Contributions aux problèmes de l'étalonnage extrinsèque d'affichages semi-transparents pour la réalité augmentée et de la mise en correspondance dense d'images

Jim Braux-Zin

UNIVERSITÉ BLAISE PASCAL — UNIVERSITÉ D'AUVERGNE

2014 — N° d'ordre 13-DOC

ECOLE DOCTORALE
DES SCIENCES POUR L'INGENIEUR
N° d'ordre : 13-DOC

Thèse

Présentée à l'Université d'Auvergne

pour l'obtention du grade de DOCTEUR
(Décret du 5 juillet 1984)

Spécialité Vision par Ordinateur

soutenue le 26 septembre 2014

Jim BRAUX-ZIN

Contributions aux problèmes de l'étalonnage extrinsèque d'affichages semi-transparents pour la réalité augmentée et de la mise en correspondance dense d'images

Mme. Lourdes Agapito	Président et Rapporteur
M. Adrien Bartoli	Directeur de thèse
M. Romain Dupont	Encadrant
M. Simon Lacroix	Examinateur
M. Étienne Mémin	Rapporteur
M. Mohamed Tamaazousti	Encadrant

CEA, LIST

Résumé

La réalité augmentée consiste en l'insertion d'éléments virtuels dans une scène réelle, observée à travers un écran. Les systèmes de réalité augmentée peuvent prendre des formes différentes pour obtenir l'équilibre désiré entre trois critères : *précision*, *latence* et *robustesse*. Il est possible d'identifier trois composants principaux : *localisation*, *reconstruction* et *affichage*. Les contributions de cette thèse se concentrent sur l'affichage et la reconstruction.

Les systèmes de réalité augmentée utilisent habituellement des écrans classiques, opaques, car ils sont largement disponibles (notamment sur *smartphones* et tablettes). Cependant, pour des applications critiques telles que l'aide à la chirurgie ou à la conduite, l'utilisateur ne peut être à aucun moment isolé de la réalité. Nous proposons un système sous forme de « tablette augmentée » avec un écran semi transparent pour répondre à ces problèmes, au prix d'un étalonnage adapté. En effet, afin d'assurer un alignement correct entre les augmentations et la réalité, il faut connaître à chaque instant les poses relatives de l'utilisateur et de la scène observée par rapport à l'écran. Deux dispositifs de localisation (de l'utilisateur et de la scène) sont donc nécessaires et l'étalonnage consiste à calculer la pose de ces dispositifs par rapport à l'écran.

Le protocole d'étalonnage mis en place dans le cadre de cette thèse est le suivant : l'utilisateur renseigne les projections apparentes dans l'écran de points de référence d'un objet 3D connu ; les poses recherchées minimisent la distance 2D entre ces projections et celles calculées par le système. Ce problème est toutefois non convexe et difficile à optimiser. Pour obtenir une estimation initiale, nous développons une méthode directe par l'étalonnage intrinsèque et extrinsèque de caméras virtuelles. Ces dernières sont définies par leurs centres optiques, confondus avec les positions de l'utilisateur, ainsi que leur plan focal, constitué par l'écran. Les projections saisies par l'utilisateur constituent alors les observations 2D des points de référence dans ces caméras virtuelles. Un raisonnement symétrique permet de considérer des caméras virtuelles centrées sur les points de référence de l'objet, « observant » les positions de l'utilisateur. Ces estimations initiales sont ensuite raffinées par ajustement de faisceaux. Des expériences sur données synthétiques et réelles montrent le bien-fondé de cette approche. De plus, la méthode est générique puisqu'elle n'est

liée à aucune hypothèse sur le type de dispositifs de localisation, la technologie d'affichage utilisée ou la géométrie du système.

La reconstruction 3D, quant à elle, est fondamentalement basée sur la triangulation de correspondances entre images. Ces correspondances peuvent être éparses lorsqu'elles sont établies par détection, description et association de primitives géométriques (coins, segments de droite...) ou denses lorsqu'elles sont établies par minimisation d'une fonction de coût sur toute l'image. Un champ dense de correspondance est préférable car il permet une reconstruction de surface, utile notamment pour une gestion réaliste des occultations en réalité augmentée. Cependant, les méthodes d'estimation d'un tel champ sont habituellement basées sur une optimisation variationnelle (minimisation d'une fonction de coût convexe en utilisant des informations locales), précise mais sensible aux minimums locaux et limitée à des images peu différentes. A l'opposé, l'emploi de descripteurs discriminants peut rendre les correspondances éparses très robustes.

Nous proposons de combiner les avantages des deux approches par l'intégration d'un coût basé sur des correspondances éparses de primitives à une méthode d'estimation variationnelle dense. Cela permet d'empêcher l'optimisation de tomber dans un minimum local sans dégrader la précision. Notre terme basé correspondances éparses est adapté aux primitives à coordonnées non entières, et peut exploiter des correspondances de points ou de segments tout en filtrant implicitement les correspondances erronées. Nous proposons de plus une détection et gestion complète des occultations pour pouvoir mettre en correspondance des images éloignées. En particulier, nous avons adapté et généralisé une méthode locale de détection des auto-occultations. Notre méthode produit des résultats compétitifs avec l'état de l'art, tout en étant plus simple et plus rapide, pour les applications de flot optique 2D et de stéréo à large parallaxe. Cela montre que nos contributions permettent d'appliquer les méthodes variationnelles à de nouvelles applications sans dégrader leur performance. De plus, le faible couplage des modules (primitives, coût dense, optimisation), permet une grande flexibilité et généricité. Ainsi, cela nous permet de transposer notre méthode pour le recalage de surfaces déformables avec des résultats surpassant l'état de l'art, ouvrant de nouvelles perspectives.

Cette thèse a fait l'objet de trois publications dans des conférences internationales : 3DIMPVT 2012, BMVC 2013, ICCV 2013 et trois publications nationales : congrès ORASIS 2013 et RFIA 2014, journal Traitement du Signal 2014.

Mots-clefs : étalonnage, affichage semi transparent, caméras virtuelles, alignement d'images, correspondances, flot optique, stéréo, surfaces déformables.

Abstract

Augmented reality is the process of inserting virtual elements into a real scene, observed through a screen. Augmented Reality systems can take different forms to get the desired balance between three criteria: *accuracy*, *latency* and *robustness*. Three main components can be identified: *localization*, *reconstruction* and *display*. The contributions of this thesis are focused on display and reconstruction.

Most augmented reality systems use non-transparent screens as they are widely available (especially on smartphones and tablet computers). However, for critical applications such as surgery or driving assistance, the user cannot be ever isolated from reality. We answer this problem by proposing a new "augmented tablet" system with a semi-transparent screen. Such a system needs a suitable calibration scheme: to correctly align the displayed augmentations and reality, one need to know at every moment the poses of the user and the observed scene with regard to the screen. Two tracking devices (user and scene) are thus necessary, and the system calibration aims to compute the pose of those devices with regard to the screen.

The calibration process set up in this thesis is as follows: the user indicates the apparent projections in the screen of reference points from a known 3D object ; then the poses to estimate should minimize the 2D on-screen distance between those projections and the ones computed by the system. This is a non-convex problem difficult to solve without a sane initialization. We develop a direct estimation method by computing the extrinsic parameters of virtual cameras. Those are defined by their optical centers which coincide with user positions, and their common focal plane consisting of the screen plane. The user-entered projections are then the 2D observations of the reference points in those virtual cameras. A symmetrical thinking allows one to define virtual cameras centered on the reference points, and "looking at" the user positions. Those initial estimations can then be refined with a bundle adjustment. Synthetic and real-world experiments show the merits of this approach. Moreover, the method is generic, not making any assumption on the type of tracking devices, the display technology or the system geometry.

Meanwhile, 3D reconstruction is fundamentally based on the triangulation of matches between images. Those matches can be sparse when computed by detec-

tion and description of image features (corners, line segments...) or dense when computed through the minimization of a cost function of the whole image. A dense correspondence field is better because it makes it possible to reconstruct a 3D surface, useful especially for realistic handling of occlusions for augmented reality. However, such a field is usually estimated thanks to variational methods, minimizing a convex cost function using local information. Those methods are accurate but subject to local minima, thus limited to small deformations. In contrast, sparse matches can be made very robust by using adequately discriminative descriptors.

We propose to combine the advantages of those two approaches by adding a feature-based term into a dense variational method. It helps prevent the optimization from falling into local minima without degrading the end accuracy. Our feature-based term is suited to feature with non-integer coordinates and can handle point or line segment matches while implicitly filtering false matches. We also introduce comprehensive handling of occlusions so as to support large deformations. In particular, we have adapted and generalized a local method for detecting self-occlusions. Results on 2D optical flow and wide-baseline stereo disparity estimation are competitive with the state of the art, with a simpler and most of the time faster method. This proves that our contributions enables new applications of variational methods without degrading their accuracy. Moreover, the weak coupling between the components (features, dense cost, optimization) allows great flexibility and genericness. This is the reason we were able to also transpose the proposed method to the problem of non-rigid surface registration and outperforms the state of the art methods.

This thesis has been the subject of three publications in international conferences: 3DIMPVT 2012, BMVC 2013, ICCV 2013 and three national publications: ORASIS 2013 and RFIA 2014 congresses, *Traitement du Signal* 2014 journal.

Keywords: calibration, semi-transparent display, virtual cameras, image matching, feature matches, optical flow, stereo, non-rigid surface registration.

Table des matières

Chapitre 1

Introduction

Cette thèse a été effectuée entre juin 2011 et juin 2014 au laboratoire LVIC (Laboratoire Vision et Ingénierie des Contenus) du CEA LIST, à Saclay. L'ensemble des travaux ont été réalisés en cotutelle avec l'unité de recherche ISIT (Institut des Sciences de l'Image pour les Techniques interventionnelles, UMR 6284 UdA, CNRS) à Clermont-Ferrand.

Contexte

Dans le cadre de cette thèse, nous nous attachons à la réalisation d'un système de *réalité augmentée*. La réalité augmentée consiste en l'insertion d'éléments virtuels (que nous appellerons *augmentations*) dans la réalité grâce à un écran. Au moment de la rédaction de ce mémoire, la grande majorité des applications commercialisées restent les jeux et autre démonstrations ludiques mais nous considérons ici des applications plus ambitieuses qui permettent d'envisager le réel intérêt du domaine (voir figure 1.1) :

— aide à la vente : la visualisation immédiate des différentes configurations possibles d'un produit (par exemple une voiture, un meuble...) sans contrainte de stock, pourrait aider le choix de l'utilisateur et contribuer à une vente réussie tout en réduisant les coûts associés aux points de vente ;

— aide à la maintenance : les systèmes industriels complexes nécessitent des procédures de maintenance complexes, la réalité augmentée pourrait simplifier la formation des techniciens et augmenter la sécurité en affichant les instructions pas à pas sur le système réel ;

— aide à la navigation : que ce soit à l'intérieur ou à l'extérieur, un « GPS amélioré » pourrait être utile : permettant de signaler précisément l'itinéraire à suivre et d'autres informations importantes ainsi que les dangers imminents ;

— aide à la chirurgie : la plupart des opérations lourdes font l'objet d'une acquisition 3D préliminaire des organes concernés (scanner...), l'affichage de ces informations directement sur le patient en réalité augmentée pourrait grandement aider le praticien en le guidant vers la zone à traiter et en mettant

(a) *AR Games* (Nintendo) (b) Selltic (Diotasoft) (c) ARMAR (Columbia)

(d) AR-Surg (IRCAD) (e) NavVis (f) Concept BMW

FIGURE 1.1 – Diverses applications de réalité augmentée : jeu, aide à la vente, à la maintenance, à la chirurgie, à la navigation intérieure et extérieure. Seuls (a) et (b) sont actuellement commercialisées, les autres sont des concepts ou prototypes.

en évidence les zones à éviter, par exemple les artères et les nerfs.

Un système de réalité augmentée est complexe et fait intervenir des composants variés, que l'on peut classer en trois briques principales : *reconstruction*, *localisation* et *affichage*. Les définitions et relations associées à ces briques sont détaillées en chapitre 3, ainsi que le détail de notre positionnement. Dans ce mémoire, nous nous concentrerons sur les problématiques d'affichage et reconstruction, considérant que les solutions de localisation existantes sont matures et satisfaisantes, notamment le SLAM contraint de Tamaazousti *et al.* (2011). Le chapitre 2 introduit les notions de base utiles à la compréhension du mémoire : mathématiques (algèbre, géométrie projective), vision par ordinateur (localisation, reconstruction, mise en correspondance d'images) et optimisation numérique.

Problématiques et contributions

Le chapitre 4 est dédié à l'affichage. Les systèmes de réalité augmentée utilisent habituellement des écrans classiques, opaques, car ils sont largement disponibles (notamment *smartphones* et tablettes). Cependant, pour les applications critiques envisagées (chirurgie, maintenance, navigation), l'utilisateur ne peut être à aucun moment isolé de la réalité. Pour répondre à ce problème nous introduisons les contributions suivantes :

— un système sous forme de « tablette augmentée » avec un écran semi transparent et deux dispositifs de localisation par rapport à l'écran (de l'utilisateur et de la scène), requis pour un alignement correct des augmentations ;
— un nouveau protocole d'étalonnage où l'utilisateur renseigne les projections apparentes dans l'écran de points de référence d'un objet 3D connu ;
— une méthode directe d'étalonnage initial grâce à l'introduction de caméras virtuelles ;
— un raffinement non-linéaire optimal.

Ces contributions ont fait l'objet d'une publication internationale : conférence 3DIMPVT (2012) et deux publications nationales : congrès ORASIS (2013a) et journal Traitement du Signal (2014a).

Ensuite, dans le chapitre 5, nous nous penchons sur le problème de la reconstruction 3D. Cette dernière est fondamentalement basée sur la triangulation de correspondances entre images. Ces correspondances peuvent être éparses par détection, description et association de primitives géométriques (coins, segments de droite…) ou denses par minimisation d'une fonction de coût sur toute l'image. Un champ dense de correspondance est préférable car il permet une reconstruction de surface, utile notamment pour une gestion réaliste des occultations en réalité aug-

mentée. Cependant, les méthodes d'estimation d'un tel champ sont habituellement basées sur une optimisation variationnelle, précise mais sensible aux minimums locaux et limitée à des images peu différentes. A l'opposé, l'emploi de descripteurs discriminants peut rendre les correspondances éparses très robustes.

Nous proposons de combiner les avantages des deux approches par l'intégration d'un coût basé correspondances éparses de primitives à une méthode d'estimation variationnelle dense. Notre proposition consiste en les contributions suivantes :

— un nouveau terme basé correspondance de primitives dans la fonction de coût pour éviter les minimums locaux, capable de gérer des correspondances ponctuelles ou non (segments de droite), à précision sous-pixellique, même en présence d'une forte proportion de correspondances erronées ;
— un traitement complet des occultations (auto-occultations, occultations externes, bordure de l'image) pour gérer les images éloignées ;
— l'implémentation de notre méthode dans deux algorithmes de mise en correspondance dense (paramétrique et non paramétrique) avec des résultats compétitifs pour le calcul de flot optique, la stéréo à large parallaxe et l'alignement de surfaces déformables.

Ce chapitre a fait l'objet de deux publications internationales : conférences BMVC (2013b) et ICCV (2013c) et d'une publication nationale : congrès RFIA (2014b).

Chapitre 2

Notions de base

Dans ce chapitre, nous introduisons les notions de base et notations nécessaires à la compréhension du mémoire : mathématiques (algèbre, géométrie projective), vision par ordinateur (localisation, reconstruction, mise en correspondance d'images) et optimisation numérique.

