)$.

FIG. 2.12 – *Résultat du filtrage majoritaire. À gauche est montré le résultat du filtrage et à droite le résultat de l'élimination des petites régions. La taille du filtre majoritaire est 9×9 pixels et la taille minimale d'une région est de 70 pixels.*

Le résultat de ce filtrage peut être vu sur la figure 2.12. On peut voir que la plupart des défauts de la carte initiale ont été supprimés. La région fine séparant le bras doit de la manche a disparu. De même pour les régions autour du col. Le mince filet prolongeant la région des cheveux jusque dans l'épaule à droite a été supprimé. De même pour le filet en haut de l'image. Les frontières ont été lissées et se prêteront ainsi mieux à une approximation polygonale.

2.3.2 Extraction des points triples et quadruples

Pour les paragraphes qui suivent, nous nous plaçons dans le cadre de l'espace inter-pixels (voir la figure 2.13). Dans cet espace se trouvent des points et des éléments de contour. De plus nous avons choisi de nous placer dans le cas particulier d'une carte de segmentation en 4-connexité.

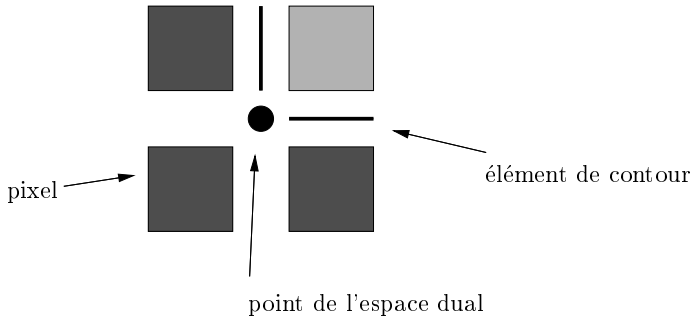


FIG. 2.13 – *Points et éléments de contour dans l'espace inter-pixels.*

Dans ce cadre, on peut définir les points multiples de la carte d'étiquettes comme étant les points de l'espace inter-pixels reliés à plus de 2 éléments de contour. Ils ne peuvent être que de deux types : les points triples et les points quadruples (voir la figure 2.14). Les points triples sont au contact de 3 régions et les points quadruples sont au contact soit de 4 régions, soit de 3 dans les situations telles que celle de la figure 2.3.

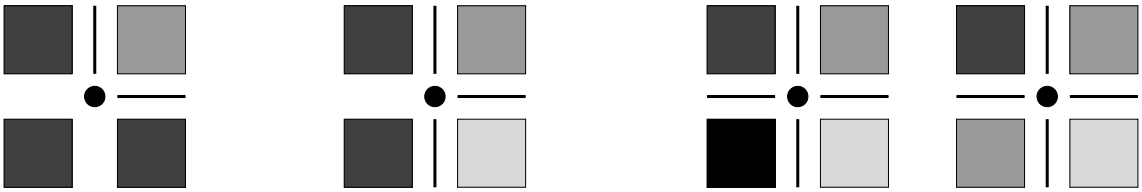


FIG. 2.14 – *Points triples et quadruples.*

Leur détection est alors très simple, puisqu'il suffit de parcourir la carte d'étiquettes pour trouver les éléments de contour, puis de rechercher les points reliés à 3 ou 4 éléments de contour. La complexité de cette étape est dans $O(n_{pixels} + n_{ptm})$ ce qui donne en fait $O(n_{pixels})$.

2.3.3 Extraction des chaînes de contours

Pour cette phase, on part d'un point multiple et l'on choisit une direction de départ empruntant un élément de contour parmi ceux qui n'ont pas déjà été parcourus. Cet élément de contour sépare 2 régions R_1 et R_2, l'une étant à gauche et l'autre à droite. On va suivre la frontière séparant R_1 et R_2 jusqu'à ce que l'on retombe sur un point multiple.

L'algorithme de cheminement utilise un automate à 4 états correspondant aux 4 directions cardinales : N, S, E et O. Supposons que l'on est sur un point de l'espace dual p, que l'on emprunte la direction D. L'automate est dans l'état D et l'on arrive sur un nouveau point p'. Il faut alors repartir de p' selon l'une des 4 directions N, S, E, O. Or en p' aboutissent 2 éléments de contour puisqu'il ne s'agit pas d'un point multiple : celui de direction $-D$ opposée à D et un autre de direction D'. On se sert de l'état de l'automate pour choisir la direction D' au lieu de $-D$.

On obtient ainsi une chaîne d'éléments de contour, orientée, ressemblant à une chaîne de Freeman [Freeman 61] mais dans l'espace dual. On marque alors les 2 points multiples et l'on itère jusqu'à ce que tous les points multiples soient marqués un nombre de fois égal à leur multiplicité. On obtient ainsi toutes les frontières ouvertes.

Il reste à trouver les frontières fermées. Pour cela, on cherche un élément de contour non encore parcouru, et on applique le même algorithme. Seul le test d'arrêt change : au lieu de tester si l'on arrive sur un point multiple, il suffit de tester si l'on retombe sur le point de départ.

La complexité de cette étape est dans $O(n_{pixels} + n_{fr})$ ce qui donne en fait $O(n_{pixels})$.

2.3.4 Approximation polygonale

Dans cette étape, chaque chaîne de contours subit une approximation polygonale. Pour cela nous transformons la chaîne de directions N, S, E, O en polygone tel que les sommets sont à coordonnées entières et tel que la différence des coordonnées de 2 sommets consécutifs est -1, 0 ou 1. Nous pouvons alors utiliser un algorithme d'approximation polygonale d'un polygone. Ces algorithmes sont très nombreux (voir par exemple [Wall et Danielsson 84]).

Pour des raisons de complexité, nous nous restreignons aux algorithmes qui sélectionnent un sous ensemble minimal de points parmi les n points du polygone initial, tout en satisfaisant un certain critère d'écart. Les 2 critères généralement utilisés sont :

Critère de distance : On définit une bande ou tube, d'une largeur que l'on se fixe, autour du polygone initial. Le polygone approchant doit être inclus dans cette bande.

Critère d'aire : L'aire de la surface contenue entre les 2 polygones doit être inférieure à un seuil que l'on se fixe.

Ces algorithmes sont basés sur la construction d'un graphe à n sommets et $O(n^2)$ arêtes, dans lequel on cherche un chemin optimal. La construction du graphe, par une méthode directe, a une complexité dans $O(n^3)$ et la recherche du chemin optimal est dans $O(n^2)$, ce qui donne au total un algorithme dans $O(n^3)$. Dans [Melkman et O'Rourke 88] les auteurs montrent un algorithme pour la construction du graphe en $O(n^2 \log n)$ et ce résultat a été amélioré par [Chan et Chin 92] qui présentent un algorithme en $O(n^2)$. Au total la complexité est donc dans $O(n^2)$.

Pour notre part, nous avons préféré un algorithme moins complexe, mais sous-optimal. Nous avons opté pour l'algorithme utilisé dans [Pateux et Labit 97] qui fonctionne par divisions successives du polygone initial jusqu'à ce que le critère d'approximation soit satisfait. À chaque étape une relaxation est effectuée sur la position des points du polygone approchant. Par ailleurs, nous avons choisi le critère d'aire car dans le cadre de l'application

à l'interpolation compensée en mouvement pour le codage, l'erreur sur l'image interpolée est liée au nombre de pixels qui changent de région à cause de l'approximation polygonale.

FIG. 2.15 – *Résultat de l'approximation polygonale. Nous autorisons une erreur d'approximation d'une surface de 0,3 pixels carré par élément de contour de la chaîne initiale.*

Le résultat de toutes ces étapes est montré dans la figure 2.15. Notre structure de représentation est plaquée sur l'image originale. Les frontières sont en blanc et les points multiples y apparaissent comme des croix noires (3×3 pixels). Les sommets des polygones apparaissent comme des points noirs (1×1 pixel).

2.3.5 Contours des régions et arbre d'homotopie

À ce stade, nous avons dans notre représentation des points multiples et des frontières, reliés par un graphe, mais sans structure topologique explicite. Dans cette phase, nous n'extrayons pas d'informations supplémentaires de la carte d'étiquettes initiale, mais nous enrichissons notre représentation d'une topologie explicite.

La première chose à faire est de construire les contours des régions. Nous rappelons qu'il s'agit simplement de la liste ordonnée des frontières rencontrées en faisant le tour d'une région. Pour cela nous utilisons l'algorithme «de la main gauche» décrit dans la section 2.1.4 à propos de la conversion entre représentations (RFO—→ RCF). Mais il s'agit d'une version simplifiée puisque nous disposons déjà des étiquettes des régions droite et

gauche pour chaque frontière. Le passage d'une frontière à la suivante consiste juste à choisir l'unique frontière délimitant la même région.

Ce parcours nous permet en plus d'identifier les composantes connexes du graphe qui sous-tend notre représentation. Les contours fermés de notre représentation (les boucles) sont aussi considérées comme composantes connexes du graphe. Nous pouvons donc maintenant déterminer l'arbre d'homotopie de notre représentation, qui est l'arbre d'«inclusion» des régions les unes dans les autres. Pour cela, on considère chaque composante connexe et on choisit un point lui appartenant. Ensuite il suffit de tester si ce point appartient au contour fermé polygonal d'une autre région.

2.3.6 Simplification et ajustement des frontières

Dans cette étape, nous procédons à une simplification de notre structure de représentation. Les points multiples trop proches sont fusionnés (pour les détails de l'opération, voir la section 2.4.1). Ici, on peut constater qu'une fusion a eu lieu au niveau de l'intersection entre les cheveux et l'épaule à gauche. De même à droite de l'œil à droite et au milieu de la manche à droite. L'élimination des petites régions, qui est doit être faite à chaque modification de notre représentation, est donc aussi appliquée ici. Seule une région du bras à droite, dans le coin en bas à droite de l'image est supprimée.

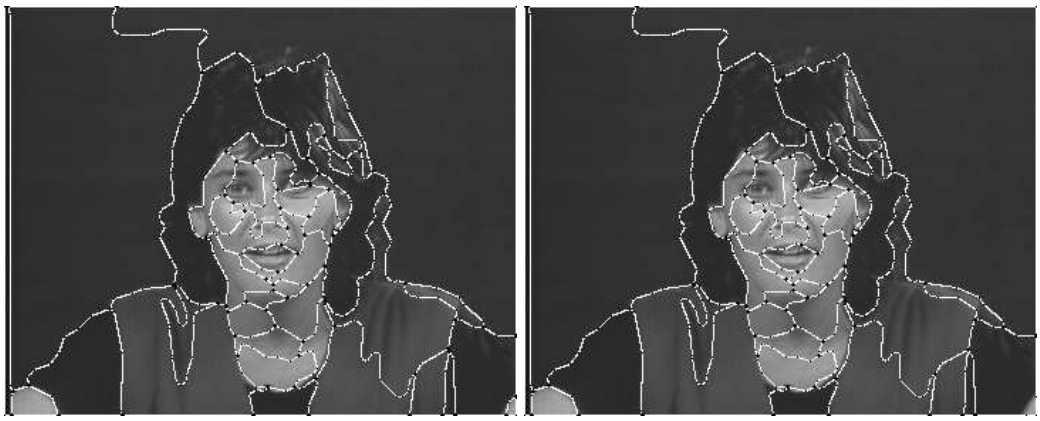


FIG. 2.16 – *Fusion des points multiples. À gauche est montrée la segmentation avant fusion et à droite le résultat de la fusion, après élimination des petites régions.*

Ensuite nous appliquons un algorithme de recalage des frontières sur le gradient spatial de l'image. Cet algorithme est le même que celui qui nous servira à l'ajustement temporel des frontières et sera décrit dans la section 3.6.1. Pour l'instant disons juste qu'il se compose de deux étapes : un recalage de chaque frontière par un mouvement affine, puis une relaxation locale de chaque sommet de l'approximation polygonale. Les résultats de ces deux phases sont montrés dans la figure 2.17. On note une petite amélioration au niveau de l'oreille à gauche et sur la partie à gauche du col. C'est cependant la deuxième phase qui apporte les améliorations les plus nettes. Toute l'encolure est maintenant parfaitement en place. La petite imperfection sur la lèvre supérieure a été corrigée.

FIG. 2.17 – *Résultat de l'ajustement spatial. À gauche est montré le résultat de l'ajustement affine des frontières et à droite le résultat de l'ajustement local des sommets.*

On constate par ailleurs un «flottement» des frontières qui n'ont pas lieu d'être, car elle ne sont pas situées le long d'une ligne de fort gradient. C'est ce qui se produit avec les frontières du cou et certaines frontières du visage. Certaines se rapprochent alors d'un vrai contour et cela donne lieu à la disparition d'une région inutile. C'est par exemple le cas de la région entre la manche à droite et le buste. On remarque enfin que les contours qui avaient été adoucis par le filtrage majoritaire retrouvent parfois des angles prononcés.

2.4 Opérations de base sur la structure

Cette section décrit quelques opérations simples que l'on peut réaliser sur notre représentation. Ces opérations sont la fusion de deux points multiples et la fusion de deux régions.

Les opérations plus complexes comme le suivi et l'interpolation temporelle seront vues ultérieurement dans le chapitre 3 et dans la section 4.4.2 portant sur la prédiction bidirectionnelle de segmentation.

2.4.1 Fusion des points multiples

Cette première opération consiste à prendre 2 points multiples proches et reliés par une frontière. On les regroupe alors en un seul point multiple, en mettant à jour le graphe de la représentation en conséquence. Cette opération est intéressante car elle simplifie le graphe, ce qui peut réduire sont coût de codage (voir la section 4.3), et supprime les frontières trop courtes pour être suivies de façon fiable. C'est aussi l'un des avantages de notre représentation sur la représentation par carte d'étiquettes, sur laquelle une telle opération n'est pas réalisable facilement.

Dans le cas le plus général, nous partons de 2 points multiples p_1 et p_2, de multiplicités respectives m_1 et m_2. Si p_1 et p_2 ont un seul arc en commun, le point multiple résultant p

a pour multiplicité $m = m_1 + m_2 - 2$. Par exemple, la fusion de 2 points triples donne un point quadruple (voir la figure 2.18).

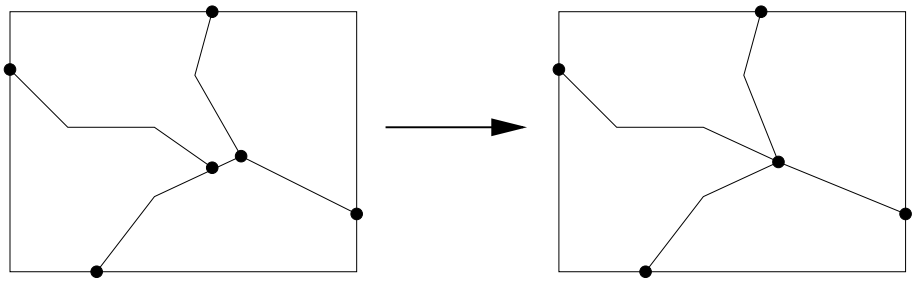


FIG. 2.18 – *Fusion des points multiples : cas général.*

Dans le premier cas particulier illustré par la figure 2.19(a), les 2 points multiples ont plusieurs arcs adjacents en commun, il est nécessaire de supprimer ces arcs, mais aussi la ou les régions dont les contours sont formés par ces arcs.

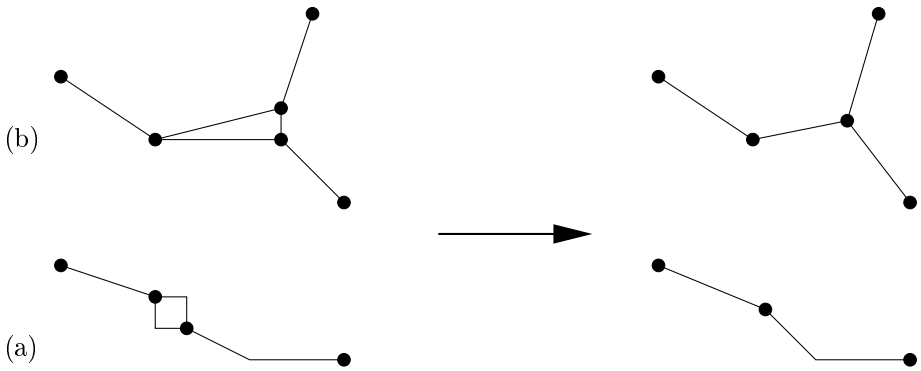


FIG. 2.19 – *Fusion des points multiples : 2 cas particuliers.*

Dans le deuxième cas particulier illustré par la figure 2.19(b), il faut tester si une région n'est pas «aplatie» par la fusion des 2 points multiples. Pour cela, on considère l'ensemble E_i des sommets adjacents à p_i. On regarde si les ensembles E_1 et E_2 ont une intersection contenant un point p différent de p_1 et de p_2. On appelle alors F_i la frontière reliant p et p_i et F celle qui relie p_1 et p_2. Si de plus, F_1 et F_2 sont toutes deux de simples segments, alors il faut supprimer la région dont le contour est formé par F, F_1 et F_2.

Ces deux cas particuliers apparaissent lorsqu'il y a des régions de petite surface. À première vue ces situations ne devraient pas se produire, car les petites régions ont été supprimées dans une étape précédente. Cependant, elles sont possibles car il est difficile d'ajuster les deux seuils parfaitement : le seuil de fusion des points multiples et le seuil d'élimination d'une petite région. De plus elles peuvent résulter de l'évolution temporelle de la structure. Le traitement de ces cas particuliers est donc nécessaire.

2.4.2 Fusion des régions

Cette opération est utilisée pour supprimer une région qui devient trop petite au cours du suivi temporel. Elle est alors fusionnée avec l'une des régions adjacentes. Cette opération est extrêmement simple avec une carte d'étiquettes car il suffit de remplacer une étiquette par une autre. Mais avec notre représentation, la mise en œuvre est plus compliquée.

Dans le cas général, les 2 régions sont séparées par une ou plusieurs frontières qu'il faut d'abord supprimer. Pour chaque frontière de séparation F, on considère les points multiples p_1 et p_2 qui sont à ses extrémités. Leur multiplicité doit être décrémentée de 1. Si $m_i \geq 4$, p_i subsiste. Par contre, si $m_i = 3$, il faut aussi supprimer p_i. Dans ce cas, les 2 frontières qui restent attachées à p_i doivent aussi être mise bout à bout. Il faut alors faire attention à leur orientation et éventuellement changer cette orientation avant de les fusionner.

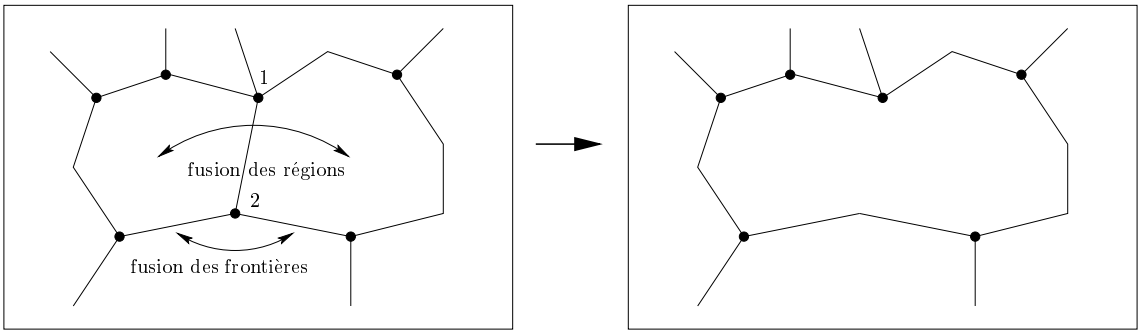


FIG. 2.20 – *Fusion des régions : cas général.*

Dans le premier cas particulier, illustré par la figure 2.21(a), ce sont 3 frontières qu'il faut mettre bout à bout. En fait ce cas particulier peut être traité avec l'algorithme général, si celui-ci est conçu avec soin.

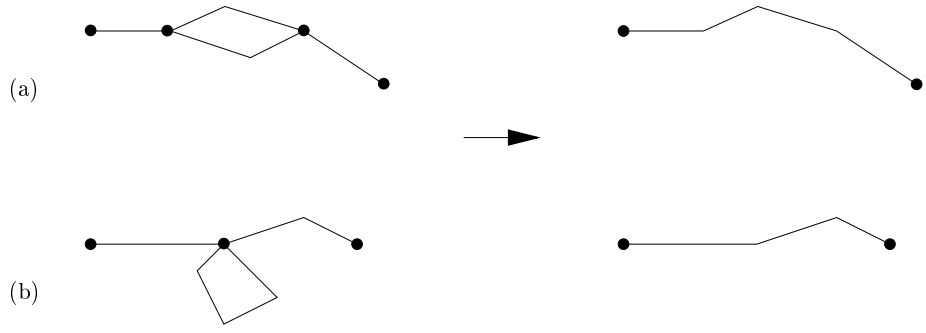


FIG. 2.21 – *Fusion des régions : 2 cas particuliers.*

Dans le deuxième cas particulier, illustré par la figure 2.20(b), on a $p_1 = p_2$ qui est un point multiple de multiplicité $m \geq 4$. Il faut donc décrémenter m de 2 unités. De plus,

si $m = 4$, les 2 frontières restantes doivent fusionner. Par contre, si $m \geq 5$, le point multiple résultant subsiste.

2.5 Résultats

Cette section montre les mêmes résultats que pour la séquence *«Miss America»*, mais pour la séquence *«Interview»*.

FIG. 2.22 – *Segmentation spatiale utilisant le* MDL *et la fusion de régions.*

Pour cette séquence nous avons utilisé la fusion de régions au sens du mouvement pour éviter d'avoir un trop grand nombre de régions. La figure 2.22 montre la segmentation dont nous partons. On peut y voir les mêmes défauts que pour l'autre séquence, à savoir des frontières très irrégulières et de minces filets.

La figure 2.23 montre le résultat du filtrage par bloc. Il corrige les petits défauts comme la frontière crénelée qui se trouve au milieu de la jambe gauche (causée par les rayures ou les carreaux du pantalon).

Le résultat du filtrage majoritaire est montré dans la figure 2.24. Les filets horizontaux au niveau de l'épaule droite et du bras gauche ont été supprimés. Les filets en haut et en bas ont étés réduits mais sont toujours là. Nous avons donc fait une deuxième itération de ce filtrage.

La figure 2.25 montre la deuxième itération. Les filets des bords haut et bas de l'image ont cette fois ci disparu. Le reste de la segmentation est à peu près inchangée.

Le résultat de l'approximation polygonale est montré dans la figure 2.26. Pour cette

FIG. 2.23 – *Résultat du filtrage par bloc. Ici la taille minimale d'une région est de 110 pixels.*

image, la fusion des points multiples ne donne rien puisque, contrairement à l'autre séquence, les régions sont déjà assez grosses et les points multiples bien séparés.

La figure 2.27 montre le résultat de l'ajustement affine des frontières. L'effet le plus notable est observé sur les deux frontières qui délimitent les cheveux et le fond, qui se repositionnent plus près du vrai contour.

Le résultat de l'ajustement local des sommets est montré dans la figure 2.28. Les quelques petites imperfections des contours de la tête sont corrigées. De même pour les frontières entre le canapé et le fond.

La figure 2.29 montre le résultat final obtenu sur la même séquence *«Interview»* avec un autre jeu de paramètres de l'algorithme de segmentation basé sur le MDL. Les figures 2.30 et 2.31 montrent les résultats finaux obtenus sur les séquences *«Tennis»* et *«Flower Garden»*.

2.6 Conclusion partielle

Dans ce chapitre, nous avons défini une structure de représentation pour une partition quelconque d'une image. Partant de la notion de graphe d'arrangement utilisée en géométrie algorithmique, nous avons abouti à une représentation par des frontières ouvertes reliant des points multiples. Les régions et leurs contours n'en sont alors que des éléments dérivés. La représentation est aussi enrichie d'informations supplémentaires comme l'arbre d'homotopie de la partition.

FIG. 2.24 – *Résultat du filtrage majoritaire. À gauche est montré le résultat du filtrage et à droite le résultat de l'élimination des petites régions. La taille du filtre majoritaire est 13×13 pixels.*

Nous avons ensuite discuté le problème de l'initialisation de cette représentation dans la première image de la séquence. Nous avons conclu que le niveau de détail qui convenait pour un suivi temporel ultérieur était une segmentation spatiale assez grossière, ou une segmentation spatio-temporelle légèrement sur-segmentée. Sur l'exemple le plus difficile de notre base de tests (séquence *«Miss America»*), nous avons constaté que les méthodes de segmentation spatiale basées contours ne convenaient pas. Après avoir testé plusieurs algorithmes utilisant la coopération contours/régions, nous avons arrêté notre choix sur une variante particulière utilisant le critère du MDL. Mais certains de ses défauts nous ont obligés à effectuer un post-traitement à base de filtrage majoritaire.

Enfin, nous avons détaillé les algorithmes nécessaires pour effectuer quelques opérations de base sur cette représentation. La fusion des points multiples est une opération intéressante de simplification d'une segmentation puisqu'elle n'est pas faisable facilement sur une représentation classique par carte d'étiquettes. Par contre, la fusion de deux régions, triviale avec des étiquettes, nécessite un algorithme plus compliqué.

Bien que cette représentation véhicule la même information que les autres représentations et que des conversions sont possibles entre elles, la différence importante réside dans les algorithmes de suivi temporel qui lui sont applicables. Nous allons donc maintenant présenter l'algorithme que nous utiliserons.

FIG. 2.25 – *Résultat de la deuxième itération du filtrage majoritaire (mêmes paramètres).*

FIG. 2.26 – *Résultat de l'approximation polygonale. Nous autorisons une erreur d'approximation d'une surface de 0,3 pixels carré par élément de contour de la chaîne initiale.*

FIG. 2.27 – *Résultat de l'ajustement affine des frontières.*

FIG. 2.28 – *Résultat de l'ajustement local des sommets.*

FIG. 2.29 – *Segmentation finale obtenue avec un autre jeu de paramètres.*

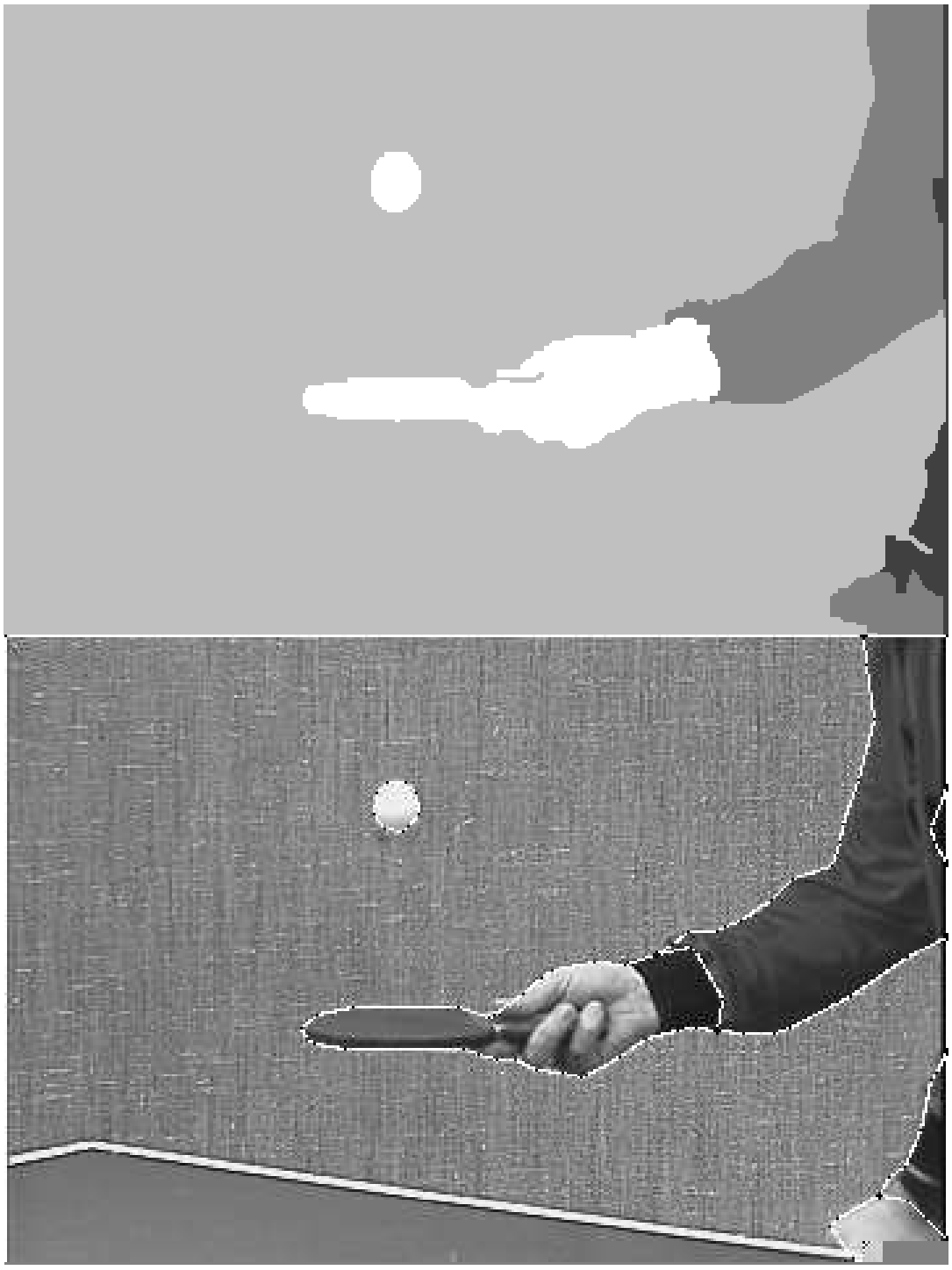


FIG. 2.30 – *Segmentations* MDL *et finale.*

FIG. 2.31 – *Segmentations* MDL *et finale, retouchée manuellement.*

Chapitre 3

Suivi temporel d'objets multiples

Introduction

Dans la section 1.3.1, nous avons présenté trois arguments qui militent en faveur d'un **suivi temporel d'une segmentation** dans une séquence. Il s'agit du gain en temps de calcul, de la stabilité temporelle accrue et de l'enrichissement de la description de la séquence que le suivi permet.

À la première raison, nous verrons dans la section 4.3, que nous pourrons rajouter un gain en coût de codage de la segmentation. Ce gain est évidemment appréciable pour une application de codage vidéo. Quant aux deux autres raisons, elles sont essentielles pour le codage interpolatif de séquences dont nous parlerons dans le chapitre 4.

Notre représentation de la segmentation au sens du mouvement comporte différents éléments sur lesquels on peut *a priori* essayer d'effectuer un suivi temporel : les points multiples, les frontières, les régions et leurs contours. Les approches opérant sur la représentation RCE [Odobez 94] [Garcia-Garduño 96] [Nzomigni 95] suivent les régions et les approches opérant sur la représentation RCF [Wu 95] suivent leurs contours. Dans le cas de notre représentation RFO, il est naturel de travailler soit sur les points multiples, soit sur les frontières ouvertes.

Or nous verrons dans la suite de ce chapitre (section 3.6.2, figure 3.16) que les points multiples ne sont pas stables au cours du temps. Un mouvement faible des objets peut engendrer un mouvement de grande amplitude des points multiples. L'approche choisie est donc le **suivi des frontières**, en nous aidant cependant du mouvement de la texture à l'intérieur des régions, tel que cela est fait dans [Bascle et al. 94].

Notons enfin que le fonctionnement de notre algorithme de suivi se fait image par image. Il est causal et «récursif» : les images sont traitées dans l'ordre croissant des indices temporels et chaque image est traitée sans délai (autre que le temps de calcul), dès qu'elle connue du système de codage. C'est une différence importante par rapport aux algorithmes d'interpolation que nous développons dans le chapitre 4, qui fonctionnent de façon non causale sur un groupe d'images (GOP). Certains algorithmes de suivi travaillent sur un GOP [Hall et al. 97], grâce à un modèle de surface active. Ils offrent potentiellement plus de robustesse grâce à la prise en compte de plusieurs images simultanément. Cependant,

ils semblent difficilement adaptables au suivi multi-objets, à cause de la complexité du problème de l'interaction entre plusieurs surfaces.

Plan du chapitre

La première section (3.1) expose les hypothèses que nous faisons sur la séquence d'image traitée pour que notre algorithme de suivi temporel fonctionne correctement.

Dans une deuxième section (3.2), nous présentons une vue d'ensemble du suivi temporel, comme l'enchaînement de modules décrits par leurs entrées et leurs sorties.

Les sections suivantes (3.3 à 3.5) détaillent ensuite le fonctionnement des modules utilisés.

Enfin, la section 3.7 montrent les résultats obtenus sur nos séquences de test.

3.1 Hypothèses nécessaires au suivi

Un algorithme de suivi temporel fait nécessairement un certain nombre d'hypothèses sur la séquence traitée. Nous présentons ici les 3 hypothèses qui sont nécessaires à notre algorithme. Pour chaque hypothèse, nous indiquons son utilité et discutons de sa validité.

Hypothèse H1

La première hypothèse est la plus évidente puisqu'elle suppose que, globalement, les mêmes régions se retrouvent d'une image à l'autre. Elles peuvent éventuellement se retrouver à des positions différentes à cause de leur mouvement, mais cela exclut les changements de plan (au sens cinématographique du terme).

Cette hypothèse est tout à fait valide dans notre cas, puisque nous travaillons sur des séquences déjà découpées en plans. Dans le cas contraire, des algorithmes fiables existent pour ce premier traitement de découpage. Elle peut par exemple être réalisée par un algorithme de détection de changement dans les images, après une compensation du mouvement dominant, obtenu grâce à un estimateur robuste [Cherfaoui 95] [Bouthemy et Ganansia 96] [Bouthemy et al. 97].

Cette hypothèse nous autorise à réaliser une prédiction de la segmentation d'une image vers la suivante. Notre algorithme rentre donc dans le cadre des algorithmes de prédiction/ajustement, et ce schéma sera appliqué aux frontières des régions.

De plus, nous supposons que les objets sont en nombre décroissant. Ainsi, les objets sortant de la scène ou complètement occultés seront traités, mais ni les objets entrant dans la scène, ni les objets entièrement cachés puis se découvrant. Cette hypothèse est assez restrictive, mais c'est en partie grâce à elle, que nous obtiendrons un suivi sable.

Hypothèse H2

La deuxième hypothèse, déjà évoquée dans la section 2.2, suppose que les contours spatio-temporels dans une image (ou contours en mouvement) sont un sous-ensemble des contours spatiaux de l'image.

Cette hypothèse nous autorise à réaliser l'ajustement des frontières de notre représentation sur les gradients spatiaux de l'image. Comme notre algorithme travaille sur la frontière

dans son ensemble, cette hypothèse peut en pratique être assouplie. Il suffit en fait qu'une partie seulement de la frontière soit à proximité des gradients spatiaux pour espérer un ajustement correct.

Cette hypothèse est en général fausse. Des contre-exemples classiques sont des images formées de régions de texture aléatoire, en mouvement et sans séparation nette. L'œil est capable de distinguer les différents mouvements et de délimiter les régions, malgré l'absence de gradient spatial les séparant. Sur ces images, notre algorithme ne fonctionnera pas, mais il faut noter que ces exemples sont des images synthétiques, que l'on retrouve rarement en pratique. Les problèmes les plus gênants peuvent être posés par les conditions photométriques particulières d'acquisition des images. L'éclairement de la scène peut être tel que les limbes des objets sont atténués et se confondent avec le fond. Mais dans ces situations, un ajustement utilisant des informations de mouvement aura autant de difficultés qu'un ajustement utilisant des informations spatiales.

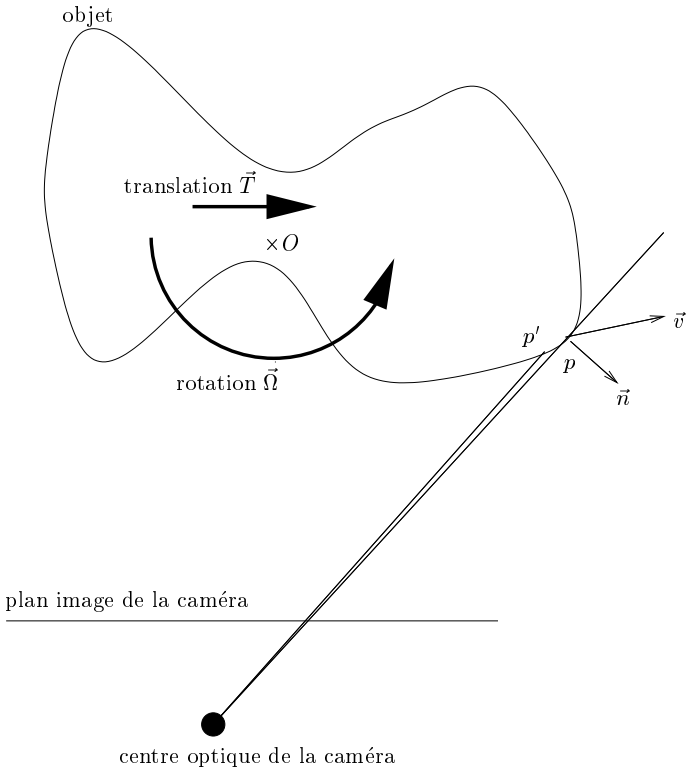


FIG. 3.1 – *Mouvements de la texture et des limbes.*

Hypothèse H3

La troisième hypothèse suppose que le mouvement d'une frontière entre deux régions est égal au mouvement de la texture de l'une de ces deux régions, en général la région d'avant plan.

Pour une frontière positionnée sur un contour interne d'un objet, cette hypothèse est toujours vraie. En effet, la frontière est définie par la texture, donc elle bouge avec elle. Dans le cas d'un limbe d'un objet, sa validité est moins évidente. La figure 3.1 illustre le cas d'un objet solide. Le point p est un point appartenant au limbe. Sa position est définie par l'intersection entre la surface de l'objet et la tangente passant par le centre de la caméra. Son mouvement dépend donc du mouvement de l'objet, mais aussi de la forme de la surface. Le point p' est le même point (distingué sur la figure pour des raisons de clarté) mais on considère qu'il fait partie de la texture de l'objet. Son mouvement est donc déterminé par le torseur cinématique de l'objet $(\vec{T}, \vec{\Omega})$. Ces vecteurs vitesse sont différents, mais leurs projections sur la normale $\vec{n}$, seule observée dans l'image, est égale. L'hypothèse est aussi valide pour une très large classe d'objets déformables.

Cette hypothèse est valide pour le mouvement de la texture située dans le voisinage d'une frontière. Lorsque le mouvement d'un objet est approché par un modèle affine, l'hypothèse risque d'être mise en défaut, si l'objet a un mouvement trop peu uniforme. Mais pour de petites régions ou pour des régions de mouvement homogène, cette hypothèse nous autorise à effectuer la prédiction d'une frontière grâce au mouvement de l'une des deux régions qu'elle délimite. Le choix entre ces deux régions est effectué selon un critère qui sera exposé dans la section 3.5.