2.1 Notations

Algèbre linéaire

x	Scalaire
$\mathbb{R}^n$	Espace vectoriel de dimension n
$\boldsymbol{x}$	Vecteur
$x_1 \dots$	Coordonnées
$\|.\|$	Norme euclidienne L^2
$\|.\|_\Sigma$	Norme de Mahalanobis
$\|.\|$	Norme L^1
$\wedge$	Produit vectoriel
$[\boldsymbol{x}]_\times$	Matrice antisymétrique
$\mathbf{M}$	Matrice
$M_{11} \dots$	Coefficients
$\mathbf{M}^\top$	Transposée
$\mathbf{M}^{-1}$	Inverse
$\mathbf{M}^\dagger$	Pseudo-inverse
$\mathrm{diag}(\boldsymbol{x})$	Matrice diagonale avec $\boldsymbol{x}$ ses éléments diagonaux
$0_{m\times n}$	Matrice nulle de dimensions $m \times n$
$\mathbf{I}$	Matrice identité

Géométrie euclidienne

$\boldsymbol{q}$	Point 2D (vecteur de $\mathbb{R}^2$)
$\boldsymbol{Q}$	Point 3D (vecteur de $\mathbb{R}^3$)
$\mathbf{R}$	Matrice de rotation
$\boldsymbol{T}$	Vecteur de translation
$\mathscr{W}$	Repère monde
$\mathscr{C}$	Repère associé à la caméra $\mathscr{C}$
$\boldsymbol{Q}^{(\mathscr{C})}$	Point 3D exprimé dans le repère $\mathscr{C}$

Géométrie projective

$\mathbb{P}^n$	Espace projectif de dimension n, dont les éléments ont $n+1$ coordonnées
$\dot{\boldsymbol{x}}$	Représentation canonique de $\boldsymbol{x} = (x_1, \ldots, x_n)$ en coordonnées homogènes si $\boldsymbol{x} = (x_1, \ldots, x_n)$, alors $\dot{\boldsymbol{x}} = (x_1, \ldots, x_n, 1)$
π	Fonction de déhomogénisation $\mathbb{P}^n \to \mathbb{R}^n$
$\mathbf{K}$	Matrice 3×3 d'étalonnage de caméra (paramètres intrinsèques)

Images

I	Image
Ω_I	Domaine de l'image
$I(\boldsymbol{q})$	Intensité du pixel $\boldsymbol{q}$
$H_I^{\mathbf{H}}$	Application de l'homographie de matrice $\mathbf{H}$ à l'image I
$T_I^{\boldsymbol{t}}$	Application de la translation de vecteur $\boldsymbol{t}$ à l'image I

Estimation

C	Fonction de coût
$\widetilde{\boldsymbol{x}}$	Observation bruitée
$\widehat{\boldsymbol{x}}$	Estimation

2.2 Géométrie projective

Sur l'espace vectoriel $\mathbb{R}^{n+1}$, il est possible de définir la relation d'équivalence suivante :

$$\boldsymbol{u} \sim \boldsymbol{v} \Leftrightarrow \exists \lambda \in \mathbb{R}^* \mid \boldsymbol{u} = \lambda \boldsymbol{v}. \tag{2.1}$$

L'ensemble des classes d'équivalence de $\mathbb{R}^{n+1}$ pour cette relation « $\sim$ » définit un espace appelé *espace projectif*. Cet espace, de dimension n, sera noté $\mathbb{P}^n$. Si des études théoriques de ces espaces existent, nous nous intéresserons dans nos travaux à la *géométrie projective* qui leur est associée et qui permet en particulier de formaliser la notion de point à l'infini dans les espaces affines.

Un vecteur de l'espace projectif $\mathbb{P}^n$ aura pour coordonnées :

$$\boldsymbol{h} = (h_1 \ \dots \ h_{n+1})^\top \tag{2.2}$$

avec les h_i non tous nuls. Si h_{n+1} est non nul, ce vecteur $\boldsymbol{h}$ représente le vecteur $\boldsymbol{x}$ de $\mathbb{R}^n$ avec $\boldsymbol{x} = (h_1/h_{n+1} \ \dots \ h_n/h_{n+1})^\top$. Dans le cas contraire, le vecteur $\boldsymbol{h}$ décrit un point à l'infini. Les coordonnées de $\boldsymbol{h}$ sont appelées *coordonnées homogènes* de $\boldsymbol{x}$. La représentation canonique d'un vecteur de $\mathbb{R}^n$ en coordonnées homogènes est notée avec un point et définie comme suit :

$$\forall\ \boldsymbol{x} = (x_1 \ \dots \ x_n)^\top \in \mathbb{R}^n : \qquad \dot{\boldsymbol{x}} = (x_1 \ \dots \ x_n, 1)^\top \in \mathbb{P}^n. \tag{2.3}$$

Nous appellerons π la fonction de *déhomogénisation* permettant de passer des coordonnées homogènes aux coordonnées euclidiennes, à savoir :

$$\begin{array}{cccc} \pi : & \mathbb{P}^n & \to & \mathbb{R}^n \\ & (x_1 \ \dots \ x_{n+1})^\top & \mapsto & (x_1/x_{n+1} \ \dots \ x_n/x_{n+1})^\top. \end{array} \tag{2.4}$$

En particulier $\forall \boldsymbol{x} \in \mathbb{R}^n : \pi(\dot{\boldsymbol{x}}) = \boldsymbol{x}$. Nous nous intéressons maintenant aux cas particulier de cette géométrie en deux puis trois dimensions.

2.2.1 Le plan projectif

L'espace projectif de dimension 2 est appelé *plan projectif*. Un point de $\mathbb{P}^2$ est représenté par un vecteur de dimension 3 : $\boldsymbol{h} = (x \ y \ w)^\top$. De même, une droite d'équation $ax + by + c = 0$ peut être représentée par le vecteur $\boldsymbol{l} = (a \ b \ c)^\top$. Cette notation homogène permet de définir simplement la notion d'appartenance du point $\boldsymbol{h}$ à la droite $\boldsymbol{l}$, à savoir :

$$\boldsymbol{l}^\top \boldsymbol{h} = 0. \tag{2.5}$$

L'équation de la droite $\boldsymbol{l}$ passant par les points $\boldsymbol{h}_1$ et $\boldsymbol{h}_2$ est obtenue en calculant leur produit vectoriel :

$$\boldsymbol{l} = \boldsymbol{h}_1 \wedge \boldsymbol{h}_2. \tag{2.6}$$

Dans l'espace $\mathbb{P}^2$, droites et points jouent un rôle équivalent : c'est ce qu'on appelle le *principe de dualité*. En particulier, à partir de l'équation duale de l'équation

précédente, il est possible de calculer le point d'intersection de deux droites $\boldsymbol{l}_1$ et $\boldsymbol{l}_2$:

$$\boldsymbol{h} = \boldsymbol{l}_1 \wedge \boldsymbol{l}_2. \tag{2.7}$$

2.2.2 L'espace projectif 3D

Un point $\boldsymbol{Q}$ de $\mathbb{R}^3$ aura pour coordonnées homogènes dans $\mathbb{P}^3$ le vecteur

$$\boldsymbol{H} = (X\ Y\ Z\ W)^\top.$$

Dans cet espace de dimension 3, le dual du point $\boldsymbol{Q}$ est le plan Π d'équation $aX + bY + cZ + d = 0$ qui est représenté par le vecteur $\Pi = (a\ b\ c\ d)^\top$. L'appartenance du point $\boldsymbol{Q}$ au plan Π est alors donnée par la relation :

$$\Pi^\top \boldsymbol{H} = 0. \tag{2.8}$$

2.3 Images

2.3.1 Définition et notations

Au niveau informatique une image est un tableau à deux dimensions contenant un nombre fini de pixels. Sauf mention contraire, nous considérons des images en niveau de gris à valeurs réelles. Une image I de dimensions $w \times h$ est donc une application bidimensionnelle :

$$\begin{aligned} I : [\![1, w]\!] \times [\![1, h]\!] &\to \mathbb{R} \\ (x, y) &\mapsto I[x, y]. \end{aligned} \tag{2.9}$$

Pour simplifier les expressions mathématiques, il est possible de considérer I définie sur le domaine continu :

$$\Omega_I = [1, w] \times [1, h] \tag{2.10}$$

en interpolant les valeurs entre les pixels voisins pour les coordonnées non entières. On a alors :

$$\begin{aligned} I : \quad \Omega_I &\to \mathbb{R} \\ (x, y) &\mapsto I(x, y). \end{aligned} \tag{2.11}$$

Notons que l'on utilise les crochets pour les coordonnées discrètes et les parenthèses pour les coordonnées réelles.

2.3.2 Opérations

Les opérations suivantes sur les images seront utiles par la suite :

Homographie. Une homographie décrit une transformation linéaire dans l'espace projectif. L'homographie $H_I^{\mathbf{H}}$ de l'image I décrite par la matrice $\mathbf{H} \in \mathbb{R}^{3\times3}$ est définie par :

$$\forall \boldsymbol{q} \quad \text{t.q.} \quad \pi(\mathbf{H}\dot{\boldsymbol{q}}) \in \Omega_I \quad : \quad H_I^{\mathbf{H}}(\boldsymbol{q}) = I\left(\pi(\mathbf{H}\dot{\boldsymbol{q}})\right). \tag{2.12}$$

Deux observations d'une même surface rigide sont toujours liées par une homographie.

Homographie affine. Une homographie est dite affine si sa matrice $\mathbf{H}$ vérifie : $H_{31} = H_{32} = 0 \quad \text{et} \quad H_{33} = 1$. Elle peut donc s'écrire :

$$\forall \boldsymbol{q} \quad \text{t.q.} \quad \pi(\overline{\mathbf{H}}\boldsymbol{q} + \boldsymbol{t}_{\mathbf{H}}) \in \Omega_I \quad : \quad H_I^{\mathbf{H}}(\boldsymbol{q}) = I\Big(\pi(\overline{\mathbf{H}}\boldsymbol{q} + \boldsymbol{t}_{\mathbf{H}})\Big) \tag{2.13}$$

avec :

$$\overline{\mathbf{H}} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \quad \boldsymbol{t}_{\mathbf{H}} = \begin{pmatrix} H_{13} \\ H_{23} \end{pmatrix}.$$

Translation. La translation de l'image I de vecteur $\boldsymbol{t} \in \mathbb{R}^2$ est notée $T_I^{\boldsymbol{t}}$ et définie telle que :

$$\forall \boldsymbol{q} \quad \text{t.q.} \quad (\boldsymbol{q} + \boldsymbol{t}) \in \Omega_I \quad : \quad T_I^{\boldsymbol{t}}(\boldsymbol{q}) = I(\boldsymbol{q} + \boldsymbol{t}). \tag{2.14}$$

Convolution. La convolution, notée $*$ est une opération permettant d'appliquer un filtre linéaire à une image. En coordonnées discrètes, une convolution sur l'image I est définie par une image W de domaine Ω_W, appelée *masque* ou *noyau* :

$$\begin{aligned} &\forall W : [\![-m, m]\!] \times [\![-n, n]\!] \to \mathbb{R} \\ &(I * W)[x, y] = \sum_{i=-m}^{m} \sum_{j=-n}^{n} W[i, j] \cdot I[x - i, y - j]. \end{aligned} \tag{2.15}$$

En coordonnées réelles la définition est similaire :

$$\forall W : \Omega_W \to \mathbb{R} \qquad (I * W)(x, y) = \iint_{(x',y')\in\Omega_W} W(x', y') \cdot I(x - x', y - y')\, dx\, dy. \tag{2.16}$$

Différentiation. Le gradient de l'image I est défini et noté comme suit :

$$\forall (x,y) \in \Omega_I \quad : \quad \nabla I(x,y) = \begin{pmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{pmatrix}. \tag{2.17}$$

Le laplacien est la somme des dérivées partielles au second ordre :

$$\forall (x,y) \in \Omega_I \quad : \quad \nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2}. \tag{2.18}$$

2.3.3 Champs bidimensionnels

Nous définissons trois champs bidimensionnels qui seront utiles par la suite, définis sur un domaine continu Ω :

- $\mathbb{U} = \{u : \Omega \to \mathbb{R}\}$, espace des fonctions « images » précédemment définies,
- $\mathbb{V} = \left\{\boldsymbol{v} = (v_1, v_2) : \Omega \to \mathbb{R}^2\right\}$ espace des champs vectoriels de dimension 2,
- $\mathbb{W} = \left\{\boldsymbol{W} : \Omega \to \mathrm{Sym}^2(\mathbb{R})\right\}$, l'ensemble des champs de tenseurs symétriques 2×2, assimilables à trois champs scalaires w_{11}, w_{12} et w_{22} tels que :

$$\forall \boldsymbol{W} \in \mathbb{W}, \forall \boldsymbol{q} \in \Omega \quad : \quad \boldsymbol{W}(\boldsymbol{q}) = \begin{pmatrix} w_{11}(\boldsymbol{q}) & w_{12}(\boldsymbol{q}) \\ w_{12}(\boldsymbol{q}) & w_{22}(\boldsymbol{q}) \end{pmatrix}. \tag{2.19}$$

2.4 Caméras perspectives

Dans le cadre de nos travaux, nous travaillerons sur les caméras perspectives. Ces caméras respectent le modèle des *caméras sténopé* idéales. L'idée de ce modèle de caméras est de considérer que l'ensemble des rayons lumineux passent par un seul et unique point appelé le centre optique (voir figure 2.1).

Dans la suite, nous présenterons tout d'abord les différents paramètres qui caractérisent ce type de caméras. Nous présenterons alors la géométrie reliant les images observées par plusieurs caméras. Enfin, nous étudierons les méthodes permettant de retrouver le déplacement de ces caméras ainsi que la structure de l'environnement à partir des images qu'elles observent.

2.4.1 Projection perspective

La projection perspective vise à calculer, pour tout point $\boldsymbol{Q}$ de $\mathbb{R}^3$, la position 2D $\boldsymbol{q}$ de sa projection dans l'image. Basée sur la projection centrale, cette transformation consiste à calculer l'intersection du plan de la *rétine* de la caméra (*i.e.* le capteur)

avec le *rayon de projection* de $\boldsymbol{Q}$. Ce dernier est défini comme étant la droite reliant le point $\boldsymbol{Q}$ au centre de la caméra (figure 2.1).

Cette projection peut être vue comme un enchaînement de trois transformations géométriques (figure 2.1) :

— La première transformation est un changement de repère qui consiste à exprimer les coordonnées de $\boldsymbol{Q}$ dans le repère lié à la caméra. Ce changement de repère est défini par les *paramètres extrinsèques* de la caméra. Dans la suite du mémoire, en cas d'ambiguité, les points 3D seront indicés $\mathscr{W}$ ou $\mathscr{C}$ en fonction du repère dans lequel sont exprimées leurs coordonnées (respectivement monde ou caméra).

— La deuxième transformation est la *projection centrale* du point 3D. Elle revient à passer du point 3D (exprimé dans le repère caméra) au point d'intersection du rayon de projection et du capteur. Les coordonnées 2D du point résultant sont alors exprimées dans le plan de la rétine.

— La troisième transformation est un changement de repère 2D qui vise à passer du repère rétine (repère lié à la physique du capteur et où les coordonnées sont exprimées en mm) au repère *image* de la caméra (repère géométrique où les coordonnées sont exprimées en pixels). Cette transformation est définie par les *paramètres intrinsèques* de la caméra.