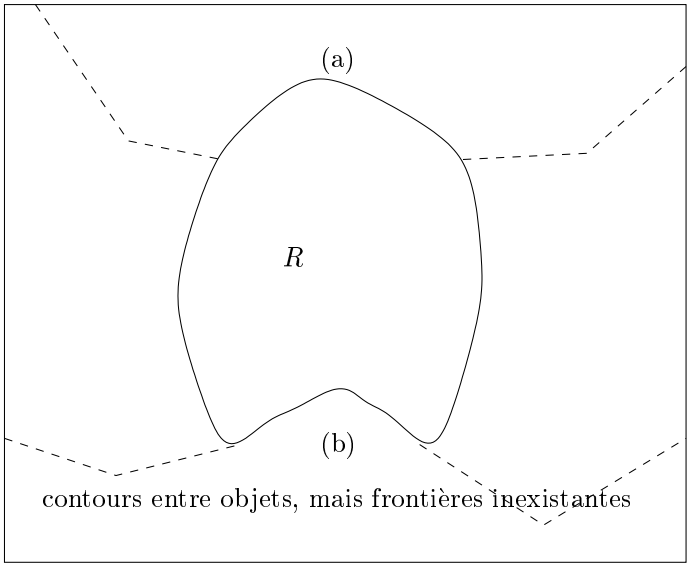


FIG. 3.2 – *Frontière pouvant appartenir à deux régions.*

Si cette hypothèse est vraie sur de petits segments d'une frontière, elle ne l'est pas forcément globalement sur la frontière entière. Si l'image est sous-segmentée (voir la figure 3.2), on peut imaginer qu'une partie (a) de la frontière appartienne à la région R et une partie (b) à une autre région. Notre algorithme marchera donc correctement si la segmentation initiale est située au bon niveau de détail, voire si elle est légèrement sur-segmentée (à propos du choix d'un niveau de détail approprié, voir aussi la section 2.2).

3.2 Vue d'ensemble du suivi

Comme nous l'avons indiqué, notre algorithme fonctionne selon le paradigme général de prédiction/ajustement. Une prédiction de l'objet suivi est générée à partir de l'observation de son passé. Puis les paramètres de cet objet sont ajustés grâce à l'observation de l'image courante. Dans notre cas, ce principe est appliqué aux frontières entre objets, mais aussi à la texture des objets. En effet, le mouvement de la texture des objets aide fortement au suivi des frontières.

Nous commençons par un bref rappel sur les deux sens de description du mouvement possibles et sur les notations que nous utilisons. Puis nous décrivons les modules de base du suivi en termes d'entrée et de sortie. Ensuite nous détaillons deux algorithmes de suivi temporel qui diffèrent par le sens de description du mouvement utilisé.

3.2.1 Les deux sens de description du mouvement

Soit deux images I_{t_1} et I_{t_2}, observées respectivement aux instants t_1 et t_2, avec $t_1 < t_2$. Lorsque l'on souhaite décrire le mouvement entre ces deux images pour un codage prédictif, il y a deux possibilités :

Prédiction de I_{t_2} à partir de I_{t_1} : Il s'agit d'un parcours causal où l'on utilise le champ déplacement de I_{t_2} vers I_{t_1} : $\vec{d}^{-}_{t_2 \to t_1}()$. L'exposant $^-$ indique que le mouvement considéré va dans le sens décroissant du temps. La formule de reconstruction est $\widehat{I_{t_2}}(p) = I_{t_1}(p + \vec{d}^{-}_{t_2 \to t_1}(p))$. Comme nous modélisons le déplacement par un modèle affine par régions, le champ dense $\vec{d}^{-}_{t_2 \to t_1}()$ est généré par ce modèle. Pour une région R, le vecteur des paramètres du déplacement affine $[a, b, c, d, t_x, t_y]$ est noté $\Theta^{-R}_{t_2 \to t_1}$. La même notation désigne aussi la transformation affine elle même :

$$\forall p \in R, \Theta^{-R}_{t_2 \to t_1}(p) = p + \vec{d}^{-}_{t_2 \to t_1}(p)$$

Prédiction de I_{t_1} à partir de I_{t_2} : Il s'agit d'un parcours anti-causal, employé pour une interpolation, où l'on utilise le champ déplacement de I_{t_1} vers I_{t_2} : $\vec{d}^{+}_{t_1 \to t_2}()$. L'exposant $^+$ indique que le mouvement considéré va dans le sens croissant du temps. La formule de reconstruction est $\widehat{I_{t_1}}(p) = I_{t_2}(p + \vec{d}^{+}_{t_1 \to t_2}(p))$. Pour une région R, le vecteur des paramètres du déplacement affine $[a, b, c, d, t_x, t_y]$ est noté $\Theta^{+R}_{t_1 \to t_2}$. La même notation désigne aussi la transformation affine elle même :

$$\forall p \in R, \Theta^{+R}_{t_1 \to t_2}(p) = p + \vec{d}^{+}_{t_1 \to t_2}(p)$$

On peut passer d'un descripteur de mouvement $\Theta^{-R}_{t_2 \to t_1}$ dans le sens $^-$ à un descripteur dans le sens $^+$ (et inversement) en considérant la transformation affine réciproque $[\Theta^{-R}_{t_2 \to t_1}]^{-1}$ (voir l'annexe B). Mais pour optimiser la qualité de prédiction par compensation de mouvement, il est nécessaire de réestimer un mouvement $\Theta^{+R}_{t_1 \to t_2}$ à partir de cette initialisation. En effet, les supports d'estimation ne sont pas les mêmes et il est nécessaire de s'appuyer sur la grille discrète des pixels échantillonnés dans l'image que l'on veut prédire et non pas dans l'image qui sert à la prédiction.

3.2.2 Description des modules du suivi

Nous suivons deux types d'objets (les frontières et les régions) et nous avons deux types d'opérations (la prédiction et l'ajustement). Cela fait donc quatre modules algorithmiques distincts à décrire. Nous supposons connues les images de $t = 0$ à $t - 1$; les modules de prédiction fournissent une prédiction pour l'instant t et les modules d'ajustement utilisent l'image I_t. Nous allons par la suite décrire deux algorithmes de suivi, l'un travaillant sur des mouvements du type $\Theta^{-R}_{t\to t-1}$ et l'autre sur des mouvements du type $\Theta^{+R}_{t-1\to t}$. Nous resterons donc général dans la description qui suit, sans présupposer le sens choisi.

Prédiction du mouvement des textures Il s'agit d'un prédicteur qui fonctionne sur les paramètres des mouvements affines des régions $\Theta^{+R}_{t-1\to t}$. Il prend en entrée le mouvement à l'instant précédent $\Theta^{+R}_{t-2\to t-1}$ et fournit en sortie une prédiction [1] $\widehat{\Theta}^{+R,t|t-1,}_{t-1\to t}$. Les mouvements de chaque région sont prédits indépendamment les uns des autres. Les différents prédicteurs seront décrits dans la section 3.3.

Prédiction des frontières Ce prédicteur est le plus simple puisqu'il prend en entrée un mouvement affine $\Theta^{+}_{t-1\to t}$ et l'applique à la frontière $\widehat{F}_{t-1|t-1}$ pour obtenir la frontière prédite $\widehat{F}_{t|t-1}$. Le mouvement servant à la prédiction est celui de l'une des deux régions délimitées par F. Il peut s'agir soit d'un mouvement prédit, soit d'un mouvement estimé, selon le type de suivi effectué. Les critères permettant de décider de quelle région il faut utiliser le mouvement seront montrés dans la section 3.5.

Ajustement du mouvement des textures Ce module est en fait un estimateur de mouvement. Il prend en entrée les deux images I_{t-1} et I_t, un support d'estimation dans l'une des deux images et l'initialisation $\widehat{\Theta}^{\pm R,t|t-1,}_{t_i\to t_j}$ du mouvement à trouver. On a soit $t_i = t$ et $t_j = t - 1$, soit $t_i = t - 1$ et $t_j = t$, selon le type de suivi effectué. Il fournit en sortie le mouvement estimé $\widehat{\Theta}^{\pm R,t|t}_{t_i\to t_j}$. L'estimateur utilisé sera détaillé dans la section 3.4.

Ajustement des frontières Ce module prend en entrée les frontières $\widehat{F}_{t|t-1}$ et l'image I_t. Il les ajuste sur l'image pour fournir les frontières finales $\widehat{F}_{t|t}$. L'algorithme utilisé sera exposé dans la section 3.6.

3.2.3 Suivi rétrograde (ou en mode arrière)

Il s'agit de l'approche utilisée dans [Bonnaud et Labit 94] et [Wu 95] à une variante près et avec la différence importante que ces deux références travaillent sur la représentation RCF. Ce qui la caractérise est l'estimation de descripteurs de mouvement dans le sens $^-$. La raison de ce choix est l'utilisation du suivi dans un schéma de codage causal par compensation de mouvement, pour lequel seuls des descripteurs de la forme $\Theta^{-R}_{t\to t-1}$ sont nécessaires.

L'algorithme part d'une segmentation ajustée sur la deuxième image de la séquence. En effet, dans un schéma de codage causal, la segmentation de la première image ne sert à

1. Une notation du type $\widehat{A}_{t|t-1}$ s'interprète ainsi : estimateur de la grandeur A pour l'image I_t, connaissant l'image I_{t-1}.

rien. On suppose que l'on a aussi estimé les mouvements $\Theta_{2\to 1}^{-R}$ en initialisant l'estimateur sur un mouvement nul. Maintenant, pour passer de l'image I_{t-1} à l'image I_t, les différentes phases se succèdent ainsi :

1. **Prédiction du mouvement des textures** À partir des mouvements $\widehat{\Theta}_{t-1\to t-2}^{-R,t-1|t-1}$ estimés entre le couple d'images précédent, on inverse le sens de description pour obtenir $\widehat{\Theta}_{t-2\to t-1}^{+R,t-1|t-1} = [\widehat{\Theta}_{t-1\to t-2}^{-R,t-1|t-1}]^{-1}$. Ensuite, on génère une prédiction $\widehat{\Theta}_{t-1\to t}^{+R,t|t-1}$ grâce au prédicteur de mouvement.
2. **Prédiction des frontières** On applique les mouvements **prédits** $\widehat{\Theta}_{t-1\to t}^{+R,t|t-1}$ aux frontières $\widehat{F}_{t-1|t-1}$ pour obtenir les frontières prédites $\widehat{F}_{t|t-1}$.
3. **Ajustement des frontières** On ajuste les prédictions $\widehat{F}_{t|t-1}$ sur l'image I_t pour obtenir les frontières ajustées $\widehat{F}_{t|t}$.
4. **Ajustement du mouvement des textures** Les frontières ajustées permettent maintenant de définir des supports d'estimation R dans l'image I_t. Il faut de nouveau inverser le sens du mouvement pour obtenir l'initialisation du mouvement nécessaire à l'estimateur : $\widehat{\Theta}_{t\to t-1}^{-R,t|t-1} = [\widehat{\Theta}_{t\to t-1}^{+R,t|t-1}]^{-1}$. L'estimation de mouvement nous donne alors $\widehat{\Theta}_{t\to t-1}^{-R,t|t}$.

3.2.4 Suivi direct (ou en mode avant)

C'est l'approche utilisée dans [Meyer et Bouthemy 92] et [Bascle et al. 94] sur une représentation RCF mono-objet. Ce qui la caractérise est l'estimation de descripteurs de mouvement dans le sens $^+$, de la forme $\Theta_{t-1\to t}^{+R}$. Dans ces deux références, ce choix est justifié car leur application est l'analyse de la séquence, et non pas le codage de celle-ci au moindre coût. Cette approche est plus simple que la précédente puisqu'elle ne nécessite pas de retournement du temps. Nous verrons aussi, qu'elle permet une meilleure prédiction des frontières.

L'algorithme part d'une segmentation ajustée sur la première image de la séquence. Pour la prédiction et l'estimation du mouvement vers la deuxième image, on utilise un mouvement nul. Maintenant, pour passer de l'image I_{t-1} à l'image I_t, les différentes phases se succèdent ainsi :

1. **Prédiction du mouvement des textures** À partir des mouvements $\widehat{\Theta}_{t-2\to t-1}^{+R,t-1|t-1}$ estimés à l'instant précédent, on génère directement $\widehat{\Theta}_{t-1\to t}^{+R,t|t-1}$ grâce au prédicteur.
2. **Ajustement du mouvement des textures** L'estimation de mouvement se fait sur les supports définis par les frontières $\widehat{F}_{t-1|t-1}$, déjà connues. Pour initialisation on prend simplement $\widehat{\Theta}_{t-1\to t}^{+R,t|t-1}$. L'estimateur fournit alors $\widehat{\Theta}_{t-1\to t}^{+R,t|t}$.
3. **Prédiction des frontières** On applique les mouvements **estimés** $\widehat{\Theta}_{t-1\to t}^{+R,t|t}$ aux frontières $\widehat{F}_{t-1|t-1}$ pour obtenir les frontières prédites $\widehat{F}_{t|t-1}$.
4. **Ajustement des frontières** On ajuste les prédictions $\widehat{F}_{t|t-1}$ sur l'image I_t pour obtenir les frontières ajustées $\widehat{F}_{t|t}$.

Comme il s'agit d'une boucle qui effectue ces quatre phases sur chaque image, il est indifférent de commencer par telle ou telle phase. En pratique, dans notre mise en œuvre, nous avons préféré commencer l'enchaînement de cette boucle par la phase 3, car la phase 2 est de loin la plus longue.

3.2.5 Comparaison des deux modes de suivi

Application à l'analyse

La différence la plus importante entre les deux algorithmes précédents réside dans la phase de prédiction des frontières. Dans le suivi direct, la prédiction est effectuée avec un mouvement estimé, alors que dans le suivi indirect, c'est un mouvement prédit qui est utilisé. Or le mouvement estimé est forcément meilleur que le mouvement prédit puisqu'il tient compte de l'observation de l'image courante I_t. La prédiction des frontières a donc toutes les chances d'être de meilleure qualité, ce qui conduira à des frontières plus précises après l'ajustement.

Application au codage

Le seul intérêt du suivi indirect est donc qu'il a une complexité réduite dans le cadre d'une application au codage par compensation de mouvement **monodirectionnelle**. Si l'on effectuait un suivi direct, il faudrait réestimer les descripteurs $\Theta^{-R}_{t\to t-1}$ à partir des descripteurs $\Theta^{+R}_{t-1\to t}$. Cela doublerait donc le temps passé à l'estimation de mouvement, qui est déjà la phase la plus coûteuse en temps de calcul de l'algorithme.

Mais dans le cadre de notre application au codage par compensation de mouvement **bidirectionnelle**, le suivi direct n'est pas pénalisant. En effet, nous verrons dans le chapitre 4 que nous aurons besoin à la fois des descripteurs $\Theta^{-R}_{t\to t-1}$ et des descripteurs $\Theta^{+R}_{t-1\to t}$. C'est donc pour le suivi direct que nous avons finalement opté dans la suite.

3.3 Prédiction du mouvement des textures

Cette phase de prédiction est effectuée conformément à l'hypothèse H1 qui nous assure que les objets présents dans l'image I_{t-1} sont encore présents dans l'image I_t. Cette prédiction est indispensable dans le suivi indirect, puisqu'elle sert aussi à prédire les frontières. Elle est aussi très utile dans le suivi direct, puisqu'elle sert à initialiser l'estimateur de mouvement, réduisant ainsi le problème des minima locaux. Il est donc important de choisir un bon prédicteur.

3.3.1 Choix d'un mode de prédiction

Dans [Bonnaud et Labit 94] nous avions proposé 4 prédicteurs différents :

- Prédiction par un mouvement nul : on utilise le descripteur de la transformation identité $\widehat{\Theta}^{+R,t|t-1}_{t-1\to t} = I_d$
- Prédiction à court terme : on utilise le mouvement estimé entre le couple d'images précédent $\widehat{\Theta}^{+R,t|t-1}_{t-1\to t} = \widehat{\Theta}^{+R,t-1|t-1}_{t-2\to t-1}$

- Prédiction à court terme lissée : on utilise l'estimateur *a posteriori* d'un filtre de Kalman pour l'instant précédent $t-1$ (voir l'annexe A).
- Prédiction à long terme : on utilise l'estimateur *a priori* d'un filtre de Kalman pour l'instant t [Meyer 93] (voir l'annexe A).

Nous initialisions l'estimateur de mouvement avec ces 4 prédicteurs et obtenions ainsi 4 valeurs d'EQM de prédiction pour l'image. Nous choisissions parmi ces 4 prédicteurs *a posteriori* en fonction de l'EQM obtenue en sortie de l'estimateur de mouvement. Dans cette approche, ces initialisations multiples étaient nécessaires car le prédicteur du filtre de Kalman, utilisé seul, amenait à une divergence du suivi temporel. Cette divergence peut s'expliquer ainsi : l'estimation du mouvement, injectée comme mesure dans le filtre de Kalman, n'est pas indépendante de la prédiction du filtre. En effet, cette prédiction sert d'initialisation à l'estimation et l'estimateur de mouvement dépend fortement de la prédiction. Dès qu'une estimation s'écarte des précédentes, la prédiction pour l'instant suivant amplifie cet écart. S'il n'est pas dû à une vraie variation du mouvement de l'objet, mais à une variation aléatoire, alors l'estimation se trouve mal initialisée. Dans ce cas, l'estimateur de mouvement tombe dans un minimum local très proche de la mauvaise initialisation. Ainsi, la prédiction se trouve confirmée. Le phénomène peut donc se poursuivre avec une prédiction encore plus éloignée de la réalité ; en quelques images, les paramètres de mouvement prennent des valeurs aberrantes.

Dans notre approche actuelle, nous avons pu réduire la complexité globale, tout en obtenant des résultats satisfaisants, grâce à deux simplifications :

- Nous nous limitons à deux prédicteurs : le mouvement nul et l'un des trois autres, de façon indifférente. Nous n'avons jamais observé de divergence avec aucun des 3 autres prédicteurs, et tous donnent des résultats similaires.
- Le test de sélection entre prédicteurs est fait avant l'estimation du mouvement, ce qui évite des estimations coûteuses inutiles.

La sélection du prédicteur est faite, région par région, de la façon suivante :

1. Calcul de l'EQM avec un mouvement nul :

$$\text{EQM}_0 = \sum_{p \in R} [I_{t-1}(p) - I_t(p)]^2$$

2. Calcul de l'EQM avec un mouvement prédit :

$$\text{EQM}_{pred} = \sum_{p \in R} [I_{t-1}(p) - I_t(\widehat{\Theta}^{+R,t|t-1}_{t-1 \to t}(p))]^2$$

3. Si $\text{EQM}_0 < \text{EQM}_{pred}$ alors on utilise le mouvement nul, sinon on utilise le prédicteur.

Ces simplifications sont rendues possibles par l'utilisation d'un meilleur estimateur de mouvement que dans [Bonnaud et Labit 94] et par le suivi direct au lieu du suivi indirect. Malgré ces simplifications, nous obtenons une bonne prédiction comme on pourra en juger sur les résultats de l'estimation dans la section suivante (section 3.4).

3.3.2 Mise à jour du centre du mouvement

Dans l'utilisation d'un descripteur de mouvement (affine ou affine simplifié) en vue d'obtenir un champ dense, intervient un point C_{t-1} qui est le centre du mouvement $\Theta^{+R}_{t-1\to t}$ (voir l'annexe B). Ses deux coordonnées ne font pas partie des paramètres du descripteur car il s'agit de paramètres redondants par rapport aux autres. Mais comme nous allons le voir, ils ont une grande importance vis à vis de la prédiction.

Le choix qui est fait classiquement est de prendre C_{t-1} confondu avec le centre de gravité G_{t-1} de la région à chaque instant. Cela marche très bien tant que l'on suit une région en avant-plan. Mais quand une région est sous une autre et qu'elle subit un découvrement ou un recouvrement, le centre de gravité de ses pixels visibles se déplace. Or un changement du point C affecte la valeur des paramètres de translation (voir l'annexe B). Donc si l'on repérait le mouvement par rapport à G, qui est fortement modifié en cas d'occultation, les paramètres de translation subiraient une évolution qui ne serait pas bien prise en compte par le modèle de prédiction du filtre de Kalman. Pour contourner cette difficulté, le point C_{t-1} est mis à jour par l'application du mouvement prédit :

$$C_t = \widehat{\Theta}^{+R,t|t-1}_{t-1\to t}(C_{t-1})$$

ce qui revient à lui appliquer la translation ce même mouvement

$$C_t = C_{t-1} + \begin{pmatrix} \hat{t}^+_x \\ \hat{t}^+_y \end{pmatrix}.$$

Ainsi, tant qu'il n'y a pas d'occultation, le point C reste confondu avec le centre de gravité, ou du moins suit la même évolution. En cas d'occultation, il continue à se trouver à l'endroit où G aurait été si l'occultation n'avait pas eu lieu.

3.4 Ajustement du mouvement des textures

Cette phase d'estimation de mouvement est effectuée par la mise en cascade de deux estimateurs différents :

- un estimateur robuste multirésolution (estimateur nommé RMR) [Odobez 94],
- un estimateur non robuste et monorésolution, spécialisé pour le codage (estimateur nommé COD).

L'estimateur RMR fonctionne selon le principe rappelé dans la section 1.1. Lorsque la taille de la région le permet, nous utilisons au maximum 4 niveaux de résolution. La fonction de pondération robuste utilisée est celle de Tukey. Cette fonction dépend d'un paramètre réglant le taux de rejet des *outliers*, qui s'interprète comme une variance de la DFD. Nous avons pris 8 comme valeur finale de cette constante, dans le niveau de résolution le plus bas.

L'estimateur COD réalise simplement une minimisation de l'EQM suivante :

$$\text{EQM}(\Theta^{+R}_{t-1\to t}) = \sum_{p\in R}[I_{t-1}(p) - I_t(\Theta^{+R}_{t-1\to t}(p))]^2$$

Lorsque le mouvement déplace le point p sur un point p' situé en dehors du domaine de définition de l'image I_t, il est nécessaire de prolonger celle-ci. L'algorithme de *padding* employé est alors similaire à ceux utilisé dans MPEG. Les pixels des bords sont dupliqués dans la direction perpendiculaire au bord et les coins ont une valeur uniforme égale à celle du pixel correspondant. Mathématiquement cela revient à utiliser la projection P sur le rectangle de définition de l'image, au sens où le projeté de p' est le point le plus proche du rectangle :

$$I_t(p') = I_t[P(p')]$$

La minimisation suivante

$$\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{COD}} = \arg \min_{\Theta^{+R}_{t-1\to t}} \mathrm{EQM}(\Theta^{+R}_{t-1\to t})$$

est effectuée par un algorithme d'optimisation choisi parmi ceux proposés dans [Press et al. 92] pour des fonctions de plusieurs variables réelles, et qui ont une complexité raisonnable. Il s'agit de la méthode des directions conjuguées de Powell, d'une méthode de gradient conjugué de Fletcher – Reeves – Polak – Ribiere (FRPR) ou d'une méthode de type quasi-Newton de Broyden – Fletcher – Goldfarb – Shanno (BFGS). Dans nos expériences, nous avons constaté que la première méthode est la plus fiable, mais qu'elle est nettement plus lente car elle n'utilise pas les dérivées partielles de l'EQM. Pour une étude détaillée des mérites comparés de différents algorithmes de minimisation appliqués au problème de l'estimation de mouvement, nous renvoyons à [Sanson 95].

Les deux derniers algorithmes de minimisation cités utilisent les dérivées partielles de la fonction objectif par rapport aux paramètres de la transformation affine. Ces dérivées s'expriment analytiquement en fonction du gradient de I_t. Une amélioration de notre algorithme par rapport à ceux utilisés dans [Nicolas 92] et [Garcia-Garduño 96] est la prise en compte de la fonction d'interpolation servant à passer de I_t définie sur un rectangle de $\mathbb{N}^2$ à une fonction continue définie sur $\mathbb{R}^2$. Dans les travaux cités, le gradient est calculé en chaque pixel par un filtre de gradient (un filtre de Prewitt par exemple) puis interpolé par une interpolation bilinéaire. Dans notre méthode, le gradient est calculé comme le vecteur les dérivées partielles de la fonction I_t interpolée, ce qui nous donne des dérivées partielles exactes de l'EQM. Nous avons par ailleurs le choix entre une interpolation bilinéaire et une interpolation bicubique. Cette dernière est préférable puisqu'elle conserve mieux les hautes fréquences de l'image interpolée, et assure un gradient continu car la fonction interpolée I_t est alors de classe D^2.

La mise en cascade des deux algorithmes est effectuée ainsi :

1. On initialise l'estimateur RMR avec $\widehat{\Theta}^{+R,t|t-1}_{t-1\to t}$; le résultat est noté $\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{RMR}}$. On calcule l'EQM suivante :

$$\mathrm{EQM}_{\mathrm{RMR}} = \sum_{p\in R}[I_{t-1}(p) - I_t(\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{RMR}})]^2$$

2. Si $\mathrm{EQM}_{\mathrm{RMR}} < \mathrm{EQM}_{pred}$ (avec EQM_{pred}, l'EQM obtenue par le meilleur prédicteur de la section précédente) alors on initialise l'estimateur suivant avec $\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{RMR}}$, sinon

on utilise $\widehat{\Theta}^{+R,t|t-1}_{t-1\to t}$. Cette comparaison est nécessaire puisque l'estimateur RMR ne cherche pas à minimiser l'EQM mais une version robuste de celle-ci. Elle peut donc très bien augmenter.

3. Le deuxième estimateur fournit alors $\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{COD}}$ et l'EQM correspondante

$$\mathrm{EQM}_{\mathrm{COD}} = \sum_{p\in R}[I_{t-1}(p) - I_t(\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{COD}})]^2$$

4. Comme on a toujours $\mathrm{EQM}_{\mathrm{COD}} < \min(\mathrm{EQM}_{\mathrm{RMR}}, \mathrm{EQM}_{pred})$, on prend comme estimation finale

$$\widehat{\Theta}^{+R,t|t}_{t-1\to t} = \widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{COD}}$$

Cette mise en cascade a deux avantages :

- Grâce au premier estimateur, nous bénéficions de l'optimisation multirésolution pour les mouvements de grande amplitude, et de la robustesse pour les occultations entre régions.

- Le deuxième estimateur, n'étant pas robuste, réduit encore plus l'EQM, ce qui sera bénéfique pour l'interpolation. Mais comme il est monorésolution, il ne risque pas de modifier très fortement le mouvement estimé précédemment, ce qui serait préjudiciable au suivi temporel.

Les résultats de l'estimation pour les séquences *«Miss America»* et *«Flower Garden»* sont montrés dans les figures 3.3 et 3.4. Les courbes présentent l'EQM plutôt que le PSNR, pour ne pas laisser penser qu'il s'agit d'une image décodée. Il faudrait encore prendre en compte le codage de l'image d'erreur, ce qui sera fait dans le chapitre 4. Nous avons toutefois choisi une échelle logarithmique, comme dans le PSNR.

Dans la séquence *«Miss America»*, l'EQM sans compensation de mouvement est relativement basse au début et à la fin de la séquence car le mouvement est faible. Par contre, un mouvement plus important se produit entre les images 30 et 60. On constate que le suivi temporel assure une compensation de mouvement de très bonne qualité, avec une EQM comprise entre 5 et 10. En outre, notre algorithme est légèrement meilleur que le *block-matching* de MPEG.

Toutes proportions gardées, on observe un résultat similaire pour la séquence *«Flower Garden»*. L'EQM y est nettement plus forte car le mouvement est bien plus important. Mais le gain en EQM est proportionnellement plus élevé. L'amélioration par rapport au *block-matching* est plus nette. Ceci est en partie dû aux fortes occultations dans la séquence qui sont mieux traitées par notre algorithme, même sans interpolation. Dans la séquence *«Miss America»*, le gain est plus faible car le fond occulté est quasiment uniforme, ce qui ne pénalise pas un mouvement incorrect.

Les résultats pour les autres séquences de test seront montrés dans la section 3.7.

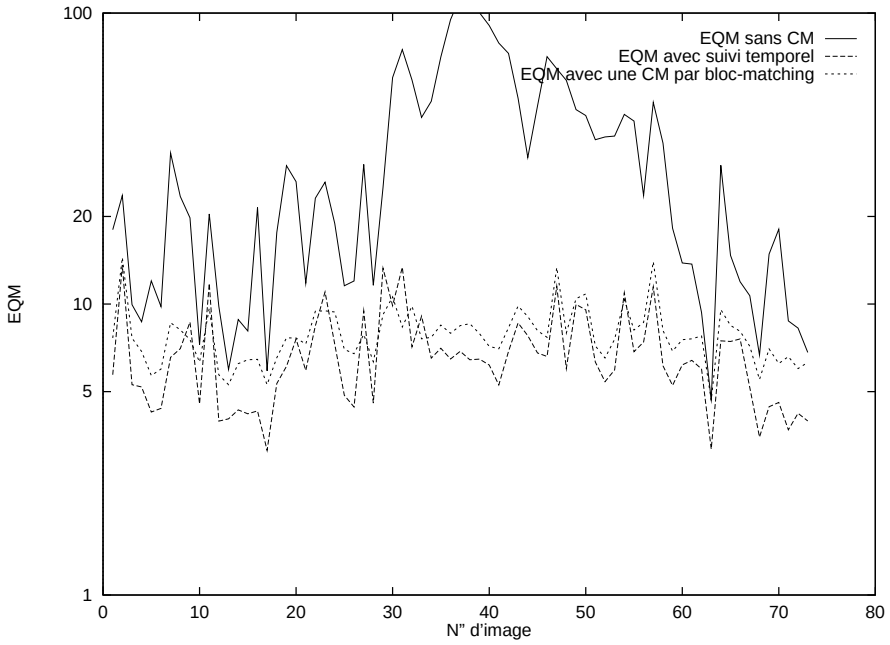


FIG. 3.3 – EQM *de prédiction par compensation de mouvement des images de la séquence* «Miss America»*, avec notre algorithme de suivi temporel et par un algorithme de* block-matching *(*16×16 *pixels, mouvement translationnel, estimé au pixel près).*

3.5 Prédiction et affectation des frontières

La prédiction d'une frontière est effectuée en lui appliquant le mouvement de l'une des deux régions qu'elle délimite :

$$\widehat{F}_{t|t-1} = \widehat{\Theta}^{+R_i,t|t}_{t-1\to t}(\widehat{F}_{t-1|t-1})$$

avec $R_i = R_g$ ou R_d, la région gauche ou droite. Ceci est fait en accord avec l'hypothèse H3 imposant l'identité entre mouvements apparents des frontières et des textures (voir la section 3.1). La prédiction s'applique à toutes les frontières, sauf celles du bord de l'image. En effet, nous souhaitons qu'elles restent à leur place puisque le bord lui-même ne bouge pas.

Remarquons qu'après application d'un mouvement affine, la frontière prédite a des coordonnées réelles. Nous ne repassons pas immédiatement en coordonnées entières, car ce n'est pas gênant pour la phase suivante de l'algorithme, et ainsi toute la précision est conservée.

Les frontières issues d'un même point multiple vont subir des mouvements différents, dans cette phase et dans la phase suivante d'ajustement affine. Il n'est donc plus possible de représenter l'extrémité commune à ces frontières par un unique point multiple. Celui-ci est donc dupliqué un nombre de fois égal à sa multiplicité et chaque frontière est augmentée d'un exemplaire à l'une de ses extrémités.

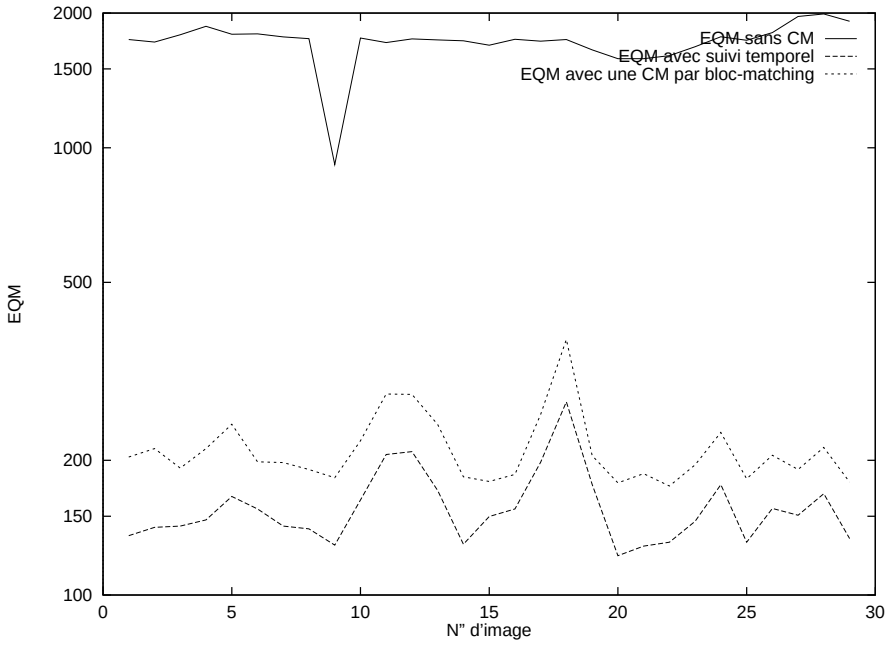


FIG. 3.4 – EQM *de prédiction par compensation de mouvement (CM) des images de la séquence* «Flower Garden»*, avec notre algorithme de suivi temporel et par un algorithme de* block-matching.

Lorsque la frontière que l'on considère est causée par une discontinuité du champ de profondeur, la région à laquelle la frontière doit être affectée est celle du dessus. Ainsi, avec cette information supplémentaire notre représentation s'enrichit pour passer d'un modèle de mouvement 2D à un modèle $2D\frac{1}{2}$ de la scène.

Dans les algorithmes de suivi pour les représentations RCE des critères sont définis pour déterminer l'ordre de superposition des régions. Dans [Pardas et al. 94] et [Wu 95], si deux régions se recouvrent, la région du dessus est celle dont le mouvement permet de prédire au mieux leur intersection. Pour les découvrements, [Wu 95] distingue les découvrements de zones minces et les découvrements de zones épaisses. Les zones minces sont réaffectées au régions existantes selon un critère spatial de luminance ou de contraste. Les zones épaisses sont resegmentées spatialement ; de nouvelles régions sont donc crées. Dans [Begen et Meyer 98], les auteurs proposent un critère basé sur la densité des *outliers* d'un estimateur robuste de mouvement, après pondération par le gradient spatial. Ils mesurent cette densité dans deux fines bandes de part et d'autre de la frontière entre deux régions. Si une région a une forte densité d'*outliers*, cela signifie qu'elle est en dessous.

Pour notre part, nous définissons deux critères : un critère spatial et un critère basé mouvement. Le premier est spécifique à notre représentation et spécialisé pour une utilisation dans l'algorithme de suivi temporel. Le deuxième est similaire à [Wu 95] dans le cas des recouvrements, mais les découvrements sont traités de façon symétrique, comme dans [Morier et al. 98] [Morier 98]. Nos critères sont définis sur une frontière et non pas sur un

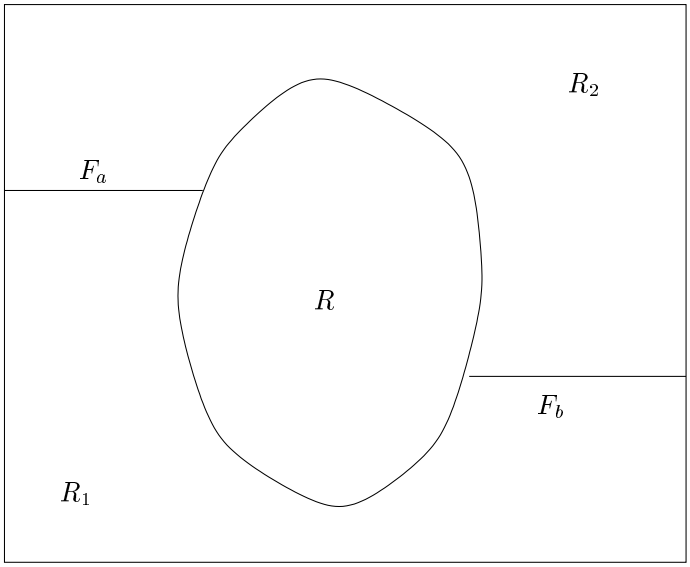


FIG. 3.5 – *Affectation des frontières et ordre de superposition des régions.*

couple de régions. Cela permet une description un peu plus fine de la scène, comme on peut le voir sur la figure 3.5. Si on note $R_1 \prec R_2$ la relation R_1 «est en dessous de» R_2 et $F \sqsubset R_i$ la relation F «est affectée» à la région R_i, on peut donc avoir $F_a \sqsubset R_1$ et $F_b \sqsubset R_2$. Dans ce cas on n'a ni $R_1 \prec R_2$, ni $R_2 \prec R_1$.

3.5.1 Critère spatial

Ce critère très simple part de l'idée que la frontière prédite servira à la phase d'ajustement spatial. Sans trop anticiper sur cette phase, on peut dire que l'ajustement spatial est effectué en maximisant l'intégrale du gradient spatial de l'image I_t le long des frontières. Il est donc souhaitable de partir d'un gradient déjà élevé, signe qu'un contour est présent à l'endroit de la prédiction.

Pour une frontière F, définie paramétriquement (on suppose le paramétrage uniforme) par la fonction

$$F : \left| \begin{array}{rcl}]0,1[& \longrightarrow & \mathbb{R}^2 \\ s & \longmapsto & F(s) \end{array} \right.$$

on définit donc la quantité suivante :

$$GR(F) = \int_0^1 \|\vec{\nabla} I_t\|(F(s))\,\mathrm{d}s$$

Soit R_g et R_d, les régions respectivement à gauche et à droite de F. Si

$$GR(\widehat{\Theta}^{+R_g,t|t}_{t-1\to t}(F)) > GR(\widehat{\Theta}^{+R_d,t|t}_{t-1\to t}(F))$$

alors $F \sqsubset R_g$ et

$$\widehat{F}_{t|t-1} = \widehat{\Theta}^{+R_g,t|t}_{t-1\to t}(\widehat{F}_{t-1|t-1})$$

L'avantage de ce critère est qu'il est très peu complexe à calculer. Sa complexité est proportionnelle à la longueur de la frontière car nous décomposons l'intégrale sur les segments du polygone et nous l'approchons sur chaque segment par la méthode des trapèzes avec un nombre d'échantillons proportionnel à la longueur du segment.

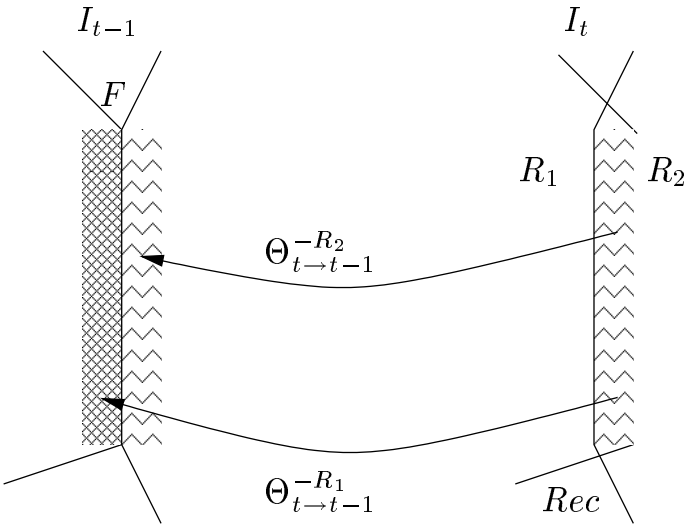


FIG. 3.6 – *Affectation des frontières : recouvrement simple.*

3.5.2 Critère basé mouvement

L'idée de départ de ce critère est illustrée sur les figures 3.6 et 3.7 représentant une frontière F entre deux régions R_1 et R_2, que l'on cherche à prédire de l'image I_{t-1} vers l'image I_t. Sur ces deux figures, on a supposé que $R_1 \prec R_2$.