La projection perspective est donc une transformation projective $\mathbf{A} : \mathbb{P}^3 \to \mathbb{P}^2$. En pratique, elle sera représentée par une *matrice de projection* $\mathbf{P}$ de dimension (3×4). La projection perspective s'exprime alors par la relation matricielle suivante :

$$\dot{\boldsymbol{q}} \sim \mathbf{P}\dot{\boldsymbol{Q}}_{\mathscr{W}}. \tag{2.20}$$

La matrice $\mathbf{P}$ se décompose selon les trois transformations citées précédemment :

$$\mathbf{P} = \mathbf{K} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \\ 0_{1\times 3} & 1 \end{pmatrix} \tag{2.21}$$

où

— $\mathbf{K}$ est la *matrice d'étalonnage* (de taille 3×3) de la caméra ;

— $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ la matrice de *projection centrale* ;

— $\begin{pmatrix} \mathbf{R} & \boldsymbol{T} \\ 0_{1\times 3} & 1 \end{pmatrix}$ la *matrice de pose* ou *matrice des paramètres extrinsèques*.

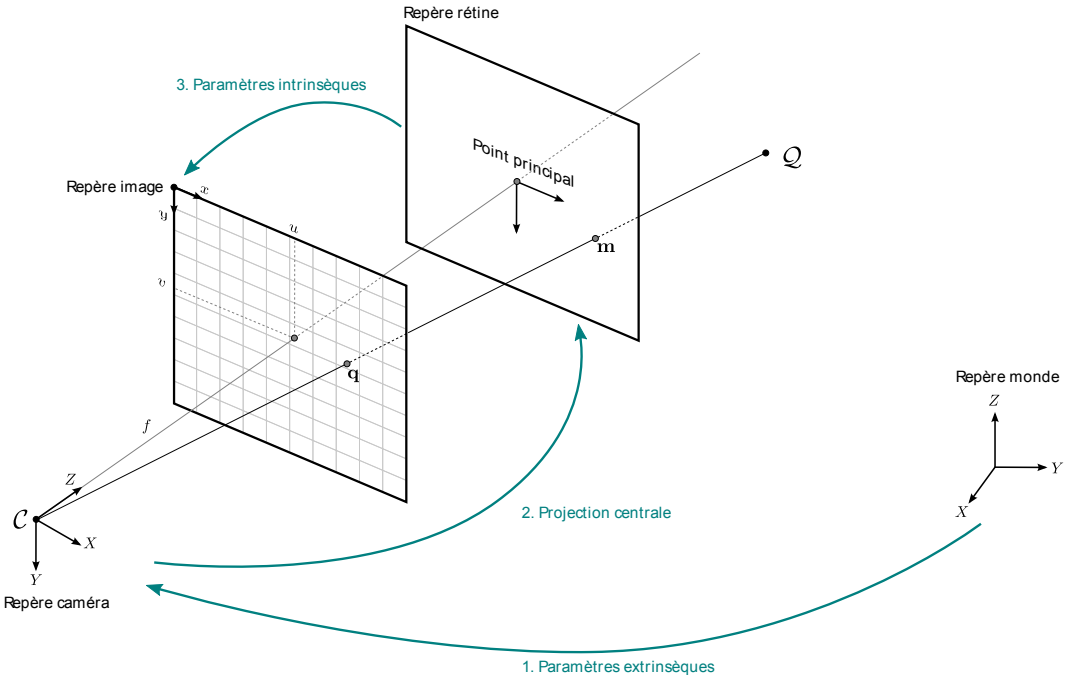


FIGURE 2.1 – La projection perspective peut être vue comme trois transformations géométriques consécutives pour les points 3D.

Paramètres extrinsèques

Les paramètres extrinsèques d'une caméra caractérisent la pose de celle-ci dans le repère monde. La pose d'une caméra possède 6 degrés de liberté :

— la position 3D du centre optique, décrit par le vecteur $\boldsymbol{T} = (t_x\ t_y\ t_z)^\top$;

— l'orientation 3D de la caméra, représentée en pratique sous la forme d'une matrice de rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$ (voir annexe 2.A pour d'autres représentations), vérifiant en particulier $\det\mathbf{R} = 1$ et $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$.

Les paramètres extrinsèques de la caméra permettent d'établir les changements de repère monde/caméra, à savoir :

$$\dot{\boldsymbol{Q}}_{\mathscr{C}} \sim \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \\ 0_{1\times3} & 1 \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}} \tag{2.22}$$

$$\dot{\boldsymbol{Q}}_{\mathscr{W}} \sim \begin{pmatrix} \mathbf{R}^\top & -\mathbf{R}^\top\boldsymbol{T} \\ 0_{1\times3} & 1 \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{C}}. \tag{2.23}$$

Projection centrale

Lorsqu'on utilise les coordonnées homogènes, la projection centrale d'un point $\boldsymbol{Q}_{\mathscr{C}}$ en le point $\boldsymbol{q}$ est une fonction linéaire de $\mathbb{P}^3 \mapsto \mathbb{P}^2$ caractérisée par la matrice de dimension 3×4 :

$$\dot{\boldsymbol{q}} \sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{C}}. \tag{2.24}$$

Dans de nombreuses publications, le changement de repère 3D et la projection centrale sont vus comme une unique projection centrale à partir d'un point 3D dans le repère monde. Cela s'écrit matriciellement :

$$\begin{aligned} \dot{\boldsymbol{q}} &\sim \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \\ 0_{1\times3} & 1 \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}} \\ &\sim \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}}. \end{aligned} \tag{2.25}$$

La projection du monde dans l'image (équation (2.20)) s'écrit donc généralement sous cette forme :

$$\dot{\boldsymbol{q}} \sim \mathbf{K} \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}} \quad \text{ou encore} \quad \boldsymbol{q} = \pi\left[\mathbf{K} \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}}\right]. \tag{2.26}$$

Paramètres intrinsèques et distorsion

Les paramètres intrinsèques définissent les propriétés géométriques du capteur de la caméra. Ils constituent la matrice d'étalonnage sous la forme suivante :

$$\mathbf{K} = \begin{pmatrix} f/m_x & s & u \\ 0 & f/m_y & v \\ 0 & 0 & 1 \end{pmatrix} \tag{2.27}$$

avec :

- m_x et m_y les dimensions d'un pixel en x et en y ;
- f la *distance focale* décrivant la distance orthogonale entre le centre et la rétine de la caméra (figure 2.1), on note $f_x = f/m_x$ et $f_y = f/m_y$ les distances focales exprimées en pixels ;
- $(u\ v)^\top$ le *point principal*. Souvent approximé comme étant le centre du capteur, il est plus précisément l'intersection entre l'axe optique et la rétine de la caméra (figure 2.1) ;
- s l'*obliquité* entre les axes x et y, la plupart du temps $s = 0$ pour des axes orthogonaux.

Dans le cas, fréquent, où les pixels sont parfaitement carrés, $f_x = f_y = f_0$ et $s = 0$. La matrice d'étalonnage prend donc la forme suivante :

$$\mathbf{K} = \begin{pmatrix} f_0 & 0 & u \\ 0 & f_0 & v \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.28}$$

Il est important de noter que les capteurs à courte focale peuvent présenter un phénomène de distorsion important. Ceci se traduit visuellement par une déformation des lignes droites dans l'image sous forme de courbes. Pour corriger cela, il est possible d'ajouter à l'étalonnage de la caméra des paramètres de distorsion permettant de passer de la position observée d'un point 2D dans l'image à sa position réelle, c'est à dire corrigée de toute distorsion.

Dans le cadre de ce mémoire, nous considérerons que les entrées de nos algorithmes ont été préalablement corrigées en distorsion. Pour plus de renseignements sur l'étalonnage des caméras, nous invitons le lecteur à se référer à l'article de Lavest *et al.* (1998).

2.4.2 Notion de rétroprojection

La *rétroprojection* peut être vue comme l'opération inverse de la projection. Son but est d'inférer la position d'un point 3D $\boldsymbol{Q}$ à partir de son observation $\boldsymbol{q}$ dans

l'image. Néanmoins, à partir d'un seule image, il est impossible d'obtenir la position exacte du point 3D. En effet, l'utilisation d'une seule caméra ne permet pas de retrouver la profondeur à laquelle se situe ce point. La rétroprojection d'un point de l'image se traduit sous la forme du rayon optique qui passe à la fois par le centre de la caméra $\mathscr{C}$ et par l'observation $\boldsymbol{q}$. La position du point 3D est donc exprimée à un facteur λ près qui reflète la profondeur du point sur ce rayon :

$$\boldsymbol{Q}(\lambda) = \lambda \mathbf{P}^{\dagger} \dot{\boldsymbol{q}} \tag{2.29}$$

où $\mathbf{P}^{\dagger}$ désigne la pseudo-inverse de la matrice $\mathbf{P}$:

$$\mathbf{P}^{\dagger} \;=\; \mathbf{P}^{\top}(\mathbf{P}\mathbf{P}^{\top})^{-1} \tag{2.30}$$

de telle sorte que $\mathbf{P}\boldsymbol{Q}(\lambda) = \lambda(\mathbf{P}\mathbf{P}^{\dagger})\dot{\boldsymbol{q}} = \lambda\dot{\boldsymbol{q}}$.

2.4.3 Étalonnage par associations 3D-2D

Un des problèmes fondamentaux de la vision par ordinateur est l'étalonnage de caméras perspectives, c'est à dire l'estimation des paramètres intrinsèques et extrinsèques. Pour cela nous considérons ici le cas d'associations 3D-2D connues : soient n points 3D de la scène observée dont les coordonnées $\boldsymbol{Q}_{i=1\ldots n}$ exprimées dans le repère monde sont connues, ainsi que leurs projections $\boldsymbol{q}_{i=1\ldots n}$ dans le plan image de la caméra. D'après l'équation (2.26) on a les relations suivantes :

$$\begin{aligned} &\forall i \in [\![1,n]\!] \quad \boldsymbol{q}_i = \pi\Big[\mathbf{K}\begin{pmatrix}\mathbf{R} & \boldsymbol{T}\end{pmatrix}\dot{\boldsymbol{Q}}_i\Big] \\ \Leftrightarrow\; &\exists\, \lambda_i \in \mathbb{R} \text{ tel que } \dot{\boldsymbol{q}}_i = \lambda_i \mathbf{K}\begin{pmatrix}\mathbf{R} & \boldsymbol{T}\end{pmatrix}\dot{\boldsymbol{Q}}_i. \end{aligned} \tag{2.31}$$

Dans les paragraphes suivants nous détaillons la méthode classique d'étalonnage telle que décrite par Hartley et Zisserman (2004).

Estimation de la matrice de projection

La première étape de l'étalonnage est l'estimation de la matrice de projection $\mathbf{P} = \mathbf{K}\begin{pmatrix}\mathbf{R} & \boldsymbol{T}\end{pmatrix}$, de dimensions 3×4. Pour ceci, l'équation (2.31) est transformée comme suit :

$$\Big(\exists\, \lambda_i \in \mathbb{R} \text{ tel que } \boldsymbol{q}_i = \lambda_i \mathbf{P}\dot{\boldsymbol{Q}}_i\Big) \Leftrightarrow \dot{\boldsymbol{q}}_i \wedge \mathbf{P}\dot{\boldsymbol{Q}}_i = 0. \tag{2.32}$$

Chacune de ces équations se traduit par trois équations (une par coordonnée) linéaires en les coefficients de $\mathbf{P}$. En coordonnées homogènes, seulement deux de ces équations sont indépendantes. Soit

$$\boldsymbol{p} = (P_{11}, P_{12}, \ldots, P_{34}) \tag{2.33}$$

l'ensemble des coefficients de $\mathbf{P}$, on obtient un système linéaire homogène de la forme :

$$\mathbf{A}\boldsymbol{p} = 0 \tag{2.34}$$

avec $\mathbf{A}$ de dimensions $2n \times 4$. Cette transformation s'appelle *Direct Linear Transform* (DLT, Abdel-Aziz et Karara, 1971).

La matrice $\mathbf{P}$ possède 12 coefficients à estimer mais est définie à un facteur scalaire près. Il y a donc 11 inconnues et il est possible d'estimer la matrice $\mathbf{P}$ avec un minimum de $n = 6$ correspondances 3D-2D. En effet, une décomposition spectrale de $\mathbf{A}$ (par décomposition en valeurs singulières, voir section 2.8.1) permet d'identifier son noyau $\ker \mathbf{A}$, de dimension 1 hors cas dégénérés.

Il est à noter que le cas où les points 3D sont coplanaires constitue un cas dégénéré. Mais si l'on sait *a priori* qu'ils sont coplanaires (utilisation d'un motif d'étalonnage plan), alors on peut utiliser une méthode dédiée, telle que celle proposée par Zhang (2000).

Décomposition de la matrice de projection

La deuxième étape consiste à extraire les paramètres intrinsèques et extrinsèques de la matrice de projection. Pour cela considérons la sous-matrice 3×3 de $\mathbf{P}$ suivante :

$$\bar{\mathbf{P}} = \begin{pmatrix} P_{11} & \cdots & P_{13} \\ \vdots & & \vdots \\ P_{31} & \cdots & P_{33} \end{pmatrix} \sim \mathbf{KR}. \tag{2.35}$$

La matrice $\mathbf{O} = \bar{\mathbf{P}}\bar{\mathbf{P}}^\top$ vérifie :

$$\mathbf{O} = \bar{\mathbf{P}}\bar{\mathbf{P}}^\top \sim \mathbf{K}\left(\mathbf{R}\mathbf{R}^\top\right)\mathbf{K}^\top = \mathbf{K}\mathbf{K}^\top \tag{2.36}$$

car les matrices de rotations sont orthogonales par définition. Or, d'après l'équation (2.27) de la matrice d'étalonnage :

$$\mathbf{K}\mathbf{K}^\top = \begin{pmatrix} f_x^2 + s^2 + u^2 & f_y s + uv & u \\ f_y s + uv & f_y^2 + v^2 & v \\ u & v & 1 \end{pmatrix} \tag{2.37}$$

La procédure suivante permet alors d'estimer les paramètres intrinsèques :

1. calcul du facteur d'échelle :

$$\alpha = O_{33} \tag{2.38}$$

2. extraction du point principal :

$$u = \frac{O_{13}}{\alpha} \qquad \text{et} \qquad v = \frac{O_{23}}{\alpha} \tag{2.39}$$

3. extraction de la focale f_y :

$$f_y = \sqrt{\frac{O_{22}}{\alpha} - v^2} \tag{2.40}$$

4. extraction de l'obliquité :

$$s = \frac{\frac{O_{12}}{\alpha} - uv}{f_y} \tag{2.41}$$

5. extraction de la focale f_x :

$$f_x = \sqrt{\frac{O_{11}}{\alpha} - s^2 - u^2}. \tag{2.42}$$

Il est possible d'imposer certaines contraintes de manière dure lors de l'extraction de la matrice de paramètres extrinsèques (voir équation (2.28) pour des pixels carrés par exemple) mais cela peut entraîner des instabilités si la matrice $\mathbf{P}$ est imparfaite.

La matrice de paramètres intrinsèques $\mathbf{K}$ est inversible, il est donc possible d'extraire les paramètres extrinsèques. Pour la rotation on a, d'après l'équation (2.35) :

$$\mathbf{K}^{-1}\bar{\mathbf{P}} \sim \mathbf{R} \quad \Leftrightarrow \quad \exists \lambda \in \mathbb{R}, \lambda \mathbf{K}^{-1}\bar{\mathbf{P}} = \mathbf{R} \tag{2.43}$$

ce qui implique l'égalité des déterminants, or $\mathbf{R}$ est une matrice de rotation donc $\det \mathbf{R} = 1$:

$$\begin{aligned} \det\left(\lambda \mathbf{K}^{-1}\bar{\mathbf{P}}\right) &= \det(\mathbf{R}) \\ \lambda^3 \det\left(\mathbf{K}^{-1}\bar{\mathbf{P}}\right) &= 1 \\ \lambda &= \sqrt[3]{\frac{1}{\det\left(\mathbf{K}^{-1}\bar{\mathbf{P}}\right)}} \end{aligned} \tag{2.44}$$

puis :

$$\mathbf{R} = \sqrt[3]{\frac{1}{\det\left(\mathbf{K}^{-1}\bar{\mathbf{P}}\right)}}\mathbf{K}^{-1}\bar{\mathbf{P}}. \tag{2.45}$$

Après l'estimation, il est important de vérifier que $\mathbf{R}$ est une matrice de rotation valide, c'est à dire qu'elle vérifie $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$. Enfin, la translation $\boldsymbol{T}$ est triviale à calculer en utilisant la relation

$$\lambda\mathbf{K}^{-1}\mathbf{P} = \begin{pmatrix}\mathbf{R} & \boldsymbol{T}\end{pmatrix}. \tag{2.46}$$

Raffinement non linéaire

Nous avons jusqu'ici considéré les observations 2D $\boldsymbol{q}_{i=1\ldots n}$ comme étant les projections parfaites des points 3D $\boldsymbol{Q}_{i=1\ldots n}$ dans le plan image. En pratique, ces observations sont souvent bruitées. On note $\widetilde{\boldsymbol{q}}_{i=1\ldots n}$ les observations bruitées et $\widehat{\mathbf{K}}$, $\widehat{\mathbf{R}}$, $\widehat{\boldsymbol{T}}$ les paramètres estimés. Il est possible d'obtenir une solution optimale aux moindres carrés en résolvant le système non linéaire suivant :

$$\widehat{\mathbf{K}},\widehat{\boldsymbol{r}},\widehat{\boldsymbol{T}} = \arg\min \sum_{i=1}^{n} \left\| \widetilde{\boldsymbol{q}}_i - \pi\left[\widehat{\mathbf{K}}\begin{pmatrix}\mathbf{R}(\widehat{\boldsymbol{r}}) & \widehat{\boldsymbol{T}}\end{pmatrix}\boldsymbol{Q}_i\right]\right\|^2. \tag{2.47}$$

où $\widehat{\boldsymbol{r}}$ est la représentation angle-axe de la rotation estimée (voir annexe 2.A). La méthode directe décrite précédemment peut servir d'initialisation à une méthode d'optimisation convexe (voir section 2.8).

Si certaines contraintes sont connues *a priori*, il est préférable de les intégrer seulement au stade de raffinement sous la forme de contraintes douces. Par exemple si les pixels de l'image sont carrés, les paramètres extrinsèques désirés doivent vérifier $f_x = f_y$ et $s = 0$. La contrainte douce correspondante prend la forme suivante :

$$\widehat{\mathbf{K}},\widehat{\boldsymbol{r}},\widehat{\boldsymbol{T}} = \arg\min \sum_{i=1}^{n} \left\| \widetilde{\boldsymbol{q}}_i - \pi\left[\widehat{\mathbf{K}}\begin{pmatrix}\mathbf{R}(\widehat{\boldsymbol{r}}) & \widehat{\boldsymbol{T}}\end{pmatrix}\boldsymbol{Q}_i\right]\right\|^2 + w\left(\|f_x - f_y\|^2 + \|s\|^2\right). \tag{2.48}$$

où Hartley et Zisserman (2004) propose d'initialiser le poids $w \in \mathbb{R}$ à zéro et de le faire croître à chaque itération de l'optimisation jusqu'à convergence.

2.5 Géométrie multi-vue

Lorsqu'une même scène est observée de différents points de vue, il est possible d'estimer le déplacement relatif entre les différentes caméras et de calculer la

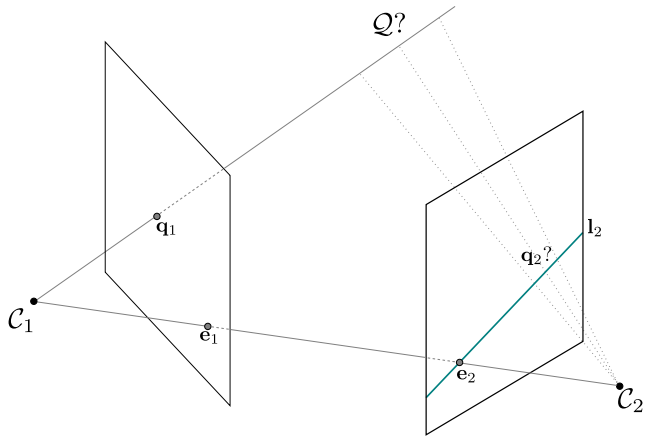


FIGURE 2.2 – La géométrie épipolaire définit des contraintes géométriques entre les différentes observations d'un même point de l'espace.

géométrie 3D de l'environnement observé. Ce cas de figure peut apparaître dans différentes configurations :

configuration spatiale si plusieurs caméras observent simultanément une même scène à partir de différents points de vue ;

configuration temporelle quand une seule caméra se déplace dans l'environnement, l'ensemble des vues correspondant alors aux points de vue de la caméra capturés à des instants différents.

Il est important de noter que ces deux configurations, *dans le cas d'une scène rigide*, sont équivalentes et peuvent être traitées de façon identique. Cette partie se consacrera à l'étude de la géométrie entre deux vues. Des méthodes complémentaires sur 3 et N vues peuvent être trouvées dans le livre de Hartley et Zisserman (2004).

2.5.1 Géométrie épipolaire

La *géométrie épipolaire* décrit les contraintes reliant les observations d'une même scène observée par deux caméras, notées $\mathscr{C}_1$ et $\mathscr{C}_2$ (figure 2.2). Ces contraintes sont directement liées au déplacement relatif (également appelé positionnement relatif) entre les deux caméras mais sont totalement indépendantes de la structure de la scène. Toutefois, il est important de rappeler que dans le cas du déplacement d'une caméra (*i.e.* dans le cas de la configuration temporelle), la géométrie épipolaire est uniquement vérifiée si la scène observée est rigide.

Matrice fondamentale

La *matrice fondamentale* exprime la relation épipolaire dans le cas où les paramètres internes des caméras sont inconnus. Ainsi, pour un point $\boldsymbol{q}_1$ de l'image de la première caméra, il est possible de calculer la droite $\boldsymbol{l}_2$ sur laquelle se situe l'observation correspondante dans la deuxième caméra (figure 2.2) :

$$\boldsymbol{l}_2 \sim \mathbf{F}\dot{\boldsymbol{q}}_1 \tag{2.49}$$

La droite $\boldsymbol{l}_2$ est appelée *droite épipolaire* associée à $\boldsymbol{q}_1$. De plus, si deux observations $\boldsymbol{q}_1$ et $\boldsymbol{q}_2$ correspondent au même point de l'espace, elles vérifient :

$$\dot{\boldsymbol{q}}_2^\top \mathbf{F}\dot{\boldsymbol{q}}_1 = 0 \tag{2.50}$$

Cette relation permet d'estimer la matrice fondamentale à partir d'associations 2D entre deux images. En pratique, $\mathbf{F}$ peut se calculer à l'aide de 8 points (Hartley, 1997) ou à partir de 7 points sous certaines hypothèses (Torr et Murray, 1997).

Dans chacune des images, un point joue un rôle particulier. Il s'agit des deux *épipôles* $\boldsymbol{e}_1$ et $\boldsymbol{e}_2$. Elles correspondent à la projection dans l'image du centre optique de l'autre caméra. Les épipôles présentent deux caractéristiques intéressantes. Tout d'abord, elles définissent le noyau de $\mathbf{F}$: $\mathbf{F}\dot{\boldsymbol{e}}_1 = \mathbf{F}\dot{\boldsymbol{e}}_2 = 0$. De plus, les épipôles correspondent aux points d'intersection de toutes les droites épipolaires de chacune des images.

Matrice essentielle

La *matrice essentielle* $\mathbf{E}$ peut être vue comme le cas particulier de la matrice fondamentale dans le cas où l'étalonnage des caméras ($\mathbf{K}_1$ et $\mathbf{K}_2$) est connu. La relation entre matrice essentielle et matrice fondamentale est la suivante :

$$\mathbf{E} \sim {\mathbf{K}_2}^\top \mathbf{F}\mathbf{K}_1 \tag{2.51}$$

L'équation (2.50) devient dans ce cas :

$$\dot{\boldsymbol{q}}_2^\top \left(\mathbf{K}_2^{-\top}\mathbf{E}\mathbf{K}_1^{-1}\right)\dot{\boldsymbol{q}}_1 = 0 \tag{2.52}$$

où $\mathbf{K}_2^{-\top}$ est la transposée inverse de $\mathbf{K}_2$. Pour estimer la matrice essentielle, Nistér (2004) a proposé un algorithme efficace appelé *algorithme des 5 points*.

Relation entre matrice essentielle et déplacement relatif

La matrice essentielle peut avoir différentes utilisations. En effet, il existe une relation qui lie la matrice essentielle d'un couple de caméras ($\mathscr{C}_1$, $\mathscr{C}_2$) au déplacement relatif entre ces caméras. Le déplacement relatif est défini par le couple

$(\mathbf{R}_{1\to2}, \boldsymbol{T}_{1\to2})$ dont une formalisation en sera faite à la section 2.5.2. La relation entre $\mathbf{E}$, $\mathbf{R}_{1\to2}$ et $\boldsymbol{T}_{1\to2}$ s'écrit :

$$\mathbf{E} = [\boldsymbol{T}_{1\to2}]_\times \mathbf{R}_{1\to2} \tag{2.53}$$

où $[\boldsymbol{T}]_\times$ est la matrice antisymétrique construite à partir du vecteur $\boldsymbol{T}$, à savoir :

$$[\boldsymbol{T}]_\times = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix} \tag{2.54}$$

Une estimation de la matrice essentielle (grâce à l'appariement d'au moins 5 points) permet donc de retrouver le déplacement relatif entre les caméras. Cette notion sera développée dans la section 2.5.2.

2.5.2 Calcul de la géométrie de la scène observée

Dans cette section, nous allons présenter l'ensemble des outils mathématiques élémentaires qui permettent de calculer la géométrie d'une scène 3D, à savoir la pose des différentes caméras ainsi que le nuage de points 3D associés aux points d'intérêt observés.

Poses de caméras et déplacement relatif

Le but de cette section est de formaliser la notion de *déplacement relatif* entre deux caméras ainsi que les notations associées. Comme nous l'avons vu précédemment, la pose des caméras peut être vue comme un changement de repère entre le repère monde et les repères attachés aux caméras :

$$\boldsymbol{Q}_{\mathscr{C}_1} = \begin{pmatrix} \mathbf{R}_1 & \boldsymbol{T}_1 \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}} \tag{2.55}$$

$$\boldsymbol{Q}_{\mathscr{C}_2} = \begin{pmatrix} \mathbf{R}_2 & \boldsymbol{T}_2 \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{W}} \tag{2.56}$$

avec $\boldsymbol{Q}_{\mathscr{W}}$ un point 3D exprimé dans le repère monde et $\boldsymbol{Q}_{\mathscr{C}_1}$ et $\boldsymbol{Q}_{\mathscr{C}_2}$ ses coordonnées respectives dans les repères liés aux caméras $\mathscr{C}_1$ et $\mathscr{C}_2$. Afin de fixer les notations, nous appellerons $(\mathbf{R}_{1\to2}, \boldsymbol{T}_{1\to2})$ le déplacement relatif entre les caméras, c'est à dire la transformation permettant de passer du repère lié à $\mathscr{C}_1$ à celui lié à $\mathscr{C}_2$:

$$\boldsymbol{Q}_{\mathscr{C}_2} = \begin{pmatrix} \mathbf{R}_{1\to2} & \boldsymbol{T}_{1\to2} \end{pmatrix} \dot{\boldsymbol{Q}}_{\mathscr{C}_1} \tag{2.57}$$

Des équations (2.55) et (2.56), on obtient :

$$\begin{cases} \mathbf{R}_{1\to2} = \mathbf{R}_2 {\mathbf{R}_1}^\top \\ \boldsymbol{T}_{1\to2} = \boldsymbol{T}_2 - \mathbf{R}_2 {\mathbf{R}_1}^\top \boldsymbol{T}_1 \end{cases} \tag{2.58}$$

Calcul du déplacement relatif par associations 2D/2D

Lorsque la structure de l'environnement est inconnue, il est tout de même possible de calculer le déplacement relatif entre deux caméras. Cela nécessite d'associer les observations des deux caméras qui correspondent aux mêmes points 3D de l'environnement. Comme nous l'avons vu précédemment (section 2.5.1), ceci permet de calculer la matrice fondamentale (algorithme des 8 points, Hartley, 1997) ou essentielle (algorithme des 5 points, Nistér, 2004). Il est alors possible d'extraire d'une de ces matrices le déplacement inter-caméra ($\mathbf{R}_{1\to2}, \boldsymbol{T}_{1\to2}$).

Dans le cas de caméras non étalonnées, $\mathbf{R}_{1\to2}$ et $\boldsymbol{T}_{1\to2}$ sont calculés à partir de la matrice fondamentale (Hartley et Zisserman, 2004). Dans ce cas, le déplacement inter-caméra ne peut être retrouvé qu'à une transformation projective près. En particulier, ceci induit qu'il est impossible de retrouver les rapports de distance et les angles.

Si l'étalonnage des caméras est connu, il est préférable d'utiliser la matrice essentielle. La décomposition en valeurs singulières (Faugeras, 1993) de celle-ci permet en effet d'en extraire 4 couples de solutions possibles pour $\mathbf{R}_{1\to2}$ et $\boldsymbol{T}_{1\to2}$. Parmi ces 4 couples, on retient celui permettant de reconstruire les 5 points ayant servi au calcul de $\mathbf{E}$ devant les deux caméras. Le détail de cette décomposition peut être trouvé dans l'article de Nistér (2004).

Dans le cas étalonné, le déplacement relatif entre les deux caméras (et donc toute la structure 3D sous-jacente) est défini à un facteur près. En effet, dans le cas du calcul du déplacement par associations 2D/2D, le facteur d'échelle de la scène (c'est à dire sa métrique) n'est pas observable. En pratique, cette échelle est donc fixée arbitrairement.

Notons également que seul le déplacement relatif est défini mais pas la pose des caméras dans le repère monde. En effet, aucune information de localisation absolue n'est fournie de sorte que les deux caméras obtenues sont positionnées à une rotation et une translation près dans le monde. Ainsi, si le déplacement relatif est défini à un facteur près, la pose absolue des caméras est définie à 7 degrés près. Une transformation 3D possédant ces 7 degrés de liberté est appelée *similitude* et peut être représentée par la matrice homogène suivante :

$$\mathbf{S} \sim \begin{pmatrix} s\mathbf{R} & \boldsymbol{T} \\ 0_{1\times3} & 1 \end{pmatrix} \tag{2.59}$$

avec s le facteur d'échelle, $\mathbf{R}$ la rotation et $\boldsymbol{T}$ la translation.

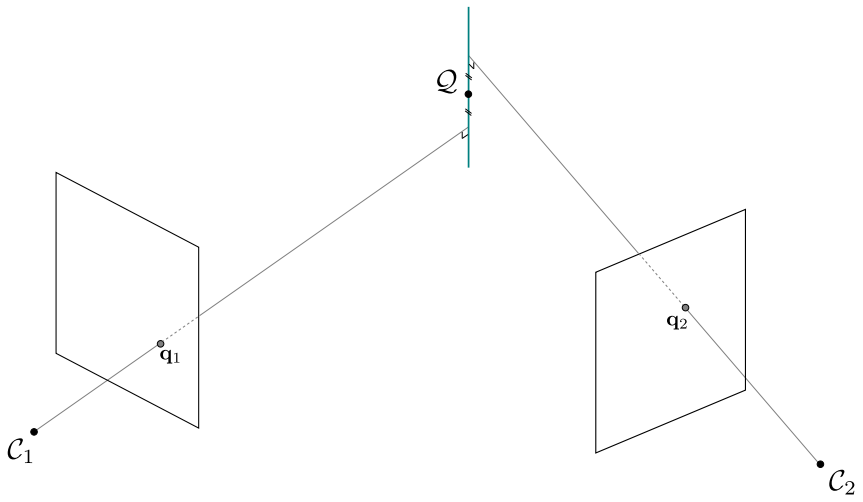


FIGURE 2.3 – La structure de l'environnement peut être obtenue par triangulation des observations dans les images.