La figure 3.6 montre que dans le cas d'un recouvrement, la zone d'intersection entre les deux régions, notée Rec, a pour mouvement celui de la région R_2. Cette zone existant dans les images I_t et I_{t-1}, le mouvement peut être pris entre ces deux instants. Le descripteur de mouvement correct est donc $\Theta^{-R_2}_{t\to t-1}$. Le critère correspondant se base donc sur les erreurs quadratiques suivantes :

$$\text{EQ}_i^{Rec} = \sum_{p\in Rec} [I_t(p) - I_{t-1}(\widehat{\Theta}^{-R_i,t|t}_{t\to t-1}(p))]^2$$

et s'écrit :

$$\text{si } \text{EQ}_2^{Rec} < \text{EQ}_1^{Rec} \text{ alors } F \sqsubset R_2.$$

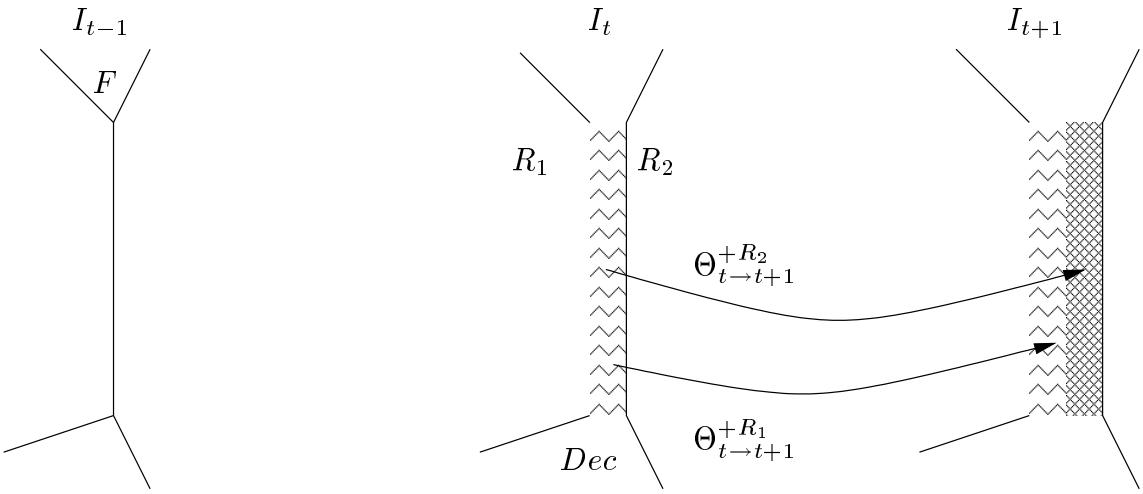


FIG. 3.7 – *Affectation des frontières : découvrement simple.*

Dans le cas d'un découvrement (figure 3.7), la zone découverte, notée Dec, a pour mouvement celui de la région R_1. Mais contrairement au cas précédent, cette zone n'est observable que sur les images I_t et I_{t+1}. On est donc obligé de regarder dans l'image I_{t+1} et le descripteur de mouvement correct est donc $\Theta^{+R_1}_{t\to t+1}$. De plus, dans cette phase de l'algorithme, les mouvements $\widehat{\Theta}^{+R_i,t+1|t+1}_{t\to t+1}$ n'ont pas encore été estimés car les supports des régions dans I_t sont encore en cours de calcul. Il faut donc se contenter des prédictions $\widehat{\Theta}^{+R_i,t+1|t}_{t\to t+1}$. Le critère correspondant se base donc finalement sur les erreurs quadratiques suivantes :

$$\text{EQ}_i^{Dec} = \sum_{p\in Dec} [I_t(p) - I_{t+1}(\widehat{\Theta}^{+R_i,t+1|t}_{t\to t+1}(p))]^2$$

et s'écrit :

$$\text{si } \text{EQ}_1^{Dec} < \text{EQ}_2^{Dec} \text{ alors } F \sqsubset R_2.$$

Dans le cas général, une frontière peut être le lieu à la fois d'un recouvrement et d'un découvrement. C'est par exemple le cas illustré par la figure 3.8 d'une frontière en rotation. Il faut donc déterminer quelles sont ces zones. Dans cette figure, on a supposé que le mouvement de R_1 était nul, mais cela se fait sans perte de généralité car seul le mouvement relatif est important pour cette détermination. On appelle F_1 et F_2 les frontières prédites avec les mouvements des régions R_1 et R_2 :

$$F_i = \widehat{\Theta}^{+R_i,t|t}_{t-1\to t}(F)$$

L'union des zones recouvertes et découvertes est le polygone formé par F_1, F_2 et les deux segments permettant de fermer ce polygone. Maintenant, il faut orienter F_1 et F_2 de sorte que la région R_i soit à gauche de la frontière F_i quand on la parcourt selon son orientation. Pour distinguer entre recouvrement et découvrement il suffit de regarder l'orientation des

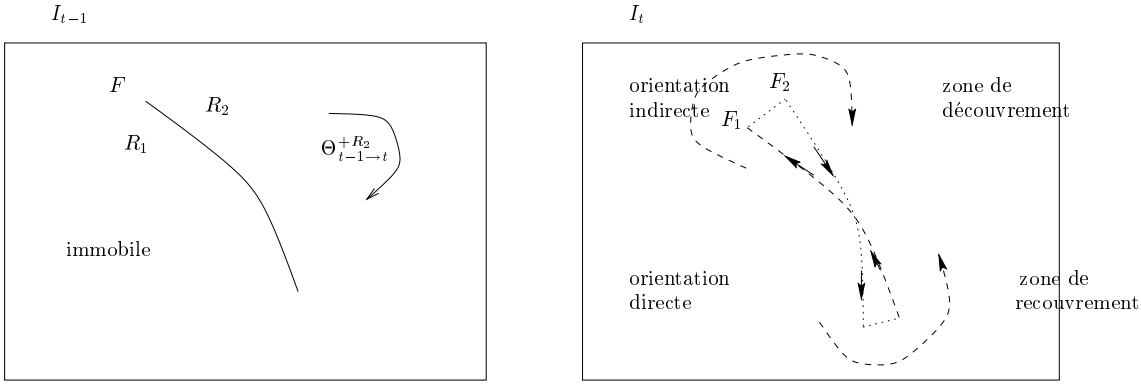


FIG. 3.8 – *Affectation des frontières : occultation complexe (recouvrement et découvrement).*

composantes connexes de l'intérieur du polygone. Si une composante connexe est orientée dans le sens trigonométrique direct, il s'agit d'une zone de recouvrement. Si elle est orientée dans le sens trigonométrique indirect, il s'agit d'une zone de découvrement. Remarquons que cette distinction ne dépend pas des mouvements et qu'elle est purement géométrique.

Le critère final prenant en compte à la fois recouvrements et découvrements s'écrit donc :

$$\text{si } \mathrm{EQ}_2^{Rec} + \mathrm{EQ}_1^{Dec} < \mathrm{EQ}_1^{Rec} + \mathrm{EQ}_2^{Dec} \text{ alors } F \sqsubset R_2.$$

Le calcul de ce critère est un peu plus complexe que le précédent car il nécessite un remplissage de polygone et un calcul d'erreur de compensation de mouvement.

3.5.3 Frontières ambiguës et choix du critère

Le critère basé mouvement semble au premier abord plus intéressant que le critère spatial. En effet, il utilise toute l'information disponible qui est l'ensemble des zones d'occultation, alors que l'autre n'utilise que l'information disponible le long de la frontière. Mais avec ce critère, trois problèmes se posent :

- Pour les zones découvertes, il est nécessaire d'utiliser un mouvement prédit, ce qui nuit à la fiabilité du critère. L'alternative serait d'utiliser un mouvement estimé, mais elle aurait deux inconvénients. D'abord il faudrait estimer le mouvement sur un support dans l'image I_t qui serait prédit. Ceci n'est pas trop gênant si on emploie un estimateur robuste, et si la région a une surface supérieure à la surface des erreurs de prédiction de son support. Ensuite, et c'est l'inconvénient majeur, l'estimation de mouvement aurait un coût de calcul important.

- Toujours pour les zones découvertes, il est nécessaire de disposer de l'image I_{t+1}. Cela introduit un retard d'une image dans le traitement de la séquence. Mais ce n'est pas gênant puisque, pour notre algorithme d'interpolation, nous introduisons déjà un retard de plusieurs images, car nous interpolons des images de type B entre deux images de type I ou de type P. Le seul problème se pose pour la dernière image I ou P.

FIG. 3.9 – *Prédiction des frontières (*«Flower Garden»*, image 1 —→ 2). Sur la deuxième image, les points multiples sont montrés dans leur position antérieure.*

Pour celle là, nous pouvons soit utiliser le critère spatial, soit réutiliser l'affectation de la frontière dans l'image précédente.

- Si l'occultation se limite à une zone découverte, le critère ne marche que si la région du dessous est texturée. En effet, si elle est uniforme, qu'on lui applique le mouvement de la région du dessus ou celui de la région du dessous, l'erreur quadratique sera proche de 0.

- La fiabilité de ce critère est très dépendante de la surface des zones d'occultation. En effet, si les erreurs quadratiques sont calculées sur un faible nombre de pixels, elles sont très bruitées. Dans ce cas, on peut soit utiliser le critère spatial, soit classer la frontière comme «ambiguë». Sauf dans le cas peu probable où la frontière est une ligne droite et que le mouvement relatif entre les deux régions lui est parallèle, cela signifie que le mouvement relatif est petit. On peut donc effectuer la prédiction avec la moyenne des deux mouvements.

Dans nos expériences, nous avons donc choisi le critère en fonction de la séquence. Par exemple, dans la séquence *«Miss America»* (voir la figure 3.21), le mouvement est suffisamment lent pour que l'affectation des frontières ne soit pas essentielle. De plus, le fond étant uniforme, le critère basé mouvement est difficilement utilisable. Dans la séquence *«Interview»* (voir la figure 3.22), seules quelques frontières ont un mouvement important, mais là encore l'algorithme fonctionne avec une prédiction par le mouvement moyen des deux régions séparées par la frontière à prédire.

Par contre, dans la séquence *«Tennis»* (voir la figure 3.23), le mouvement de la balle est très rapide et une bonne prédiction est essentielle. Le contour de la balle étant bien marqué, le critère spatial est suffisant. Dans la séquence *«Flower Garden»* (voir la figure 3.9), le mouvement de l'arbre est lui aussi assez rapide et nécessite une bonne prédiction des frontières. Mais dans cette séquence, le fond est très texturé et se prête donc bien au critère basé mouvement.

3.6 Ajustement des frontières

L'ajustement des frontières est une phase indispensable pour atteindre notre objectif de précision de localisation des objets. Même si la prédiction est effectuée avec un mouvement estimé (et non pas prédit), elle ne peut pas toujours prendre en compte des déformations complexes d'objets non rigides.

De nombreuses techniques et variantes existent pour effectuer cette tâche d'ajustement. Citons par exemple la technique des ECM (Éléments de Contours en Mouvement) [Bouthemy 89] [Ricquebourg et Bouthemy 95] dans laquelle il s'agit de maximiser la corrélation le long de petits segments perpendiculaires à la frontière, calculée entre les images I_{t-1} et I_t. Cette technique a l'avantage d'être rapide, mais a pour inconvénient de ne pas prendre en compte les fonds texturés. On peut donc difficilement l'appliquer à notre cas où nous ne faisons aucune hypothèse sur la texture des objets.

Une autre grande famille de méthodes est celle des contours actifs, déjà évoquée dans la section 1.3.3. Nous ne pouvons pas utiliser la technique telle quelle car elle est prévue à

l'origine pour une initialisation manuelle du contour actif à l'extérieur de l'objet à extraire. Un terme d'énergie force ensuite le contour à se rétracter pour épouser la forme de l'objet. Or dans un suivi temporel la prédiction peut se trouver dans une position quelconque par rapport au contour idéal. Si la prédiction est à l'intérieur de l'objet, le contour actif risque alors de se rétracter complètement jusqu'à se réduire à un point.

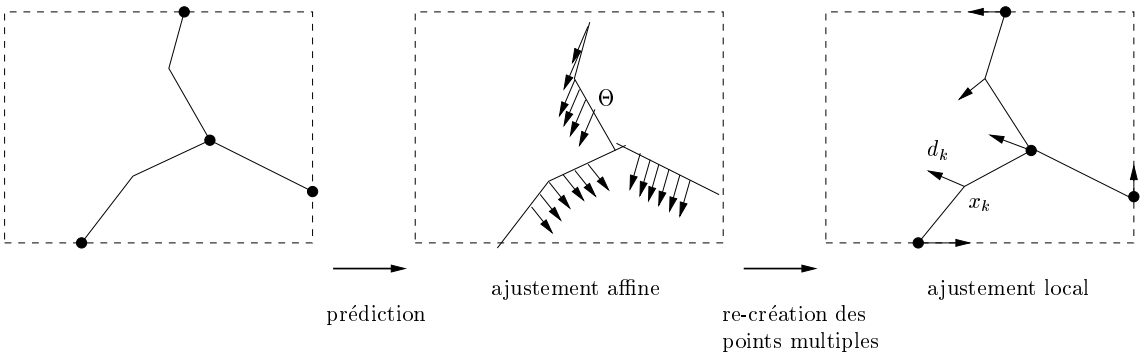


FIG. 3.10 – *Vue d'ensemble de l'ajustement.*

Nous utilisons donc une technique inspirée de [Bascle et al. 94] où l'ajustement est décomposé en deux étapes : une étape d'ajustement affine où le contour subit une déformation affine et une étape d'ajustement local où la déformation est libre. Cette technique a été adaptée, pour le suivi de plusieurs objets simultanément, au cas des frontières ouvertes, avec la prise compte de l'interaction entre frontières (voir la figure 3.10). L'adaptation a consisté en l'insertion d'une étape intermédiaire de re-création des points multiples, par la définition d'une fonction d'énergie calculée sur l'ensemble de notre structure de représentation et par l'adoption d'une méthode d'optimisation plus adaptée au codage de la structure.

Les observations que nous tirons de l'image I_t pour l'ajustement des frontières sont les simples gradients spatiaux $\vec{\nabla} I_t$. Certains travaux [Verbeek 92] utilisent plutôt comme observations une carte de distance par rapport à des contours extraits de l'image. L'avantage de cette méthode est que le contour actif initial peut être initialisé assez loin du contour souhaité. Mais dans notre algorithme de suivi, nous avons vu que la prédiction donne des frontières déjà assez proches de la bonne position. De plus, cette technique nécessite la binarisation de l'image des gradients $\vec{\nabla} I_t$ et donc le choix d'un seuil. Cela est acceptable dans le cas de l'extraction d'un unique objet. Mais dans le cas de l'ajustement de plusieurs objets, si un contour de faible contraste est en dessous du seuil, la frontière correspondante va être irrémédiablement attirée sur le contour d'un autre objet. Or notre algorithme doit pouvoir fonctionner même sur des séquences aux contours peu marqués, comme la séquence *«Miss America»*, par exemple.

D'autres travaux utilisent des information spatio-temporelles pour l'ajustement de contours actifs. Par exemple, dans [Unk2] la fonction d'énergie contient des termes calculés à partir des dérivées partielles de l'image I_t par rapport au temps. C'est utile dans le cas où les informations purement spatiales viennent à manquer. Mais ces techniques ont

des inconvénients lorsque les mouvements sont de grande amplitude, et entraînent alors des imprécisions sur la localisation des frontières. Nous préférons donc nous en tenir à notre hypothèse H2 et aux gradients spatiaux.

3.6.1 Ajustement affine des frontières (AAF)

Cette 1ère étape d'ajustement consiste en un recalage global de la frontière par une transformation affine. L'intérêt d'utiliser un tel recalage est qu'il dispense des termes de régularisation employés classiquement dans la fonction d'énergie d'un contour actif. Pour des contours actifs définis par une courbe paramétrique de régularité C^1, la régularisation consiste par exemple à pénaliser une forte courbure du contour [Kass et al. 87]. Pour un contour polygonal, il est possible de remplacer la courbure par une approximation par différences finies [Amini et al. 90], mais cela suppose un échantillonnage fin du contour, ce qui va à l'encontre d'un codage efficace de celui-ci. Une autre alternative consiste à rajouter un terme énergétique inspiré d'un modèle mécanique à base de ressorts [Unk1]. Mais dans notre cas, avec un ajustement par application d'une transformation affine, la frontière garde son allure générale inchangée et donc la régularisation est inutile.

Pour une frontière F, définie paramétriquement (on suppose le paramétrage uniforme) par la fonction

$$F : \left| \begin{array}{rcl}]0,1[& \longrightarrow & \mathbb{R}^2 \\ s & \longmapsto & F(s) \end{array} \right.$$

on définit donc la quantité suivante :

$$E_F^d(\Theta_{t\to t}^F) = -\int_0^1 \|\vec{\nabla} I_t\|[\Theta_{t\to t}^F(F(s))]\,\mathrm{d}s$$

On effectue alors la minimisation de cette fonction :

$$\Theta_{t\to t}^{\mathrm{AAF},F} = \arg\min_{\Theta_{t\to t}^F} E_F^d(\Theta_{t\to t}^F)$$

et l'ajustement lui-même est réalisé par l'application du mouvement optimal :

$$\widehat{F}_{t|t}^{\mathrm{AAF}} = \Theta_{t\to t}^{\mathrm{AAF},F}(\widehat{F}_{t|t-1})$$

L'algorithme de minimisation employé est une simple descente de gradient, du même type que celle utilisée pour l'estimation de mouvement. La frontière ajustée tombe ainsi dans le minimum local le plus proche. Ce qui est souvent considéré comme un défaut de la méthode est pour nous un avantage. En effet, si deux objets ont des contours proches l'un de l'autre, il ne faut pas que le contour le plus marqué attire les deux frontières. Pour cette même raison, l'algorithme marche même si le contour sur lequel la frontière doit s'ajuster est de faible contraste.

Un autre défaut classique des méthodes basées sur les contours actifs est la sensibilité à un fond très texturé ou contenant de très forts gradients. Nous avons donc apporté deux améliorations par rapport à la fonction d'énergie de [Bascle et al. 94] :

Gradient normal : Dans la fonction d'énergie E_F^d, au lieu de considérer la norme de $\vec{\nabla} I_t$ nous ne considérons que sa projection sur la normale à la frontière. Appelons $\vec{N}(s)$

FIG. 3.11 – *Ajustement affine des frontières (*«Flower Garden»*, image 2). En haut : avant ajustement. En bas : après ajustement.*

le vecteur normal à F en $F(s)$. Dans l'énergie de F, nous utilisons donc

$$|\,\vec{\nabla} I_t[\Theta^F_{t\to t}(F(s))]\,.\,\Theta^F_{t\to t}(\vec{N}(s))\,|$$

Gradient seuillé : Pour limiter l'influence relative des parties de fort contraste le long du contour, nous effectuons un seuillage des composantes du gradient $\vec{\nabla} I_t$. Un seuillage brutal aurait pour effet une perte de précision dans la localisation de la frontière, donc nous avons préféré un seuillage «doux» avec la fonction suivante :

$$t : \left| \begin{array}{l} \mathbb{R} \longrightarrow \mathbb{R} \\ x \longmapsto t(x) = \begin{cases} \log(1+x) & \text{si } x \geq 0 \\ -\log(1-x) & \text{si } x \leq 0 \end{cases} \end{array} \right.$$

Par convention, t s'applique aussi sur un vecteur, en s'appliquant sur chaque composante : $t\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t(x) \\ t(y) \end{pmatrix}$. Nous préférons appliquer t aux composantes du gradient plutôt qu'au produit scalaire pour des raisons de rapidité : cela permet de précalculer les images des composantes de $t(\vec{\nabla} I_t)$.

En définitive, l'expression complète de la fonction d'énergie est

$$E^d_F(\Theta^F_{t\to t}) = -\int_0^1 t(\vec{\nabla} I_t[\Theta^F_{t\to t}(F(s))]).\Theta^F_{t\to t}(\vec{N}(s))\,\mathrm{d}s$$

De même que pour la prédiction des frontières, l'ajustement affine ne s'applique pas aux frontières du bord de l'image.

La complexité de cet algorithme est relativement faible car elle ne dépend pas du nombre de pixels de l'image. Elle est proportionnelle à la longueur totale des segments dans les frontières polygonales de notre représentation (nous échantillonnons les intégrales comme dans la section 3.5.1). Elle dépend donc plus de la complexité de la scène que du nombre de pixels. De plus, les minimisations effectuées pour chaque frontière sont indépendantes, ce qui permet de les exécuter en parallèle.

3.6.2 Re-création des points multiples

À ce stade de l'algorithme, les points multiples n'ont plus d'existence puisque les frontières ont été déconnectées entre elles par les mouvements de la prédiction et de l'ajustement affine. Les points multiples ont été dupliqués et intégrés aux frontières avant la phase de prédiction. Pour obtenir de nouveau une structure complète pour la suite, il est nécessaire de recréer les points multiples dans une $2^{\text{ème}}$ étape. Cette étape nécessite d'une part de couper les frontières qui s'intersectent et d'autre part de prolonger les frontières qui se sont éloignées des autres frontières connectées au même point multiple d'origine.

Pour effectuer ce traitement, il est possible d'utiliser différents types d'informations :

- Informations image : Ce sont les informations pouvant potentiellement donner la re-création la plus précise. Cependant nous avons choisi de ne pas nous en servir car nous réutiliserons le même algorithme à deux reprises pour l'interpolation temporelle, dans

l'interpolation par prédiction bidirectionnelle de segmentation (section 4.4.2) et dans l'interpolation par prédiction du mouvement (section 4.4.3). Or dans ces contextes, l'image ne sera pas disponible.

- Informations de mouvement : Ces informations sont insuffisantes et même non pertinentes. En effet, peu importe si une frontière se rapproche ou s'éloigne d'une autre. Ce qui compte est sa position instantanée et surtout sa profondeur relative par rapport aux autres frontières.

- Informations de profondeur : Ce serait les informations les plus utiles, mais malheureusement elles ne sont pas toujours disponibles, à cause du problème des frontières ambiguës. Il serait possible de traiter les cas particuliers où cette information est absente, partiellement disponible ou complète, mais cela complexifierait encore la procédure. L'algorithme que nous présentons ci-dessous n'utilise pas cette information, mais nous indiquerons comment il serait possible d'en tirer partie.

- Informations géométriques et topologiques : Ce sont les informations de plus bas niveau, mais elles ont l'avantage d'être fiables et toujours disponibles. Combinées à quelques heuristiques, nous verrons qu'elles sont suffisantes.

Finalement, la re-création des points multiples est effectuée par des techniques de géométrie algorithmique plutôt que par un traitement au niveau des pixels, ce qui est conforme à notre idée de n'effectuer des transformations que sur la représentation RFO, sans passer par une représentation RCE (voir la section 2.1.4 et les figures 2.5 et 2.6). L'éventuelle perte de précision causée par la non utilisation des informations image et des informations de profondeur, n'est pas très grave, puisqu'elle sera rattrapée grâce à l'étape suivante d'ajustement local.

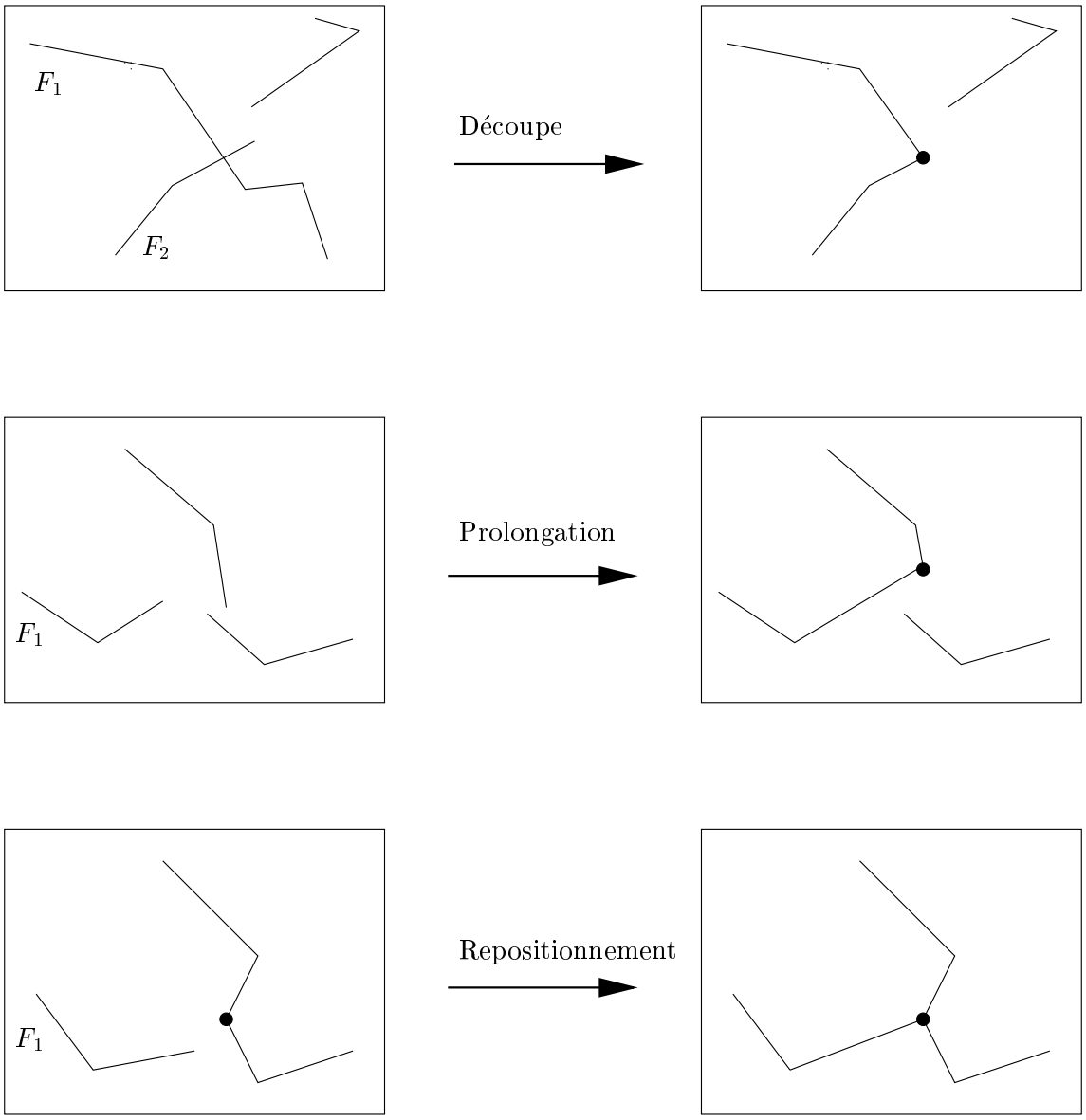


FIG. 3.12 – *Opérations de base pour la re-création des points multiples.*

Trois opérations de base sont disponibles pour cette opération (voir la figure 3.12) :

La découpe Lorsqu'une frontière F_1 est coupée par une autre frontière F_2, il faut la couper entre le point d'intersection et l'une des extrémités. Pour cela, les sommets concernés du polygone sont supprimés et le segment où a lieu l'intersection est coupé. La partie la plus délicate est en fait la détermination de la bonne extrémité à supprimer. Deux cas se présentent :

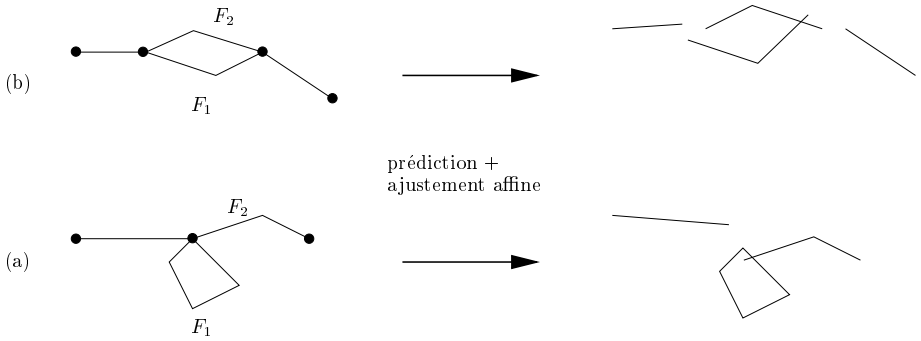


FIG. 3.13 – *Cas particuliers pour la découpe des frontières.*

- Soit les deux frontières étaient déjà reliées au même point multiple. On a alors gardé la mémoire de quelle extrémité correspondait à ce point multiple et l'on peut s'en resservir. Notons que cette information n'est utilisable que si les deux extrémités de F_1 étaient reliées à des points multiples différents (cas particulier illustré dans la figure 3.13a). Il faut de plus que F_2 ne soit pas reliée à ces deux mêmes points multiples. Ce serait par exemple le cas dans la situation illustrée dans la figure 3.13b qui nous oblige à considérer un multigraphe au lieu d'un simple graphe.
- Soit le contact entre ces deux frontières est nouveau et l'on s'en remet à l'heuristique suivante : on choisit l'extrémité la plus proche du point d'intersection. Cette heuristique est aussi utilisée dans les cas particuliers du premier cas.

La prolongation Lorsque l'une des deux extrémités d'une frontière est déconnectée des autres, l'un des moyens pour la reconnecter est de prolonger le dernier segment du polygone. On le prolonge alors jusqu'à ce qu'il rencontre une autre frontière. La justification de cette méthode est qu'il s'agit juste d'une fermeture des frontières à courte distance. Il n'est donc pas justifié de rajouter un segment supplémentaire dans le polygone de la frontière, qui irait de l'extrémité à une autre frontière.

Mais cette technique est délicate à manier car le prolongement du segment peut très bien ne rencontrer aucune autre frontière, ou une frontière distante sans aucun rapport. Il faut alors faire l'hypothèse que cette frontière doit se reconnecter avec les mêmes que précédemment. On choisit donc deux frontières (le problème du choix est

discuté ci-dessous) et on va prolonger leur dernier segment. Deux cas se présentent alors :

- La prolongation de l'un des segments intersecte l'autre frontière. Il faut donc couper cette frontière comme expliqué précédemment.
- Les droites supportant les segments aux extrémités s'intersectent en un point p ne faisant pas partie des frontières. Il faut alors prolonger les deux frontières jusqu'à ce point.

Le repositionnement Il s'agit du deuxième moyen pour reconnecter une frontière aux autres. Au lieu de prolonger le segment extrémité, on déplace tout simplement le point extrémité vers un point d'une autre frontière. La justification de cette méthode est que le changement d'angle qui va en résulter est assez faible.
De même que précédemment, il faut faire l'hypothèse que la frontière doit se reconnecter aux mêmes frontières qu'à l'instant précédent. Là encore, deux cas se présentent :

- Un point multiple complet ou incomplet (dans les deux cas de multiplicité ≥ 2) existe déjà. Il peut avoir été produit par l'une des opérations ci-dessus. Il suffit alors de connecter la frontière à ce point.
- Aucun point multiple n'existe encore. Soit m la multiplicité du point multiple préexistant. On calcule alors l'isobarycentre des extrémités des m frontières, et leurs extrémités sont repositionnées sur ce point.

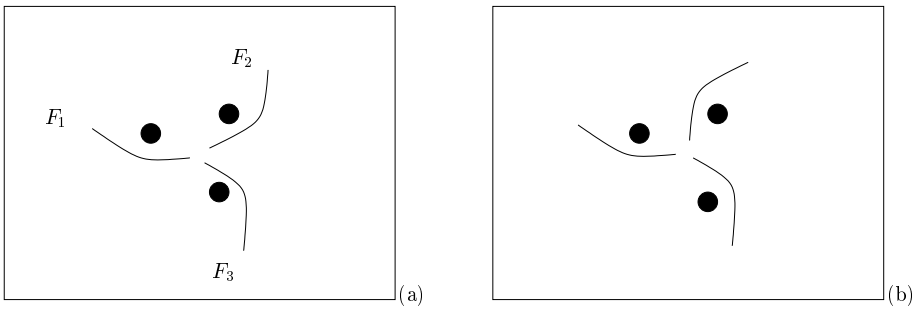


FIG. 3.14 – *Point triple et informations de profondeur. Sur cette figure, l'affectation d'une frontière à sa région est représentée par un disque noir. Quand toutes les frontières sont affectées à une région, il y a $2^3 = 8$ configurations possibles. Mais en prenant en compte les permutations d'indices, deux cas seulement sont représentatifs.*

Parmi ces trois opérations, la prolongation et le repositionnement remplissent la même fonction. Il est donc possible de substituer l'un à l'autre. Comme la prolongation est la plus délicate à manier, nous avons préféré n'utiliser que le repositionnement. En effet, l'utilisation de la prolongation nécessite le choix de deux frontières à prolonger. Le bon choix ne peut être fait que si les informations de profondeur sont connues. Prenons par exemple le cas le plus simple qui est celui du point triple (voir la figure 3.14). Supposons que toutes les informations de profondeur relatives sont connues. Intéressons nous maintenant

au cas de la figure 3.14(a), qui est de loin le plus courant. Dans cette situation, la meilleure chose à faire est probablement de prolonger F_3 en calculant son intersection avec soit F_1, soit F_2. Mais un problème se pose lorsque une ou plusieurs frontières n'ont pu être affectées de façon certaine à une région (frontières ambiguës). Se pose aussi le problème des points de multiplicité m supérieure à 3. On voit que le nombre de situations possible (3^m) explose. Il faudrait alors identifier les situations les plus typiques et définir un traitement pour chacune. Nous avons donc préféré un algorithme générique.

Finalement, l'algorithme de re-création des points multiples se décompose en deux parties :

1. Les intersections entre frontières sont d'abord détectées et donnent la localisation de nouveaux points multiples. Les frontières sont coupées en conséquence.
2. Ensuite, les frontières toujours déconnectées sont repositionnées sur les éventuels points multiples créés précédemment, ou sur des nouveaux points créés par un calcul de barycentre.

Les frontières sur le bord sont encore un cas particulier supplémentaire. Il n'est pas possible d'effectuer un repositionnement de leur extrémité sur un point multiple qui ne serait pas sur le bord. Lors d'un calcul de barycentre, on ne prend donc en compte dans la sommation que les deux frontières du bord. Ainsi le point multiple obtenu sera forcément sur le bord. De plus, le cas des coins de l'image doit être pris en compte. Si une extrémité à repositionner et le point cible ne sont pas situés sur le même bord, le repositionnement n'est plus direct. Il faut introduire autant de points intermédiaires que de coins rencontrés dans le parcours du bord.

Dans le domaine de la géométrie algorithmique, la précision du calcul des coordonnées de l'intersection de deux segments est un problème classique. Pour notre application, même en prenant des nombres réels en double précision, nous avons rencontrés des difficultés. Nous avons donc effectué tous ces calculs avec des nombres rationnels exacts. Ce choix a cependant un inconvénient potentiel : comme les coordonnées des points multiples sont reprises d'une image à l'autre, au bout de plusieurs images, il serait possible que les dénominateurs des coordonnées rationnelles prennent des valeurs très grandes. Nous évitons ce problème complètement en arrondissant ces coordonnées à l'entier le plus proche. Mais en fait, la vraie raison qui justifie l'arrondissement des coordonnées est le codage efficace de notre représentation (voir la section 4.3).

La complexité de cette étape est relativement faible car elle ne dépend pas du nombre de pixel, mais uniquement de la complexité de la scène et de la finesse de l'approximation polygonale des frontières. Elle est dominée par l'algorithme de recherche d'intersections entre frontières. Il s'agit d'un algorithme qui balaye le plan par une ligne horizontale [Bentley et Ottman 79], et nous avons utilisé la mise en œuvre dans [Mehlhorn et Näher 89]. Si on appelle n_{segm} le nombre de segments dans la représentation ($n_{segm} \sim n_{somm}$) et n_{inters} le nombre d'intersections, sa complexité est dans $O((n_{segm} + n_{inters}) \log n_{segm})$ [2]. Les autres opérations ont une complexité dans $O(n_{ptm})$.

2. Il existe un autre algorithme de complexité $O(n_{inters} + n_{segm} \log n_{segm})$ *en moyenne* [Myers 85]. Mais sa complexité dans le pire des cas est la même que celle de l'algorithme utilisé, et sa mise en œuvre est nettement plus difficile.

FIG. 3.15 – *Re-création des points multiples (*«Flower Garden»*, image 2). En haut : frontières ajustées et position antérieure des points multiples. En bas : points multiples recréés.*

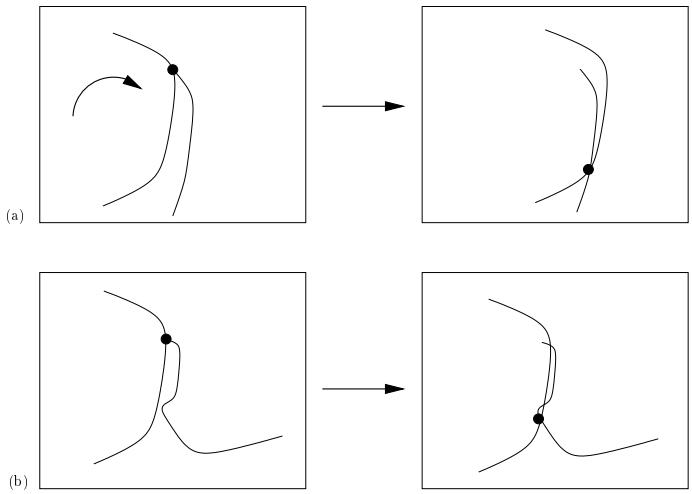


FIG. 3.16 – *Instabilité des points multiples au cours du temps.*

La figure 3.15 montre le résultat de l'opération pour le premier couple d'images de la séquence *«Flower Garden»*. Les autres résultats sont montrés dans la section 3.7. Une remarque générale sur les résultats obtenus est que l'on observe dans certains cas une instabilité de la position des points multiples. La figure 3.16(a) montre qu'un petit déplacement relatif de deux frontières peut engendrer un grand déplacement du point multiple correspondant, si la géométrie des frontières s'y prête. Il peut même de produire une discontinuité dans le mouvement du point multiple, comme l'illustre la figure 3.16(b). C'est en partie pour ces raisons que nous avons préféré suivre les frontières plutôt que les points multiples.

3.6.3 Ajustement local des sommets (ALS)

La $3^{\text{ème}}$ étape est une déformation quelconque de l'ensemble de la structure. Il s'agit du dernier raffinement dont le but est de prendre en compte des mouvements non rigides et de corriger un éventuel biais causé par la re-création des points multiples. On fait en sorte que cette déformation soit suffisamment petite pour ne pas changer la topologie de la structure de sorte qu'il est possible de tirer partie de la connaissance de cette topologie. Elle va s'effectuer par la minimisation d'une fonction d'énergie composée de 3 termes prenant en compte ces différents éléments. Pour cela, nous nous sommes inspirés de [Mardia et al. 92].

Soit S l'ensemble des sommets de l'approximation polygonale (à ne pas confondre avec X, l'ensemble des sommets du graphe; on a $X \subset S$). Pour chaque sommet $x_k \in S$, le raffinement est un vecteur de déplacement d_k à coordonnées entières. On appelle D la liste de tous les raffinements individuels : $D = (d_k)_{k \in S}$. De plus, les voisins de x_k sont notés $(x_l)_{l \in vois(k)}$. Un point multiple de multiplicité m a m voisins et un sommet d'une frontière a 2 voisins.