Calcul de la structure de l'environnement (triangulation)

Nous avons vu que la rétroprojection d'une observation 2D d'un point de l'espace permet d'obtenir sa position 3D à la profondeur près (section 2.4.2). Dès lors qu'au moins deux caméras dont la pose et l'étalonnage sont connus observent ce point, la profondeur du point peut être estimée. On parle alors de *triangulation* du point. L'idée de la triangulation est de calculer l'intersection des rayons optiques issus des deux observations. En pratique, à cause des bruits sur les différentes données (étalonnage, pose des caméras, position des observations...), les rayons ne s'intersectent pas. Dans le cas de deux caméras, le résultat de la triangulation est le point équidistant des deux rayons (figure 2.3).

Calcul de pose par associations 2D/3D

Une fois la structure de l'environnement partiellement connue, il est possible de calculer la pose d'une caméra tiers à partir d'associations réalisées entre les observations 2D de son image et la position 3D de 3 points de l'environnement. De nombreuses méthodes ont été proposées pour résoudre ce problème. Une comparaison de certaines de ces méthodes peut être trouvée dans l'article de Haralick *et al.* (1994). Plus récemment, Lepetit *et al.* (2009) ont proposé une nouvelle approche plus performante (en temps de calcul et en précision) du calcul de pose.

L'utilisation d'associations 2D/3D plutôt que 2D/2D présente plusieurs avantages.

Tout d'abord, il est à noter que le calcul de pose 2D/3D est beaucoup plus rapide que le calcul de pose 2D/2D (l'estimation de la matrice essentielle étant une étape coûteuse). De plus, nous avons vu précédemment que l'extraction des paramètres à partir de la matrice essentielle ne permet pas d'estimer le facteur d'échelle et donc en particulier la norme de la translation entre les différentes caméras. Avec l'approche 2D/3D, le facteur d'échelle peut être estimé à partir de l'observation de la distance entre les différents points de l'espace. Enfin, Tardif *et al.* (2008) ont montré que l'utilisation de l'approche 2D/3D offre un calcul plus précis de la position de la caméra.

Erreur de reprojection et ajustement de faisceaux

Lorsqu'un ensemble de points 3D et de caméras sont reconstruits à l'aide des méthodes définies précédemment, il est nécessaire de définir une erreur permettant de mesurer la qualité de cette reconstruction. L'idée principale de cette erreur est de mesurer la distance entre l'endroit où le point est détecté dans l'image et sa position estimée. Si des erreurs 3D ont été proposées (par exemple mesurer la distance entre le rayon optique issu de l'observation et le point 3D), il a été montré qu'il est généralement préférable d'utiliser une erreur 2D (Lu *et al.*, 2000), en particulier pour éviter que les points 3D au loin aient une erreur plus importante du fait de leur profondeur.

La solution couramment retenue est *l'erreur de reprojection* (figure 2.4), notée r. Elle consiste à mesurer la distance 2D entre l'observation du point 3D dans l'image (c'est à dire la position 2D du point d'intérêt) et la projection du point 3D reconstruit dans cette même image :

$$r = \|\widetilde{\boldsymbol{q}} - \pi(\widehat{\mathbf{P}}\dot{\widehat{\boldsymbol{Q}}})\| \tag{2.60}$$

où $\widetilde{\boldsymbol{q}}$ est l'observation 2D bruitée, $\widehat{\mathbf{P}}$ la matrice de projection estimée et $\widehat{\boldsymbol{Q}}$ le point 3D reconstruit.

Les méthodes de calcul de pose des caméras et de la structure de l'environnement telles qu'elles ont été présentées précédemment ne fournissent pas une solution optimale au problème de reconstruction et localisation simultanées. Pour corriger cela, il est possible de raffiner l'ensemble des paramètres de la scène (à savoir les 6 paramètres de pose de chaque caméra et les 3 paramètres de la position de chaque point 3D) en cherchant à minimiser l'erreur de reprojection pour chacun des couples caméra-point 3D observé. On parle alors d'*ajustement de faisceaux*. La fonction à minimiser s'écrit donc :

$$C(\widehat{\mathbf{C}}_1^E, \ldots, \widehat{\mathbf{C}}_N^E, \widehat{\boldsymbol{Q}}^1, \ldots, \widehat{\boldsymbol{Q}}^M) = \sum_{1 \leq j \leq N} \sum_{i \in \mathcal{A}_j} \|\widetilde{\boldsymbol{q}}_j^i - \pi(\widehat{\mathbf{P}}_j \dot{\widehat{\boldsymbol{Q}}}^i)\|^2 \tag{2.61}$$

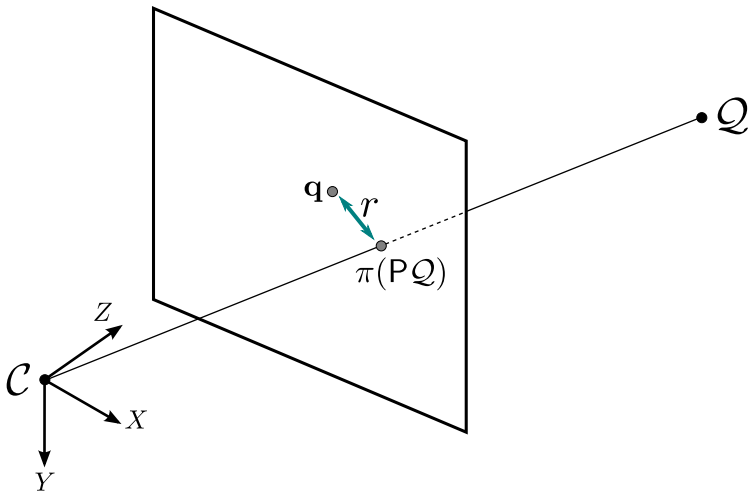


FIGURE 2.4 – L'erreur de reprojection est la distance r entre l'observation $\boldsymbol{q}$ d'un point $\boldsymbol{Q}$ et sa projection dans l'image $\pi(\mathbf{P}\dot{\boldsymbol{Q}})$.

où les $\widehat{\mathbf{C}}^{E}_{i=1\ldots N}$ sont les 6 paramètres extrinsèques, $\widehat{\mathbf{P}}_{j=1\ldots N}$ les matrices de projection des caméras, $\widehat{\boldsymbol{Q}}^{i=1\ldots M}$ les points 3D et $\widetilde{\boldsymbol{q}}^{i}_{j}$ l'observation du point i dans la caméra j. L'ensemble $\mathcal{A}_j$ contient les indices des points 3D vus par la caméra j. Afin de minimiser cette fonction de coût, on utilisera un algorithme de minimisation non linéaire. Ce type d'algorithme sera décrit dans la section 2.8.

2.5.3 Cas d'une scène plane

Dans cette section, nous présentons ce que deviennent les relations qui existent entre deux caméras dans le cas où la scène observée est planaire.

Nous nous plaçons dans le cas de la figure 2.5 : deux caméras $\mathscr{C}_1$ et $\mathscr{C}_2$ observant un plan Π. La projection de ce plan de l'espace dans le plan image (et réciproquement) définit une homographie 2D. Un résultat intéressant est alors que, pour tout point 3D $\boldsymbol{Q}$ appartenant au plan Π, la fonction passant des coordonnées de son observation $\boldsymbol{q}_1$ dans la première image aux coordonnées de son observation $\boldsymbol{q}_2$ dans la deuxième image est également une homographie. En effet, la composition de 2 homographies est une homographie. Le lien entre les observations peut donc s'écrire :

$$\dot{\boldsymbol{q}}_2 \sim \mathbf{H}_{1\to 2}\dot{\boldsymbol{q}}_1. \tag{2.62}$$

Il est possible de définir une relation entre le déplacement relatif des caméras

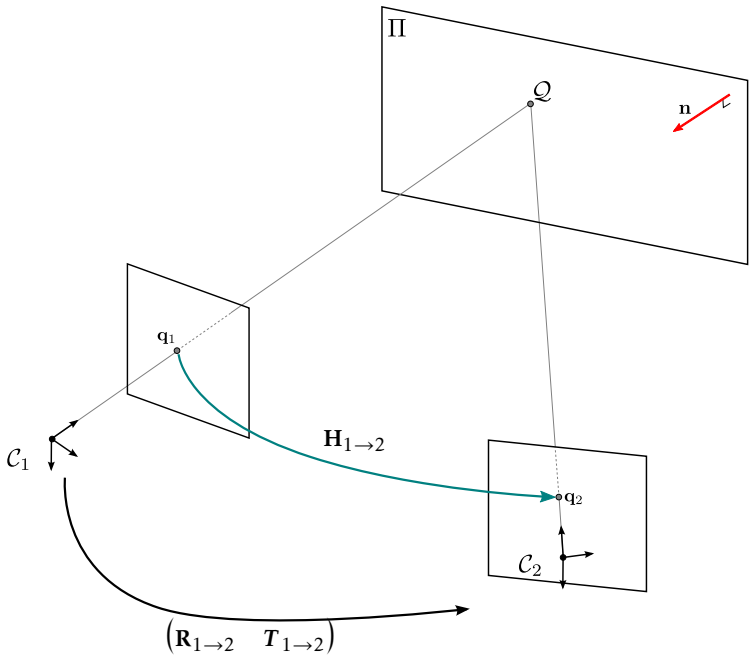


FIGURE 2.5 – Les coordonnées des observations correspondantes de points 3D situés sur un même plan de l'espace sont reliées par une homographie 2D.

$(\mathbf{R}_{1\to2}, \boldsymbol{T}_{1\to2})$, l'équation du plan observé Π et cette homographie $\mathbf{H}_{1\to2}$ liant les observations de ce plan dans les 2 images.

Pour la démonstration, on se place dans le repère lié à $\mathscr{C}_1$. Soit un point 3D $\boldsymbol{Q}$ appartenant au plan Π, alors son observation dans $\mathscr{C}_1$ vérifie :

$$\boldsymbol{q}_1 = \pi(\mathbf{K}_1\boldsymbol{Q}) \quad \Leftrightarrow \quad \exists\lambda \in \mathbb{R} \text{ tel que } \boldsymbol{Q} = \lambda\mathbf{K}_1^{-1}\dot{\boldsymbol{q}}_1, \tag{2.63}$$

De plus, soit $\boldsymbol{n}$ la normale au plan Π, toujours exprimée dans le repère lié à $\mathscr{C}_1$, l'équation du plan Π est $(\boldsymbol{n}^\top\ d)^\top$, où d est la distance de $\mathscr{C}_1$ au plan. $\boldsymbol{Q} \in \Pi$ vérifie donc :

$$\boldsymbol{n}^\top\boldsymbol{Q} + d = 0. \tag{2.64}$$

En injectant l'équation (2.63) on peut déterminer la profondeur de $\boldsymbol{Q}$:

$$\lambda = -\frac{d}{\boldsymbol{n}^\top\mathbf{K}_1^{-1}\dot{\boldsymbol{q}}_1}. \tag{2.65}$$

Par ailleurs la projection de $\boldsymbol{Q}$ dans $\mathscr{C}_2$ vérifie (en utilisant l'équation (2.57)) :

$$\dot{\boldsymbol{q}}_2 \sim \mathbf{K}_2\begin{pmatrix}\mathbf{R}_{1\to2} & \boldsymbol{T}_{1\to2}\end{pmatrix}\dot{\boldsymbol{Q}} \sim \mathbf{K}_2(\mathbf{R}_{1\to2}\boldsymbol{Q} + \boldsymbol{T}_{1\to2}), \tag{2.66}$$

puis en injectant (2.63) et (2.65) :

$$\dot{\boldsymbol{q}}_2 \sim \mathbf{K}_2 \left(\lambda \mathbf{R}_{1\to 2} \mathbf{K}_1^{-1} \dot{\boldsymbol{q}}_1 + \boldsymbol{T}_{1\to 2} \right) \tag{2.67}$$

$$\sim \mathbf{K}_2 \left(\frac{d}{\boldsymbol{n}^\top \mathbf{K}_1^{-1} \dot{\boldsymbol{q}}_1} \mathbf{R}_{1\to 2} \mathbf{K}_1^{-1} - \begin{pmatrix} 0_{3\times 2} & \boldsymbol{T}_{1\to 2} \end{pmatrix} \right) \dot{\boldsymbol{q}}_1 . \tag{2.68}$$

D'après l'équation (2.62), l'homographie $\mathbf{H}_{1\to 2}$ peut donc s'exprimer :

$$\mathbf{H}_{1\to 2} \sim \mathbf{K}_2 \left(\frac{d}{\boldsymbol{n}^\top \mathbf{K}_1^{-1} \dot{\boldsymbol{q}}_1} \mathbf{R}_{1\to 2} \mathbf{K}_1^{-1} - \begin{pmatrix} 0_{3\times 2} & \boldsymbol{T}_{1\to 2} \end{pmatrix} \right) \tag{2.69}$$

ce qui fait le lien avec le déplacement inter-caméra.

Faugeras (1993) a montré qu'il est possible d'utiliser cette relation pour extraire les paramètres du déplacement relatif et les paramètres liés au plan. Il est à noter qu'il y a une ambigüité entre la norme de $\boldsymbol{T}_{1\to 2}$ et la distance au plan d. Cela revient à dire que, comme pour la matrice essentielle, la norme du déplacement entre les deux caméras n'est pas directement estimable. Pour cela, il est nécessaire de connaître *a priori* la distance d.

2.6 Correspondances de primitives géométriques

Une primitive géométrique est un objet mathématique déterminé par un nombre fini de paramètres (point, ligne droite, courbe de Bézier...). Les correspondances de primitives géométriques entre deux images permettent d'obtenir des correspondances 2D-2D, utiles à la plupart des problèmes évoqués dans la section précédente. La mise en correspondance suit le plus souvent le schéma suivant :

- *détection* de primitives dans les deux images,
- *description* par association d'un vecteur (*descripteur*) identifiant chaque primitive,
- *mise en correspondance* des primitives avec leur plus proche voisin et un éventuel filtrage des correspondances ambigües.

Chaque étape est détaillée dans les sections suivantes.

2.6.1 Détection de primitives géométriques

La détection de primitives consiste à extraire d'une image les primitives géométriques les plus saillantes. Deux qualités principales sont importantes pour les détecteurs : la précision des paramètres estimés et la répétabilité *i.e.* la capacité a détecter les mêmes primitives après diverses perturbations (bruit, changement de luminosité, distorsions affines et perspectives...). Nous considérons dans ce mémoire deux types de primitives : les points et les segments de droites.

Points

Les points sont les primitives géométriques les plus simples, et de loin les plus utilisées. Trois types de primitives sont en fait englobés dans ce terme :

Pixel : Un pixel est défini par des coordonnées entières dans le plan image. Ce type de primitive est principalement utilisé pour des approches denses (chaque pixel de l'image est mis en correspondance) ou semi denses utilisant une grille régulière.