Comme pour l'ajustement affine, on définit un terme énergétique d'attache aux données $E_d(D)$ pour l'ensemble des déplacements D. Pour un segment $[x_k, x_l]$, on définit une énergie

de façon similaire à l'énergie des frontières :

$$E_d(x_k, x_l) = -\int_0^1 \mid t(\vec{\nabla} I_t[s.x_k + (1-s).x_l]) \,.\, \vec{N} \mid \mathrm{d}s$$

avec $\vec{N}$ un vecteur normal au segment $[x_k, x_l]$. Puis on tient compte des déplacements D et une somme est effectuée sur tous les sommets :

$$E_d(D) = \sum_{k \in S} \sum_{l \in vois(k)} E_d(x_k + d_k, x_l + d_l)$$

Chaque terme correspondant à un segment est en fait la valeur moyenne du gradient seuillé et projeté. Il n'y a pas de pondération dépendant de la longueur du segment. Nous avons par ailleurs fait des essais en pondérant chaque segment par sa longueur courante $d(x_k + d_k, x_l + d_l)$ ou par sa longueur originale $d(x_k, x_l)$. Le résultat est que les segments situés sur de forts gradients ont tendance à se rallonger au détriment des segments voisins situés sur des gradients plus faibles. Certaines frontières se transforment même en un simple segment. Avec la pondération par la longueur courante, l'effet apparaît dès la première image. La pondération par la longueur initiale du segment, freine un peu le phénomène, mais au bout de quelques images, le résultat est le même.

Ensuite on définit une énergie favorisant de petits déplacements par

$$E_p(D) = \lambda_p \sum_{k \in S} \rho_p(\|d_k\|)$$

avec λ_p un coefficient de pondération des termes énergétiques et ρ_p une fonction positive croissante. Il s'agit donc d'une pénalisation des grands déplacements. Le choix de ces paramètres sera explicité plus tard.

Enfin le dernier terme est une énergie de régularisation, qui pénalise des déplacements différents entre sommets voisins. Elle est définie par

$$E_r(D) = \lambda_r \sum_{k \in F} \sum_{l \in vois(k)} \rho_r(\|d_k - d_l\|)$$

avec de même que précédemment λ_r un coefficient de pondération des énergies et ρ_r une fonction positive croissante.

Finalement, l'ajustement local consiste à effectuer la minimisation suivante :

$$\widehat{D} = \arg\min_D E_d(D) + E_p(D) + E_r(D)$$

Telle que l'énergie a été écrite, chaque sommet contribue dans la sommation de $E_d(D)$ à un nombre de termes égal à sa multiplicité (2 ou plus). Chaque point multiple contribue à $m > 2$ termes, ce qui renforce son poids relatif par rapport aux simples sommets et contribue ainsi à régulariser la solution de façon structurelle. De plus, chaque terme est local à un sommet, mais leur somme traduit des contraintes globales à l'ensemble de la structure. Une modification de la position d'un sommet peut changer la position optimale

d'un autre sommet éloigné, pour peu qu'ils appartiennent à la même composante connexe du graphe G.

La minimisation de fonctions d'énergie définies sur des contours actifs peut se faire de nombreuses manières. Dans [Bascle 94], les variables sur lesquelles portent la minimisation sont continues et l'algorithme utilisé est une descente de gradient. Dans [Wu 95], il s'agit de polygones ajustables : des segments de droites sont ajustés de façon indépendante, puis leurs intersections sont calculées pour reformer un polygone. Dans notre cas, les variables $(d_k)_{k \in S}$ à minimiser sont discrètes, donc les algorithmes de type gradient sont difficilement applicables. Une méthode qui a été employée dans ce cas est la programmation dynamique [Amini et al. 90] [Geiger et al. 95]. Elle peut être appliquée à des contours ouverts, et elle a aussi été adaptée à des contours fermés. Cependant, dans notre cas où un graphe complexe de dépendances existe entre les variables à minimiser, son utilisation serait assez délicate. Un autre inconvénient est sa complexité : si on a n_{somm} variables à traiter et que pour chaque variable on envisage p possibilités d'ajustement, la complexité d'une itération de cet algorithme est $O(p^2.n_{somm})$. La plus petite valeur possible de p est le cas où l'on considère un 4-voisinage. On a alors $p = 5$. Autrement on peut avoir $p = 9$ ou même $p = 25$ pour un voisinage d'ordre supérieur. Nous avons donc préféré utiliser un algorithme de relaxation déterministe dont la complexité est plus faible.

Parmi les algorithmes de relaxation déterministes, nous avons préféré l'algorithme HCF [Chou et Brown 88] à l'algorithme ICM car il donne de meilleurs résultats pour une complexité à peine plus élevée. Son principe est d'effectuer la modification d'une variable d_k qui apporte la plus grande diminution d'énergie, alors que l'algorithme ICM considère les variables dans l'ordre de leurs indices k. Pour toutes les variables d_k, on définit un ensemble de valeurs à considérer, sur lequel on calcule les variations d'énergie. Nous avons choisi de considérer une fenêtre carrée, ce qui donne 9 déplacements élémentaires. Des déplacements plus grands sont alors obtenus en cumulant plusieurs déplacements élémentaires. Une possibilité pour réduire cet ensemble serait de s'inspirer des travaux de [Geiger et al. 95] qui font une recherche le long de la normale au contour. Dans notre cas, il s'agirait de ne considérer, pour les sommets non multiples, que des positions sur la bissectrice de l'angle formé par les deux segments se coupant en ce sommet. Les points multiples conserveraient, eux, une fenêtre de recherche. Cette réduction aurait un grand intérêt si l'on s'intéressait à de grands déplacements comme dans le cas de la segmentation spatiale initialisée manuellement. Mais dans notre cas, la norme des vecteurs d_k est censée être faible, donc la technique présente moins d'intérêt. De plus, il est difficile de définir une bissectrice de façon discrète pour un petit déplacement. Mentionnons là encore le cas particulier des sommets situés sur le bord de l'image. Leur seul déplacement autorisé est soit horizontal soit vertical.

La convergence de l'algorithme est assurée par la décroissance stricte de la fonction d'énergie. Cependant, nous avons introduit une petite modification à l'algorithme HCF original. Lorsque deux sommets se rapprochent et que leur distance descend en dessous d'un seuil convenu (1 à 3 pixels), nous les fusionnons. Ceci peut amener une augmentation de l'énergie. Il n'y a plus alors de décroissance stricte de l'énergie. L'algorithme converge tout de même car il n'y a que des diminutions d'énergie avec un nombre fini d'augmentations.

Intéressons nous maintenant au choix des paramètres λ_p et λ_r, et des fonctions ρ_p et ρ_r. Le réglage de ces paramètres est un problème classique dans le domaine des contours ac-

tifs. La plupart des auteurs les règlent de façon empirique, en fonction de la régularité des contours et du niveau de bruit dans l'image. D'autres travaux en cherchent une méthode automatique : par exemple, dans [Gennert et Yuille 88], c'est une technique de type minimax qui esquive le problème puisqu'elle n'a pas besoin d'un paramètre explicite, et dans [Shahraray et Anderson 89] c'est une méthode statistique de validation croisée. Dans notre problème, nous contournons la difficulté pour chaque terme d'énergie. L'énergie de régularisation E_r est là pour introduire une corrélation entre déplacements de sommets de la même frontière. Or l'étape précédente d'ajustement affine peut justement être vue comme une étape de décorrélation. Si la frontière avait besoin d'un mouvement d'ensemble pour son ajustement, celui-ci a déjà été effectué par l'ajustement affine. Donc le seul ajustement qui reste à effectuer est tel que les d_k sont décorrélés. Nous avons donc pris $\lambda_r = 0$. Quant à l'énergie E_p qui pénalise de petits déplacements, nous avons choisi une simple fonction échelon dépendant d'un seuil s_p :

$$\rho_p : \left| \begin{array}{l} \mathbb{R}^+ \longrightarrow \mathbb{R}^+ \\ \quad a \longmapsto \rho_p(a) = \begin{cases} 0 & \text{si } a < s_p \\ +\infty & \text{si } a > s_p \end{cases} \end{array} \right.$$

Ainsi nous autorisons n'importe quel ajustement, sans le biais généralement causé par la régularisation, tout en ayant un garde-fou contre les déplacements aberrants. De plus cela nous évite le problème du réglage de λ_p puisque sa valeur n'a plus d'influence. Il reste à fixer le seuil s_p que nous prenons entre 2 et 5 pixels en fonction de la taille des images et des mouvements non affines présents dans la séquence.

La complexité de cet algorithme est assez faible. Initialement il faut calculer pour chaque sommet $p - 1$ variations possibles d'énergie. Parmi ces variations, on retient la meilleure et, si elle est négative, on la met dans un tas ordonné (appelée aussi «pile» d'ordonnancement ou d'instabilité). Ceci nécessite au plus n_{somm} insertions. Grâce à la structure de tas de Fibonacci [Fredman et Tarjan 87], ceci peut être effectué en au plus $n_{somm} . \log n_{somm}$ opérations et grâce à une mise en œuvre efficace [Mehlhorn et Näher 89], cela est très rapide. Ensuite, à chaque modification d'une variable d_k, le tas peut être mis à jour de façon incrémentale : il faut recalculer les nouvelles variations d'énergie, enlever l'ancienne variation d'énergie qui était au sommet du tas et insérer la nouvelle variation la meilleure pour le sommet qui a bougé. Comme de plus en plus de sommets atteignent un minimum d'énergie, leurs variations possibles d'énergie sont positives. Ils disparaissent donc du tas, dont la taille diminue par conséquent. L'algorithme itère tant que le tas n'est pas vide. Le nombre de ces itérations n'a pas de majorant *a priori*, bien qu'il soit fini. Mais nous n'avons jamais observé un nombre d'itérations extrêmement important, résultat prévisible par le fait que l'initialisation des sommets est déjà très proche d'un minimum local. Contrairement aux autres phases du suivi temporel, cette étape ne peut pas se paralléliser trivialement comme les autres par un traitement parallèle de l'ensemble des régions ou des frontières. Mais on peut cependant noter que des algorithmes parallèles de relaxation existent [Mémin 93].

FIG. 3.17 – *Ajustement local des sommets (*«Flower Garden»*, image 2). En haut : avant ajustement. En bas : après ajustement.*

3.6.4 Comparaison avec une représentation par contours fermés

La plus grande différence entre la représentation par contours fermés et notre représentation par frontières ouvertes est l'ajustement. Dans [Wu et al. 95], chaque frontière est ajustée deux fois car elle appartient aux deux contours des deux régions adjacentes. Dans notre algorithme, chaque frontière est ajustée une fois seulement. Cela nous apporte évidemment un gain de complexité, mais a aussi une grande influence sur le traitement des zones d'occultation. On se reportera à la figure 3.18, qui est à comparer avec la figure 3.10.

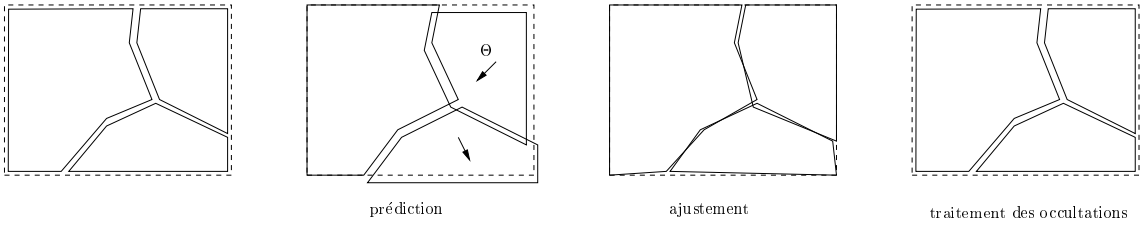


FIG. 3.18 – *Vue d'ensemble d'un suivi par contours fermés.*

Dans [Wu et al. 95], les zones d'occultation sont en fait des zones de non recouvrement ou des zones d'intersection (voir la section 1.3.4 pour la distinction avec les zones découvertes et recouvertes). Leur traitement a trois caractéristiques principales : il est effectué 1^o après la prédiction, 2^o au niveau des pixels et 3^o à chaque image. Sur ces trois points notre algorithme de suivi temporel apporte une amélioration :

1. Le traitement des occultations est fait avant la prédiction, grâce à l'attribution de chaque frontière à une région, ce qui permet de faire une prédiction «intelligente».
2. Il se fait au niveau des polygones, ce qui nous évite de convertir notre représentation en une carte d'étiquettes, et de perdre dans ce passage les informations de haut niveau qu'elle contient.
3. Pour la plupart des séquences il peut n'être fait qu'une fois pour toute la séquence.

À propos du 1^{er} point, notre algorithme fonctionne très bien pour une région en cours de recouvrement. Sa frontière va se raccourcir progressivement jusqu'à ce que la surface de la région devienne inférieure au seuil des petites régions. Elle est alors éliminée, ce qui est conforme à sa disparition derrière d'autres régions. Le cas d'un découvrement est plus complexe. Considérons une région R_d en cours de découvrement et une région R_r qui la recouvre partiellement. Notre algorithme fonctionne correctement tant que R_d reste en contact avec R_r. Mais si après plusieurs images ces régions se séparent, notre méthode n'est plus valable. Pour détecter cette situation, il faudrait détecter l'apparition d'une $3^{ème}$ région en dessous ou entre les deux premières. Nous ne le faisons pas, mais pour cela, on pourrait vérifier l'EQM calculée par l'estimation de mouvement de la texture de R_d. Si elle augmente suffisamment, on pourrait recalculer une segmentation spatiale de R_d pour détecter les régions nouvellement apparues dans l'espace entre R_d et R_r. On réaliserait tout de même un gain de complexité puisque cette situation ne se produit que dans une image de la séquence alors que les cas où notre suivi marche se produisent dans toutes les autres images. Mais le gain le plus important apparaît dans les résultats. Sur les séquences

comportant des mouvements de grande amplitude, on observe ce que [Wu et al. 95] appelle des «zones de découvrement épaisses». Ces zones sont segmentées spatialement, ce qui crée un grand nombre de nouvelles régions, et ce à chaque image. Ainsi la traînée de ces objets est fortement sur-segmentée à cause de toutes ces nouvelles régions. Quant à nous, grâce à notre hypothèse de cohérence temporelle, toutes ces zones découvertes sont automatiquement affectées aux régions du dessous. Le meilleur exemple en est la séquence *«Flower Garden»*.

À propos du 3$^{\text{ème}}$ point, dans [Wu 95], paragraphe 5.4.2.1, il est question d'une information de superposition entre régions. Cette information est acquise par un test sur une seule image au début de la séquence. Bien que cela ne soit pas fait, cette information pourrait servir au traitement des recouvrements avant l'étape d'ajustement, un peu comme dans notre algorithme. Même si ces remarques peuvent sembler similaires à notre algorithme, deux différences importantes subsistent :

- Cette information de superposition ne sert pas pour le traitement des découvrements. Il est donc nécessaire de les traiter à chaque image.
- L'affectation de frontière à une région est plus générale dans le cas où deux régions ont plusieurs frontières en commun (voir la figure 3.5).

Mais de toute façon, dans le cadre d'une RCF, on ne peut pas traiter les occultations et découvrements avant l'ajustement, car celui-ci crée de nouvelles intersections et des zones non recouvertes à cause de la duplication des frontières.

3.7 Résultats

3.7.1 EQM de compensation du mouvement

Les figures 3.19 et 3.20 montrent l'EQM de compensation de mouvement obtenue par notre algorithme sur les séquences *«Interview»* et *«Tennis»*. Sur ces séquences, le nombre de régions n'est pas suffisant pour obtenir une meilleure EQM qu'avec le *block-matching*. Pour les autres séquences, notre algorithme donne de meilleurs résultats d'EQM (voir la partie 4, figure 4.15).

3.7.2 Prédiction des frontières

Les figures 3.21 à 3.23 montrent la prédiction de la segmentation sur le premier couple d'images des séquences de test.

3.7.3 Ajustement des frontières

Les figures 3.24 à 3.27 montrent l'ajustement de la segmentation sur le premier couple d'images des séquences de test.

La figure 3.28 montre un exemple supplémentaire intéressant tiré de la séquence *«Flower Garden»*, où l'ajustement permet de rattraper une erreur importante située sur la branche à droite de l'arbre du premier plan.

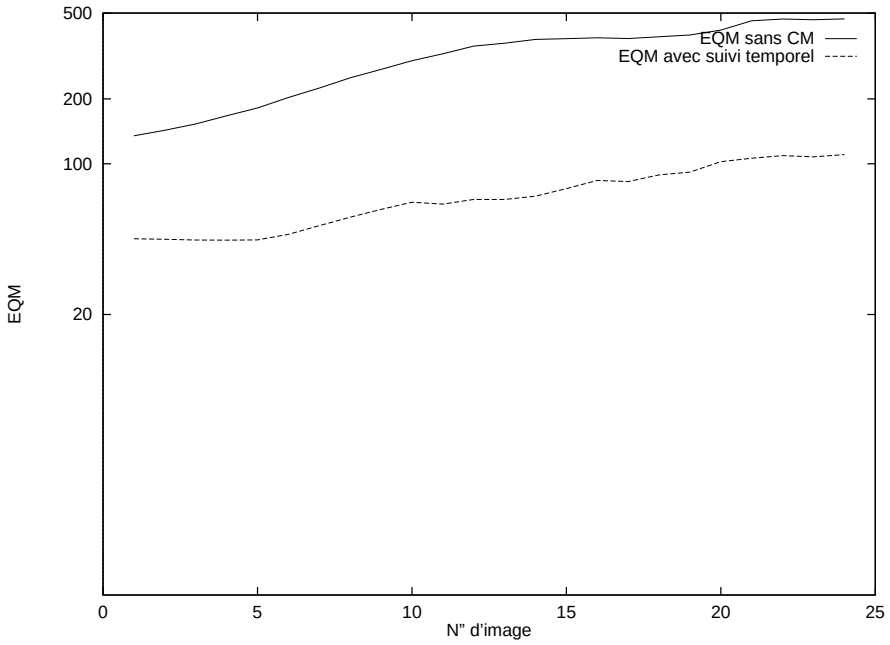


FIG. 3.19 – EQM *de prédiction par compensation de mouvement des images de la séquence* «Interview»*, avec notre algorithme de suivi temporel.*

3.7.4 Suivi sur la séquence entière

Les figures 3.29 à 3.33, montrent le résultat du suivi temporel sur les séquences entières. Pour la séquence *«Interview»*, nous montrons deux résultats, obtenus par des initialisations différentes du suivi sur la première image.

L'observation principale à faire sur ces résultats est la grande stabilité temporelle des segmentations obtenues. C'est particulièrement vrai pour la séquence *«Miss America»*, qui est très longue (73 images) et dans laquelle le mouvement est tantôt très faible, tantôt relativement important. Sur cette séquence, les régions sont globalement conservées. On note juste une forte approximation polygonale des frontières sur les contours très peu visibles entre les cheveux et le fond, alors que, par exemple, l'encolure reste parfaitement détaillée.

Sur la séquence *«Interview»*, on remarque le suivi correct du mouvement articulé de la personne, grâce à sa segmentation en plusieurs régions et à l'articulation des frontières qui en résulte. Par contre, le bras qui se découvre n'est pas segmenté, car initialement, le bouquet le fleurs qui le recouvre partiellement ne l'est pas non plus.

La séquence *«Flower Garden»* montre un exemple extrême d'occultation puisqu'en traversant toute le cadre de la caméra, l'arbre recouvre presque entièrement la moitié gauche du fond et découvre une grande partie de la moitié droite.

Les résultats sur la séquence *«Tennis»* montrent le comportement de l'algorithme sur un mouvement de grande amplitude. En effet, la balle se déplace de 15 pixels dès la première

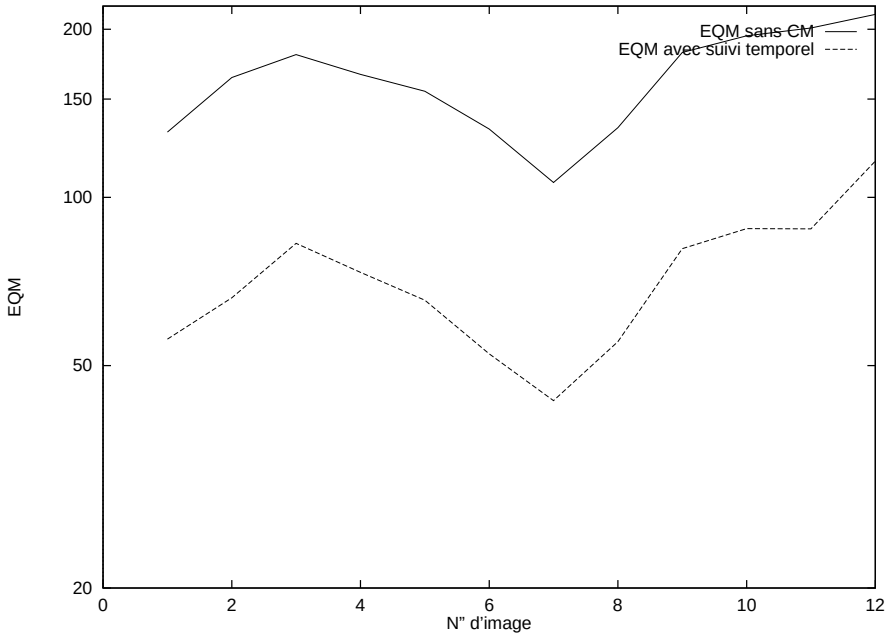


FIG. 3.20 – EQM *de prédiction par compensation de mouvement des images de la séquence* «Tennis»*, avec notre algorithme de suivi temporel.*

image, puis son mouvement s'annule en haut de sa trajectoire, et repart rapidement vers le bas. Ici, c'est le suivi par hypothèses multiples qui permet de détecter l'annulation du mouvement. Alors qu'un suivi avec une seule prédiction ne marche pas toujours, l'ajout du prédicteur par mouvement nul a permis une détection à coup sûr. Cette séquence montre aussi la disparition d'un objet (le bras gauche du joueur) qui sort du cadre de la caméra.

FIG. 3.21 – *Prédiction des frontières (*«Miss America»*, image 1 ⟶ 2)*

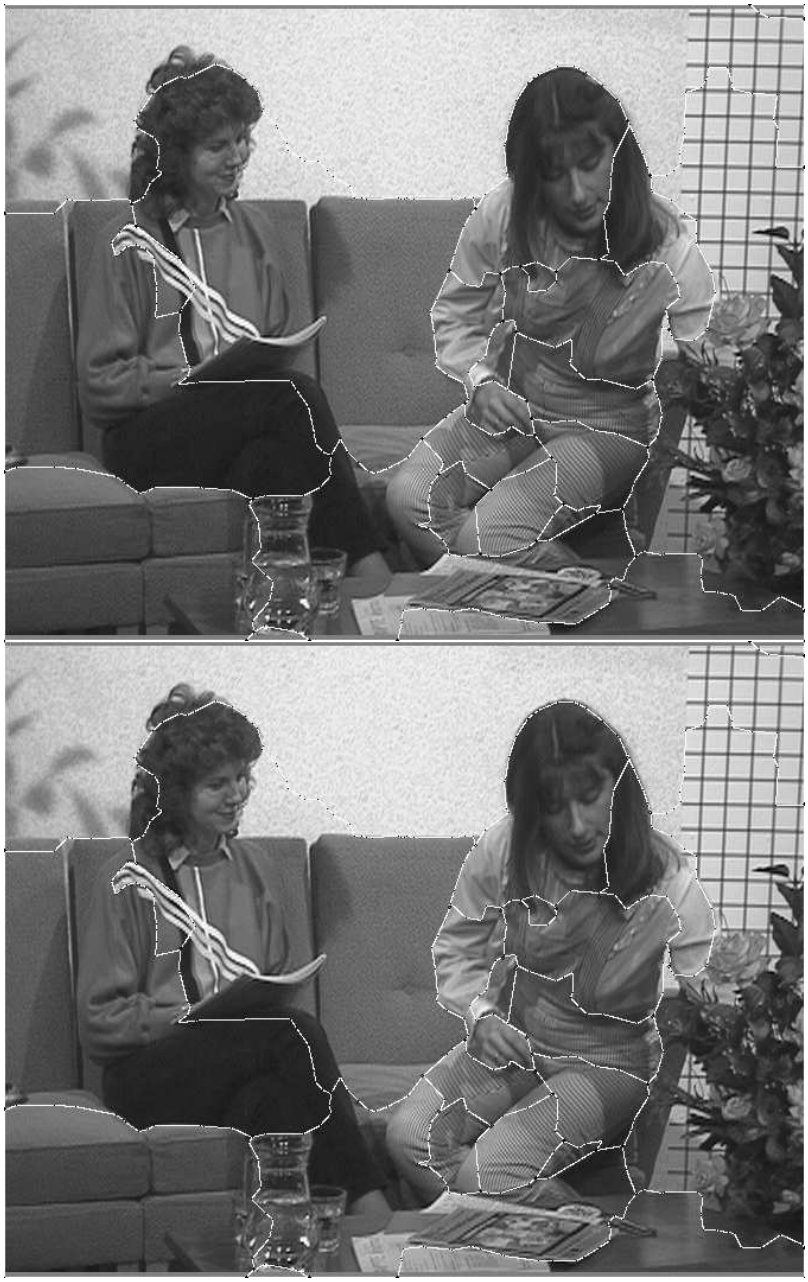


FIG. 3.22 – *Prédiction des frontières (*«Interview»*, image 1 ⟶ 2)*

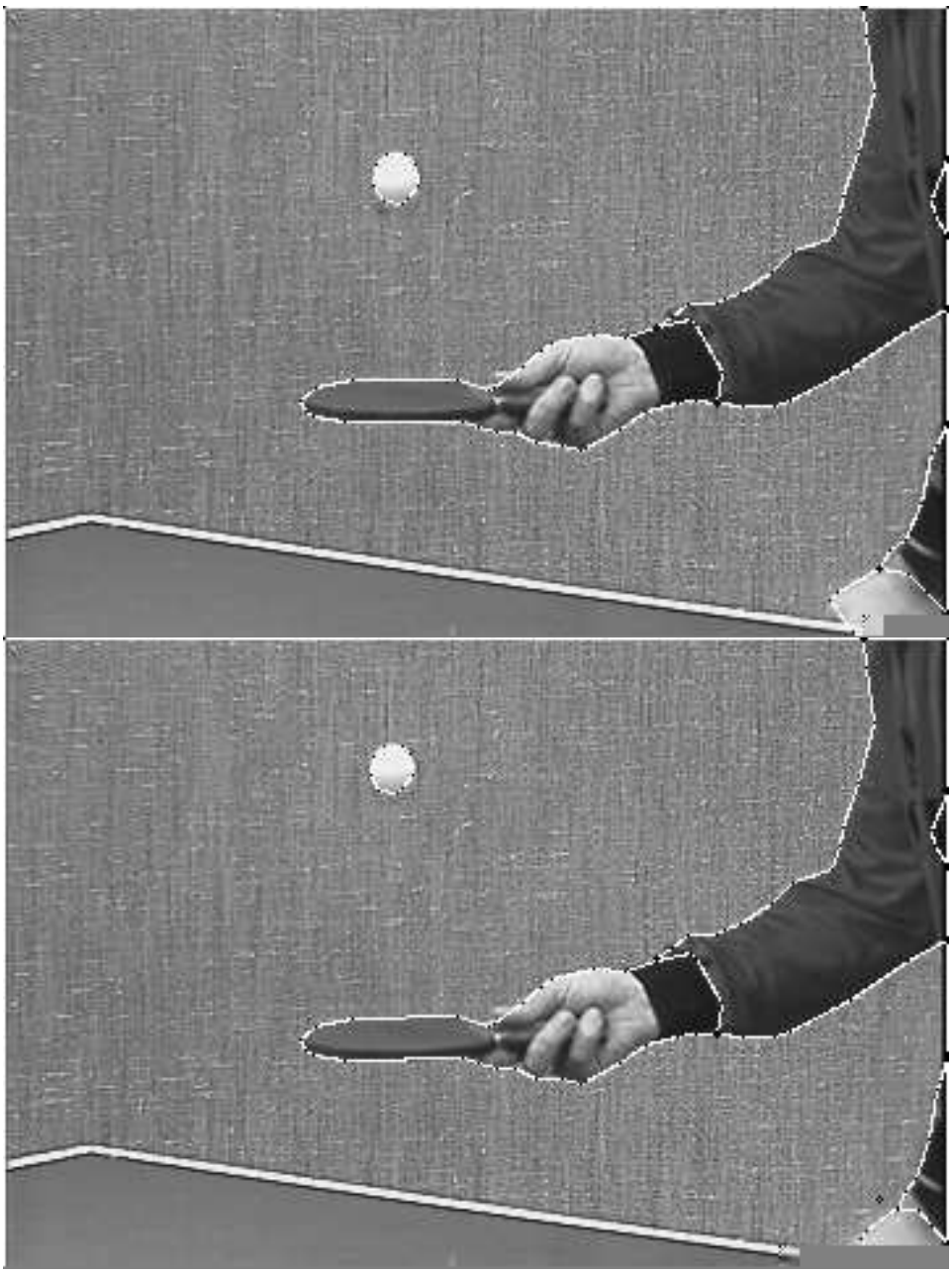


FIG. 3.23 – *Prédiction des frontières (*«Tennis»*, image 1 —→ 2)*

FIG. 3.24 – *Ajustement des frontières, séquence* «Miss America»*, image 2. Dans l'ordre, ajustement affine et ajustement local.*

FIG. 3.25 – *Ajustement des frontières, séquence* «Interview»*, image 2. Dans l'ordre, ajustement affine et ajustement local.*

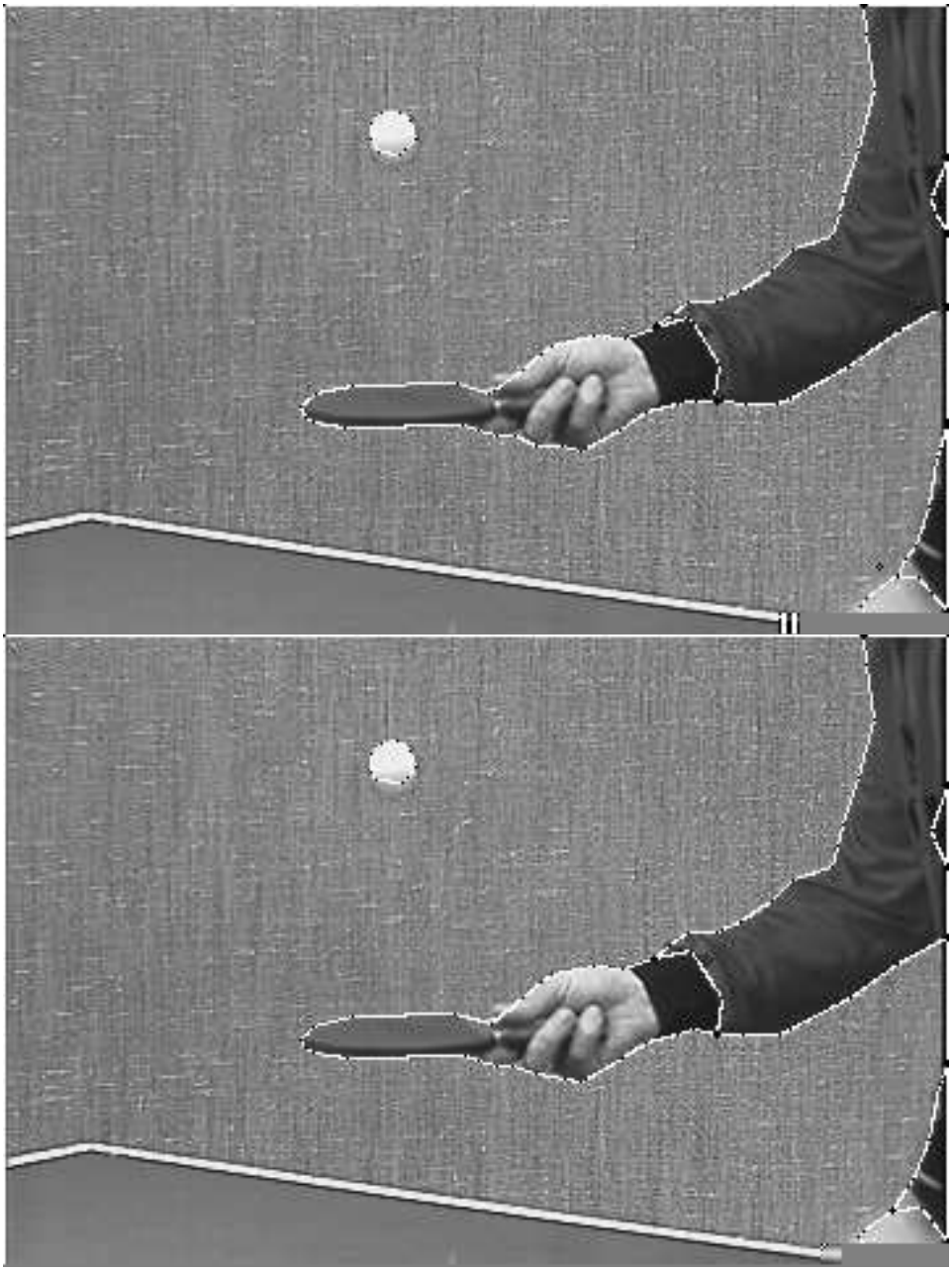


FIG. 3.26 – *Ajustement des frontières, séquence* «Tennis»*, image 2. Dans l'ordre, ajustement affine et ajustement local.*

FIG. 3.27 – *Ajustement des frontières, séquence* «Flower Garden»*, image 2. Dans l'ordre, ajustement affine et ajustement local.*

FIG. 3.28 – *Ajustement des frontières, séquence* «Flower Garden», *image 24. Dans l'ordre, prédiction, ajustement affine et ajustement local.*

FIG. 3.29 – *Suivi sur la séquence entière (*«Miss America»*, image 1 ⟶ 73). Une image sur 5 est montrée. Il y a 33 régions initialement.*

FIG. 3.30 – *Suivi sur la séquence entière (*«Interview»*, image 1 —→ 17). Une image sur 3 est montrée. Il y a 22 régions initialement.*

FIG. 3.31 – *Suivi sur la séquence entière (*«Interview»*, image 1 —→ 17). Une image sur 3 est montrée. La segmentation spatiale initiale est différente. Il y a 21 régions initialement.*

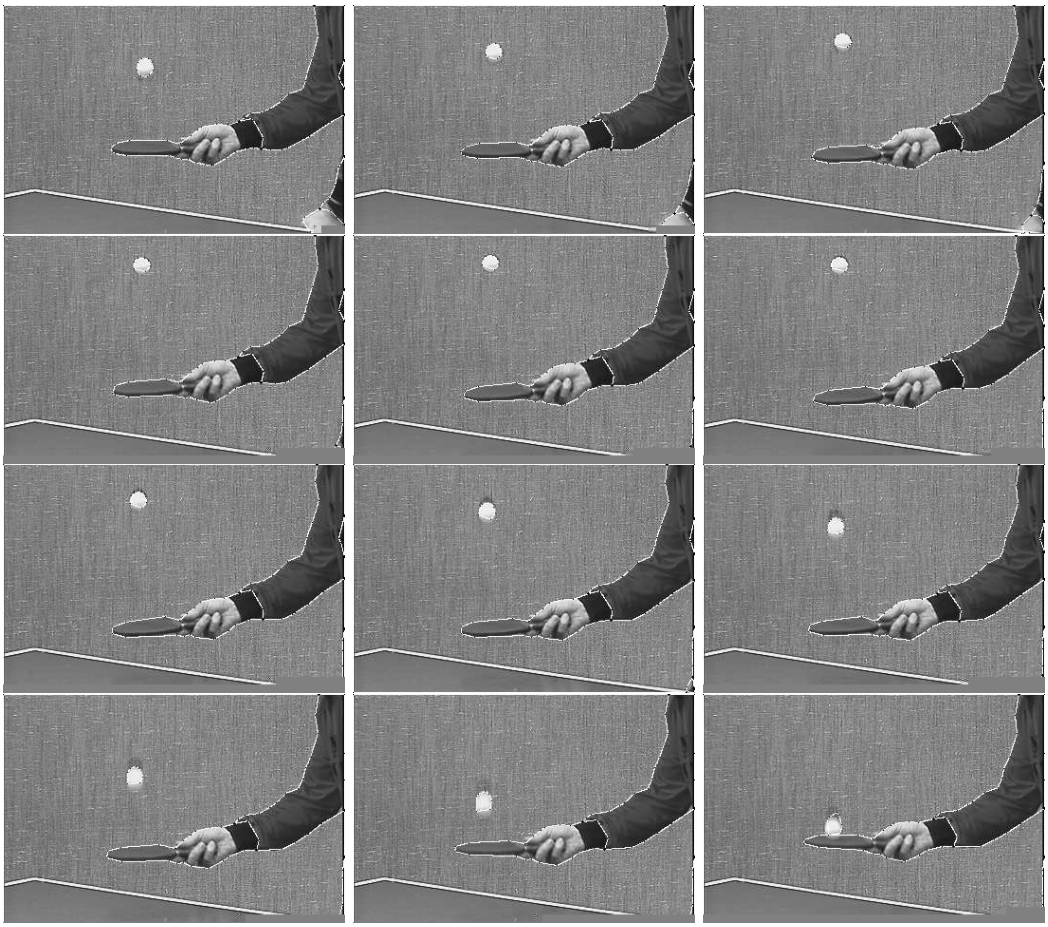


FIG. 3.32 – *Suivi sur la séquence entière (*«Tennis»*, image 1 —→ 12). Toutes les images sont montrées. Il y a 7 régions initialement.*

FIG. 3.33 – *Suivi sur la séquence entière (*«Flower Garden»*, image 1 ⟶ 29). Une image sur 2 est montrée. Il y a 10 régions initialement.*

3.8 Conclusion partielle

Dans ce chapitre, nous avons fait quelques hypothèses sur les séquences traitées, en vu de pouvoir développer un algorithme de suivi temporel de segmentation garantissant une grande stabilité temporelle des partitions obtenues. Dans la version actuelle de l'algorithme, la plus forte hypothèse est la stabilité du graphe de la représentation au cours de la séquence. Il est possible de lever cette hypothèse, mais cela risquerait de compromettre la stabilité temporelle.

Les points multiples étant trop instables au cours du temps, le suivi temporel opère sur les frontières des régions. Nous avons comparé deux algorithmes de suivi utilisant le principe de prédiction/ajustement. Tous deux estiment d'une part les mouvements de la texture à l'intérieur des régions et d'autre part le mouvement des frontières des régions, et font coopérer les deux estimations. Mais ils diffèrent dans l'ordre d'enchaînement des étapes d'estimation. Dans cette thèse, nous avons préféré l'algorithme de suivi en mode avant, plus adapté à une application pour l'interpolation, plutôt que le suivi en mode arrière que nous avions testé dans des travaux antérieurs.

Pour la prédiction du mouvement des régions, l'algorithme de suivi en mode avant, nous a permis de nous limiter à un choix *a posteriori* entre deux prédicteurs, ce qui réduit la complexité par rapport au suivi en mode arrière. La prédiction des frontières, effectuée grâce au mouvement des régions a nécessité le développement de deux critères permettant d'affecter une frontière à une région et de déterminer ainsi un ordre partiel de superposition des régions. Une étape de re-création des points multiples permet de rendre à la représentation sa cohérence qu'elle avait perdu après l'application aux frontières du mouvement des régions. Cette étape est effectuée sur la base de traitements purement géométriques appliqués aux frontières déconnectées. L'ajustement des frontières, basé sur la technique générale des contours actifs et donc sur les gradients spatiaux de l'image, est effectué en deux étapes. Un premier ajustement affine global sur chaque frontière permet un recalage grossier sur les contours des objets. Un deuxième ajustement local, qui utilise les spécificités de notre représentation pour régulariser la solution obtenue, permet un dernier recalage précis.

Une comparaison de notre algorithme avec un algorithme opérant sur une représentation par contours fermés a montré un gain de complexité d'un facteur 2 pour l'étape d'ajustement des frontières et un meilleur traitement des occultations, si les hypothèses que nous avons faites sont vérifiées. En particulier, les découvrements sont traités sans création artificielle de nouvelles régions dans les zones découvertes.

Les résultats obtenus confirment la stabilité temporelle de notre algorithme, même sur des séquences très longues. Mais le point le plus délicat est la sensibilité à la segmentation initiale. Si la première image est sur-segmentée, les régions sont trop petites et les frontières trop proches, ce qui amène une relative instabilité temporelle dans les premières images, où régions et frontières fusionnent. Si elle est sous-segmentée, l'une de nos hypothèses n'est plus vérifiée et l'on obtient une mauvaise prédiction des frontières.

Chapitre 4

Interpolation temporelle

Introduction

Types d'interpolation et leurs applications

L'interpolation dans son sens le plus général est la reconstruction d'échantillons manquants au sein d'échantillons observés. Il peut s'agir d'**interpolation spatiale** lorsque l'on cherche à augmenter la résolution spatiale d'une image, ou à obtenir une valeur de niveau de gris en une position quelconque entre les pixels d'une image. Il peut aussi s'agir d'interpolation de vues 3D [Faugeras et Laveau 94] [Blanc et Mohr 97] où l'on cherche à produire l'image d'une scène tridimensionnelle, comme si elle était observée depuis un point de vue intermédiaire fictif, à partir de deux ou plusieurs vues prises sous des angles différents. Mais dans notre étude, il s'agit plutôt d'une **interpolation temporelle** au sein d'une séquence d'images. Nous verrons en fait que ces types d'interpolation sont liés puisque l'interpolation temporelle doit souvent faire appel à une interpolation spatiale.

Avec des algorithmes similaires, nous nous intéresserons à deux problèmes distincts. D'une part l'**interpolation pure** telle qu'elle a été définie ci-dessus, c'est-à-dire dans le but de créer des images non observées à partir des images précédentes et suivantes. D'autre part le **codage interpolatif**, c'est-à-dire des techniques de codage utilisant l'interpolation temporelle comme un moyen d'obtenir une prédiction de l'image, dont la différence avec l'image originale sera ensuite encodée.

Une autre façon de distinguer les différents types d'interpolation est de considérer la localisation du dispositif d'interpolation dans une chaîne de compression – transmission – décompression. L'interpolation peut avoir lieu avant ou après la transmission c'est-à-dire dans le dispositif de codage (**avant la compression**) ou celui de décodage (**après la décompression**).

Si l'on regarde maintenant les applications de l'interpolation temporelle, on peut faire un tableau récapitulatif qui résume le type d'interpolation à utiliser et l'endroit où elle doit intervenir (voir ci-dessous). Nous avons considéré trois applications possibles. D'abord la **conversion de standards** où il s'agit de convertir entre standards vidéo de fréquences d'échantillonnage différentes. Ensuite la **compression** de séquences d'images où l'interpolation intervient comme une prédiction bidirectionnelle. Enfin la **diffusion multipoints**

où la fréquence d'affichage du terminal de restitution est différente de la fréquence des images transmises.

Application	Type d'interpolation	Position de l'interpolation
conversion de standards	interpolation pure	avant compression
compression	codage interpolatif	après décompression
diffusion multipoints	interpolation pure	après décompression

Utilité de l'interpolation

La conversion de standards reste toujours très utilisée. Les standards analogiques ont des fréquences d'affichage différentes, par exemple SECAM/PAL (50Hz) et NTSC (60Hz). Les standards numériques actuels (MPEG2, MPEG4, H261, H263) n'ont pas arrangé les choses, même si tous les acteurs du domaine se sont mis d'accord pour les utiliser. En effet, ces standards ne spécifient que les algorithmes de décompression, mais pas le format, c'est à dire les résolutions spatiales et temporelles du flux vidéo.

Même en cas d'accord sur les formats, par exemple autour de futures normes de TVHD, les terminaux multimédia de restitution sont de natures multiples : les différents récepteurs de télévision, les écrans d'ordinateur, les terminaux portables... L'affichage d'une même séquence d'images numériques, diffusée en mode multipoints, sur des terminaux ayant des fréquences de rafraîchissement différentes nécessite une interpolation temporelle.

Quant à la compression, il a été démontré par MPEG que le codage interpolatif, par l'utilisation de B-*frames*, apporte une amélioration du compromis débit/distortion par rapport à un codage purement causal (utilisation exclusive de I- et P-*frames*).

Schémas d'interpolation proposés

Pour la conversion de standards, des dispositifs industriels performants existent depuis des années. Il s'agit de machines spécialisées effectuant une estimation en temps réel d'un champ dense de mouvement, en prenant en compte les occultations (voir la section 4.1). Le mouvement étant estimé sur les images originales, avant compression, il est potentiellement d'assez bonne qualité pour assurer une interpolation satisfaisante. Ces machines sont probablement perfectibles, mais la recherche dans ce domaine est moins prometteuse que pour les deux autres applications de l'interpolation temporelle. Par ailleurs, cette technique présente un défaut inhérent qui est la transmission d'images redondantes si la fréquence vidéo en sortie est supérieure à la fréquence en entrée.

Nous nous sommes donc particulièrement intéressés aux deux autres applications. Or ce qui les caractérise est l'interpolation dans le décodeur, après transmission. Une interpolation temporelle de qualité nécessite le «vrai» mouvement observé dans la séquence, dont l'obtention est coûteuse en temps de calcul. Nous proposons donc un schéma d'interpolation n'utilisant que les informations de mouvement transmises par le codeur, sans avoir à les calculer dans le décodeur. Ainsi, le décodeur serait d'un coût plus faible qu'un décodeur devant effectuer l'estimation de mouvement lui-même. Le codeur resterait de complexité élevée, mais son coût est divisé par le nombre de décodeurs recevant la transmission. De

plus, même si le coût n'avait pas d'importance, il resterait toujours un argument technique en faveur du schéma proposé. L'estimation de mouvement sur les images décodées, n'est pas très fiable à cause des dégradations causées par la compression. Le phénomène est d'autant plus marqué que le système fonctionne à bas débit. Il est donc préférable d'effectuer l'estimation de mouvement sur l'image originale, au niveau du codeur.

Dans le chapitre précédent nous avons vu un algorithme de suivi temporel de segmentation spatio-temporelle d'objets ayant un mouvement uniforme. Nous allons donc utiliser cette modélisation du mouvement et la segmentation obtenue à des fins de codage interpolatif dans un schéma de codage par régions, et si besoin est, pour l'interpolation pure au niveau du décodeur. Nous utiliserons à la fois les **supports des objets** (partie géométrique) et les **mouvements paramétriques** estimés sur ces objets (partie cinématique).

Plan du chapitre

Dans une première section, nous ferons une brève revue bibliographique de quelques algorithmes d'interpolation temporelle, en les classant selon la modélisation du mouvement sur laquelle ils reposent.

La deuxième section portera sur la technique de base qui nous servira dans nos algorithmes d'interpolation, à savoir la compensation de mouvement bidirectionnelle basée régions. Nous y présenterons un estimateur de mouvement spécialisé, conçu spécialement pour l'interpolation basée régions. Nous y aborderons aussi le problème du traitement des occultations, et nous y étudierons le gain de codage attendu par cette méthode d'interpolation.

Ensuite, dans la troisième section, nous décrirons les techniques de codages utilisées pour les différentes informations transmises dans notre schéma de codage, à savoir la segmentation, les mouvements et l'image d'erreur. Nous détaillerons particulièrement la partie concernant la segmentation puisqu'elle est spécifique à la représentation que nous avons adoptée, qui se prête particulièrement bien à un codage prédictif efficace.

Enfin, dans la quatrième section, nous présenterons trois modes de codage interpolatifs possibles, selon la nature des informations que l'on veut transmettre entre le codeur et le décodeur. L'interpolation pure est alors un cas particulier de l'un de ces modes.

4.1 Approches existantes

La technique d'«interpolation» temporelle la plus simple est la répétition d'images. Elle est encore utilisée pour des conversions de standards 50 Hz $\longleftrightarrow$ 60 Hz. Le résultat est bien évidemment saccadé. Une variante consiste à effectuer une interpolation linéaire entre images de référence, sans compensation de mouvement. Le résultat donne alors un flou de bougé.

Il est donc indispensable d'effectuer une compensation de mouvement pour obtenir des résultats satisfaisants. Une technique connexe, mais d'applications différentes est le *morphing* [Ruprecht et Muller 95] [Lee et al. 96]. Il ne s'agit plus de mouvement mais de déformations d'un objet en un autre. Elles sont obtenues à partir d'un champ dense défini par des points singuliers ou de lignes, et ce manuellement de façon interactive.

Pour revenir au domaine qui nous intéresse, l'interpolation est donc relative à une modélisation du mouvement. Pour chaque modélisation, nous citons quelques articles de référence.

Champ dense : Ce modèle est utilisé pour la conversion de standards [Konrad 88] et pour le désentrelacement [Depommier et Dubois 92], effectués en studio, avant transmission. Dans [Huang et Mersereau 94] et [Tom et Katsaggelos 95] le mouvement est une translation sub-pixellique et l'interpolation est vue comme un processus de restauration. Dans le même ordre d'idées, [Kokaram et Godsill 97] [Kawaguchi et Mitra 97] utilisent des modèles AR tridimensionnels.

Blocs : Les normes MPEG1 et MPEG2 [Le Gall 91, Le Gall 92] utilisent l'interpolation par blocs pour la prédiction des B-*frames*. D'autres travaux utilisent aussi cette modélisation, par exmple [Bergeron et Dubois 90, Bergeron et Dubois 91], [Kim et Park 92] et [Richardson et al. 96] pour le post-processing.

Maillage : Les travaux suivants interpolent une image entière avec un maillage, ce qui ne permet pas de traiter les occultations : [Huang et Hsu 94] [Dudon et al. 95] [Dudon 96]. Par contre, dans [Hsu et Liu 97], les occultations sont prises en compte par la détection des mailles dans lesquelles plusieurs mouvements sont présents.

Parmi tous les autres travaux que nous avons recensés, aucun n'utilise une modélisation du mouvement par régions : [Bierling et Thoma 86] [Thoma et Bierling 89] [Cafforio et al. 90] [Cafforio et al. 90] [Tubaro et Rocca 93] [Robert 92] [RC et Sklansky 93] [Puri et Haskell 92] [Pearlman et Abdel-Malek 92] [Mori et al. 91] [Yang et al. 90] [Kharathisvili et al. 92] [Hobson et Carmen 91] [Hanen et Barba 93] [Gupta et Gersho 92] [Xie et al. 95] [Lagendijk et Sezan 92].

Les seuls travaux utilisant cette modélisation sont les nôtres [Bonnaud et al. 95], suivis par ceux-ci : [Han et Woods 97] [Fan et Gan 98].

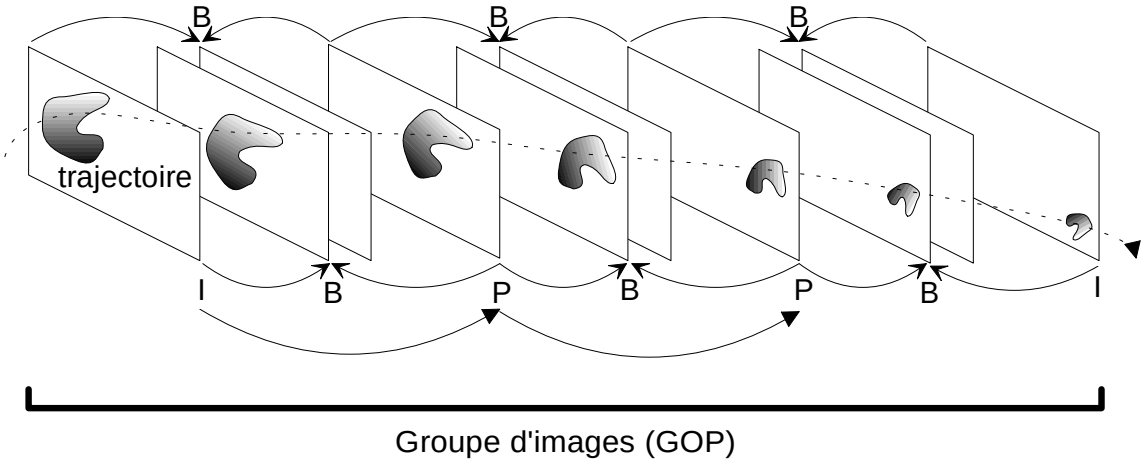


FIG. 4.1 – *Images de type I, P ou B et trajectoire d'un objet. Les flèches représentent la relation «sert pour la prédiction de» entre deux images.*

4.2 Compensation de mouvement bidirectionnelle basée sur les objets

Une nouveauté de notre approche par rapport à celles décrites dans la section précédente est que nous travaillons sur une modélisation du mouvement par **régions**. Cela va nous permettre d'effectuer une interpolation temporelle au niveau du décodeur, dans le cadre d'un schéma de codage par régions. En fait, on peut aller plus loin et dire que notre approche travaille sur les **objets** de la séquence, ce mot ayant la signification décrite ci-dessous.

Parmi les approches précédentes, la plupart travaillent sur un seul couple d'images. Les images interpolées entre deux images de références le sont en utilisant le mouvement entre ce couple d'images. Comme le mouvement est complètement réinitialisé à chaque instant, l'interpolation obtenue n'est pas forcément visuellement stable dans le temps. Au contraire, notre approche utilise la propriété de stabilité temporelle de la segmentation obtenue par notre algorithme. Il ne s'agit plus seulement de régions mais d'objets vidéo que l'on retrouve dans tout le GOP. Leur intersection avec les plans image forme une sorte de «tube», comme cela est illustré sur la figure 4.1. Ainsi, la stabilité visuelle de l'interpolation a plus de chances d'être assurée au sein du GOP.

Le choix des images du début et de fin du GOP (I-*frames*) ne doit pas être fait au hasard. La première image doit être celle qui suit immédiatement la détection d'un changement de plan (*cut*). La dernière image doit soit appartenir au même plan si sa longueur est supérieure à celle du GOP, soit être la dernière du plan. Dans ce cas, le GOP sera plus court que le GOP de base.

Ensuite on pourrait n'utiliser que des B-*frames*, comme le proposent [Kiranyaz et Onural 97]. Le schéma de prédiction prend alors la forme d'un arbre binaire. Une image de type B est d'abord interpolée entre les 2 images de type I. Puis cette image B sert à son tour d'image de référence pour prédire d'autres images intermédiaires par interpolation, et ainsi de suite récursivement. L'avantage est que l'on obtient une plus grande efficacité de codage qu'avec des images P. Mais l'inconvénient est que la structure est moins souple : les GOP ne peuvent avoir pour taille que $(2^n - 1)_{n \in \mathbb{N}}$. Si l'on intercale des images P, on peut retrouver l'une des structures classiques de GOP avec 3 images B consécutives, mais celle avec 2 images B consécutives.

Pour notre étude, nous avons donc conservé la structure de GOP classique de MPEG, comprenant aussi des images de type P. Mais notre étude porte principalement sur les images de type B entre 2 images de référence de type I ou P. Les structures de GOP testées dans la suite de l'étude sont répertoriées dans le tableau 4.1.

désignation du GOP	Numéro d'image												
	1	2	3	4	5	6	7	8	9	10	11	12	13
IP	I	P	P	P	P	P	P	P	P	P	P	P	I
1B	I	B	P	B	P	B	P	B	P	B	P	B	I
2B	I	B	B	P	B	B	P	B	B	P	B	B	I
3B	I	B	B	B	P	B	B	B	P	B	B	B	I

TAB. 4.1 – *Structures de* GOP *testées.*

4.2.1 Initialisation des descripteurs de mouvement

L'estimation de mouvement que nous allons effectuer entre l'image I_t et les images de référence I_{t_1} et I_{t_2} nécessite une initialisation. Pour cela nous allons reconstituer la trajectoire de chaque objet à partir de ses mouvements élémentaires entre images successives. Ces mouvements sont ceux estimés lors de la phase de suivi : il s'agit des mouvements issus de l'estimateur COD qui sont de la forme $\widehat{\Theta}^{+R,t|t}_{t-1\to t,\mathrm{COD}}$. Pour alléger l'écriture, nous les noterons simplement $\Theta^{+R}_{t-1\to t}$.

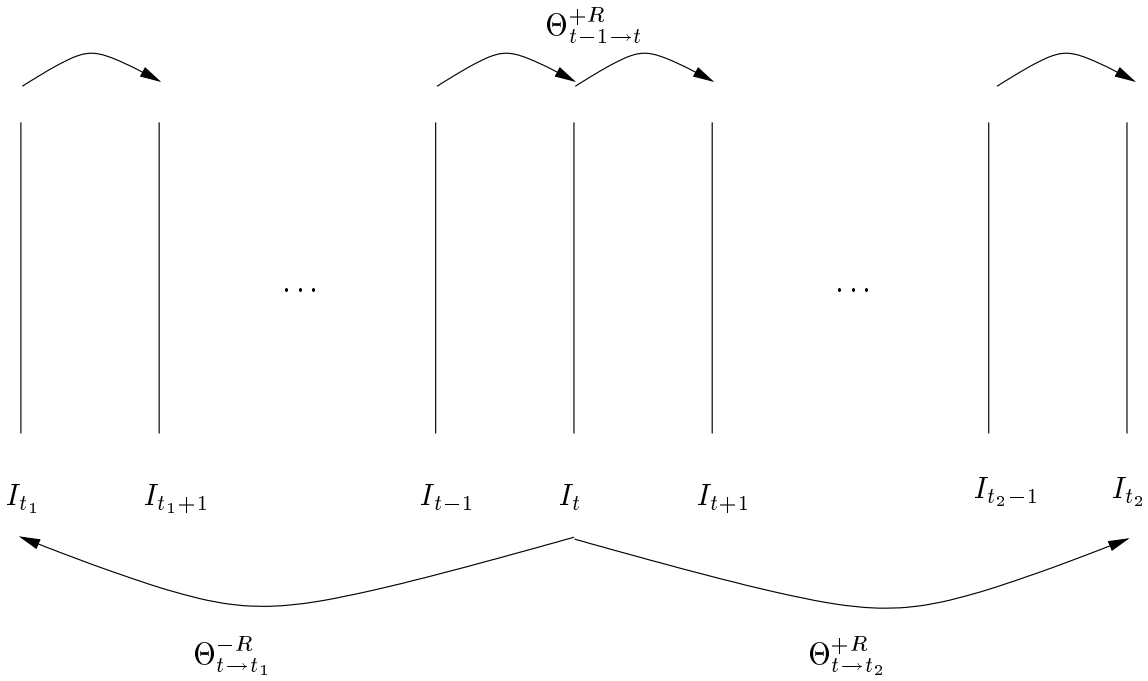


FIG. 4.2 – *Initialisation des descripteurs de mouvement.*

Pour chaque région R de l'image I_t l'initialisation du mouvement vers l'image I_{t_2} est calculée ainsi :

$$\Theta^{+R}_{t\to t_2,\mathrm{INIT}} = \Theta^{+R}_{t_2-1\to t_2} \circ \Theta^{+R}_{t_2-2\to t_2-1} \circ \ldots \circ \Theta^{+R}_{t+1\to t+2} \circ \Theta^{+R}_{t\to t+1}$$

L'opérateur $\circ$ désigne la composition entre descripteurs de mouvement (voir l'annexe B). Pour pouvoir composer deux mouvements, il faut qu'ils soient du même type. Deux mouvements affines donnent un mouvement affine, de même pour les mouvements affines simplifiés ou translationnels. Bien que ce ne soit pas le cas dans notre étude, on peut imaginer que l'algorithme de suivi temporel de segmentation a pu sélectionner le modèle de mouvement optimal entre chaque couple d'images consécutives, comme cela est fait dans [Nicolas 92].

Pour l'initialisation du mouvement vers l'image I_{t_1}, il est nécessaire d'inverser le sens

des descripteurs. Pour cela, on pose

$$\Theta^{-R}_{t\to t-1} = [\Theta^{+R}_{t-1\to t}]^{-1}$$

et le calcul est le suivant :

$$\Theta^{-R}_{t\to t_1,\text{INIT}} = \Theta^{-R}_{t_1+1\to t_1} \circ \Theta^{-R}_{t_1+2\to t_1+1} \circ \ldots \circ \Theta^{-R}_{t-1\to t-2} \circ \Theta^{-R}_{t\to t-1}$$

4.2.2 Estimation du mouvement sur 3 images

Pour effectuer une compensation de mouvement entre l'image I_t et les images de référence I_{t_1} et I_{t_2}, on pourrait se contenter d'estimer les mouvements $\widehat{\Theta}^{-R}_{t\to t_1}$ et $\widehat{\Theta}^{+R}_{t\to t_2}$. Mais il est possible de faire mieux en estimant conjointement ces deux mouvements, grâce à une fonction d'énergie globale prenant en compte les 3 images et le mode d'interpolation utilisé.

Des techniques d'estimation de mouvement multi-images ont été développées pour des modélisation du mouvement par champ dense [Huang et Mersereau 94]. Dans [Chahine et Konrad 94, Chahine et Konrad 95] il s'agit d'une technique markovienne qui permet de prendre en compte 3 ou même 5 images. Les occultations sont détectées et incluses dans la formulation énergétique du problème.

La technique classique d'interpolation temporelle consiste à prédire un pixel de I_t par une combinaison linéaire des niveaux de gris des images I_{t_1} et I_{t_2}, après compensation de mouvement. On a donc

$$\widehat{I}_t(p) = \alpha I_{t_1}(\Theta^{-R}_{t\to t_1}(p)) + \beta I_{t_2}(\Theta^{+R}_{t\to t_2}(p)) \tag{4.1}$$

avec $\alpha + \beta = 1$. On peut choisir

$$\alpha = \beta = 0,5$$

ou tenir compte du temps en prenant

$$\alpha = \frac{t_2 - t}{t_2 - t_1}, \beta = \frac{t - t_1}{t_2 - t_1}$$

Ce dernier choix s'impose pour des séquences présentant des variations d'illumination. Mais même sur nos séquences où l'illumination est constante, les coefficients variables donnent de meilleurs résultats (voir les figures 4.13 et 4.14). Une dernière possibilité, non testée, serait d'estimer ces coefficients en même temps que l'estimation de mouvement, comme dans [Nicolas et al. 93].

Dans notre cas, l'estimation doit se réaliser sur les régions de l'image I_t. Pour chaque région R dans I_t, nous définissons donc la fonction d'énergie suivante :

$$\text{EQM}(\Theta^{-R}_{t\to t_1}, \Theta^{+R}_{t\to t_2}) = \sum_{p\in R} [\alpha I_{t_1}(\Theta^{-R}_{t\to t_1}(p)) + \beta I_{t_2}(\Theta^{+R}_{t\to t_2}(p)) - I_t(p)]^2$$

L'estimation consiste à effectuer la minimisation suivante

$$(\widehat{\Theta}^{-R}_{t\to t_1}, \widehat{\Theta}^{+R}_{t\to t_2}) = \arg \min_{(\Theta^{-R}_{t\to t_1}, \Theta^{+R}_{t\to t_2})} \text{EQM}(\Theta^{-R}_{t\to t_1}, \Theta^{+R}_{t\to t_2})$$

Les autres paramètres sont réglés comme pour l'estimation de mouvement faite lors du suivi temporel (voir la section 3.4). La prolongation des images en dehors de leur rectangle de définition est la même que le *padding* de MPEG. L'interpolation spatiale est soit bilinéaire, soit bicubique (voir la figure 4.12 pour une comparaison). La méthode d'optimisation est soit celle de Powell, soit les méthodes FRPR ou BFGS [Press et al. 92] (voir la section 3.4).

La fonction d'énergie que nous avons définie ne tient pas compte des occultations. Leur prise en compte dans cette fonction serait possible puisque nous disposons des segmentations des 3 images (voir la section suivante 4.2.3). Cependant, nous avons préféré ne pas en tenir compte pour deux raisons :

- Si on en tient compte, la fonction à minimiser n'est plus dérivable par rapport aux variables que l'on cherche à optimiser. Il n'est donc plus possible d'employer les méthodes d'optimisation simples reposant sur le gradient. Il faudrait alors avoir recours à des techniques plus complexes reposant sur le sous-gradient.

- De plus, certains pixels ne peuvent pas être prédits par interpolation. Comme nous le verrons dans la section 4.2.3.3, leur prédiction est alors spatiale et dépend des autres régions. On ne pourrait plus alors effectuer la minimisation région par région, mais il faudrait l'effectuer pour toutes les régions simultanément, ce qui aurait un coût très important. Une autre solution serait de ne prendre en compte dans la sommation de la fonction à minimiser que les pixels prédictibles. Mais cela introduit un biais qui pourrait pousser l'algorithme de minimisation à prédire le moins de pixels possible, pourvu qu'ils aient une très faible EQM.

4.2.3 Traitement des objets multiples et occultations

La compensation de mouvement bidirectionnelle décrite dans l'équation 4.1 est une interpolation linéaire des images. Nous allons la rendre non-linéaire pour tenir compte des occultations intervenant lorsque de multiples objets sont présents dans la scène. L'équation 4.1 n'utilise qu'une partie de l'information disponible, à savoir les descripteurs de mouvement. Pour prendre en compte les phénomènes d'occultations, il est nécessaire d'utiliser l'ensemble des information fournies par le suivi temporel, c'est-à-dire les cartes de segmentation.

Au sein d'une même région, les pixels peuvent être de natures différentes. Ils peuvent être prédits par l'équation 4.1 (section précédente), ou être des pixels recouverts ou découverts, auquel cas cette équation n'est plus adaptée. Chaque pixel p doit donc subir un traitement différent des autres pixels de la même région. Pour cela, on remplace donc les coefficients d'interpolation globaux (α, β) par des coefficients spécifiques à chaque pixel (α_p, β_p). Le choix de ces coefficients est détaillé dans les sections suivantes et résumé dans la figure 4.3 et le tableau 4.2.

4.2.3.1 Zones de recouvrement

Il s'agit de pixels appartenant à une région R dans l'image I_t, mais appartenant à une autre région R_2 dans l'image I_{t_2}, après compensation par le mouvement $\widehat{\Theta}^{+R}_{t\to t_2}$. Par contre,

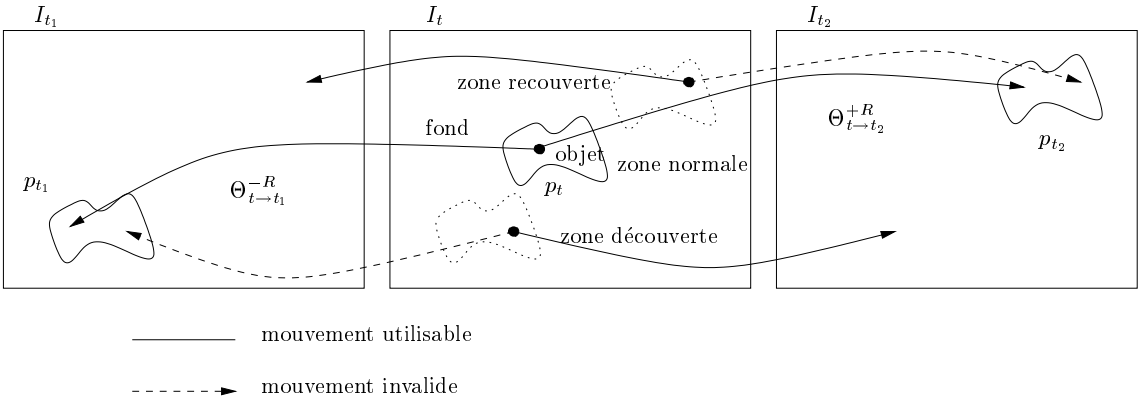


FIG. 4.3 – *Traitement des occultations.*

		position dans l'image 2	
		$p_{t_2} \in R_{t_2}$	$p_{t_2} \notin R_{t_2}$
position dans l'image 1	$p_{t_1} \in R_{t_1}$	zone «normale» $(\alpha_p, \beta_p) = (\alpha, \beta)$	zone recouverte $(\alpha_p, \beta_p) = (1, 0)$
	$p_{t_1} \notin R_{t_1}$	zone découverte $(\alpha_p, \beta_p) = (0, 1)$	zone non prédictible par CM

TAB. 4.2 – *Type de zone et coefficients d'interpolation* (α_p, β_p) *en fonction de la position du point interpolé dans les images de référence.*

l'application du mouvement $\Theta^{-R}_{t\to t_1}$ donne un pixel p_{t_1} dans l'image I_{t_1} qui appartient lui aussi à la région R. Le test est donc le suivant :

$$p \in R \text{ est recouvert} \iff \Theta^{-R}_{t\to t_1}(p) \in R \text{ et } \widehat{\Theta}^{+R}_{t\to t_2}(p) \notin R$$

Ces pixels ne peuvent donc être prédits qu'avec la texture de l'image I_{t_1}. Leur reconstruction est donc effectuée ainsi :

$$\widehat{I}_t(p) = I_{t_1}(\Theta^{-R}_{t\to t_1}(p))$$

4.2.3.2 Zones de découvrement

Il s'agit de pixels appartenant à une région R dans l'image I_t, mais appartenant à une autre région R_1 dans l'image I_{t_1}, après compensation par le mouvement $\Theta^{-R}_{t\to t_1}(p)$. Par contre, l'application du mouvement $\widehat{\Theta}^{+R}_{t\to t_2}$ donne un pixel p_{t_2} dans l'image I_{t_2} qui appartient lui aussi à la région R. Le test est donc le suivant :

$$p \in R \text{ est découvert} \iff \Theta^{-R}_{t\to t_1}(p) \notin R \text{ et } \widehat{\Theta}^{+R}_{t\to t_2}(p) \in R$$

Ces pixels ne peuvent donc être prédits qu'avec la texture de l'image I_{t_2}. Leur reconstruction est donc effectuée ainsi :

$$\widehat{I}_t(p) = I_{t_2}(\Theta^{+R}_{t\to t_2}(p))$$

4.2.3.3 Zones non prédictibles

L'algorithme précédent fonctionne dans la plupart des cas. Il est cependant des situations, où il peut être mis en défaut. Cela est dû au fait qu'à chaque nouvelle image de référence on «oublie» les textures observées dans le passé.

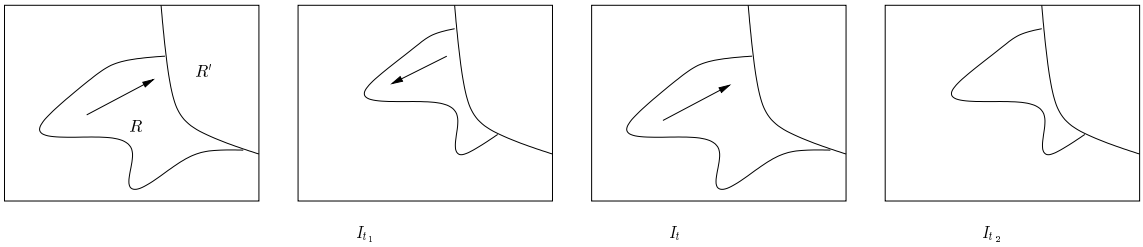


FIG. 4.4 – *Mise en défaut de la prédiction des zones découvertes (1). Si un objet R de second plan se découvre entre l'image de référence I_{t_1} et l'image interpolée I_t, puis est recouvert entre I_t et I_{t_2}, alors sa partie découverte ne peut être interpolée.*

C'est le cas lorsqu'il se produit un aller-retour d'une région sous une autre région, pendant l'intervalle de temps entre les 2 images de référence (voir la figure 4.4). Une zone de cette région est alors successivement découverte puis recouverte. Comme on ne mémorise donc pas les zones recouvertes dans l'intervalle entre les deux images de référence précédentes, même si la texture manquante a été observée dans le passé, elle ne sera pas disponible pour la prédiction de l'image courante.

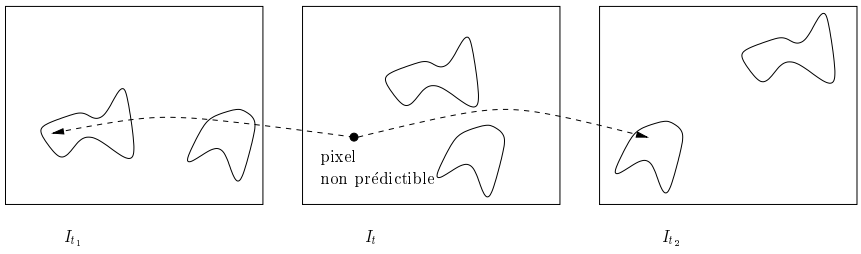


FIG. 4.5 – *Mise en défaut de la prédiction des zones découvertes (2). Si deux régions occupent successivement la même zone du fond dans les deux images de référence, cette zone ne peut être interpolée.*

Une autre situation possible est lorsque deux régions différentes recouvrent successivement la même zone du fond dans les deux images de référence (voir la figure 4.5). Cette zone n'est alors prédictible dans aucune image interpolée.

Il faut cependant remarquer que ces situations sont fort peu probables puisque l'intervalle de temps séparant les images de référence est assez faible (de 2 à 4 intervalles inter-images). Il y a donc peu de chances qu'une région puisse inverser son mouvement dans cet intervalle ou que deux régions différentes se succèdent pour en cacher une troisième.

De plus, ces situations sont détectées, donc nous pouvons essayer d'y pallier pour obtenir malgré tout une prédiction plausible. Nous faisons cela par une prédiction spatiale. Il s'agit soit d'une extrapolation de la texture de la région à partir des pixels voisins de la même région, soit d'une interpolation spatiale à partir des pixels voisins de toutes les régions.

4.2.4 Gains de codage attendus

Dans cette section, nous nous intéressons aux gains théoriques en coût de codage que l'on peut attendre de l'interpolation. Les gains réels, confirmant cette étude théorique, seront montrés dans la section 4.6. La comparaison est faite par rapport à un algorithme classique de codage par compensation de mouvement par régions et monodirectionnelle, c'est-à-dire où l'on n'utilise que l'image précédente pour effectuer une prédiction causale de l'image courante.

Le premier avantage est que l'interpolation limite les effets de dérive des textures. À bas débit, les images sont fortement dégradées et si le débit est insuffisant pour remettre à jour correctement les images de type P, celles-ci accumulent de plus en plus d'erreurs. L'introduction d'image de type B a pour effet de réduire le nombre d'images de type P au sein d'un GOP (voir le tableau 4.1). La dérive des images de type P est ainsi réduite. Ceci n'est évidemment valable que pour les zones non découvertes, puisque si le débit est trop faible, le codage de l'image d'erreur ne suffit pas à corriger suffisamment les erreurs dues à la mauvaise prédiction des zones découvertes.

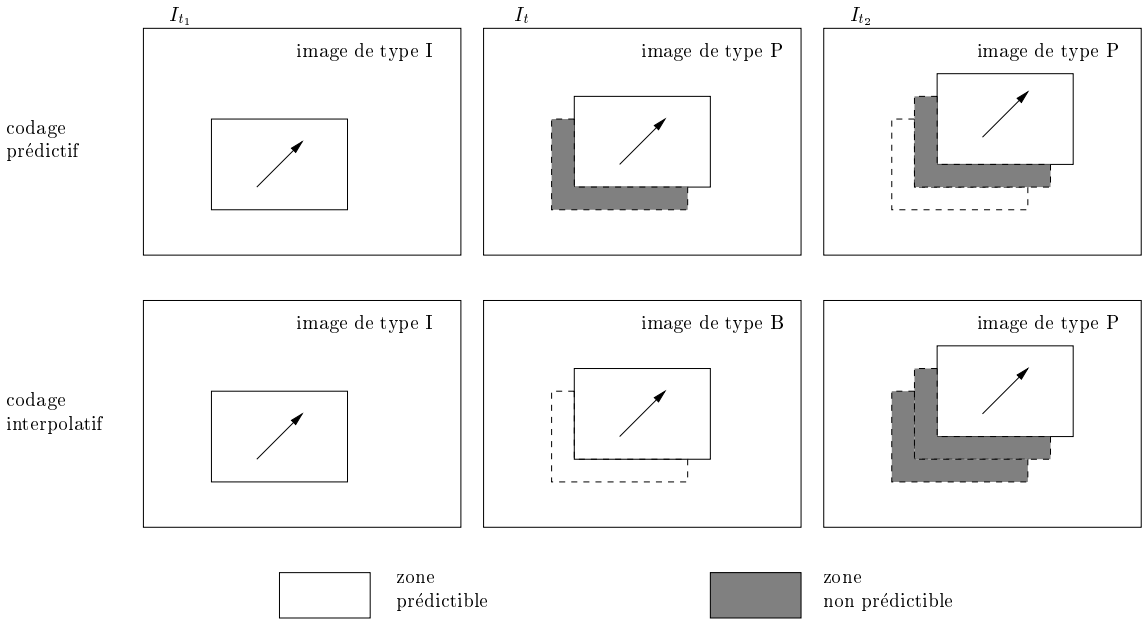


FIG. 4.6 – *Gain du codage prédictif par rapport au codage interpolatif dans les zones découvertes. Les zones découvertes, non-prédictibles dans les images de type P, sont regroupées dans le cas du codage interpolatif dans l'image* I_{t_2}.

Le deuxième effet bénéfique de l'interpolation est un codage plus efficace des zones découvertes, même dans les images de type P. Comme cela est montré dans la figure 4.6, une image de type P succédant à plusieurs images de type B contient toutes les zones découvertes successives. Il y a donc autant d'information à coder que si l'on avait codé chaque zone découverte à chaque instant, dans des images de type P. Cependant toutes ces nouvelles informations sont regroupées de façon spatialement cohérente dans la même image I_{t_2}. Il est donc possible d'exploiter la corrélation spatiale entre ces zones dans leur codage. Ceci est plus efficace que des codages séparés car ces zones appartenant probablement au même objet sont probablement fortement corrélées.

Le troisième effet est une meilleure prédiction des zones «normales» grâce à l'interpolation, comme nous allons le voir avec un modèle très simple. Pour cela, on modélise l'erreur de prédiction par compensation de mouvement par un bruit additif d'espérance nulle, qui prend en compte à la fois l'erreur liée au mouvement et l'erreur liée à la non conservation du niveau de gris au cours du temps. On peut alors écrire dans le cas d'une prédiction monodirectionnelle entre les instants $t-1$ et t :

$$I_t(p) = I_{t-1}(\Theta^{-R}_{t\to t-1}(p)) + b$$

Dans le cas d'une prédiction bidirectionnelle, on peut écrire :

$$I_t(p) = \alpha[I_{t_1}(\Theta^{-R}_{t\to t_1}(p)) + b_1] + \beta[I_{t_2}(\Theta^{+R}_{t\to t_2}(p)) + b_2]$$

Pour comparer ces deux prédictions, il faut se placer sur une image de type B succédant immédiatement à une image de type I ou P, et prendre $t-1=t_1$. Si l'on fait l'hypothèse que les variances des bruits b_1 et b_2 sont égales à la variance σ_P^2 de b, et que les bruits b_1 et b_2 ne sont pas corrélés, la variance σ_B^2 de la prédiction par interpolation est égale à $\alpha^2\sigma_P^2 + \beta^2\sigma_P^2$. Elle est donc toujours inférieure à σ_P^2. En effet, le rapport des variances

$$\frac{\sigma_B^2}{\sigma_P^2} = \alpha^2 + (1-\alpha)^2$$

est toujours < 1. De plus, il est minimal pour $\alpha = \beta = 0,5$, c'est-à-dire pour l'interpolation à coefficients fixes.