Coin : Un coin est l'intersection de deux lignes. Les coins sont souvent détectés à partir du tenseur de structure (Harris et Stephens, 1988; Shi et Tomasi, 1994) :

$$\mathbf{T}(x,y) = \begin{pmatrix} \left(\frac{\partial I(x,y)}{\partial x}\right)^2 & \frac{\partial I(x,y)}{\partial x}\frac{\partial I(x,y)}{\partial y} \\ \frac{\partial I(x,y)}{\partial x}\frac{\partial I(x,y)}{\partial y} & \left(\frac{\partial I(x,y)}{\partial y}\right)^2 \end{pmatrix} * W \tag{2.70}$$

où W est un masque (voir section 2.3.2), par exemple gaussien. Les valeurs propres de ce tenseur représentent la distribution du gradient dans la fenêtre considérée. Soient $\lambda_1 \geq \lambda_2$ ces valeurs propres pour un couple (x,y) donné, les trois cas suivants sont possibles :
— l'image est constante sur la fenêtre : $\lambda_1 \approx \lambda_2 \approx 0$,
— une ligne est présente dans la fenêtre : $\lambda_1 \gg \lambda_2 \approx 0$,
— un coin est présent dans le fenêtre : $\lambda_1 \gg 0$ et $\lambda_2 \gg 0$.
Le principal avantage des coins est qu'ils peuvent être définis avec une précision sous-pixellique (Lucas et Kanade, 1981).

Blob : Un blob est un point associé à une échelle. Les méthodes classiques détectent les maximum 3D (position et échelle) de la réponse d'opérateurs différentiels. La représentation espace-échelle est définie comme suit :

$$I^*(x,y;\sigma) = I(x,y) * \mathbf{G}(\sigma) \tag{2.71}$$

où $\mathbf{G}$ est le noyau gaussien défini par :

$$\mathbf{G}(x,y;\sigma) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \tag{2.72}$$

Le laplacien de gaussienne normalisé est un des opérateurs les plus utilisés :

$$\nabla^2_{\text{norm}} I^*(x,y;\sigma) = \sigma\left(\frac{\partial^2 I^*(x,y;\sigma)}{\partial x^2} + \frac{\partial^2 I^*(x,y;\sigma)}{\partial y^2}\right). \tag{2.73}$$

Pour une échelle σ donnée, le laplacien de gaussienne a une réponse maximale pour des blobs de dimension $\sqrt{2}\sigma$. L'opérateur différence de gaussiennes (Marr et Hildreth, 1980; Lowe, 2004) permet d'approximer le laplacien sans dérivées spatiales.

L'inconvénient des blobs est qu'ils sont moins précisément localisés que les coins et peuvent se révéler instables dans certaines situations. Cependant ils peuvent grandement améliorer la mise en correspondance en la rendant invariante aux changements d'échelle entre les images.

Pour le lecteur intéressé, Mikolajczyk et Schmid (2004) font une comparaison détaillée des détecteurs usuels de points.

Segments de droite

Beaucoup de scènes observées contiennent des structures construites par l'Homme. Ces structures ont tendance à posséder peu de texture, rendant difficile la détection de points, mais affichent de nombreuses arrêtes droites. Il peut être pertinent de détecter de telles primitives qui couvrent une zone plus large qu'un point et sont potentiellement plus discriminantes. Plusieurs méthodes existent pour extraire des segments de droite d'une image, par exemple en étendant la méthode d'extraction de contours de Canny (1986) (Wang *et al.*, 2008), ou en utilisant un vote par région et une validation *a contrario* (Von Gioi *et al.*, 2012).

2.6.2 Description et mise en correspondance

Après avoir détecté des primitives géométriques, il convient de les identifier de la manière la plus discriminante possible pour obtenir des correspondances non ambigües. Il y a un compromis à effectuer entre robustesse (conservation du descripteur en présence de perturbations) et pouvoir discriminant (les descripteurs de deux primitives différentes doivent être le plus distants possible). Par exemple, l'utilisation d'un voisinage de grande taille rend le descripteur très discriminant mais moins robuste. Les techniques principales utiles à la compréhension du mémoire sont décrites ici pour les primitives considérées. On appelle I_1 et I_2 les deux images que l'on cherche à mettre en correspondance.

Points

Pour les points, il est souvent nécessaire de considérer un voisinage pour une description riche en information, usuellement une fenêtre carrée centrée sur le point considéré.

Par souci de clarté, on appelle $\boldsymbol{q}_1 \in \Omega_{I_1}$ et $\boldsymbol{q}_2 \in \Omega_{I_2}$ les deux points comparés, et W un masque (section 2.3.2) représentant le voisinage considéré. On note $S(\boldsymbol{q}_1, \boldsymbol{q}_2, I_1, I_2)$ la distance entre les points dans l'espace des descripteurs.

Différences absolues. La description la plus simple est l'intensité de l'image en ce point. Lors de la mise en correspondance on dit qu'on minimise les *différences absolues*, *i.e.* la valeur absolue de la différence d'intensité :

$$S_{\text{AD}}(\boldsymbol{q}_1, \boldsymbol{q}_2, I_1, I_2) = |I_1(\boldsymbol{q}_1) - I_2(\boldsymbol{q}_2)|. \tag{2.74}$$

L'extension de cette mesure à un voisinage est nommée *somme des différences absolues* :

$$S_{\text{SAD}}(\boldsymbol{q}_1, \boldsymbol{q}_2, I_1, I_2) = \left| \, (I_1 - T_{I_2}^{\boldsymbol{q}_2 - \boldsymbol{q}_1}) * W \, \right|. \tag{2.75}$$

où $T_{I_2}^{\boldsymbol{q}_2 - \boldsymbol{q}_1}$ est l'image I_2 translatée par le vecteur $\boldsymbol{q}_2 - \boldsymbol{q}_1$ (voir 2.3.2). Le principal inconvénient de ces descripteur est leur grande sensibilité aux variations d'illuminations. Les différences absolues par pixel sont très peu discriminantes et doivent être associées à des informations additionnelles pour une mise en correspondance plus fiable. La somme des différences absolues est plus discriminante mais moins robuste aux déformations affines autres que la translation pure.

Gradient. Le gradient de l'image encode la structure de l'image sous la forme de deux informations : la direction et l'amplitude des variations d'intensité. Le coût associé à une mise en correspondance est :

$$S_{\text{grad}}(\boldsymbol{q}_1, \boldsymbol{q}_2, I_1, I_2) = \|\nabla I_1(\boldsymbol{q}_1) - \nabla I_2(\boldsymbol{q}_2)\|. \tag{2.76}$$

En tant que descripteur, il est plus discriminant que l'intensité et robuste aux variations additives de luminosité. Par contre il est très sensible à toute transformation affine, ainsi qu'à la présence de bruit. Il est rarement utilisé directement mais il est par contre intrinsèquement lié aux descripteurs suivants.

Census. La transformée de Census (Zabih et Woodfill, 1994) est une représentation compacte de la structure locale autour d'un pixel. Elle peut être considérée comme une version généralisée et discrétisée du gradient (Hafner *et al.*, 2013). Nous utiliserons la version ternaire de Ranftl *et al.* (2012), définie comme suit :

$$\Theta\, I(x, y, \epsilon, n) = \bigoplus_{\substack{-n \le i \le n \\ -n \le j \le n}} \begin{cases} 0 & \text{si} \quad I(x+i, y+j) - I(x, y) < -\epsilon \\ 1 & \text{si} \quad -\epsilon \le I(x+i, y+j) - I(x, y) \le \epsilon \\ 2 & \text{si} \quad I(x+i, y+j) - I(x, y) > \epsilon \end{cases} \tag{2.77}$$

(a) (b) (c)

FIGURE 2.6 – Une fenêtre 3×3 (a), sa transformée de Census en représentation numérique (b) et binaire (c).

où $\bigoplus$ est l'opérateur de concaténation, $\epsilon \in \mathbb{R}$ un seuil et $n \in \mathbb{N}^*$ la taille de la fenêtre considérée. La distance de Census est :

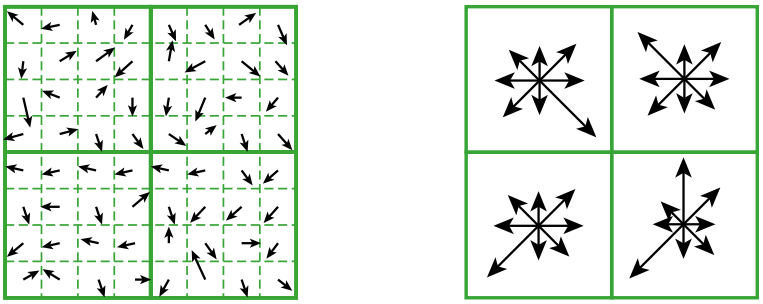
$$S^{(\epsilon,n)}_{\text{Census}}(\boldsymbol{q}_1, \boldsymbol{q}_2, I_1, I_2) = |\Theta I_1(\boldsymbol{q}_1, \epsilon, n) - \Theta I_2(\boldsymbol{q}_2, \epsilon, n)|. \tag{2.78}$$

En pratique une représentation binaire est utilisée (voir figure 2.6) avec la distance de Hamming (1950) pour des comparaisons plus efficaces.

Census est robuste à tout changement d'illumination monotone. Il est, ainsi que d'autres descripteurs similaires (Ojala *et al.*, 2002) très utilisé quand la rapidité de calcul est importante.

HOG. Le descripteur HOG (*Histogram of Oriented Gradients*, Dalal et Triggs, 2005) accumule sur une fenêtre les informations de direction du gradient au sein d'un histogramme. Chaque pixel contribue à la classe de l'histogramme correspondant à la direction du gradient avec un poids proportionnel à son amplitude. De tels descripteurs sont surtout utilisés de manière dense pour la détection d'objets.

SIFT. Le descripteur SIFT (*Scale-Invariant Feature Transform*, Lowe, 2004) est une amélioration du descripteur HOG, plus robuste et plus discriminante, dédiée à la description de points d'intérêts épars. Des descripteurs HOG sont calculés dans le voisinage du point d'intérêt considéré (voir figure 2.7) puis concaténés. Le descripteur obtenu est de grande dimension (128 dans l'implémentation originale) ce qui lui permet d'être très discriminant. La normalisation de ce vecteur, ainsi que d'autres opérations (lissage gaussien, seuillage de l'amplitude du gradient) permettent de rendre ce descripteur robuste à la plupart des variations d'illumination et aux changements de point de vue jusqu'à environ 50°.



(a) (b)

FIGURE 2.7 – Champ de gradients en (a) et les histogrammes de gradients orientés utilisés pour le descripteur SIFT en (b).

ASIFT. ASIFT (*Affine SIFT*, Yu et Morel, 2011) est une extension de SIFT. Des transformations affines variées sont appliquées à l'image et pour chaque transformation un descripteur SIFT est calculé pour chaque point d'intérêt. Cela permet de rendre la mise en correspondance robuste à des déformations affines très importantes, au prix d'un temps de calcul accru.

DAISY. Le descripteur DAISY (Tola *et al.*, 2010) est une concaténation d'histogrammes de gradients orientés, comme SIFT, mais il couvre une large zone circulaire, de diamètre entre 10 et 30 pixels. Les histogrammes sont échantillonnés sur des cercles concentriques, avec des degrés de lissage plus importants en fonction de la distance au centre (voir figure 2.8). Ce fonctionnement, inspiré de la vision humaine, permet une robustesse aux déformations perspectives, sous réserve de détecter les occultations qui peuvent sinon grandement perturber les correspondances en raison de la taille du descripteur. Des descripteurs similaires existent par ailleurs (Mikolajczyk et Schmid, 2005, descripteur GLOH) mais DAISY a la particularité d'avoir été conçu pour un calcul dense efficace.

Espaces de couleur

Les images en couleurs peuvent être représentées par trois canaux. La manière la plus simple de construire des descripteurs de primitives en couleur est de concaténer [1] les descripteurs précédents calculés sur chaque canal comme le proposent van de Sande *et al.* (2008). Le choix de l'espace de couleur est alors déterminant.

1. Dans ce mémoire, les distances entre descripteurs couleurs sont divisées par le nombre de canaux pour garder une même amplitude que les descripteurs en niveaux de gris.

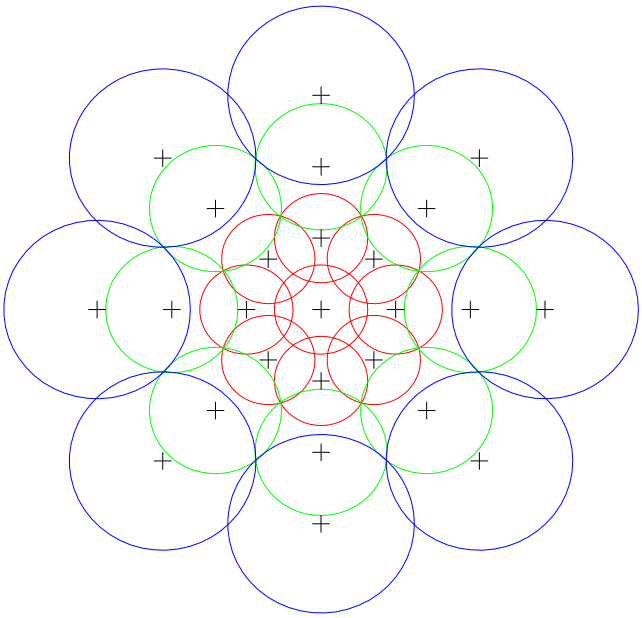


FIGURE 2.8 – Descripteur DAISY : concaténation d'histogrammes de gradients orientés en cercles concentriques. Les croix représentent les points de calcul des histogrammes. Les cercles colorés représentent la déviation standard du lissage gaussien utilisé pour chaque histogramme. Image extraite de l'article de Tola *et al.* (2010).

Nous en listons deux ici et nous référons le lecteur intéressé vers un ouvrage de référence tel que Wyszecki et Stiles (1982) pour plus de détails.

RGB. La représentation utilisée nativement par la plupart des applications informatiques, issue de la théorie de synthèse additive des couleurs, est composé d'un canal par couleur primaire : rouge, vert et bleu, abrégé en RVB (*RGB* en anglais). L'inconvénient de cette représentation est qu'elle ne correspond pas au processus de vision humaine qui interprète indépendamment la luminosité et la couleur.

HSV. La représentation teinte saturation lumière (*Hue Saturation Value* en anglais) utilise un canal (valeur : V) pour la luminosité et deux canaux (teinte : H et saturation : S) pour la couleur. Cette représentation est utile pour sélectionner une couleur dans les logiciels de dessin par exemple, mais par soucis d'efficacité elle n'utilise que des transformations simples de l'espace RGB qui la laisse peu représentative de la vision humaine (voir figure 2.9).

L*a*b*. L'espace de couleur CIE [2] L*a*b* est composé d'un canal de luminosité (L) et deux canaux pour la couleur, basés sur la théorie des couleurs opposées selon laquelle l'œil humain capte seulement des différences de couleurs. La transformation L*a*b* a été conçue pour reproduire le plus fidèlement possible la vision humaine (voir figure 2.9), c'est à dire que la distance euclidienne entre deux couleurs de cet espace est proportionnelle à leur différence apparente. La transformation de RGB vers L*a*b* est complexe mais automatisée par de nombreuses bibliothèques informatiques (Bradski, 2000). La transformation inverse n'est pas toujours possible car la représentation L*a*b* couvre toutes les couleurs visibles par l'œil humain, contrairement à la représentation RGB : la figure 2.10 représente l'espace de couleur RGB dans l'espace L*a*b*, à luminosité constante.

Segments de droite

Les segments de droite nécessitent des approches différentes pour être mis en correspondance.

Plus petite distance orthogonale. Lorsque les différences entre images sont faibles, une stratégie possible (Harris et Stennett, 1990) est d'associer simplement les segments de I_1 à l'arrête la plus proche dans I_2, sous réserve qu'ils aient une orientation similaire. En revanche cette approche est peu robuste et difficilement généralisable aux déformations importantes.