4.3 Codage des différentes informations

4.3.1 Codage des mouvements des régions

Le codage des mouvements se fait par une quantification suffisamment précise pour être considérée comme sans perte. Pour un mouvement affine, il faut distinguer entre paramètres de translation et paramètres affines. Les premiers ont besoin d'une moins grande précision, mais d'une plus grande amplitude. Le tableau 4.3 résume les coûts de codage associés à ces paramètres.

L'intervalle $[-16;16]$ utilisé pour les paramètres de translation a une amplitude supérieure à l'amplitude maximale constatée sur les séquences de test, qui est de 15 pixels pour la balle de ping-pong de la séquence *«Tennis»*. L'amplitude des paramètres affines

	translation	affine
intervalle	$[-16; 16]$	$[-0,25; 0,25]$
précision	$1/4$	$1/512$
coût	7 bits	8 bits

TAB. 4.3 – *Coût de codage des mouvements des régions.*

correspond à une limite que nous avons fixée dans l'algorithme de suivi temporel, seuil au-delà duquel le mouvement estimé est considéré comme aberrant. En effet, un paramètre de 0,25 correspond par exemple à une rotation de l'objet de 15°. La précision de 1/512 est choisie de sorte à garantir une précision sur le champ de déplacement de 1/4 de pixel pour un point situé à une distance de 128 pixels du centre de référence du mouvement, ce qui correspond à une taille de région assez grande. Pour une séquence au format TV, comportant plus de pixels que les images au format CIF, il faudrait peut-être augmenter la précision en conséquence. Au total, ce sont donc 46 bits qui sont nécessaires pour un mouvement affine, et 30 bits pour un mouvement affine simplifié.

4.3.2 Codage de la segmentation

Le codage sans pertes d'une carte de segmentation représentée par une carte d'étiquette peut se faire par la technique des codes de Freeman [Freeman 61]. Une bonne mise en œuvre [Pateux et Labit 97] effectuant une prédiction par chaîne de Markov et un codage arithmétique obtient des résultats de l'ordre de 0,8 à 1,3 bit par élément de contour. Mais dans notre cas, il est possible de faire mieux grâce à l'approximation polygonale des frontières entre régions dont nous disposons.

4.3.2.1 Codage en mode intra-image

Les résultats présentés ici s'appuient sur les méthodes de codage développées dans [Pateux et Labit 97]. Il s'agit d'un codage optimisé pour une approximation polygonale de segmentation. Notons que dans les travaux originaux, il s'agit d'un codage avec pertes, mais que dans notre cas, il s'agit d'un codage sans pertes, puisque nous partons déjà d'une représentation par polygones.

Une première transformation consiste à effectuer un certain parcours du graphe et à transformer les positions des sommets en vecteurs déplacement d'un sommet au suivant. Chaque segment, dont les extrémités sont quantifiées au pixel près, est décrit dans un système de coordonnées qui est l'équivalent des coordonnées polaires dans le cas discret. Des statistiques calculées en ligne sur la longueur des segments permettent d'effectuer un codage entropique de cette information. Les coûts obtenus par cette technique sont montrés dans les figures 4.7 à 4.10 sous la dénomination «codage intra». On observe une diminution rapide du coût au début de la séquence, grâce à l'apprentissage statistique effectué au cours du temps par l'algorithme.

4.3.2.2 Codage en mode inter-images

Il est possible d'améliorer le codage précédent en utilisant la propriété de stabilité temporelle de la segmentation. L'algorithme de suivi temporel fournit une prédiction des frontières au cours du temps, qui peut être utilisée pour un codage différentiel prédictif d'une segmentation par rapport à l'instant précédent.

Le mouvement de la texture des régions est déjà transmis pour l'interpolation par compensation de mouvement. Donc il ne reste que deux informations à transmettre : le mouvement de l'ajustement affine des frontières (AAF) et les vecteurs déplacements de l'ajustement local des sommets (ALS). La re-création des points multiples est effectuée au décodeur de la même façon que dans l'algorithme de suivi.

	translation	affine
intervalle	$[-4; 4]$	$[-0,125; 0,125]$
précision	$1/2$	$1/128$
coût	4 bits	5 bits

TAB. 4.4 – *Coût de codage des mouvements affines des frontières.*

Les paramètres de l'AAF sont quantifiés comme indiqué dans le tableau 4.4. Les pas de quantification sont calculés de sorte à assurer une précision de l'ordre du pixel. Une frontière nécessite donc 28 bits. Les vecteurs de l'ALS sont codés avec le même algorithme que les déplacements dans le mode intra, mais avec des statistiques différentes.

Les figures 4.7 à 4.10 montrent le coût de codage de l'ALS, sous la dénomination «codage inter». Pour une comparaison avec le mode intra, il faut encore rajouter le coût de l'AAF. Tout compris, le mode inter offre un gain de l'ordre de 30%.

4.3.3 Codage de l'image d'erreur

Le codage de l'image d'erreur réutilise les travaux de [Nguyen 95], mais sans utiliser l'aspect de codage par régions d'intérêt. Il s'agit d'un codage par transformée, quantification et codage entropique.

Le transformation utilisée est une transformation par ondelettes biorthogonales 7–9 [Antonini et al. 90]. Ces bancs de filtre sont généralement considérés comme les plus performants, et sont notamment meilleurs que les ondelettes orthogonale [Daubechies 88], car ils sont symétriques, donc à phase linéaire, ce qui est une propriété importante pour des signaux de type image. Nous avons utilisé 3 niveaux de décomposition et des paquets d'ondelettes (redécomposition des sous-bandes de hautes fréquences).

La quantification est une quantification scalaire uniforme dans chaque sous-bande. La contribution de chaque sous-bande à l'erreur totale subit une pondération psychovisuelle [Girod 92] prenant en compte la sensibilité du système visuel humain aux fréquences spatiales [Vandendorpe 91]. Le choix des quantificateurs est optimisé et permet une régulation en débit ou en distortion comme dans [Ramchandran et al. 94] : on considère pour chaque sous-bande (au nombre de n) un ensemble de k quantificateurs possibles. L'algorithme trouve alors une combinaison de quantificateurs qui optimise le débit sous une contrainte de

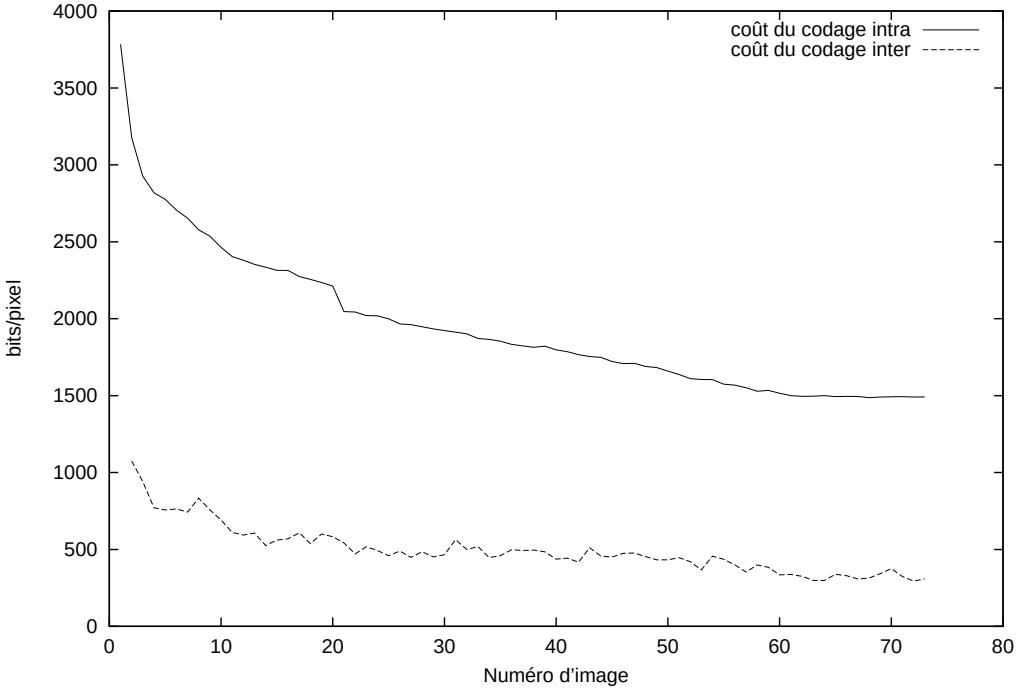


FIG. 4.7 – *Séquence* «Miss America». *Comparaison du coût de codage de la segmentation en mode intra- et en mode inter- image.*

distortion (ou l'inverse). Il s'agit d'un algorithme rapide qui trouve quasiment la meilleure combinaison parmi les k^n combinaisons possibles, mais avec une complexité proche de $n.k$.

Les indices de quantification subissent ensuite un codage entropique, sous-bande par sous-bande, selon un modèle probabiliste d'ordre 0. Au codage de Huffman [Huffman 51], nous avons préféré le codage arithmétique [Rissanen 76] [Witten et al. 87] [Langdon et Rissanen 81] qui donne des résultats légèrement meilleurs. L'implantation utilisée est celle de [Nelson et Gailly 92].

4.4 Modes de codage interpolatif et codage hiérarchique

Cette section montre comment on peut faire varier les informations de mouvement et de segmentation transmises, pour obtenir des schémas de codage différents. Cette idée peut avoir deux applications :

- On peut transformer le schéma de codage interpolatif décrit précédemment pour en faire un schéma de codage hiérarchique à 3 niveaux.

- On peut considérer ces 3 niveaux comme des modes d'interpolation indépendants. L'image d'erreur transmise dépend alors du mode d'interpolation choisi.

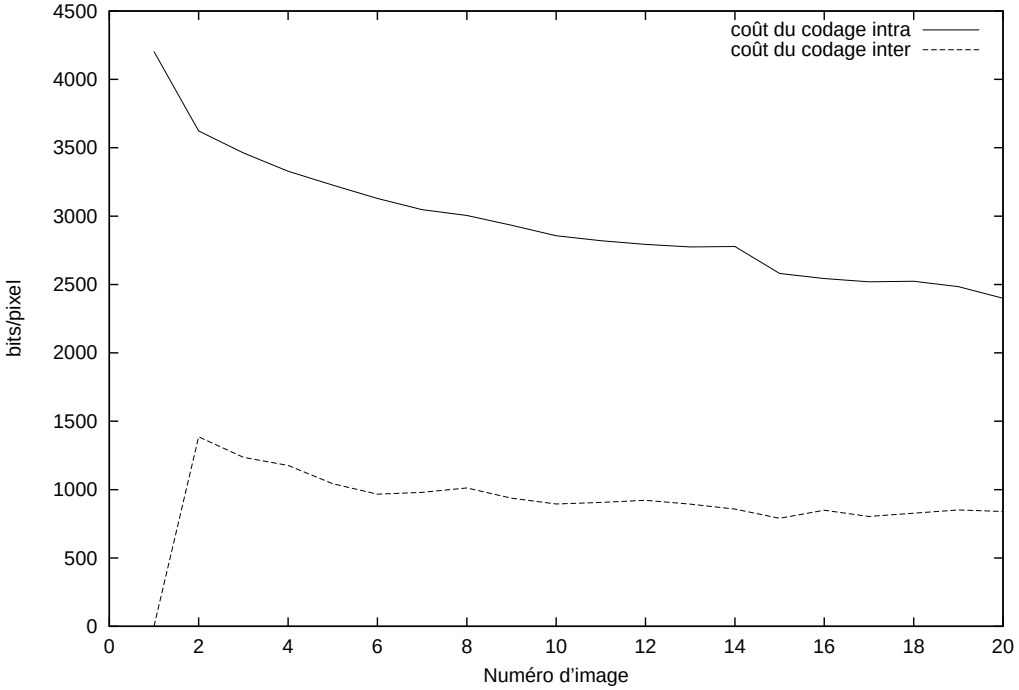


FIG. 4.8 – *Séquence* «Interview». *Comparaison du coût de codage de la segmentation en mode intra- et en mode inter- image.*

4.4.1 Interpolation par compensation bidirectionnelle de mouvement

Ce schéma est le plus simple puisqu'il suppose la transmission de toutes les informations nécessaires à l'interpolation entre le codeur et le décodeur. La segmentation de l'image I_t à interpoler est transmise, ainsi que les mouvements des régions vers les images de référence $\widehat{\Theta}^{-R}_{t\to t_1}$ et $\widehat{\Theta}^{+R}_{t\to t_2}$.

4.4.2 Interpolation par prédiction bidirectionnelle de segmentation

Dans ce mode, ce n'est plus l'image qui est interpolée, mais sa segmentation. Le but est le même que dans [Meyer 96], mais à la différence de ces travaux qui utilisent des techniques purement morphologiques, nous disposons des mouvements des régions, ce qui va nous aider fortement.

Le codeur transmet toujours $\widehat{\Theta}^{-R}_{t\to t_1}$ et $\widehat{\Theta}^{+R}_{t\to t_2}$, mais plus la segmentation de I_t. Celle-ci est alors reconstruite au mieux par le décodeur. Pour cela, il suffit de faire comme pour la prédiction des frontières dans l'algorithme de suivi temporel, en tenant compte de l'affectation des frontières aux régions. Le décodeur peut soit appliquer les mouvements $[\widehat{\Theta}^{-R}_{t\to t_1}]^{-1}$ à la segmentation de I_{t_1}, soit appliquer les mouvements $[\widehat{\Theta}^{+R}_{t\to t_2}]^{-1}$ à la segmentation de I_{t_2}.

Le décodeur obtient ainsi des frontières déconnectées pour l'image I_t, exactement comme l'algorithme de suivi temporel après les phases de prédiction et d'ajustement affine

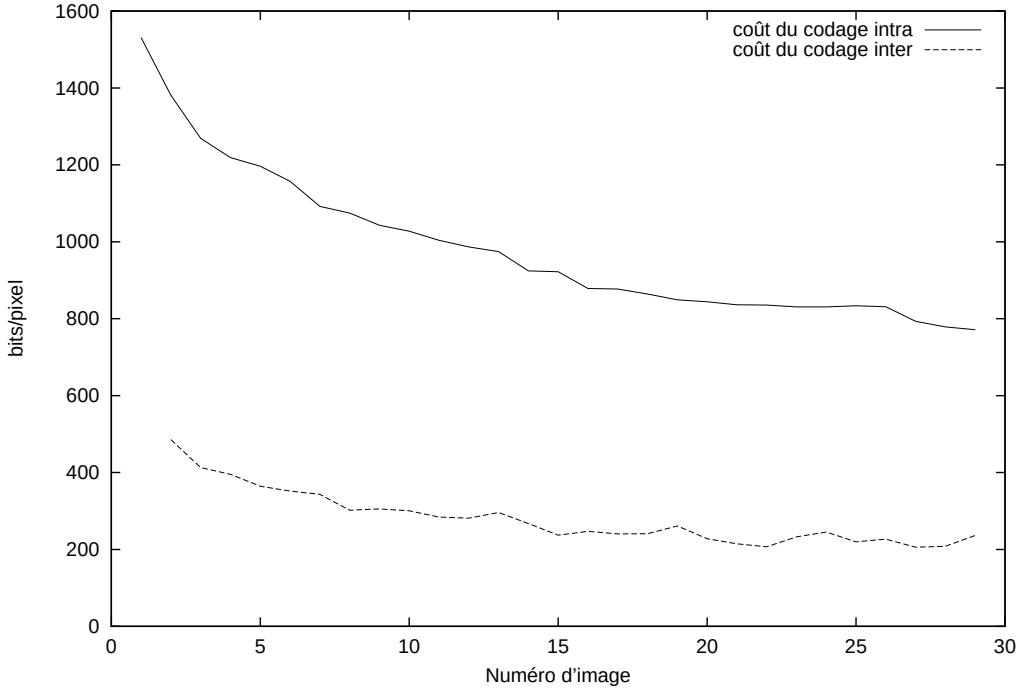


FIG. 4.9 – *Séquence* «Flower Garden». *Comparaison du coût de codage de la segmentation en mode intra- et en mode inter- image.*

des frontières. On peut donc appliquer le même algorithme de re-création des points multiples. Ce mode d'interpolation ressemble au codage inter de segmentation présenté plus haut. Mais il est plus adapté à l'interpolation, puisque les segmentations sont transmises dans le même ordre que les images : pour un GOP de type 1B par exemple, la transmission se fait dans l'ordre IPB au lieu de IBP.

4.4.3 Interpolation par prédiction du mouvement (interpolation pure)

Dans ce mode, ni la segmentation, ni les mouvements ne sont transmis pour les images de type B. Il faut donc reconstruire ces mouvements à partir des mouvements transmis pour les images de type P, à savoir $\widehat{\Theta}^{-R}_{t_2 \to t_1}$. Il existe des contraintes permettant de commencer à résoudre ce problème : la composition des mouvements servant à l'interpolation doit être égale au mouvement servant à la prédiction de l'image P. Cela se traduit par les équations suivantes :

$$\forall t, t_1 < t < t_2, \quad \widehat{\Theta}^{-R}_{t \to t_1} \circ [\widehat{\Theta}^{+R}_{t \to t_2}]^{-1} = \widehat{\Theta}^{-R}_{t_2 \to t_1}$$

Une autre série d'équations est nécessaire à la résolution du problème. Nous avons choisi de considérer que le mouvement des objets entre les images de référence est à vitesse

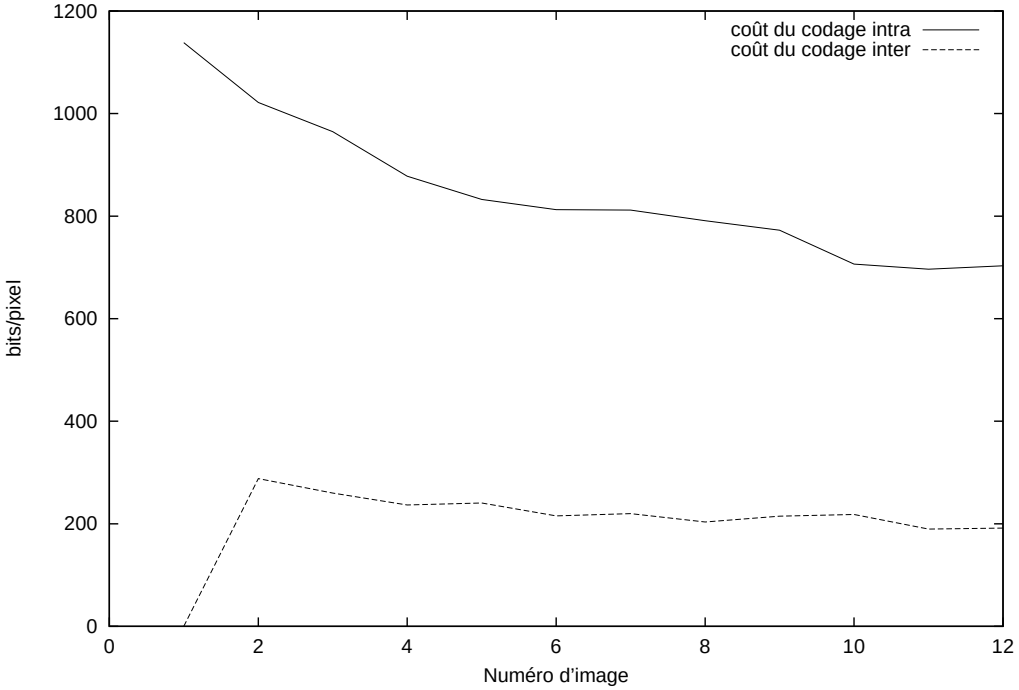


FIG. 4.10 – *Séquence* «Tennis». *Comparaison du coût de codage de la segmentation en mode intra- et en mode inter- image.*

constante, ce qui nous permet d'écrire que

$$\forall t, t_1 < t \leq t_2, \quad \widehat{\Theta}^{-R}_{t \to t-1} = \Theta_0$$

avec Θ_0 une constante.

Ainsi on obtient l'équation

$$\Theta_0^{t_2 - t_1} = \widehat{\Theta}^{-R}_{t_2 \to t_1}$$

Nous montrons comment la résoudre analytiquement dans un cas simple à l'annexe B, section B.6. Dans le cas général, il faut recourir à une méthode numérique de résolution.

Ensuite on obtient simplement les mouvements nécessaires à l'interpolation par :

$$\widehat{\Theta}^{-R}_{t \to t_1} = \Theta_0^{t - t_1} \quad \text{et} \quad \widehat{\Theta}^{+R}_{t \to t_2} = \Theta_0^{t - t_2}$$

Une fois ces mouvements connus, il faut encore appliquer le même algorithme que dans la section précédente pour obtenir la segmentation de l'image I_t.

4.5 Interpolation pure

Dans cette section, nous envisageons l'application de notre algorithme d'interpolation temporelle aux problèmes de l'interpolation d'images dans le cas de pertes lors de la transmission, et pour une diffusion multipoints. L'interpolation pure est en fait identique au

mode d'interpolation appelé «Interpolation par Prédiction du Mouvement» dans la section précédente. Mais dans ces applications, l'image interpolée ne sert pas comme prédiction pour un codage efficace, mais elle est montrée directement à l'utilisateur du système. Il importe donc que ses caractéristiques visuelles soient les meilleures possibles.

FIG. 4.11 – *Exemple de faux contours. Séquence* «Flower Garden»*, image 19 prédite par compensation de mouvement bidirectionnelle. Le défaut principal est visible à droite de l'arbre, dans le ciel.*

Or il existe un défaut dans les images interpolées par compensation de mouvement bidirectionnelle prenant en compte les occultations. Il s'agit de faux contours parasites causés par une légère imprécision des frontières dans la segmentation. Sur la figure 4.11, on peut par exemple remarquer dans le ciel, à droite de l'arbre, un contour issu du tronc. En fait, la segmentation ne peut jamais être parfaite car il n'y a pas de séparation nette entre objets de la scène. En effet, tout système d'acquisition d'images effectue un pré-filtrage avant l'échantillonnage en pixels. Il peut être dû au système optique ou au système CCD qui moyenne la lumière incidente sur des surfaces non ponctuelles. Ce filtrage a pour effet de lisser les contours entre objets.

Pour supprimer ces faux contours, nous avons érodé les masques des régions dans les segmentations des images de référence. Les tests $p_{t_1} \in R_{t_1}$ et $p_{t_2} \in R_{t_2}$ effectués dans l'algorithme original, sont remplacés par $p_{t_1} \in \overset{\circ}{R}_{t_1}$ et $p_{t_2} \in \overset{\circ}{R}_{t_2}$, avec $\overset{\circ}{R}$ l'érodée de R. La taille de l'élément structurant à utiliser dépend des caractéristiques du pré-filtrage.

Le fait d'éroder les masques des régions augmente le nombre de pixels non prédictibles. Ceux-ci se trouvent le long des frontières entre régions. Ils sont donc interpolés spatialement, comme expliqué précédemment.

Remarquons qu'en dépit du faux contour, l'image 4.11 a un aspect visuel globalement acceptable, même si la segmentation présente un défaut important. Sur la figure 3.33, on constate qu'un morceau du ciel est rattaché à l'arbre. Sur l'image interpolée, ce défaut est quasiment invisible car le morceau de ciel en question est peu texturé.

4.6 Résultats

Dans les résultats qui suivent, nous avons encodé toutes les séquences avec une qualité constante. Ce sont donc les débits que nous comparons. L'intérêt de cette approche est que nos comparaisons sont indépendantes du difficile problème de l'allocation de débit entre types d'images I, P ou B.

4.6.1 Comparaison des interpolations bilinéaire et bicubique

La figure 4.12 montre une comparaison les interpolations bilinéaire et bicubique. Cette dernière est meilleure, ainsi que dans les autres séquences. Mais comme elle est plus complexe, il s'agit de trouver un compromis. Pour les images de type P, le gain est moins évident car il est masqué par les grandes erreurs de prédiction dans les zones découvertes.

4.6.2 Comparaison de l'interpolation avec coefficients fixes ou variables

Les figures 4.13 et 4.14 montrent une comparaison entre interpolation avec coefficients fixes et coefficients variables. Contrairement à notre étude théorique de la section 4.2.4, ce sont les coefficients variables qui donnent de meilleurs résultats sur cette séquence et sur les autres que nous avons testées. Cela est probablement dû à l'hypothèse que nous avons faite selon laquelle les variances des erreurs provenant des deux images de référence sont égales, ce qui est faux en pratique, puisque l'on peut s'attendre à une erreur d'autant plus grande que l'image de référence est plus éloignée.

4.6.3 Comparaison des interpolations basée régions et blocs

La figure 4.15 montre une comparaison entre notre schéma de codage interpolatif et un autre schéma où tout est identique, sauf la compensation de mouvement qui est effectuée par blocs, comme dans MPEG. On remarque que la prédiction des images de type B est nettement meilleure, mais l'amélioration pour les images P est plus faible. Cela s'explique par le fait que les images P concentrent toute l'erreur causée par les zones découvertes (EQM entre 500 et 800). Donc toute amélioration dans les zones prédictibles (de l'ordre de 50 en EQM) sera masquée par ces erreurs.

Il s'agit d'une comparaison à haut débit. Pour d'autres débits, on se reportera au tableau 4.5.

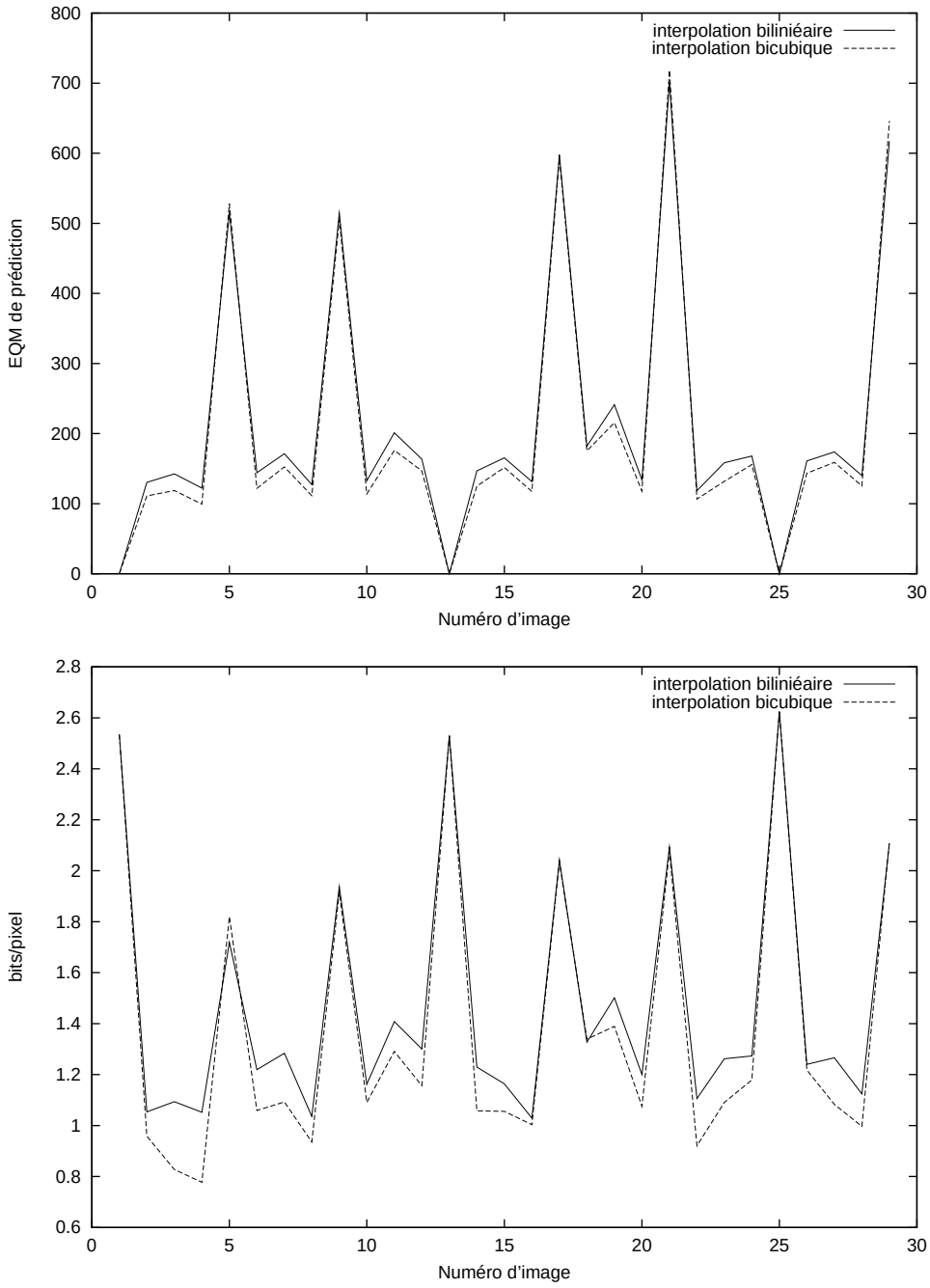


FIG. 4.12 – *Séquence* «Flower Garden». *Comparaison des interpolations bilinéaire et bicubique. Structure de* GOP *3B,* PSNR *35. En haut :* EQM *de prédiction. En bas : coût du codage de l'image d'erreur.*

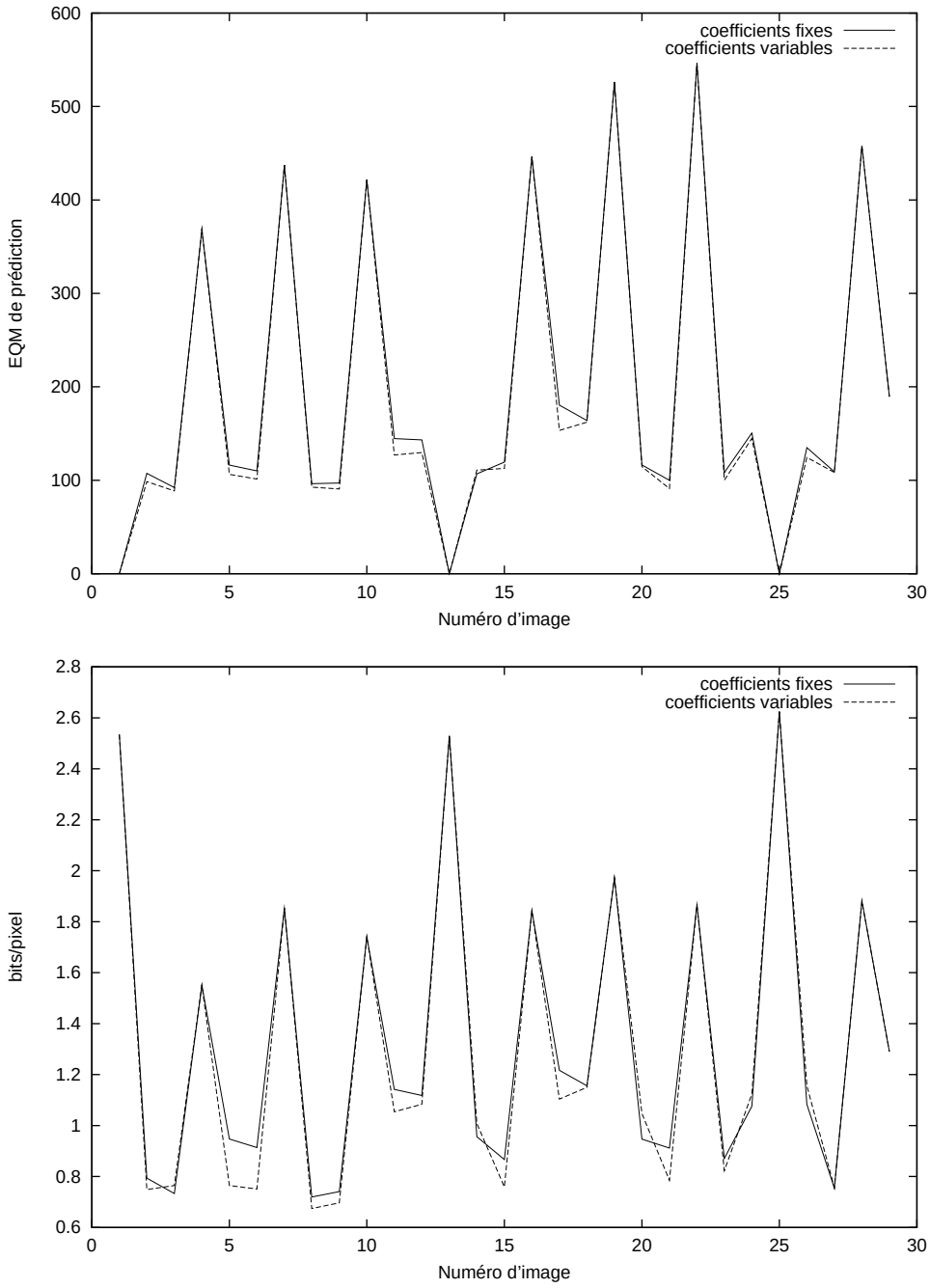


FIG. 4.13 – *Séquence* «Flower Garden». *Comparaison de l'interpolation avec des coefficients fixes ou variables. Structure de* GOP *2B,* PSNR *35. En haut :* EQM *de prédiction. En bas : coût du codage de l'image d'erreur.*

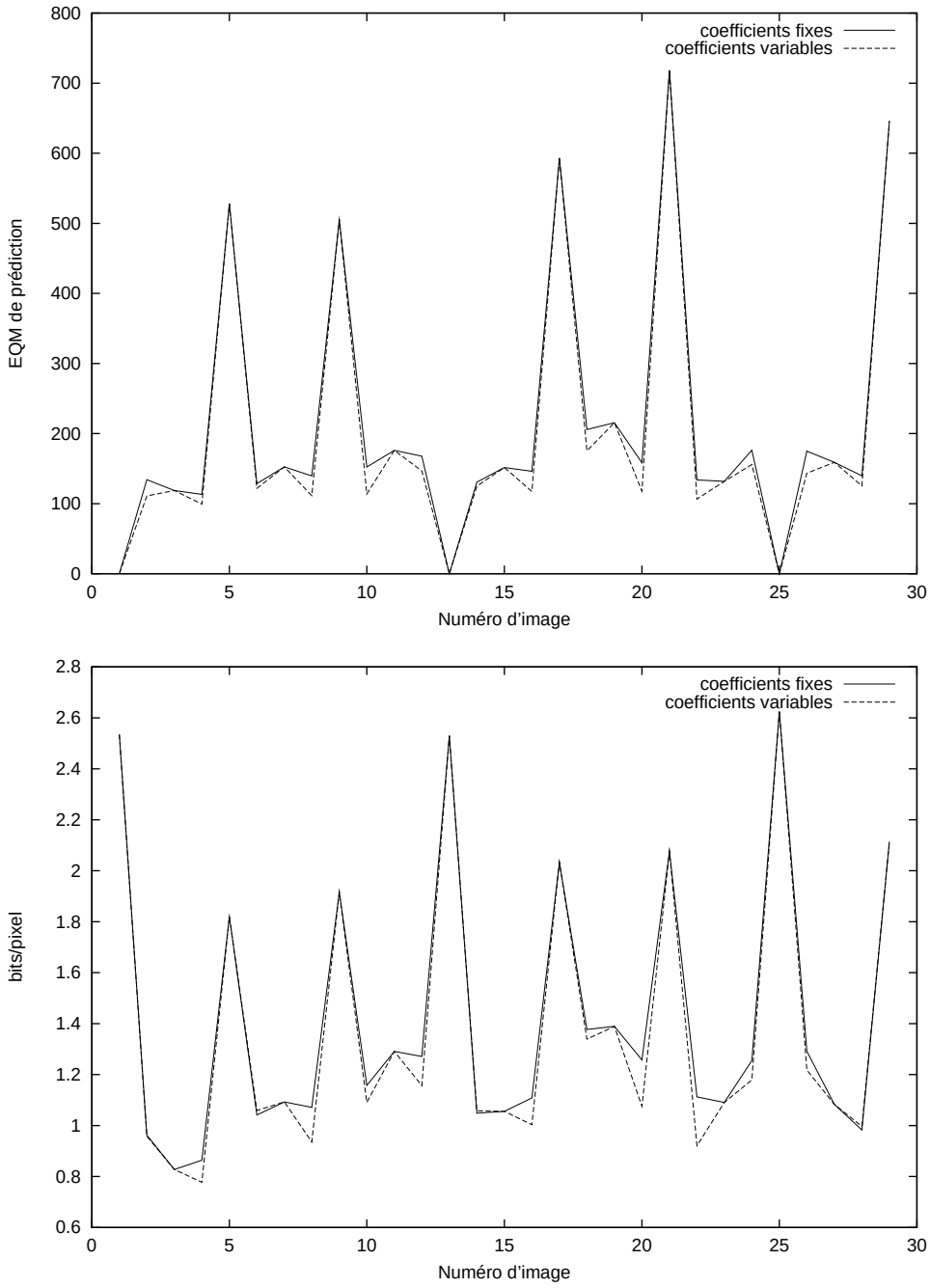


FIG. 4.14 – *Séquence* «Flower Garden». *Comparaison de l'interpolation avec des coefficients fixes ou variables. Structure de* GOP *3B,* PSNR *35. En haut :* EQM *de prédiction. En bas coût du codage de l'image d'erreur.*

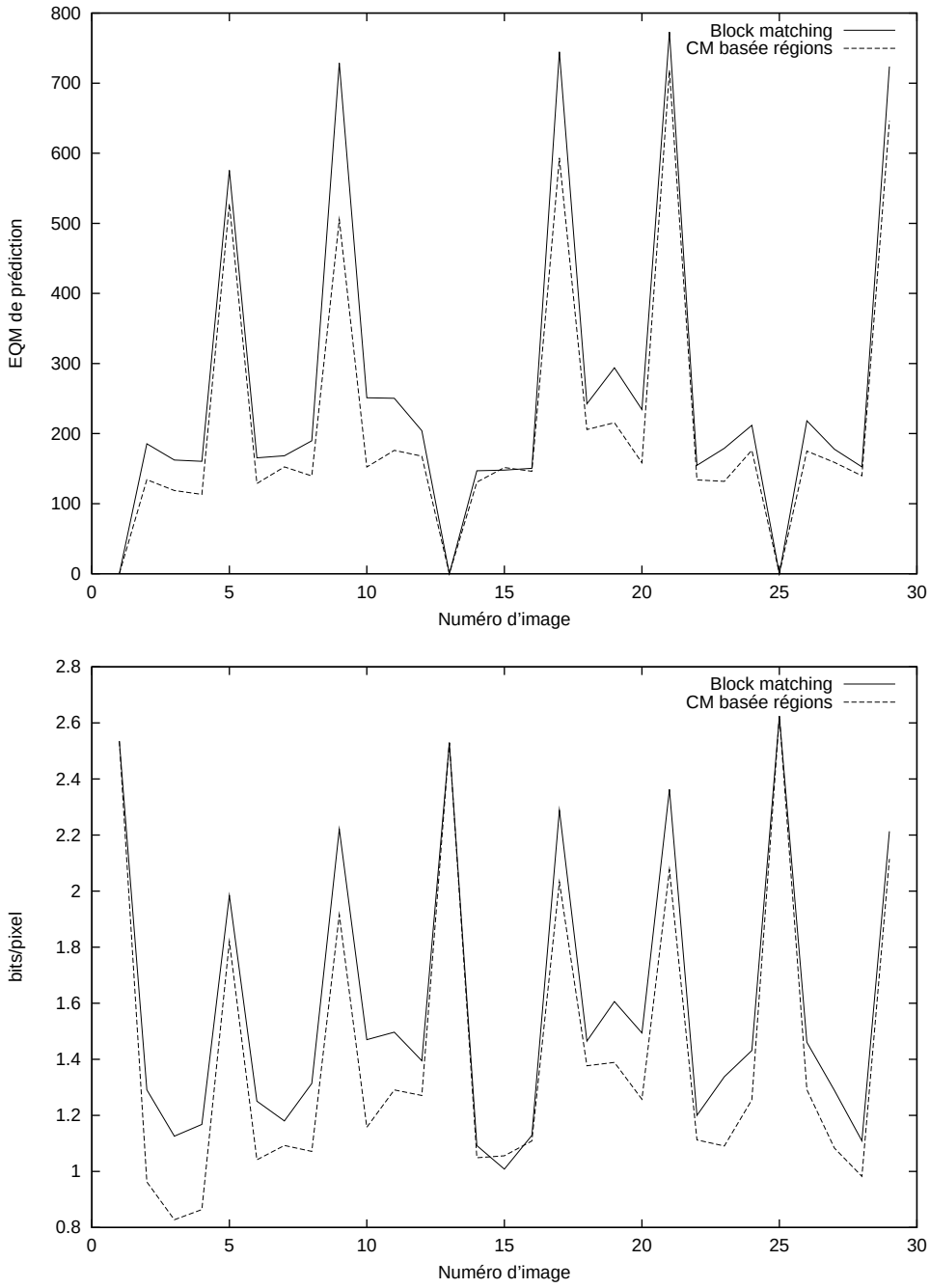


FIG. 4.15 – *Séquence* «Flower Garden». *Comparaison de l'interpolation par* block-matching *et de l'interpolation basée régions. Structure de* GOP *3B,* PSNR *35. En haut :* EQM *de prédiction. En bas : coût du codage de l'image d'erreur.*

	compensation du mouvement	
PSNR	par blocs	par régions
25	0,1207	0,1778
30	0,6714	0,5430
35	1,4453	1,1749

TAB. 4.5 – *Comparaison des compensations de mouvement par blocs et par régions. Débit moyen pour le codage de l'image d'erreur, sur les images 2 à 12. Séquence* «Flower Garden» *codée avec des* PSNR *de 25, 30 et 35 dB. Structure de* GOP *3B.*

1^{er} GOP

n° d'image	P	1B	2B	3B
2	P 1,03187	B 0,575142	B 0,7491	B 0,958191
3	P 1,08868	P 1,32827	B 0,76482	B 0,827604
4	P 1,22618	B 0,488589	P 1,5509	B 0,777225
5	P 1,06463	P 1,42145	B 0,764631	P 1,82022
6	P 1,29806	B 0,639441	B 0,7509	B 1,05904
7	P 1,20393	P 1,65706	P 1,85279	B 1,09285
8	P 1,14323	B 0,502415	B 0,674763	B 0,935369
9	P 1,20649	P 1,42221	B 0,696828	P 1,92031
10	P 1,13404	B 0,631581	P 1,74077	B 1,09058
11	P 1,2724	P 1,61918	B 1,05412	B 1,29086
12	P 1,57401	B 0,768797	B 1,08348	B 1,15658
moyenne	1,2039	1,0049	1,0621	1,1753

$2^{ème}$ GOP

n° d'image	P	1B	2B	3B
14	P 1,3438	B 0,857528	B 1,00791	B 1,05781
15	P 1,158	P 1,70724	B 0,759233	B 1,05592
16	P 1,21766	B 0,529593	P 1,84295	B 1,00336
17	P 1,35156	P 1,7241	B 1,10402	P 2,03291
18	P 1,47732	B 0,895218	B 1,15014	B 1,33897
19	P 1,59399	P 1,84427	P 1,9724	B 1,38916
20	P 1,49579	B 0,704687	B 1,04692	B 1,07571
21	P 1,09134	P 1,65014	B 0,78518	P 2,07666
22	P 1,25687	B 0,461316	P 1,86605	B 0,920786
23	P 1,17562	P 1,55592	B 0,822869	B 1,09086
24	P 1,4276	B 0,81321	B 1,12382	B 1,17874
moyenne	1,3263	1,1585	1,2256	1,2928

TAB. 4.6 – *Comparaison des structures de* GOP. *Débit nécessaire au codage de l'image d'erreur. Séquence* «Flower Garden» *codée avec un* PSNR *de 35 dB. Les images 1, 13 et 25 sont codées en mode intra-image et ne sont donc pas incluses dans cette comparaison.*

4.6.4 Comparaison des compensations de mouvement monodirectionnelle et bidirectionnelle

Les tableaux 4.6, 4.7 et 4.8 comparent les 4 structures de GOP classiques. Ils démontrent l'intérêt de la compensation de mouvement bidirectionnelle, et ce d'autant plus que l'on augmente le nombre d'images de type B, sauf sur la séquence *«Flower Garden»* à haut débit. Mais comme règle générale, on observe que l'interpolation est d'autant plus intéressante que l'on travaille à bas débit.