2. Commission Internationale de l'Éclairage

(a) Image (b) Canal L* de L*a*b* (c) Canal V de HSV

FIGURE 2.9 – Représentation des canaux de luminance des espaces colorimétriques HSV et L*a*b*. Le canal V est trop saturé sur la flamme alors que la canal L* correspond à la vision humaine. *Source : Wikipedia (HSL_and_HSV).*

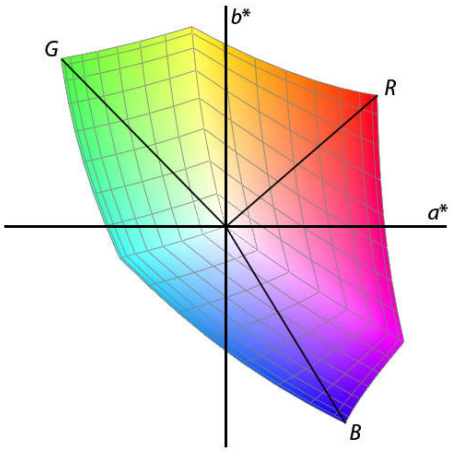


FIGURE 2.10 – Représentation de l'espace de couleur sRGB dans l'espace CIE L*a*b*. *Source : Wikipedia (Srgb-in-cielab.png).*

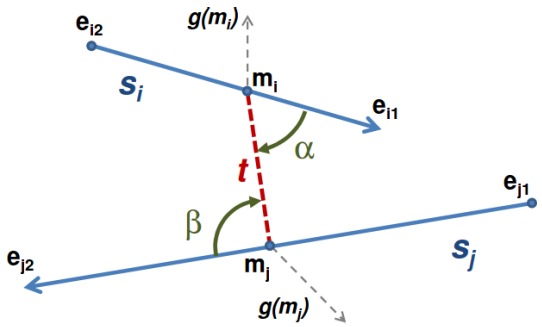


FIGURE 2.11 – Exemple de descripteur géométrique : le descripteur de segments de droite BOLD par Tombari *et al.* (2013) est basé sur les angles α et β.

Échantillonnage de points autour des segments. Il est aussi possible d'échantillonner des points d'intérêts autour du segment et d'utiliser des descripteurs de points pour la mise en correspondance (Wang *et al.*, 2009b; Hirose et Saito, 2012). En assurant une cohérence spatiale entre ces groupes de points, les fausses associations sont grandement réduites. La robustesse aux transformations géométriques ou aux variations de luminosité reste cependant contrainte par le descripteur de point utilisé.

Descripteur géométrique. La dernière catégorie de descripteur de segments de droite, que nous appelons *descripteurs géométriques*, identifie en fait des groupes de segments. Ces méthodes (Wang *et al.*, 2009a; Tombari *et al.*, 2013) utilisent pour cela des rapports d'angles (voir figure 2.11) et de longueurs, robuste aux variations de luminosité (tant que les segments restent visibles) ainsi qu'aux déformations affines.

2.6.3 Filtrage

Sans *a priori* fort sur les correspondances estimées, il est impossible de garantir l'élimination des correspondances erronées ; deux méthodes sont cependant couramment employées pour supprimer les correspondances les plus ambigües.

La première méthode utilise une validation croisée. Soient une primitive $\mathbf{f}_1$ dans l'image I_1 et $\mathbf{f}_2$ son plus proche voisin dans l'espace des descripteurs parmi les primitives de I_2. La correspondance entre $\mathbf{f}_1$ et $\mathbf{f}_2$ n'est acceptée que si $\mathbf{f}_1$ est également le plus proche de voisin de $\mathbf{f}_2$ parmi les primitives de I_1. Cette méthode peut être rendue moins sélective en considérant les $k > 1$ plus proches voisins. Elle est simple à implémenter mais nécessite deux recherches de plus proche voisin qui peuvent s'avérer coûteuses.

L'autre approche, introduite par Lowe (2004), consiste à assurer une marge entre

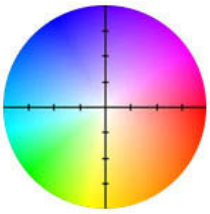


FIGURE 2.12 – Convention couleur de Baker *et al.* (2011) pour la représentation du flot optique. La teinte encode la direction du mouvement et la saturation encode son amplitude.

le premier et le plus proche voisin pour limiter l'ambiguïté. Par exemple, soient d_1 et d_2 les distances aux deux plus proches voisins d'une primitive, la mise en correspondance n'est acceptée que si $d_1 < \alpha d_2$ où $\alpha < 1$ est un facteur prédéfini.

2.7 Correspondances denses

Dans certaines situation, il est préférable d'obtenir des correspondances denses entre les images. Deux représentations équivalentes peuvent être utilisées : un champ de correspondances ($\mathbf{a} \in \mathbb{V}$) ou de déplacements ($\boldsymbol{u} \in \mathbb{V}$) tels que :

$$\forall \boldsymbol{q} \in \Omega_{I_1} \quad : \quad I_1(\mathbf{a}(\boldsymbol{q})) = I_1(\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q})) \approx I_2(\boldsymbol{q}). \tag{2.79}$$

Le champ de déplacements $\boldsymbol{u}$ est également appelé *flot optique*, surtout lorsqu'il est estimé entre deux images consécutives d'une séquence vidéo. Les champs de déplacements seront représentés dans ce mémoire en utilisant la convention couleur de Baker *et al.* (2011), reproduite en figure 2.12. Lorsqu'un champ de déplacements est restreint à une dimension on parle de *carte de disparités*.

2.7.1 Terme de données

L'estimation d'un champ de correspondances est le plus souvent effectuée en minimisant une fonction de coût définie sur toute l'image. Elle est composée au minimum d'un terme de données dense de la forme suivante :

$$C_{\text{dense}}(\boldsymbol{u}, I_1, I_2) = \iint_{\Omega_{I_1}} D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\, d\boldsymbol{q}, \tag{2.80}$$

où D est construit à partir d'une distance S entre descripteurs de point (voir section 2.6.2) :

$$D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) = S(\boldsymbol{q}, \boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q}), I_1, I_2). \tag{2.81}$$

Les descripteurs seuls ne permettent généralement pas d'obtenir un champ de correspondances correct. En effet il s'agit d'un problème mal posé. Sauf exceptions, certaines zones seront trop ambigües (peu de texture, motifs répétitifs) pour être mises en correspondance avec des informations purement locales. Nous présentons ensuite deux approches pour contraindre le champ de correspondance et lui imposer une cohérence globale.

2.7.2 Modèles non paramétriques

Les modèles non paramétriques n'imposent aucun modèle de mouvement mais utilise un *a priori* pour résoudre les ambiguïtés et obtenir un champ de correspondance cohérent. Cela se traduit par un terme R à minimiser dépendant des variations du champ de déplacements : on parle de *régularisation*. La fonction de coût prend donc la forme :

$$C_{\text{non-param}}(\boldsymbol{u}, I_1, I_2) = \iint_{\Omega_{I_1}} D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\, d\boldsymbol{q} + R(\boldsymbol{u}). \tag{2.82}$$

Dans les paragraphes qui suivent nous explicitons quatre possibilités de régularisation ainsi que leurs propriétés. Pour clarifier les expressions, nous appellons $u_1 \in \mathbb{U}$ et $u_2 \in \mathbb{U}$ les composantes du champ de déplacement telles que :

$$\forall \boldsymbol{q} \in \Omega_{I_1} \quad : \quad \boldsymbol{u}(\boldsymbol{q}) = \begin{pmatrix} u_1(\boldsymbol{q}) \\ u_2(\boldsymbol{q}) \end{pmatrix}. \tag{2.83}$$

Lissage quadratique au premier ordre. C'est la formulation adoptée par Horn et Schunck (1981) et qui a inspiré toutes les techniques suivantes :

$$R_{\text{HS}}(\boldsymbol{u}) = \iint_{\Omega_{I_1}} \|\nabla u_1(\boldsymbol{q})\|^2 + \|\nabla u_2(\boldsymbol{q})\|^2\, d\boldsymbol{q}. \tag{2.84}$$

Elle pénalise les variations du champs de déplacements pour favoriser les champs de déplacements doux. L'emploi de la norme quadratique permet une optimisation aisée mais pénalise très fortement les discontinuités. Cette propriété peut être désirable lorsqu'il est connu *a priori* que le champs estimé est doux (par exemple déformations de surfaces ou de fluides), mais empêche sa généralisation.

Variation Totale. La variation totale (Rudin *et al.*, 1992) utilise la norme L^1 pour favoriser au contraire les champs de déplacements constants par morceaux :

$$R_{\text{TV}}(\boldsymbol{u}) = \iint_{\Omega_{I_1}} |\nabla u_1(\boldsymbol{q})| + |\nabla u_2(\boldsymbol{q})|\, d\boldsymbol{q}. \tag{2.85}$$

Si elle supporte très bien les discontinuités, la variation totale a tendance à faire apparaître des « effets d'escalier » dans les zones de variations douces.

Variation Totale Généralisée. Bredies *et al.* (2010) ont proposé une extension de la variation totale qui permet d'éviter ces effets d'escalier en régularisant les dérivées du champ de déplacements à un ordre arbitraire k. La définition est faite dans le domaine dual. Soit $u \in \mathbb{U}$:

$$R_{\mathrm{TGV}^k}(u, \alpha_0, \ldots, \alpha_{k-1}) = \sup_{\mathbf{v} \in \mathcal{C}_c^k(\Omega, \mathrm{Sym}^k(\mathbb{R}))} \iint_{\Omega_{I_1}} u(\boldsymbol{q}) \operatorname{div}^k \mathbf{v}(\boldsymbol{q}) \, d\boldsymbol{q}$$
$$\text{sous contrainte : } \quad \|\operatorname{div}^l \mathbf{v}\|_\infty \leq \alpha_l, l = 0, \ldots, k-1 \quad (2.86)$$

où $\mathcal{C}_c^k(\Omega, \mathrm{Sym}^k(\mathbb{R}))$ désigne l'espace des tenseurs symétriques d'ordre k et k fois continument dérivables, et les $\alpha_{0,\ldots,k-1}$ sont des poids positifs. La formulation primale est plus explicite, en particulier pour les ordres 1 et 2, en faisant le lien avec la variation totale. Soit $\boldsymbol{u} \in \mathbb{V}$:

$$R_{\mathrm{TGV}^1}(\boldsymbol{u}, \alpha_0) = \alpha_0 \iint_{\Omega_{I_1}} |\nabla u_1(\boldsymbol{q})| + |\nabla u_2(\boldsymbol{q})| \, d\boldsymbol{q} = \alpha_0 R_{\mathrm{TV}}(\boldsymbol{u})$$
$$R_{\mathrm{TGV}^2}(\boldsymbol{u}, \alpha_0, \alpha_1) =$$
$$\min_{\boldsymbol{w_1}, \boldsymbol{w_2} \in \mathbb{V}} \left\{ \alpha_1 \iint_{\Omega_{I_1}} |\nabla u_1(\boldsymbol{q}) - \boldsymbol{w_1}(\boldsymbol{q})| + |\nabla u_2(\boldsymbol{q}) - \boldsymbol{w_2}(\boldsymbol{q})| \, d\boldsymbol{q} \right.$$
$$\left. + \alpha_0 \left(R_{\mathrm{TV}}(\boldsymbol{w_1}) + R_{\mathrm{TV}}(\boldsymbol{w_2}) \right) \right\}. \quad (2.87)$$

La variation totale généralisée à l'ordre 2 constitue un bon compromis, privilégiant un champ de déplacements doux par morceaux sans trop augmenter la complexité.

Énergie de courbure. L'énergie de courbure :

$$R_{\mathrm{courb}}(\mathbf{a}) = \iint_{\Omega_{I_1}} \left(\frac{\partial^2 \mathbf{a}(\boldsymbol{q})}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{a}(\boldsymbol{q})}{\partial xy} \right) + \left(\frac{\partial^2 \mathbf{a}(\boldsymbol{q})}{\partial y^2} \right)^2 d\boldsymbol{q} \quad (2.88)$$

est principalement utilisée pour l'alignement de surfaces déformables. En effet, si I_2 est une surface parfaitement plane, déformée en I_1, alors l'énergie de courbure correspond à l'énergie élastique associée à cette déformation. L'emploi de la norme L^2 empêche toutefois de gérer les discontinuités qui apparaîtraient en cas de pliure.

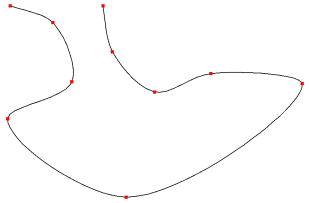


FIGURE 2.13 – Interpolation par spline cubique à 1 dimension : la courbe passe par les points de contrôle rouges, chaque section est un polynôme de degré 3 et la continuité de la dérivée est assurée entre chaque section. La FFD est une interpolation similaire en 2 dimensions.

2.7.3 Modèles paramétriques

Les méthodes paramétriques utilisent un modèle de déformation qui permet de limiter l'espace des champs de correspondances admissibles. Il est possible d'estimer les paramètres optimaux en minimisant une fonction de coût dense (équation (2.80)) ou à partir d'un nombre suffisant de correspondances éparses grâce à une optimisation aux moindres carrés par exemple.

Transformation affine. Si l'on sait que la transformation entre les images est affine :

$$\mathbf{a}^{(\text{affine})}_{\mathbf{H},t}(\boldsymbol{q}) = \mathbf{H}\boldsymbol{q} + \boldsymbol{t}, \quad \mathbf{H} \in \mathbb{R}^{2\times 2}, \boldsymbol{t} \in \mathbb{R}^2 \tag{2.89}$$

alors la mise en correspondance revient à estimer au maximum 6 paramètres, ce qui nécessite seulement 3 correspondances de points.

TPS. Le modèle de déformation TPS (*Thin Plate Spline*, Bookstein, 1989; Bartoli *et al.*, 2010) est basé sur les déplacements d'un certain nombre de points de contrôle. Le champ de déplacement résultant est l'interpolation de ces points qui minimise l'énergie de courbure (équation (2.88)). Son principal inconvénient est la complexité du calcul car chaque correspondance interpolée dépend du déplacement de tous les points de contrôle.

FFD. Le champ de correspondances associées à un modèle FFD (*Free-Form Deformation*) ne dépend que des points de contrôle voisins, ce qui simplifie grandement le calcul et les mises à jour. Nous utiliserons la formulation de Lee *et al.* (1996) à base de B-splines cubiques (voir figure 2.13).

Soit le champ discret de déplacements

$$D: \quad [\![0, n_x - 1]\!] \times [\![0, n_y - 1]\!] \to \mathbb{R}^2 \tag{2.90}$$

de $n_x \times n_y$ points de contrôle $\boldsymbol{s}_{ij}$ répartis selon une grille régulière, avec un espacement δ, le champ de correspondance interpolé est défini par :

$$FFD(\boldsymbol{q}, D) = \sum_{k=0}^{3} \sum_{l=0}^{3} B_k(v) \cdot B_l(w) \cdot (\boldsymbol{s}_{i+k,j+l} + D[i+k, j+l]) \tag{2.91}$$

où :

- $\boldsymbol{q} = (q_x, q_y)^\top$ est le point d'évaluation du champ de correspondance,
- $i = \left\lfloor \frac{q_x}{n_x} \right\rfloor$, $j = \left\lfloor \frac{q_y}{n_y} \right\rfloor$ ($\lfloor \cdot \rfloor$ dénote la partie entière),
- $v = \frac{q_x}{n_x} - \left\lfloor \frac{q_x}{n_x} \right\rfloor$, $w = \frac{q_y}{n_y} - \left\lfloor \frac{q_y}{n_y} \right\rfloor$,
- et les B_k sont les fonctions de base B-spline :

$$\begin{aligned} B_0(v) &= (1-v)^3/6 \\ B_1(v) &= (3v^3 - 6v^2 + 4)/6 \\ B_2(v) &= (-3v^3 + 3v^2 + 3v + 1)/6 \\ B_3(v) &= v^3/6. \end{aligned} \tag{2.92}$$

Cette formulation produit un champ de déplacement deux fois continument dérivable.