	Structures de GOP			
PSNR	IP	1B	2B	3B
25	0,3186	0,1949	0,1791	**0,1779**
30	0,6856	0,4991	**0,4958**	0,5430
35	1,2039	**1,0049**	1,0621	1,1753

TAB. 4.7 – *Comparaison des structures de* GOP. *Débit moyen nécessaire au codage de l'image d'erreur pour les images 2 à 12. Séquence* «Flower Garden» *codée avec des* PSNR *de 25, 30 et 35 dB.*

	Structures de GOP			
PSNR	IP	1B	2B	3B
25	0,0609	0,0592	0,0359	**0,0241**
30	0,3504	0,2871	0,2191	**0,1873**
35	0,7722	0,6580	0,6127	**0,5282**

TAB. 4.8 – *Comparaison des structures de* GOP. *Débit moyen nécessaire au codage de l'image d'erreur pour les images 2 à 11. Séquence* «Tennis» *codée avec des* PSNR *de 25, 30 et 35 dB.*

4.7 Conclusion partielle

Dans ce chapitre, nous avons vérifié que le codage interpolatif est effectivement meilleur qu'un codage prédictif causal. Ce résultat était connu pour le *block-matching*, mais nous l'avons établi pour notre algorithme d'interpolation basé régions.

Nous avons aussi montré comment utiliser notre structure de représentation pour interpoler les segmentations des images. Ceci nous permet d'appliquer notre technique à la diffusion multipoints et à la restitution d'images manquantes par suite d'erreurs de transmission.

Conclusion générale et perspectives

Le travail que nous venons de présenter apporte sa contribution au difficile problème de la segmentation d'une séquence d'images en objets vidéo, dans le cadre d'applications aux télécommunications ou plus généralement dans le domaine du multimédia. Les schémas de codage innovants qui en découlent apportent des améliorations significatives par rapport à l'existant.

Résumé des travaux effectués, contributions

Les contributions de cette thèse portent principalement sur les points suivants :

- Nous avons effectué un examen approfondi des modélisations du mouvement dans une séquence animée, en nous intéressant plus particulièrement aux problèmes de la prise en compte des discontinuités spatiales du mouvement et de sa continuité temporelle, dans la perspective d'un suivi temporel. Nous avons conclu que la modélisation par régions était la plus à même de répondre à nos besoins.

- Nous avons défini une représentation propre de la partition de l'image en régions. Il s'agit d'une représentation basée contours, de plus haut niveau que la représentation basée pixels la plus habituelle. De plus, elle constitue une amélioration de la représentation par contours fermés utilisée dans d'autres travaux antérieurs :

  ⋆ elle est non redondante puisque basée sur des frontières ouvertes ;

  ⋆ elle permet de traiter de façon plus satisfaisante le problème des occultations entre objets de la scène, et ce plus efficacement puisque le temps de calcul nécessaire est environ divisé par 2.

- Un algorithme de suivi temporel, spécifique à cette représentation, a été développé. Il opère un suivi des frontières entre objets par un mécanisme de prédiction/ajustement. Les deux caractéristiques essentielles en sont :

  ⋆ une bonne stabilité temporelle de la segmentation, grâce à quelques hypothèses assez fortes faites sur les mouvements des objets

  ⋆ une grande précision des frontières des objets obtenus, grâce à un ajustement selon un modèle de contours actifs.

On peut ainsi considérer que nos objectifs initiaux, dictés par l'application à l'interpolation, ont été atteints. Mais il faut noter la grande sensibilité de cet algorithme à la segmentation initiale de la première image de la séquence.

- L'application de ces résultats au problème de l'interpolation temporelle a ensuite été effectuée. Nous avons défini des schémas de codage interpolatif par régions, ce qui est une originalité de notre travail. Les résultats obtenus démontrent que l'interpolation permet une meilleure compression que les schémas non interpolatifs utilisés classiquement dans le codage par régions. Nous avons aussi montré comment :

  ⋆ réaliser un schéma de codage hiérarchique grâce à notre représentation, en transmettant de façon optionnelle les informations de segmentation et de mouvement.

  ⋆ utiliser notre algorithme d'interpolation temporelle dans un système de transmission multipoints. On adapte la fréquence d'affichage des images sur le terminal de visualisation par rapport à la fréquence de transmission des images, en interpolant les images nécessaires.

  ⋆ remédier aux pertes éventuelles lors de la transmission, en interpolant les images perdues.

Perspectives

Un certain nombre de problèmes relatifs aux travaux effectués restent à explorer. Nous en proposons ici quelques uns qui constituent des directions de recherche intéressantes :

- Dans le but d'assurer une stabilité temporelle maximale, la version actuelle de l'algorithme ne permet que de très faibles changements du graphe de représentation ou de topologie de la segmentation. Il serait intéressant de regarder si la tolérance de plus grands changements ne remettrait pas en cause la stabilité temporelle. De même il serait intéressant de traiter le problème des nouveaux objets qui apparaissent, qu'ils entrent dans la scène, qu'ils se mettent en mouvement, ou qu'ils soient découverts.

- Dans la version actuelle de l'algorithme de suivi temporel, nous avons uniquement considéré une application au codage et à l'interpolation. Il serait possible de l'adapter à une application pour l'analyse de séquences. Nous avons fait certains choix qu'il faudrait alors remettre en cause. Nous pensons notamment aux deux points suivants :

  ⋆ La reconstruction des points multiples après les phases de prédiction et d'ajustement affine se fait uniquement sur les informations géométriques. Il serait alors souhaitable de s'aider des informations image, par exemple en utilisant les techniques de fermeture de contours à faible distance.

  ⋆ Par souci d'économie, l'algorithme actuel ne prévoit pas possibilité de rajouter des sommets dans l'approximation polygonale des frontières. Ce serait souhaitable, surtout pour les frontières qui se découvrent progressivement.

- Il serait possible d'effectuer un codage plus performant de certains éléments :
  - ⋆ Les images d'erreur pourraient être encodées avec les techniques reconnues comme les plus efficaces actuellement (EZW ou EZT).
  - ⋆ Les paramètres de mouvement des régions ayant une forte cohérence temporelle, il serait possible de les encoder plus efficacement, par un codage prédictif (DPCM, par exemple). Des résultats préliminaires montrent que l'on pourrait descendre jusqu'à 3 bits par paramètre. Il faudrait aussi étudier l'effet de l'erreur introduite ainsi sur la qualité des images prédites par compensation de mouvement.
- On pourrait exploiter la notion de tube spatio-temporel pour effectuer un codage des informations images. Il s'agirait de réaliser une décomposition fréquentielle tridimensionnelle, orientée par le mouvement, des images de la séquence. L'idée serait de travailler région par région, sur les lignes définies par les pixels d'une région aux différents instants. Une difficulté importante de cette approche est qu'un pixel suivi dans le temps ne reste pas sur une position entière dans la grille d'échantillonnage.

Annexe A

Le filtrage de Kalman

A.1 Formalisme général

Le filtrage de Kalman consiste à estimer de façon optimale la valeur d'un vecteur d'état Ψ_t (espérance et matrice de covariance) sachant qu'à chaque instant on dispose d'un vecteur de mesure s_t qui est une combinaison linéaire bruitée des éléments de Ψ_t. L'algorithme utilise aussi la dynamique du vecteur d'état, bruitée de même. Il est optimal dans le sens où il prend en compte toutes ces informations et les combine au mieux de sorte à obtenir une estimation de variance minimale [Meditch 69] [Gelb 74] [Papoulis 84].

A.1.1 Équations d'évolution et d'observation

La première équation décrit l'évolution du vecteur d'état : il s'agit d'un modèle de variation linéaire bruité. La deuxième équation décrit les mesures bruitées que l'on effectue sur le système : il s'agit de combinaisons linéaires des composantes du vecteur d'état.

$$\left\{ \begin{array}{rcll} \Psi_{t+1} & = & A_t\Psi_t + w_t & \text{(dynamique bruitée du système)} \\ s_t & = & H_t\Psi_t + v_t & \text{(mesures bruitées)} \end{array} \right.$$

avec w_t et v_t deux bruits blancs indépendants tels que :

- $E[v_t] = 0$
- $E[w_t] = 0$
- $E[v_{t_1} v_{t_2}^T] = R\delta_{t_1 t_2}$
- $E[w_{t_1} w_{t_2}^T] = Q\delta_{t_1 t_2}$
- $E[v_t w_t^T] = 0$

où Q et R sont 2 matrices symétriques définies positives. R caractérise le bruit de mesure et Q le bruit de modèle.

A.1.2 Équations de filtrage et de prédiction

Connaissant la mesure s_t (et toutes les mesures antérieures), le problème est alors de trouver le meilleur estimateur

- de Ψ_t : il est noté $\Psi_{t/t}$ et s'appelle estimateur *a posteriori*. Notons $P_{t/t}$ sa matrice de covariance.
- de Ψ_{t+1} : il est noté $\Psi_{t+1/t}$ et s'appelle estimateur *a priori*. Notons $P_{t+1/t}$ sa matrice de covariance.

On définit par ailleurs K_t, appelé gain de Kalman qui pondère l'apport de l'innovation $s_t - H_t\Psi_{t/t-1}$ (différence entre l'observation et l'estimée *a priori* de cette observation) par rapport à la prédiction précédente $\Psi_{t/t-1}$.

Ces quantités sont données par les équations suivantes :

$$\left\{\begin{array}{rcl} K_t & = & P_{t/t-1}H_t^T(R + H_tP_{t/t-1}H_t^T)^{-1} \\ \Psi_{t/t} & = & \Psi_{t/t-1} + K_t(s_t - H_t\Psi_{t/t-1}) \\ P_{t/t} & = & (I - K_tH_t)P_{t/t-1} \end{array}\right. \qquad \text{(équations de filtrage)}$$

$$\left\{\begin{array}{rcl} \Psi_{t+1/t} & = & A_t\Psi_{t/t} \\ P_{t+1/t} & = & A_tP_{t/t}A_t^T + Q \end{array}\right. \qquad \text{(équations de prédiction)}$$

A.1.3 Initialisation

Il faut aussi fournir le vecteur d'état initial $\Psi_0 = \Psi_{0/-1}$ et sa matrice de covariance $P_0 = P_{0/-1}$.

A.2 Application au filtrage des paramètres de mouvement

A.2.1 Vecteurs d'état et d'observation

Les différents paramètres du mouvement d'une région sont filtrés indépendamment. Le modèle d'évolution du système est un mouvement à accélération constante. Si l'un quelconque des paramètres de mouvement (t_x^+, t_y^+, θ^+, k^+, a^+, b^+, c^+, $d^+ \ldots$) à l'instant t est noté ψ_t, on ne pose pas $\dddot{\psi}_t = 0$, mais cette dérivée est modélisée par un bruit gaussien centré en 0.

Le vecteur d'état est

$$\Psi_t = \begin{pmatrix} \psi_t \\ \dot{\psi}_t \\ \ddot{\psi}_t \end{pmatrix}.$$

Les différents paramètres de mouvement sont filtrés de façon indépendante : autant de filtres que $\Theta_{\mathcal{R}}^+$ a de composantes fonctionnent en parallèle. Une autre possibilité aurait été

de regrouper tous les paramètres du modèle de mouvement et leurs dérivées dans un même vecteur d'état. Nous l'avons écartée pour des raisons de simplicité opératoire.

Le vecteur d'observation est

$$s_t = (\psi_t)$$

car ψ_t est la seule sortie du système que l'on peut mesurer (estimation du mouvement). Les variables $\dot{\psi}_t$ et $\ddot{\psi}_t$ sont considérées comme non observables.

A.2.2 Équations d'évolution et d'observation

- A_t est constante au cours du temps et vaut $A = \begin{bmatrix} 1 & 1 & 1/2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$
- H_t est constante au cours du temps et vaut $H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$
- w_t et v_t deux bruits gaussiens de matrices de covariance Q et R.

A.2.3 Initialisation

Le filtre ne démarre pas à $t = 0$ mais à partir de $t = 2$. [Meyer 92] a montré que

- $Q = \sigma_w^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}$, T étant la période d'acquisition des images. Il suffit donc de fournir une seule variance σ_w pour chaque paramètre filtré.
- $\Psi_2 = \begin{pmatrix} \psi_2 \\ \frac{\frac{3}{2}\psi_2 - 2\psi_1 + \frac{1}{2}\psi_0}{T} \\ \frac{\psi_2 - 2\psi_1 + \psi_0}{T^2} \end{pmatrix}$
- $P_2 = \begin{bmatrix} \sigma_v^2 & \frac{3}{2T}\sigma_v^2 & \frac{1}{T^2}\sigma_v^2 \\ \frac{3}{2T}\sigma_v^2 & \frac{T^3}{3}\sigma_w^2 + \frac{13}{2T^2}\sigma_v^2 & \frac{9T^2}{40}\sigma_w^2 + \frac{6}{T^3}\sigma_v^2 \\ \frac{1}{T^2}\sigma_v^2 & \frac{9T^2}{40}\sigma_w^2 + \frac{6}{T^3}\sigma_v^2 & \frac{23T}{30}\sigma_w^2 + \frac{6}{T^4}\sigma_v^2 \end{bmatrix}$

Annexe B

Sur les modèles paramétriques de mouvement

B.1 Hiérarchie des modèles

Cette section reprend la hiérarchie des modèles de mouvement paramétriques définie dans [Nicolas 92]. Il s'agit de modèles du mouvement apparent entre deux images I_{t_1} et I_{t_2}, applicables sur le support d'une région. Nous considérons tantôt des modèles de transformation plane :

$$\Theta : \left| \begin{array}{c} \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} x' \\ y' \end{pmatrix} \end{array} \right.$$

tantôt des modèles de champs de déplacement 2D :

$$c : \left| \begin{array}{c} \mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} d_x \\ d_y \end{pmatrix} \end{array} \right.$$

ce qui est équivalent à une petite modification près des paramètres.

Modèle nul (0 paramètres) $\Theta = \vec{0}$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Une région suit ce modèle quand elle n'est pas en mouvement (fond immobile par exemple). Utiliser ce modèle revient à détecter les zones de I_{t_2} qui sont identiques dans I_{t_1} et à faire un rafraîchissement conditionnel. Ce modèle est utilisé quand le temps de compression/décompression doit être très faible.

Modèle constant (2 paramètres) $\Theta = [t_x, t_y]$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Ce modèle prend en compte les régions qui ont un mouvement de translation uniforme. C'est celui qui est utilisé dans la norme MPEG [Le Gall 91] [Le Gall 92], en association avec une segmentation en blocs. Cependant, MPEG se limite à $(t_x, t_y) \in \mathbb{Z}^2$ ou à $(2t_x, 2t_y) \in \mathbb{Z}^2$, alors qu'ici, $(t_x, t_y) \in \mathbb{R}^2$ et la quantification est faite ultérieurement.

Modèle affine simplifié (4 paramètres) $\Theta = [t_x, t_y, k, \Theta]$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} k & -\theta \\ \theta & k \end{bmatrix} \begin{pmatrix} x - x_C \\ y - y_C \end{pmatrix}$$

Le point C est le centre de référence du mouvement. Ses coordonnées ne font pas partie des paramètres du modèle car il ne s'agit pas de paramètres indépendants (voir la section suivante B.2).

Ce modèle prend en compte les régions qui ont un mouvement de translation, de rotation, ou qui subissent une homothétie. C'est l'un de ceux qui ont été utilisés dans cette étude, car il réalise un bon compromis entre la richesse de description du mouvement des modèles plus complexes et le coût de codage.

Le paramètre θ est appelé «angle de rotation», k est appelé «paramètre de divergence» et $\lambda = 1 + k$, un «rapport d'homothétie». Ceci est faux en toute rigueur, mais ces appellations sont justifiées par une approximation au 1^{er} ordre d'une matrice M représentant une composition de rotation et d'homothétie :

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} \lambda\cos\theta & -\lambda\sin\theta \\ \lambda\sin\theta & \lambda\cos\theta \end{bmatrix}$$

En effet, quand θ et k sont faibles (petit mouvement), les approximations suivantes sont justifiées :

$$\begin{cases} \cos\theta & \sim & 1 \\ \sin\theta & \sim & \theta \\ \lambda & \sim & 1 \end{cases}$$

Ainsi, la matrice de notre modèle qui vaut $M - I_2$ a pour approximation $\begin{bmatrix} k & -\theta \\ \theta & k \end{bmatrix}$.

Toutefois, cette approximation ne restreint pas la généralité du modèle : si l'on veut représenter exactement la rotation d'angle θ' et l'homothétie de rapport λ', il suffit de prendre

$$\begin{cases} \lambda & = & \lambda'\cos\theta' \\ \theta & = & \lambda'\sin\theta' \end{cases}$$

Modèle affine (6 paramètres) $\Theta = [t_x, t_y, a, b, c, d]$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{pmatrix} x - x_C \\ y - y_C \end{pmatrix}$$

Ce modèle rajoute au précédent des déformations comme les affinités et les cisaillements.

Modèle homographique (8 paramètres) $\Theta = [N_{xx}, N_{xy}, N_{yx}, N_{yy}, N_x, N_y, D_x, D_y]$

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} \frac{N_{xx}(x-x_C)+N_{xy}(y-y_C)+N_x}{D_x(x-x_C)+D_y(y-y_C)+1} \\ \frac{N_{yx}(x-x_C)+N_{yy}(y-y_C)+N_y}{D_x(x-x_C)+D_y(y-y_C)+1} \end{pmatrix}$$

Ce modèle est capable de décrire exactement tout mouvement d'un objet 3D plan projeté sur la rétine de la caméra. Il est donc intéressant pour des scènes comportant des objets artificiels plans (immeubles, routes, ...). Il a l'inconvénient de ne pas s'inscrire parfaitement dans la hiérarchie. Il n'a donc pas été envisagé par [Nicolas 92].

Modèle quadratique (12 paramètres) $\Theta = [t_x, t_y, a_1, a_2, a_3, a_4, a_5, b_1, b_2, b_3, b_4, b_5]$

$$\begin{aligned} \begin{pmatrix} d_x \\ d_y \end{pmatrix} &= \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{pmatrix} x - x_C \\ y - y_C \end{pmatrix} \\ &\quad + \begin{bmatrix} a_3 & a_4 \\ b_3 & b_4 \end{bmatrix} \begin{pmatrix} (x-x_C)^2 \\ (y-y_C)^2 \end{pmatrix} + \begin{bmatrix} a_5 \\ b_5 \end{bmatrix} (x-x_C)(y-y_C) \end{aligned}$$

Ce modèle est identique au modèle affine avec des termes supplémentaires du second degré. Il est rarement utilisé car les termes quadratiques sont difficiles à estimer.

Dans les sections qui suivent, nous prendront comme exemple le modèle affine, sachant qu'il est facile de généraliser au modèle affine simplifié.

B.2 Influence du déplacement du centre du mouvement

Le point C est le point dont le vecteur de déplacement vaut $\begin{pmatrix} t_x \\ t_y \end{pmatrix}$. Changer de point de référence ne change pas les paramètres de la matrice $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ mais a une influence sur les paramètres de translation $\begin{pmatrix} t_x \\ t_y \end{pmatrix}$. Si l'on veut passer de C à C', il faut effectuer le changement suivant :

$$\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} t_x - a(x_{C'} - x_C) - b(y_{C'} - y_C) \\ t_y - c(x_{C'} - x_C) - d(y_{C'} - y_C) \end{pmatrix} + \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{pmatrix} x - x_{C'} \\ y - y_{C'} \end{pmatrix}$$

B.3 Composition

Soit trois images et des mouvements Θ_1 et Θ_2 entre couples d'images consécutives. On cherche à calculer le mouvement entre les images extrêmes, qui est le mouvement

résultant de Θ_1 et Θ_2. Dans certains travaux, les auteurs se contentent d'additionner les paramètres des modèles. Il s'agit d'une approximation qui n'est valable que pour des petits mouvements. Pour l'interpolation avec des images de référence éloignées, nous pouvons être amenés à composer un grand nombre de mouvements, qui dont le résultat est de grande amplitude. En toute rigueur, nous devons donc composer les transformations planes qui sous-tendent ces mouvements. Soit p un point, $p' = \Theta_1(p)$ et $p'' = \Theta_2(p')$:

$$\Theta_1 : p' = M_1 p + t_1$$

$$\Theta_2 : p'' = M_2 p' + t_2$$

Il faut calculer la composition $\Theta_3 = \Theta_2 \circ \Theta_1$. Pour simplifier, nous avons supposer que les centres de ceux mouvements sont les mêmes, mais dans le cas contraire, il suffit juste d'effectuer une changement de centre, comme montré dans la section précédente B.2). On a alors :

$$p'' = M_2 M_1 p + M_2 t_1 + t_2$$

qui est aussi un mouvement affine de paramètres $M_3 = M_2 M_1$ et $t_3 = M_2 t_1 + t_2$.

B.4 Inversion

Soit un mouvement Θ de paramètre (M, t) :

$$\Theta : p' = Mp + t$$

Si la matrice M est inversible, on a $M^{-1}(p' - t) = p$ donc le mouvement inverse Θ^{-1} a pour paramètres $(M^{-1}, -M^{-1}t)$.

B.5 Structure de groupe

Si l'on considère l'ensemble des mouvements affines possibles, privé des mouvements non inversibles, et qu'on le munit de l'opération de composition, on obtient donc un groupe.

B.6 Racine carrée

Soit $\Theta_0 = (M_0, t_0)$ modèle affine simplifié. Le problème est de trouver $\Theta = (M, t)$ tel que $\Theta \circ \Theta = \Theta_0$. Soit p un point, $p' = \Theta(p)$ et $p'' = \Theta(p') = \Theta_0(p)$.

$$\begin{aligned} p' &= Mp + t \\ p'' &= Mp' + t \\ p'' &= M(Mp + t) + t \\ &= M^2 p + (M + I_2)t \end{aligned}$$

On procède par identification avec $p'' = M_0 p + t_0$. Or

$$M^2 = \begin{bmatrix} k^2 - \theta^2 & 2k\theta \\ -2k\theta & k^2 - \theta^2 \end{bmatrix}$$

Il faut donc résoudre le système d'équations

$$\begin{cases} k^2 - \theta^2 &= k_0 \\ 2k\theta &= \theta_0 \end{cases}$$

En substituant $\theta = \theta_0/2k$ dans (1), on obtient $4k^4 - 4k^2 k_0 - \theta_0^2 = 0$ qui est une équation bi-carrée. On pose $K = k^2$ et $K \geq 0$, et on a l'équation $4K^2 - 4Kk_0 - \theta_0^2 = 0$, ce qui donne

$$K = \frac{k_0 \pm \sqrt{k_0^2 + \theta_0^2}}{2} \text{ et } K \geq 0$$

on ne retient que la racine positive et finalement

$$\begin{cases} k &= \sqrt{\frac{k_0 + \sqrt{k_0^2 + \theta_0^2}}{2}} \\ \theta &= \frac{\theta_0}{2k} \end{cases}$$

Pour un modèle affine on arrive à une équation de degré 8 qui peut se réduire de même à une équation de degré 4, mais c'est la limite de ce que l'on peut résoudre analytiquement. Au delà, il faut avoir recours à des méthodes numériques de résolution.

Bibliographie

[Adelson et Wang 94] E.H. Adelson et J.Y.A Wang, « Representing Moving Images with layers » , *IEEE Trans. Image Process., Special Issue: Image sequence compression*, vol. 3, n° 5, septembre 1994, p. 625–638.

[Adiv 85] G. Adiv, « Determining three-dimensional motion and structure from optical flow generated by several moving objects » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, n° 4, juillet 1985, p. 384–401.

[Aho et al. 87] Aho, Hopcroft et Ullman, *Structures de données et algorithmes*, Intereditions, 1987.

[Aizawa et al. 89] K. Aizawa, T. Saito et H. Harashima, « Model-based analysis synthesis image coding system for a person's face » , *Signal Process. : Image Commun.*, vol. 1, 1989, p. 139–152.

[Altunbasak et al. 95] Y. Altunbasak, A.M. Tekalp et G. Bozdagi, « 2-D object based coding using a content-based mesh and affine motion parametrization » , *Proc. IEEE Int. Conf. Image Processing*, Washington, D.C., USA, octobre 1995.

[Altunbasak et Tekalp 96] Y. Altunbasak et A.M. Tekalp, « Occlusion-adaptative 2-D mesh tracking » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 2108–2111, Atlanta, Georgia, USA, mai 1996.

[Amini et al. 90] A.A. Amini, T.E. Weymouth et R.C. Jain, « Using dynamic programming for solving variational problems in vision » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, n° 9, septembre 1990, p. 855–867.

[Antonini et al. 90] M. Antonini, M. Barlaud, P. Mathieu et I. Daubechies, « Image coding using vector quantization in the wavelet transform domain » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 2297–2300, Albuquerque, New Mexico, avril 1990.

[Azencott 87] R. Azencott, « Markov Fields and Image Analysis » , *6ème Congrès de Rec. des Formes et Intelligence Artificielle*, AFCET-INRIA, p. 1183–1191, Antibes, France, novembre 1987.

[Bala et al. 97] L.-P. Bala, K. Talmi et Jin Liu, « Automatic Detection and Tracking of Faces and Facial Features in Video Sequences » , *Proc. Picture Coding Symposium*, p. 251–254, Berlin, Germany, septembre 1997.

[Bascle et al. 94] B. Bascle, P. Bouthemy, N. Deriche et F. Meyer, « Tracking complex primitives in an image sequence » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 426–431, Jerusalem, octobre 1994.

[Bascle 94] B. Bascle, *Contributions et applications des modèles déformables en vision par ordinateur*, Thèse de doctorat, Université de Nice Sophia-Antipolis, juillet 1994.

[Begen et Meyer 98] L. Begen et F. Meyer, « Segmentation du mouvement et détection de la profondeur relative des objets d'une scène » , *CORESA 98*, p. 42–49, CNET, Lannion, juin 1998.

[Benois et Barba 92a] J. Benois et D. Barba, « Image segmentation by region-contour cooperation as a basis for efficient coding scheme » , *Proc. SPIE (Visual Communications and Image Process.)*, p. 1218–1229, Boston, USA, novembre 1992.

[Benois et Barba 92b] J. Benois et D. Barba, « Image segmentation by region-contour cooperation for image coding » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 331–334, The Hague, The Netherlands, août 1992.

[Benois-Pineau et al. 96] J. Benois-Pineau, L. Wu et D. Barba, « Content-Based Border Preserving Coding for Structure Retrieval and Content Manipulation of Image Sequences » , *Proc. Picture Coding Symposium*, p. 553–557, Melbourne, Australia, mars 1996.

[Bentley et Ottman 79] J.L. Bentley et T.A. Ottman, « Algorithms for reporting and counting geometric intersections » , *IEEE Trans. Comput.*, vol. C-28, 1979, p. 643–647.

[Berger 91] M.O. Berger, *Les contours actifs : modélisation, comportements et convergence*, Thèse de doctorat, INPL–INRIA Lorraine, février 1991.

[Berger 93] M.O. Berger, « Tracking Rigid and non Polyhedral Objects in an Image Sequence » , *Proc. Scandinavian Conf. Image Analysis*, p. 945–957, Tromso, Norway, mai 1993.

[Bergeron et Dubois 90] C. Bergeron et E. Dubois, « Parametric block estimation of motion and application to temporal interpolation of video sequences » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 140–146, juin 1990.

[Bergeron et Dubois 91] C. Bergeron et E. Dubois, « Gradient based algorithms for bloc oriented map estimation of motion and application to motion-compensated temporal interpolation » , *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, mars 1991, p. 72–85.

[Besag 74] J. Besag, « Spatial Interaction and the Statistical Analysis of Lattice Systems » , *J. Royal Statist. Soc.*, vol. 36, Serie B, 1974, p. 192–236.

[Besag 86] J. Besag, « On the Statistical Analysis of Dirty Pictures » , *J. Royal Statist. Soc.*, vol. 48, Serie B, n° 3, 1986, p. 259–302.

[Beucher et Meyer 93] S. Beucher et F. Meyer, « The morphological approach to segmentation : the watershed transformation » , p. 433–482, *Mathematical morphology in image processing*, E.R. Dougherty ed., Marcel Dekker, 1993.

[Bierling et Thoma 86] M. Bierling et R. Thoma, « Motion Compensating Field Interpolation Using a Hierarchically Structured Displacement Estimator » , *Signal Process.*, vol. 11, 1986, p. 387–404.

[Black 92] M.J. Black, « Combining intensity and motion for incremental segmentation and tracking over long image sequences » , *Proc. European Conf. Computer Vision*, p. 485–493, Santa Margherita Liguere, Italy, mai 1992.

[Blake et al. 93] A. Blake, R. Curwen et A. Zisserman, « A Framework for Spatiotemporal Control in the Tracking of Visual Contours » , *Intern. J. Comput. Vis.*, vol. 11, n° 2, octobre 1993, p. 127–145.

[Blanc et Mohr 97] J. Blanc et R. Mohr, « Towards Fast and Realistic Image Synthesis from Real Views » , *Proc. Scandinavian Conf. Image Analysis*, Finland, 1997.

[Bonnaud et al. 95] L. Bonnaud, C. Labit et J. Konrad, « Interpolative coding of image sequences using temporal linking of motion-based segmentation » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 2265–2268, Detroit, Michigan, USA, mai 1995.

[Bonnaud et Labit 94] L. Bonnaud et C. Labit, *Étude d'algorithmes de suivi temporel de segmentation basée mouvement pour la compression de séquences d'images*, rapport technique n° 2253, INRIA, janvier 1994, Disponible à `ftp://ftp.inria.fr/-INRIA/tech-reports/RR/RR-2253.ps.gz`.

[Bouthemy et al. 97] P. Bouthemy, M. Gelgon et F. Ganansia, *A unified approach to shot change detection and camera motion characterization*, rapport technique n° 3304, Rapport de recherche INRIA, novembre 1997.

[Bouthemy et François 93] P. Bouthemy et E. François, « Motion segmentation and qualitative dynamic scene analysis from an image sequence » , *Intern. J. Comput. Vis.*, vol. 10, n° 2, 1993, p. 157–182.

[Bouthemy et Ganansia 96] P. Bouthemy et F. Ganansia, « Video partitioning and Camera motion characterization for Content-based Video Indexing » , *Proc. IEEE Int. Conf. Image Processing*, p. 905–909, Lausanne, septembre 1996.

[Bouthemy et Lalande 90] P. Bouthemy et P. Lalande, « Detection and Tracking of Moving Objects Based on a Statistical Regularization Method in Space and Time » , *Proc. European Conf. Computer Vision*, O. Faugeras ed., p. 307–311, Springer, Antibes, France, avril 1990.

[Bouthemy et Santillana Rivero 87] P. Bouthemy et J. Santillana Rivero, « A hierarchical likelihood approach for region segmentation according to motion-based criterion » , *Proc. IEEE Int. Conf. Computer Vision*, p. 463–467, London, UK, juin 1987.

[Bouthemy 89] P. Bouthemy, « A Maximum-Likelihood Framework for Determining Moving Edges » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, n° 5, mai 1989, p. 499–511.

[Buisson et al. 97] O. Buisson, B. Besserer, S. Boukir et F. Helt, « Deterioration Detection for Digital Film Restoration » , *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Porto Rico, juin 1997.

[Cafforio et al. 90] C. Cafforio, F. Rocca et S. Tubaro, « Motion compensated image interpolation » , *IEEE Trans. Commun.*, vol. 38, n° 2, février 1990, p. 215–222.

[Canny 83] J.F. Canny, *Finding edges and lines in images*, rapport technique n° 720, MIT, 1983.

[Caselles et al. 95] V. Caselles, R. Kimmel et G. Sapiro, « Goedesic active contours » , *Proc. IEEE Int. Conf. Computer Vision*, p. 694–699, Boston, MA, USA, juin 1995.

[Chahine et Konrad 94] M. Chahine et J. Konrad, « Estimation of trajectories for accelerated motion from time-varying imagery » , *Proc. IEEE Int. Conf. Image Processing*, p. 800–804, Austin, USA, novembre 1994.

[Chahine et Konrad 95] M. Chahine et J. Konrad, « Estimation and compensation of accelerated motion for temporal sequence interpolation » , *Signal Process. : Image Commun.*, vol. 7, novembre 1995, p. 503–527.

[Chan et Chin 92] W.S. Chan et F. Chin, « Approximation of polygonal curves with minimum number of line segments » , *Proc. ISAAC*, Sringer Verlag ed., p. 378–387, 1992.

[Chang et al. 94] M.M. Chang, M.I. Sezan et A.M. Tekalp, « An algorithm for simultaneous motion estimation and scene segmentation » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 221–224, Adelaide, Australia, avril 1994.

[Cherfaoui 95] M. Cherfaoui, *Indexation et consultation de documents vidéo*, Thèse de doctorat, Université de Rennes 1, janvier 1995.

[Chou et Brown 88] P.B. Chou et C.M. Brown, « Multimodal Reconstruction and Segmentation with Markov Random Fields and HCF Optimization » , *Proc. Image Understanding Workshop*, p. 214–221, Cambridge, MA, avril 1988.

[Cocquerez et Philipp 95] J.P. Cocquerez et S. Philipp, *Analyse d'images : filtrage et segmentation*, Masson, Collection enseignement de la physique, 1995.

[Cohen et Cohen 93] L.D. Cohen et I. Cohen, « Finite element methods for active contour models and balloons for 2D and 3D images » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, n° 11, novembre 1993, p. 587–600.

[Cohen 91] L. Cohen, « On active contour models and balloons » , *CVGIP: Image Underst.*, vol. 53, n° 2, 1991, p. 211–218.

[Cootes et al. 95] T.F. Cootes, D. Cooper, C.J. Taylor et J. Graham, « Active Shape Models – Their Training and Application » , *CVGIP: Image Underst.*, vol. 61, n° 1, janvier 1995, p. 38–59.

[Coster et Chermant 85] M. Coster et J.L. Chermant, *Précis d'analyse d'images*, Éditions du CNRS, 1985.

[Daubechies 88] I. Daubechies, « Orthonormal bases of compactly supported wavelets » , *Communications on Pure and Applied Math.*, vol. 41, 1988, p. 909–996.

[Davoine 95] F. Davoine, *Compression d'images par fractales basée sur la triangulation de Delaunay*, Thèse de doctorat, INPG, décembre 1995.

[de Berg et al. 97] M. de Berg, M. van Kreveld, M. Overmars et O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer-Verlag, Berlin, RFA, 1997.

[De Haan et Bellers 98] G. De Haan et E.B. Bellers, « Deinterlacing — An overview » , *Proc. IEEE*, vol. 86, n° 9, septembre 1998, p. 1839–1857.

[Delagnes et al. 95] P. Delagnes, J. Benois et D. Barba, « Active contours approach to object tracking in image sequences with complex background » , *Pattern Recognit. Lett.*, vol. 16, n° 2, 1995, p. 171–178.

[Depommier et Dubois 92] R. Depommier et E. Dubois, « Motion-compensated temporal prediction for interlaced image sequences » , *Proc. IEEE Workshop Visual Signal Process. Comm.*, p. 264–269, septembre 1992.

[Deriche et Faugeras 90] R. Deriche et O. Faugeras, « Tracking Line Segments » , *Proc. European Conf. Computer Vision*, O. Faugeras ed., p. 259–268, Springer, Antibes, France, avril 1990.

[Deriche 87] R. Deriche, « Using Canny's criteria to derive an optimal edge detector recursively implanted » , *Intern. J. Comput. Vis.*, vol. 1, n° 2, avril 1987, p. 167–187.

[Derin et Elliot 87] H. Derin et H. Elliot, « Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, n° 1, janvier 1987, p. 39–55.

[Diehl 91] N. Diehl, « Object-oriented motion estimation and segmentation in image sequences » , *Signal Process. : Image Commun.*, vol. 3, 1991, p. 23–56.

[Dubois 92] E. Dubois, « Motion-compensated filtering of time-varying images » , *Multidimens. Syst. Signal Process.*, vol. 3, 1992, p. 211–239.

[Dudon et al. 95] M. Dudon, O. Avaro et C. Roux, « Triangle-based motion estimation and temporal interpolation » , *Proc. IEEE Workshop on nonlinear signal processing*, p. 242, Halkidiki, Greece, 1995.

[Dudon 96] M. Dudon, *Modélisation du mouvement par treillis actifs et méthodes d'estimation associées. Application au codage de séquences d'images*, Thèse de doctorat, CNET/ENSTB — Université de Rennes 1, décembre 1996.

[Ebrahimi 97] T. Ebrahimi (éd.), mpeg-*4 Video Verification Model Version 8.0*, ISO/IEC JTC1/SC29/WG11 MPEG97/N1796, Stockholm, Sweden, juillet 1997.

[Fan et Gan 98] J. Fan et F. Gan, « Adaptive motion-compensated interpolation based on spatiotemporal segmentation » , *Signal Process. : Image Commun.*, vol. 12, mars 1998, p. 59–70.

[Faugeras et Laveau 94] O. Faugeras et S. Laveau, « Representing Three-Dimensional Data as a Collection of Images and Fundamental Matrices for Image Synthesis » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 689–691, Jerusalem, Israel, 1994.

[François 91] É. François, *Interprétation qualitative du mouvement partir d'une séquence d'images*, Thèse de doctorat, Irisa–Université de Rennes 1, juin 1991.

[Fredman et Tarjan 87] M.L. Fredman et R.E. Tarjan, « Fibonnacci heaps and their uses in improved network optimization algorithms » , *Journal of the ACM*, vol. 34, 1987, p. 596–615.

[Freeman 61] H. Freeman, « On the encoding of arbitrary geometric configurations » , *IEEE Trans. Comput.*, vol. C10, 1961, p. 260–268.

[Garcia-Garduño 96] V. Garcia-Garduño, *Une approche de compression orientée objets par suivi de segmentation basée mouvement pour le codage de séquences d'images numériques*, Thèse de doctorat, Irisa–Université de Rennes 1, février 1996.

[Geiger et al. 95] D. Geiger, A. Gupta, L.A. Costa et J. Vlontzos, « Dynamic programming for detecting, tracking, and matching deformable contours » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, n° 3, mars 1995, p. 294–302.

[Gelb 74] A. GELB, *Applied optimal estimation*, The MIT Press, Cambridge, 1974.

[Geman et Geman 84] S. GEMAN et D. GEMAN, « Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, n° 6, novembre 1984, p. 721–741.

[Gennert et Yuille 88] M.A. GENNERT et A.L. YUILLE, « Determining the optimal weights in multiple objective function optimization » , *Proc. IEEE Int. Conf. Computer Vision*, p. 87–89, Tarpon Springs, Florida, décembre 1988.

[Girod 92] B. GIROD, « Psychovisual aspects of image communication » , *Signal Process.*, vol. 28, 1992, p. 239–251.

[Gondran et Minoux 86] M. GONDRAN et M. MINOUX, *Graphes et algorithmes*, Eyrolles, 1986.

[Gu et Kunt 95] C. GU et M. KUNT, « Contour simplification and motion compensated coding » , *Signal Process. : Image Commun.*, vol. 7, 1995, p. 279–296.

[Gupta et Gersho 92] S. GUPTA et A. GERSHO, « Joint motion compensated prediction and interpolation of video sequences » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, IEEE, p. 457–460, IEEE New York, NY, USA, 1992.

[Hall et al. 97] J. HALL, D. GREENHILL et G.A. JONES, « Segmenting Film Sequences Using Active Surfaces » , *Proc. IEEE Int. Conf. Image Processing*, p. 751–754, Santa Barbara, CA, USA, octobre 1997.

[Han et Woods 97] S.-C. HAN et J.W. WOODS, « Frame-Rate Up-Conversion Using Transmitted Motion and Segmentation Fields for Very Low Bit-Rate Video Coding » , *Proc. IEEE Int. Conf. Image Processing*, p. 747–750, Santa Barbara, CA, USA, octobre 1997.

[Hanen et Barba 93] J. HANEN et D. BARBA, « High quality subband image coding of TV signals at 5 Mbit/s with motion compensation interpolation and visually optimized scalar quantization » , *Proc. SPIE (Visual Communications and Image Process.)*, p. 1477–1485, 1993.

[Haralick et Shapiro 92] R.M. HARALICK et L.G. SHAPIRO, *Computer and robot vision, volume 1*, chapitre 5. Mathematical morphology de , Addison Wesley, 1992.

[Healey 93] G. HEALEY, « Hierarchical segmentation-based approach to motion analysis » , *Image Vis. Comput.*, vol. 11, n° 9, novembre 1993, p. 570–576.

[Heitz et Bouthemy 90a] F. HEITZ et P. BOUTHEMY, « Motion estimation and segmentation using a global bayesian approach » , *IEEE Trans. Acoust. Speech Signal Process.*, p. 2305–2308, Albuquerque, USA, avril 1990.

[Heitz et Bouthemy 90b] F. HEITZ et P. BOUTHEMY, « Multimodal motion estimation and segmentation using Markov random fields » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 378–383, Atlantic City, USA, juin 1990.

[Heitz et Bouthemy 93] F. HEITZ et P. BOUTHEMY, « Multimodal estimation of discontinuous optical flow using Markov random fields » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, n° 12, décembre 1993, p. 1217–1232.

[Hobson et Carmen 91] W.J. HOBSON et P.R. CARMEN, « Real time TV image compression using motion-compensated frame prediction and interpolation » , *IEE Collo-*

quium on 'Adaptive Interpolation in Images' (Digest No.112), IEE, p. 3/1–3/6, IEE London, UK, 1991.

[Horn et Schunck 81] B.K.P. Horn et B.G. Schunck, « Determining optical flow » , *Artif. Intell.*, vol. 17, 1981, p. 185–203.

[Hsu et Liu 97] P. Hsu et K.J. Ray Liu, « An Adaptive Interpolation Scheme for 2-D Mesh Motion Compensation » , *Proc. IEEE Int. Conf. Image Processing*, p. 646–649, Santa Barbara, CA, USA, octobre 1997.

[Huang et Hsu 94] C.-L. Huang et C.-Y. Hsu, « A new motion compensation method for image sequence coding using hierarchical grid interpolation » , *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, n° 1, 1994, p. 42–52.

[Huang et Mersereau 94] J. Huang et R.M. Mersereau, « Multi-frame Pel-recursive Motion Estimation for Video Image Interpolation » , *Proc. IEEE Int. Conf. Image Processing*, p. 267–271, IEEE Comput. Soc. Press Los Alamitos, CA, USA, 1994.

[Huffman 51] D.A. Huffman, « A method for the construction of minimum redundancy codes » , *Proc. I.R.E.*, vol. 40, 1951, p. 1098–1101.

[Isard et Blake 98a] M. Isard et A. Blake, « Condensation – Conditional Density Propagation for Visual Tracking » , *Intern. J. Comput. Vis.*, vol. 29, n° 1, 1998, p. 5–28.

[Isard et Blake 98b] M. Isard et A. Blake, « ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework » , *Proc. European Conf. Computer Vision*, p. 893–908, 1998.

[Jain et Jain 81] J.R. Jain et A.K. Jain, « Displacement measurement and its application in interframe image coding » , *IEEE Trans. Commun.*, vol. 29, n° 12, décembre 1981, p. 1799–1808.

[Kampmann 97] M. Kampmann, « Precise Face Model Adaptation for Semantic Coding of Videophone Sequences » , *Proc. Picture Coding Symposium*, p. 675–680, Berlin, Germany, septembre 1997.

[Kass et al. 87] M. Kass, A. Witkin et D. Terzopoulos, « Snakes : Active Contour Models » , *Proc. IEEE Int. Conf. Computer Vision*, p. 259–268, London, UK, juin 1987.

[Kass et al. 88] M. Kass, A. Witkin et D. Terzopoulos, « Snakes : active contour models » , *Intern. J. Comput. Vis.*, vol. 1, 1988, p. 321–331.

[Kawaguchi et Mitra 97] K. Kawaguchi et S.K. Mitra, « Frame Rate Up-Conversion Considering Multiple Motion » , *Proc. IEEE Int. Conf. Image Processing*, p. 727–730, Santa Barbara, CA, USA, octobre 1997.

[Kervrann et Heitz 93] C. Kervrann et F. Heitz, *Segmentation non supervisée des images texturées : une approche statistique*, rapport technique n° 716, IRISA, mars 1993.

[Kervrann et Heitz 94] C. Kervrann et F. Heitz, « Robust tracking of stochastic deformable models in image sequences » , *Proc. IEEE Int. Conf. Image Processing*, p. 88–92, Seattle, USA, novembre 1994.

[Kervrann 95] C. KERVRANN, *Modèles statistiques pour la segmentation et le suivi de structures déformables bidimensionnelles dans une séquence d'images*, Thèse de doctorat, IRISA–Université de Rennes 1, novembre 1995.

[Keys 81] R.G. KEYS, « Cubic convolution interpolation for digital image processing » , *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-29, n° 6, décembre 1981, p. 1153–1160.

[Kharathisvili et al. 92] N. KHARATHISVILI, J. RONSIN, V. MATOSOV et G. KAKUBERI, « Analysis and motion estimation in coding and interpolation for the videophone » , *Proc. Int. Conf. Image Processing and Applications*, p. 73–76, IEE London, UK, 1992.

[Kim et Park 92] JOON-SEEK KIM et RAE-HONG PARK, « Local motion-adaptive interpolation technique based on block matching algorithms » , *Signal Process. : Image Commun.*, vol. 4, n° 6, 1992, p. 519–28.

[Kiranyaz et Onural 97] S. KIRANYAZ et L. ONURAL, « Motion Compensated Frame Interpolation Techniques for VLBR Coding » , *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, CA, USA, octobre 1997.

[Kokaram et Godsill 97] A.C. KOKARAM et S.J. GODSILL, « Joint Detection, Interpolation, Motion and Parameter Estimation for Image Sequences with Missing Data » , *Proc. IEEE Int. Conf. Image Processing*, p. 191–194, Santa Barbara, CA, USA, octobre 1997.

[Konrad et Dubois 88a] J. KONRAD et E. DUBOIS, « Estimation of Image Motion Fields : Bayesian Formulation and Stochastic Solution » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 1072–1075, New-York, USA, 1988.

[Konrad et Dubois 88b] J. KONRAD et E. DUBOIS, « Multigrid bayesian estimation of image motion fields using stochastic relaxation » , *Proc. IEEE Int. Conf. Computer Vision*, p. 354–362, Tarpon Springs, Florida, décembre 1988.

[Konrad et Dubois 92] J. KONRAD et E. DUBOIS, « Bayesian Estimation of Motion Vector Fields » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, n° 9, septembre 1992, p. 910–927.

[Konrad 88] JANUSZ KONRAD, *Motion-Compensated Interpolation for TV Frame Rate Conversion*, rapport technique n° 88-26, INRS-Télécommunications, Université du Québec, octobre 1988.

[Konrad 89] J. KONRAD, *Bayesian estimation of motion fields from image sequences*, Phd thesis, McGill University, Dept. Electr. Eng., juin 1989.

[Kropatsch 94] W.G. KROPATSCH, *Building irregular pyramids by dual graph contraction*, rapport technique n° 35, Dept. of Pattern Recognition and Image Processing, Institute for automation, Technical University of Vienna, juillet 1994.

[Kropatsch 95] W.G. KROPATSCH, *Equivalent Contraction kernels and the domain of dual irregular pyramids*, rapport technique n° 42, Dept. of Pattern Recognition and Image Processing, Institute for automation, Technical University of Vienna, novembre 1995.

[Kruse 96] S.-M. KRUSE, « Scene Segmentation from Dense Displacement Vector Fields using randomized Hough Transform » , *Image Commun.*, vol. 9, 1996, p. 29–41.

[Lagendijk et Sezan 92] R.L. Lagendijk et M.I. Sezan, « Motion compensated frame rate conversion of motion pictures » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 453–456, San Francisco, USA, 1992.

[Lalande et Bouthemy 90] P. Lalande et P. Bouthemy, « A statistical approach to the detection and tracking of moving objects in an image sequence » , *Proc. European Signal Process. Conf.*, p. 947–950, Barcelona, Spain, septembre 1990.

[Lalande 90] P. Lalande, *Détection du mouvement apparent dans une séquence d'images selon une approche markovienne; Application à la robotique sous-marine*, Thèse de doctorat, Irisa–Université de Rennes 1, mars 1990.

[Langdon et Rissanen 81] G. Langdon et J. Rissanen, « Compression of black-white images with arithmetic coding » , *IEEE Trans. Commun.*, vol. COM-29, n° 6, juin 1981, p. 858–867.

[Le Floch 97] H. Le Floch, *Échantillonnage irrégulier et interpolation : application a la représentation d'images fixes et animées*, Thèse de doctorat, Irisa–Université de Rennes 1, décembre 1997.

[Le Gall 91] D. Le Gall, « mpeg : a video compression standard for multimedia applications » , *Communications of the ACM*, vol. 34, avril 1991, p. 46–58.

[Le Gall 92] D. Le Gall, « The mpeg video compression algorithm » , *Signal Process. : Image Commun.*, vol. 4, avril 1992, p. 129–140.

[Leclerc 89] Yvan G. Leclerc, « Constructing Simple Stable Descriptions for Image Partitioning » , *Intern. J. Comput. Vis.*, vol. 3, 1989, p. 73–102.

[Lee et al. 96] Seung-Yong Lee, Kyung-Yong Chwa, J. Hahn et Sung Yong Shin, « Image Morphing Using Deformation Techniques » , *The Journal of Visualization and Computer Animation*, vol. 7, n° 1, janvier–mars 1996, p. 3–24.

[Leymarie et Levine 93] F. Leymarie et M.D. Levine, *Tracking Deformable Objects in the Plane Using an Active Contour Model*, 1993, To Appear in IEEE Trans. PAMI.

[Mardia et al. 92] K.V. Mardia, T.J. Hainsworth et J.F. Haddon, « Deformable templates in image sequences » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 132–135, La Haye, The Netherlands, septembre 1992.

[Maurizot 97] M. Maurizot, *Analyse du mouvement fluide 2D dans des séquences d'images numériques : localisation, caractérisation et suivi des points singuliers*, Thèse de doctorat, Irisa–Université de Rennes 1, décembre 1997.

[Meditch 69] J.S. Meditch, *Stochastic optimal linear estimation and control*, McGraw Hill, 1969.

[Mehlhorn et Näher 89] K. Mehlhorn et S. Näher, « LEDA : a library of efficient data types and algorithms » , *LNCS*, vol. 379, 1989, p. 88–106.

[Melkman et O'Rourke 88] A. Melkman et J. O'Rourke, « On polygonal chain approximation » , *Computational morphology*, G.T. Toussaint ed., North Holland, 1988.

[Meyer et Beucher 90] F. Meyer et S. Beucher, « Morphological segmentation » , *J. Vis. Commun. Image Represent.*, vol. 1, n° 1, septembre 1990, p. 21–46.

[Meyer et Bouthemy 92] F. Meyer et P. Bouthemy, « Region-based tracking in an image sequence » , *Proc. European Conf. Computer Vision*, G. Sandini ed., p. 476–484, Springer-Verlag, Santa Margherita, Italy, mai 1992.

[Meyer et Bouthemy 94] F. Meyer et P. Bouthemy, « Region-based tracking using affine motion models in long image sequences » , *CVGIP: Image Underst.*, vol. 60, n° 2, septembre 1994, p. 119–140.

[Meyer 92] F. Meyer, *Region-based tracking in an image sequence*, rapport technique n° 1723, INRIA, juillet 1992.

[Meyer 93] F. Meyer, *Suivi de régions et analyse des trajectoires dans une séquence d'images*, Thèse de doctorat, Irisa–Université de Rennes 1, juin 1993.

[Meyer 96] F. Meyer, « A morphological interpolation method for mosaic images » , *Mathematical morphology and its applications to image and signal processing, Proceedings of ISMM*, p. 337–344, Atlanta, 1996.

[Mémin 93] É. Mémin, *Algorithmes et architectures parallèles pour les approches markoviennes en analyse d'images*, Thèse de doctorat, Irisa–Université de Rennes 1, juin 1993.

[Monga 87] O. Monga, « An optimal region growing algorithm for image segmentation » , *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 1, n° 3, 1987, p. 351–375.

[Monga 88] O. Monga, *Segmentation d'images par croissance hiérarchique de régions*, Thèse de doctorat, Université de Paris 11, 1988.

[Mori et al. 91] L. Mori, F. Rocca et S. Tubaro, « Motion compensated interpolation using foreground/background segmentation » , *Proceedings of the International Conference on Digital Signal Processing*, V. Cappellini et A.G. Constantinides ed., Eur. Assoc. Signal Process.; IEEE; Univ. London; Univ. Florence; et al, p. 379–384, Elsevier Amsterdam, Netherlands, 1991.

[Morier et al. 98] F. Morier, J. Benois, D. Barba, H. Nicolas et H. Sanson, « Détection de la profondeur relative des objets en mouvement dans les séquences monoculaires » , *CORESA 98*, p. 34–41, CNET, Lannion, juin 1998.

[Morier 98] F. Morier, *Méthodes de représentation hiérarchique du contenu des séquences d'images animées*, Laboratoire systèmes électroniques et informatiques, Thèse de doctorat, IRESTE–Université de Nantes, septembre 1998.

[Moulet et Barba 88] D. Moulet et D. Barba, « Temporal following of spatial segmentation in image sequences » , *Proc. European Signal Process. Conf.*, p. 39–42, Grenoble, France, septembre 1988.

[Murray et Buxton 87] D.W. Murray et H. Buxton, « Scene segmentation from visual motion using global optimization » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, n° 2, mars 1987, p. 220–228.

[Musmann et al. 89] H. Musmann, M. Hötter et J. Ostermann, « Object-oriented analysis-synthesis coding of moving images » , *Signal Process. : Image Commun.*, vol. 1, n° 2, 1989, p. 117–138.

[Myers 85] E.W. Myers, « An $O(E \log E + I)$ expected time algorithm for the planar segment intersection problem » , *SIAM J. Comput.*, vol. 14, n° 3, août 1985, p. 625–636.

[Nagel et Enkelmann 81] H.H. Nagel et W. Enkelmann, « An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences » , *IEEE Trans. Commun.*, vol. 29, n° 12, décembre 1981, p. 1799–1808.

[Nelson et Gailly 92] M. Nelson et J.-L. Gailly, *The data compression book*, IDG Books, 1992.

[Netravali et Robbins 79] A.N. Netravali et J.D. Robbins, « Motion-compensated television coding: part I » , *Bell Syst. Tech. Journal*, vol. 58, n° 3, mars 1979, p. 631–670.

[Nguyen et Dubois 90] H.Q. Nguyen et E. Dubois, « Representation of motion information for image coding » , *Proc. Picture Coding Symposium*, p. 8.4–1 – 8.4–5, Cambridge, MA, USA, mars 1990.

[Nguyen 90] H.Q. Nguyen, *Unpublished internal report*, rapport technique, INRS-Telecom, 1990.

[Nguyen 95] E. Nguyen, *Compression sélective et focalisation visuelle : application au codage hybride de séquences d'images*, Thèse de doctorat, Irisa–Université de Rennes 1, décembre 1995.

[Nicolas et al. 93] H. Nicolas, J. Konrad et C. Labit, « Joint estimation of motion and illumination variations for coding of image sequences » , *Proc. Scandinavian Conf. Image Analysis*, p. 507–514, Tromso, Norway, mai 1993.

[Nicolas et Labit 93] H. Nicolas et C. Labit, « Motion and illumination variation estimation using a hierarchy of models: application to image sequence coding » , *J. Vis. Commun. Image Represent., Special issue: image sequence processing and motion analysis in visual communication*, vol. 6, n° 4, décembre 1993, p. 303–316.

[Nicolas 92] H. Nicolas, *Hiérarchie de modèles de mouvement et méthodes d'estimation associées. Application au codage de séquences d'images.*, Thèse de doctorat, Irisa–Université de Rennes 1, septembre 1992.

[Nzomigni 95] V. Nzomigni, *Compression sans pertes de séquences d'images biomédicales*, Thèse de doctorat, Irisa–Université de Rennes 1, décembre 1995.

[Odobez et Bouthemy 95] J.-M. Odobez et P. Bouthemy, « MRF-Based motion segmentation exploiting a 2D motion model robust estimation » , *Proc. IEEE Int. Conf. Image Processing*, p. 628–632, Washington, octobre 1995.

[Odobez 94] Jean-Marc Odobez, *Estimation, détection et segmentation du mouvement: une approche robuste et markovienne*, Thèse de doctorat, Irisa–Université de Rennes 1, décembre 1994.

[Papoulis 84] Papoulis, *Probability, Random Variables and Stochastic Processes*, Mc. Graw-Hill, 1984.

[Paragios et Deriche 97] N. Paragios et R. Deriche, « Detecting Multiple Moving Targets Using Deformable Contours » , *Proc. IEEE Int. Conf. Image Processing*, p. 183–186, Santa Barbara, CA, USA, octobre 1997.

[Paragios et Deriche 98] N. PARAGIOS et R. DERICHE, « A PDE-based level set approach for detection and tracking of moving objects » , *Proc. IEEE Int. Conf. Computer Vision*, Bombay, India, octobre 1998.

[Pardas et al. 94] M. PARDAS, P. SALEMBIER et B. GONZALEZ, « Motion and region overlapping estimation for segmentation-based video coding » , *Proc. IEEE Int. Conf. Image Processing*, p. 428–432, Austin, Texas, novembre 1994.

[Pateux et Labit 97] S. PATEUX et C. LABIT, *Codage efficace de carte de segmentation pour la compression orientée régions de séquences d'images*, rapport technique n° 1073, INRIA, janvier 1997.

[Pateux et Labit 98] S. PATEUX et C. LABIT, « Region-based video coder using the MDL formalism » , *Proc. IEEE Int. Conf. Image Processing*, p. 304–308, Chicago, Illinois, USA, octobre 1998.

[Pearlman et Abdel-Malek 92] W.A. PEARLMAN et A. ABDEL-MALEK, « Medical image sequence interpolation via hierarchical pel-recursive motion estimation » , *Proceedings. Fifth Annual IEEE Symposium on Computer-Based Medical Systems*, IEEE, p. 232–241, IEEE Comput. Soc. Press Los Alamitos, CA, USA, 1992.

[Pereira 96] F. PEREIRA, « MPEG4 : A new challenge for the representation of audio-visual information » , *Proc. Picture Coding Symposium*, p. 7–16, Melbourne, Australia, mars 1996.

[Perez 93] P. PEREZ, *Champs Markoviens et analyse multirésolution de l'image : application à l'analyse du mouvement*, Thèse de doctorat, IRISA–Université de Rennes 1, juillet 1993.

[Press et al. 92] W.H. PRESS, S.A. TEUKOLSKY, W.T. VETTERLING et B.P. FLANNERY, *Numerical Recipies in C – Second Edition*, Cambridge University Press, 1992.

[Puri et Haskell 92] A. PURI et B.G. HASKELL, « Digital HDTV coding with motion compensated interpolation » , *Signal Processing of HDTV, III. Proceedings of the Fourth International Workshop on HDTV and Beyond*, H. Yasuda et L. Chiaglione ed., p. 531–537, Elsevier Amsterdam, Netherlands, 1992.

[Ramchandran et al. 94] K. RAMCHANDRAN, A. ORTEGA et M. VETTERLI, « Bit allocation for dependant quantization with applications to multiresolution and MPEG video coders » , *IEEE Trans. Image Process.*, vol. 3, n° 5, septembre 1994.

[RC et Sklansky 93] J. RIBAS-CORBERA et J. SKLANSKY, « Interframe interpolation of cinematic sequences » , *J. Vis. Commun. Image Represent.*, vol. 4, n° 4, 1993, p. 392–406.

[Richardson et al. 96] I.E.G. RICHARDSON, I.J. HUNTER et M.J. RILEY, « Temporal smoothing of coded video data » , *Proc. Picture Coding Symposium*, p. 375–380, Melbourne, Australia, mars 1996.

[Ricquebourg et Bouthemy 95] Y. RICQUEBOURG et P. BOUTHEMY, « Subpixel estimation of normal displacements along contours using MRF-models » , *Proc. IEEE Int. Conf. Image Processing*, p. 288–291, Washington D.C., USA, octobre 1995.

[Rissanen 76] J.J. RISSANEN, « Generalized Kraft inequality and arithmetic coding » , *IBM Journal of Research and Development*, vol. 20, n° 3, mai 1976, p. 198–203.

[Robert 92] P. Robert, « Motion compensating interpolation considering occluding, appearing and disappearing areas » , *Proceedings of the Fourth International Workshop on HDTV and Beyond*, H. Yasuda et L. Chiaglione ed., p. 329–341, Elsevier Amsterdam, Netherlands, Turin, Italy, septembre 1992.

[Rougée et al. 88] A. Rougée, B. Levy et A. Willsky, « Reconstruction of two-dimensional velocity fields as a linear estimation problem » , *Proc. IEEE Int. Conf. Computer Vision*, p. 646–650, Tarpon Springs, Florida, décembre 1988.

[Ruprecht et Muller 95] D. Ruprecht et H. Muller, « Image Warping with Scattered Data Interpolation » , *IEEE Computer Graphics and Applications*, vol. 15, n° 2, mars 1995, p. 37–43.

[Salembier et Pardas 94] P. Salembier et M. Pardas, « Hierarchical morphological segmentation for image sequence coding » , *IEEE Trans. Image Process.*, vol. 3, n° 5, septembre 1994.

[Samal et Iyengar 92] A. Samal et P.A. Iyengar, « Automatic recognition and anlysis of human faces and facial expressions: a survey » , *Pattern Recognit.*, vol. 25, n° 1, 1992, p. 65–77.

[Sanson 95] H. Sanson, *Analyse de mouvement à base de régions appliquée au codage à réduction de débit de séquences télévisuelles*, Rapport de pré-soutenance, CNET/CCETT, mars 1995.

[Schmitt et Mattioli 94] M. Schmitt et J. Mattioli, *Morphologie mathématique*, Masson, 1994.

[Shahraray et Anderson 89] B. Shahraray et D.J. Anderson, « Optimal estimation of contour properties by cross-validated regularization » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, n° 6, juin 1989, p. 600–610.

[Shen et Castan 92] J. Shen et S. Castan, « An optimal linear operator for step edge detection » , *Comput. Vis. Graph. Image Process.*, vol. 54, n° 2, mars 1992, p. 13–17.

[Shi et Malik 97] J. Shi et J. Malik, « Normalized Cuts and Image Segmentation » , *Proc. IEEE Conf. Computer Vision Pattern Recognition*, Porto Rico, juin 1997.

[Shi et Malik 98] J. Shi et J. Malik, « Motion Segmentation and Tracking Using Normalized Cuts » , *Proc. IEEE Int. Conf. Computer Vision*, Bombay, India, janvier 1998.

[Stiller 93] C. Stiller, « A statistical image model for motion estimation » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 193–196, Mineapolis, USA, avril 1993.

[Sullivan et Baker 91] G.J. Sullivan et R.L. Baker, « Motion compensation for video compression using control grid interpolation » , *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, p. 2713–2716, Toronto, Canada, mai 1991.

[Taubman et Zakhor 94] D. Taubman et A. Zakhor, « Multirate 3-D subband coding of video » , *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, n° 5, septembre 1994, p. 572–589.

[Terzopoulos et Szeliski 92] D. Terzopoulos et R. Szeliski, « Tracking with Kalman Snakes » , p. 3–20, *Active Vision*, A. Blake et A. Yuille ed., MIT Press, Cambridge, MA, 1992.

[Thoma et Bierling 89] R. Thoma et M. Bierling, « Motion compensating interpolation considering covered and uncovered background » , *Signal Process. : Image Commun.*, vol. 1, n° 2, octobre 1989, p. 191–212.

[Toklu et al. 96] C. Toklu, A.M. Tekalp, A.T. Erdem et M.I. Sezan, « 2-D mesh-based tracking of deformable objects with occlusion » , *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, septembre 1996.

[Toklu et al. 97a] C. Toklu, A.M. Tekalp et A.T. Erdem, « 2-D triangular mesh-based mosaicking for object tracking in the presence of occlusion » , *Proc. SPIE (Visual Communications and Image Process.)*, p. 328–337, San Jose, CA, USA, février 1997.

[Toklu et al. 97b] C. Toklu, A.M. Tekalp et A.T. Erdem, « Object-based video coding using forward tracking 2-D mesh layers » , *Proc. SPIE (Visual Communications and Image Process.)*, p. 699–710, San Jose, CA, USA, février 1997.

[Tom et Katsaggelos 95] B.C. Tom et A.K. Katsaggelos, « Reconstruction of a high-resolution image by simultaneous registration, restoration and interpolation of low-resolution images » , *Proc. IEEE Int. Conf. Image Processing*, p. 539–542, Washingtion, D.C., USA, octobre 1995.

[Tubaro et Rocca 93] S. Tubaro et F. Rocca, « Motion field estimators and their application to image interpolation » , chapitre 6, p. 153–187, *Motion Analysis and Image Sequence Processing*, M.I. Sezan et R.L. Lagendijk ed., Kluwer Academic Publishers, 1993.

[Tziritas et Labit 94a] G. Tziritas et C. Labit, *Motion analysis for image sequence coding*, Elsevier Science, 1994, *Advances in Image Communication*, 366 pages.

[Tziritas et Labit 94b] G. Tziritas et C. Labit, *Motion analysis for image sequence coding — Motion-compensated image interpolation*, *Advances in Image Communication*, chapitre 7, p. 269–285, Advances in Image Communication, Elsevier Science, 1994.

[Tziritas et Labit 94c] G. Tziritas et C. Labit, *Motion analysis for image sequence coding — Two-dimentional motion analysis*, *Advances in Image Communication*, chapitre 3, p. 69–140, Advances in Image Communication, Elsevier Science, 1994.

[Tziritas 90] G. Tziritas, « Smoothing the displacement field for edge-based motion estimation » , *Proc. European Signal Process. Conf.*, p. 667–670, Barcelona, Spain, septembre 1990.

[Ueda et Mase 92] N. Ueda et K. Mase, « Tracking Moving Contours Using Energy-Minimizing Elastic Contour Models » , *Proc. European Conf. Computer Vision*, p. 453–457, Springer-Verlag, Santa Margherita Liguere, Italy, 1992.

[Vandendorpe 91] L. Vandendorpe, « Optimized quantization for image subband coding » , *Signal Process. : Image Commun.*, vol. 4, n° 1, 1991, p. 65–79.

[Verbeek 92] F.J. Verbeek, « Deformation correction using Euclidean contour distance maps » , *Proc. IEEE Int. Conf. Pattern Recognition*, p. 347–351, 1992.

[Vincent et Soille 91] L. Vincent et P. Soille, « Watersheds in digital spaces: an efficient algorithm based on immersion simulations » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, n° 6, juin 1991, p. 583–598.

[Walker et Rao 84] D.R. Walker et K.R. Rao, « Improved pel-recursive motion compensation » , *IEEE Trans. Commun.*, vol. 32, n° 10, octobre 1984, p. 1128–1134.

[Wall et Danielsson 84] K. Wall et P.E. Danielsson, « A Fast Sequential Method for Polygonal Approximation of Digitized Curves » , *Comput. Vis. Graph. Image Process.*, vol. 28, 1984, p. 220–227.

[Wang et Lee 93] Y. Wang et O. Lee, « Active Mesh – A Video Representation Scheme for Feature Seeking and Tracking » , *Proc. SPIE (Visual Communications and Image Process.)*, p. 1558–1569, Cambridge, Massachusetts, USA, novembre 1993.

[Witten et al. 87] I.H. Witten, R.M. Neal et J.G. Cleary, « Arithmetic coding for data compression » , *Communications of the ACM*, vol. 30, n° 6, juin 1987, p. 520–540.

[Wollborn et al. 97] M. Wollborn, M. Kampmann et R. Mech, « Content-Based Coding of Videophone Sequences Using Automatic Face Detection » , *Proc. Picture Coding Symposium*, p. 547–551, Berlin, Germany, septembre 1997.

[Woods 91] J.W. Woods (éd.), *Subband image coding*, Kluwer Academic Press, 1991.

[Wu et al. 95] L. Wu, J. Benois et D. Barba, « Spatio-temporal segmentation of image sequences for object-oriented low bit-rate coding » , *Proc. IEEE Int. Conf. Image Processing*, p. 406–409, Washington, D.C., USA, octobre 1995.

[Wu et Leahy 93] Z. Wu et R. Leahy, « An optimal graph theoretic approach to data clustering: theory and its application to image segmentation » , *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, n° 11, novembre 1993, p. 1101–1113.

[Wu 95] L. Wu, *Segmentation spatio-temporelle d'images animées en vue d'un codage à fort taux de compression*, Laboratoire systèmes électroniques et informatiques, Thèse de doctorat, IRESTE–Université de Nantes, septembre 1995.

[Xie et al. 95] Kan Xie, Luc Van Eycken et André Oosterlinck, « Determining Accurate and Reliable Motion Fields for Motion-Compensated Interpolation » , *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, n° 4, août 1995, p. 367–370.

[Yang et al. 90] Kun-Min Yang, S. Singhal et D. Le Gall, « Design of a multifunction video decoder based on a motion-compensated predictive-interpolative coder » , *Proc. SPIE (Visual Communications and Image Process.)*, p. 1530–1539, 1990.

[Yin et Basu 97] L. Yin et A. Basu, « mpeg4 Face Modeling Using Fiducial Points » , *Proc. IEEE Int. Conf. Image Processing*, p. 109–112, Santa Barbara, CA, USA, octobre 1997.

[Yulie et al. 89] A.L. Yulie, D.S. Cohen et P.W. Hallinan, « Feature extraction from faces using deformable templates » , *Proc. IEEE Conf. Computer Vision Pattern Recognition*, p. 104–109, San Diego, USA, juin 1989.

[Zheng et Blostein 95] H. Zheng et D. Blostein, « Motion-Based Object Segmentation and Estimation using the MDL Principle » , *IEEE Trans. Image Process.*, vol. 4, n° 9, septembre 1995, p. 1223–1235.

Résumé

Le cadre général de cette étude est le traitement numérique du signal, appliqué aux séquences d'images, pour des applications multimédia. Ce travail est divisé en deux contributions principales : un algorithme de segmentation d'images en objets vidéo en mouvement, et une méthode d'interpolation temporelle opérant sur ces objets.

La segmentation de la séquence est effectuée par un algorithme de suivi temporel. Un algorithme de segmentation spatio-temporelle est utilisé initialement pour obtenir des régions dans la première image de la séquence. Cette partition est ensuite suivie par une technique de contours actifs, qui opère sur une nouvelle représentation de la segmentation, composée des frontières ouvertes séparant les régions. L'algorithme estime à la fois le mouvement des frontières et celui des régions. Il est capable de suivre plusieurs objets simultanément et de traiter les occultations entre eux. Des résultats, obtenus sur des séquences d'images réelles, montrent que cet algorithme permet une bonne stabilité temporelle de la segmentation et une bonne précision des frontières.

Le but de l'algorithme d'interpolation est de reconstruire des images intermédiaires entre deux images de la séquence. Il s'agit d'un algorithme de faible complexité qui peut être utilisé à la fin d'une chaîne codeur/décodeur. L'interpolation est compensée en mouvement et utilise le mouvement des régions, estimé pendant la phase de suivi. Il est aussi basé objets, dans le sens où il utilise la segmentation pour prédire correctement les zones d'occultation. Cet algorithme peut être utilisé pour trois applications différentes : le codage interpolatif (où des images de la séquence sont prédites par interpolation), l'adaptation de la fréquence de la séquence à la fréquence d'affichage du terminal de visualisation dans une transmission multipoints et la reconstruction d'images manquantes (où l'on calcule des images non observées). Des résultats expérimentaux pour la première application montrent que pour une qualité de reconstruction donnée, la taux de compression moyen sur un groupe d'images est plus élevé en utilisant l'interpolation qu'avec une prédiction causale.

Mots clés : analyse du mouvement et segmentation, suivi temporel, interpolation temporelle, contours actifs, MPEG, compression, objets vidéo.

Abstract

The general field of this study is digital signal processing applied to image sequences for multimedia applications. This work is divided into two main contributions: an algorithm to segment images into moving video objects and a temporal interpolation method working with those objects.

The segmentation of the sequence is performed with a temporal tracking algorithm. A spatio-temporal segmentation algorithm is used to obtain initial regions in the first image of the sequence. This partition is then tracked with an active contours technique, which operates on a novel segmentation representation composed of open boundaries between regions. The algorithm estimates both the motion of boundaries and the motion of regions. It is also able to track multiple objects simultaneously and to handle occultations between them. Results obtained on real image sequences show that this algorithm achieves a good temporal stability of the segmentation and a correct accuracy of the boundaries.

The goal of the interpolation algorithm is to reconstruct frames between two images in a sequence. It is a low-complexity algorithm which can be used at the end of an object-based coder/decoder chain. The interpolation is motion-compensated, and uses the motion of regions, estimated during the tracking. It is also object-based in the sense that the segmentation is used to accurately predict occultation areas. This algorithm can be used in three different applications: interpolative coding (where known images are predicted by interpolation), adaptation of the frame-rate to the terminal display in a multicast transmission and reconstruction of missing frames (where additional frames are computed). Experimental results for the first application show that for a given reconstruction quality, the average compression is higher when using interpolation than with a causal prediction.

Keywords: motion analysis and segmentation, temporal tracking, temporal interpolation, active contours, MPEG, compression, video objects.