Autres modèles. Selon l'application visée, d'autres modèles peuvent être adoptés. Par exemple, pour l'estimation d'écoulements fluides, Dérian *et al.* (2013) réalisent une régularisation implicite grâce à une représentation sous forme d'ondelettes tronquées. Ce problème ne sera pas abordé dans le cadre de cette thèse.

Segmentation. Pour exprimer des déformations plus complexes, ces modèles peuvent être associés à une segmentation de l'image, où chaque segment possède ses propres paramètres. Par exemple Wills *et al.* (2006); Unger *et al.* (2012); Leordeanu *et al.* (2013) utilisent un modèle affine par morceaux. L'inconvénient majeur de ces approches est la sensibilité aux erreurs de segmentation.

FIGURE 2.14 – Illustration schématique des auto-occultations (en vert) et occultations externes (en jaune).

2.7.4 Occultations

Une des difficultés de la mise en correspondance dense est la présence de pixels sans correspondance. On parle alors d'occultations qui peuvent être classées en trois types :

auto-occultations : causées par un élément de la scène (présent dans les deux images) en cachant partiellement ou complètement un autre, c'est par exemple le cas lors de changements de point de vue perspective ou lorsqu'une surface déformable est repliée sur elle-même ;

bordure de l'image : le domaine de définition des images étant fini, il est possible qu'une partie de la scène visible dans une image soit hors du cadre de la deuxième, ce qui empêche la mise en correspondance ;

occultations externes : un élément extérieur apparaît dans une seule des deux images et cache une partie de la scène.

La figure 2.14 montre un exemple d'auto-occultations et occultations externes présentes dans une seule même paire d'images.

En cas de faibles déplacements, les occultations sont souvent traitées implicitement en utilisant une norme robuste, non quadratique, pour le terme de données (Wedel *et al.*, 2009; Werlberger *et al.*, 2009). Pour de plus grands déplacements, les stratégies peuvent être classées en trois catégories :

— classification explicite avec un label dédié (Tola *et al.*, 2010), approche adaptée à une optimisation discrète ;
— détection des incohérences entre flots optiques *avant* (I_0 comme image de

(a) I_2 (b) I_1 (c) I'_2 : I_2 alignée à I_1

FIGURE 2.15 – Exemple d'alignement d'images.

référence) et *arrière* (I_1 comme image de référence), ce qui nécessite le double de calcul (Alvarez *et al.*, 2002),
— détection des occultations par un critère local (Gay-Bellile *et al.*, 2010; Sun *et al.*, 2014), rapide à calculer mais surtout adapté aux auto-occultations qui ont un comportement plus prévisible.

2.7.5 Alignement d'images

En appliquant à I_2 le champ de correspondances $\mathbf{a}$ ou de déplacements $\boldsymbol{u}$ calculé, on obtient une nouvelle image :

$$\forall \boldsymbol{q} \text{ t.q. } \mathbf{a}(\boldsymbol{q}) \in \Omega_{I_2} : \quad I'_2(\boldsymbol{q}) = I_2(\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q})) = I_2(\mathbf{a}(\boldsymbol{q})) \tag{2.93}$$

qu'on appelle *alignée* à I_1 car si les champs ont été correctement estimés on a :

$$\forall \boldsymbol{q} \in \Omega_{I'_2} \quad I'_2(\boldsymbol{q}) \approx I_1(\boldsymbol{q}). \tag{2.94}$$

La figure 2.15 illustre le processus par un exemple.

2.8 Optimisation numérique

La plupart des problèmes rencontrés au cours de cette thèse peuvent se ramener à la minimisation d'une fonction de coût.

2.8.1 Méthodes linéaires

Lorsque le problème peut se formuler sous la forme d'un système d'équations linéaires, il existe des méthodes matures, robustes et rapides pour trouver la solution optimale.

Moindres carrés linéaires

Le problème le plus classique est appelé *moindres carrés linéaires*. Il consiste à résoudre une équation de la forme :

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b} \tag{2.95}$$

avec $\boldsymbol{b} \in \mathbb{R}^m$ et $\mathbf{A} \in \mathbb{R}^{m \times n}$ connus et $\boldsymbol{x} \in \mathbb{R}^n$ inconnu en minimisant :

$$C_{\mathrm{LS}}(\boldsymbol{x}, \mathbf{A}, \boldsymbol{b}) = \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2. \tag{2.96}$$

Il est possible de dériver une solution directe même si les méthodes numériques actuelles obtiennent des résultats plus stables et plus rapides en évitant l'inversion de matrice :

$$\boldsymbol{x} = \arg\min C_{\mathrm{LS}}(\boldsymbol{x}, \mathbf{A}, \boldsymbol{b}) = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \boldsymbol{b} = \mathbf{A}^\dagger \boldsymbol{b}. \tag{2.97}$$

$\mathbf{A}^\dagger$ est la pseudo-inverse de $\mathbf{A}$.

Décomposition en valeurs singulières

Une autre approche très utile pour la résolution de systèmes linéaires est la décomposition en valeurs singulières. Toute matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ admet une décomposition de la forme suivante :

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top \tag{2.98}$$

où $\mathbf{U} = \mathbf{U}^{-\top} \in \mathbb{R}^{m \times m}$, $\mathbf{S} \in \mathbb{R}^{m \times n}$ de coefficients diagonaux $\boldsymbol{s} \in \mathbb{R}^n$ et $\mathbf{V} = \mathbf{V}^{-\top} \in \mathbb{R}^{n \times n}$. Les composantes de $\boldsymbol{s}$ sont appelées *valeurs singulières*. Les colonnes de $\mathbf{U}$ et $\mathbf{V}$ sont les vecteurs singuliers respectivement à gauche et à droite. Le nombre de valeurs singulières non nulles est égal au rang de la matrice $\mathbf{A}$. La décomposition en valeurs singulières est un des algorithmes numériques les plus utilisés et bénéficie d'implémentations très efficaces.

Pseudo-inverse. La pseudo-inverse de la matrice $\mathbf{A}$ s'obtient facilement à partir de la décomposition en valeurs singulières :

$$\mathbf{A}^\dagger = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^\top. \tag{2.99}$$

Système d'équations linéaires homogène. On appelle système d'équations linéaire homogène un problème sous la forme :

$$\mathbf{A}\boldsymbol{x} = 0 \tag{2.100}$$

où $\mathbf{A}$ est connue. Résoudre le système revient à estimer le noyau de $\mathbf{A}$. La décomposition en valeur singulière permet une résolution simple. Soit r le rang de la matrice $\mathbf{A}$, la matrice $\mathbf{S}$ possède $n-r$ colonnes nulles. Les colonnes de $\mathbf{V}$ correspondant forment une base du noyau de $\mathbf{A}$.

2.8.2 Méthodes non linéaires

Lorsque la fonction est non linéaire, il est possible de résoudre le problème posé en utilisant une méthode itérative. Soit une fonction $C(\boldsymbol{x})$ à minimiser, en partant d'un jeu de paramètres $\boldsymbol{x}_n$, chaque itération produit un incrément $\Delta\boldsymbol{x}_n$ tel que $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta\boldsymbol{x}_n$ soit plus proche de la solution recherchée. Ces méthodes font toutes l'hypothèse que la fonction C est convexe. Dans le cas contraire, il est toujours possible d'utiliser ces méthodes avec une bonne initialisation mais sans garantie de convergence vers le minimum global. Voici les méthodes principalement utilisées en vision par ordinateur :

Descente de gradient

La descente de gradient est une méthode de résolution du premier ordre. La direction de déplacement choisie est directement liée au gradient de la fonction étudiée :

$$\boldsymbol{x}_{n+1}^{(\text{desc. grad.})} = \boldsymbol{x}_n + \alpha \nabla C(\boldsymbol{x}_n). \tag{2.101}$$

La longueur de pas α est généralement fixée à 1. L'avantage de cette approche est qu'elle converge efficacement même si le jeu de paramètres initial est éloigné du minimum recherché.

Newton

La méthode de Newton est une méthode du second ordre, basée sur une approximation quadratique de la fonction à minimiser. Le développement de Taylor de la fonction de coût C s'écrit :

$$C(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx C(\boldsymbol{x}) + \nabla C(\boldsymbol{x})\Delta\boldsymbol{x} + \frac{1}{2}\Delta\boldsymbol{x}^\top \mathbf{H}(\boldsymbol{x})\Delta\boldsymbol{x} \tag{2.102}$$

où la matrice hessienne $\mathbf{H}$ est définie par :

$$\mathbf{H}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial^2 C(\boldsymbol{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 C(\boldsymbol{x})}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 C(\boldsymbol{x})}{\partial x_n x_1} & \cdots & \frac{\partial^2 C(\boldsymbol{x})}{\partial x_n^2} \end{pmatrix}. \tag{2.103}$$

Un extremum est atteint si et seulement si le gradient de l'équation (2.102) par rapport à $\Delta\boldsymbol{x}$ est nul, c'est à dire :

$$\mathbf{J}(\boldsymbol{x}) + \mathbf{H}(\boldsymbol{x})\Delta\boldsymbol{x} = 0 \quad \Leftrightarrow \quad \Delta\boldsymbol{x} = -\mathbf{H}(\boldsymbol{x})^{-1}\nabla C(\boldsymbol{x}) \tag{2.104}$$

où $\mathbf{J}$ est la matrice jacobienne des dérivées partielles :

$$\mathbf{J}(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{pmatrix}. \tag{2.105}$$

L'incrément de Newton est donc :

$$\boldsymbol{x}_{n+1}^{(\text{Newton})} = \boldsymbol{x}_n - \mathbf{H}(\boldsymbol{x})^{-1}\nabla C(\boldsymbol{x}). \tag{2.106}$$

Plus sensible à la condition initiale que la descente de gradient, la méthode de Newton assure néanmoins une convergence plus efficace lorsque l'approximation quadratique est valide, ce qui est habituellement le cas lorsque les paramètres sont proches de la solution.

La variante dite de Gauss-Newton permet d'éviter le calcul coûteux de la hessienne. Elle s'applique uniquement aux problèmes de moindres carrés, c'est à dire avec un coût de la forme :

$$C(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x})^2 \tag{2.107}$$

avec m le nombre de résidus à minimiser et n le nombre de paramètres tel que $\boldsymbol{x} \in \mathbb{R}^m$. La hessienne peut alors être approximée par : $\mathbf{H} \approx \mathbf{J}^\top\mathbf{J}$. L'incrément de Gauss-Newton est donc :

$$\boldsymbol{x}_{n+1}^{(\text{Gauss-Newton})} = \boldsymbol{x}_n - \left(\mathbf{J}(\boldsymbol{x})^\top\mathbf{J}(\boldsymbol{x})\right)^{-1}\mathbf{J}^\top(\boldsymbol{x}). \tag{2.108}$$

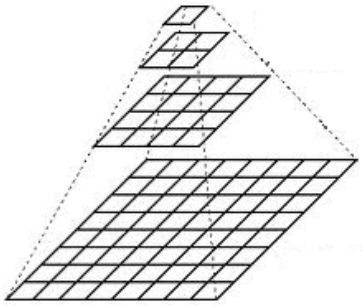


FIGURE 2.16 – Pyramide d'image pour un traitement multi-résolution. L'image de base (en bas) est successivement redimensionnée. Le traitement commence par le sommet de la pyramide et augmente progressivement la résolution pour éviter les minimums locaux.

Levenberg-Marquardt

La méthode d'optimisation non linéaire de Levenberg-Marquardt (Marquardt, 1963) combine les deux approches précédemment citées afin de profiter de leur avantage respectif. Ainsi, lorsque la solution est éloignée, c'est l'algorithme de descente de gradient qui sera privilégié. En se rapprochant de la solution, c'est la méthode de Gauss-Newton qui sera prépondérante afin d'accélérer la convergence.

Approches multi-résolution

Le principal problème des approches non linéaires est le risque de tomber dans un minimum local. L'idée des approches *multi-résolution*, ou multi-échelle, est d'effectuer d'abord l'optimisation sur une version lissée du signal puis de diminuer progressivement le lissage jusqu'au signal original. Sous l'hypothèse que le minimum global est plus large que les minimums locaux, cela permet de guider l'estimation vers le bon bassin de convergence.

En traitement d'image, on utilise des *pyramides d'images* (voir figure 2.16) construites en lissant et décimant itérativement l'image de base.

2.8.3 Variantes des moindres carrés

Un problème de minimisation aux moindres carrés minimise une fonction de la forme :

$$C(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x})^2. \tag{2.109}$$

Nous nous intéressons ici à deux problèmes qui peuvent perturber la convergence : l'échelle des données en entrée et la présence de données aberrantes.

Moindres carrés pondérés

Il peut y avoir un problème d'échelle si les f_i ont des ordres de grandeur différents, voire des unités différentes. Cela peut entraîner la domination de la fonction de coup par certains termes, les autres étant négligés. Pour palier à ce problème, on peut normaliser chaque terme f_i par son incertitude. Si on assimile les termes f_i à des variables aléatoires gaussiennes indépendantes, l'incertitude est mesurée par la déviation standard σ_i (estimée par ailleurs). La fonction de coût normalisée est donc :

$$C_{\mathrm{WLS}}(\boldsymbol{x}) = \sum_{i=1}^{m} \left(\frac{f_i(\boldsymbol{x})}{\sigma_i} \right)^2 . \tag{2.110}$$

Sa minimisation équivaut à inférer le maximum de vraisemblance.

Moindres carrés généralisés

L'idée précédente peut se généraliser à des fonctions non indépendantes en utilisant la matrice de covariance $\Sigma \in \mathbb{R}^{m \times m}$ définie par :

$$\forall (i,j) \in [\![1,m]\!]^2 \qquad \Sigma_{ij} = \mathrm{cov}\left[f_i, f_j \right] \tag{2.111}$$

qui permet d'encoder les dépendances entre les termes. La fonction de coût normalisée s'écrit dans ce cas :

$$C_{\mathrm{GLS}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})^\top \Sigma^{-1} \boldsymbol{f}(\boldsymbol{x}) = \|\boldsymbol{f}(\boldsymbol{x})\|_\Sigma^2 \tag{2.112}$$

où $\boldsymbol{f}(\boldsymbol{x}) = (f_1, \ldots, f_m)^\top$ et $\|.\|_\Sigma$ désigne la norme de Mahalanobis. Si les termes sont indépendants, alors la matrice de covariance est diagonale et les moindres carrés généralisés sont équivalents aux moindres carrés pondérés.

Optimisation robuste

Les moindres carrés sont souvent utilisés pour optimiser les paramètres d'un modèle relativement à des données en entrée. Si ces données sont imparfaites et contiennent des données non conformes au modèle (on parle de données aberrantes), ces dernières peuvent grandement perturber une minimisation aux moindres carrés : la contribution de chaque résidu étant critique, les grands résidus ont une influence

disproportionnée. Pour palier à ce problème, il existe des méthodes pour supprimer explicitement les données non conformes au modèle (Fischler et Bolles, 1981, RANSAC) mais nous nous concentrons ici sur les méthodes implicites utilisant des estimateurs robustes, ou *M-estimateurs*.

Ceux-ci prennent la forme d'une fonction ρ modulant les résidus ; la fonction de coût (2.109) modifiée est de la forme :

$$C(\boldsymbol{x}) = \sum_{i=1}^{m} \rho(f_i(\boldsymbol{x})). \tag{2.113}$$

Par exemple, avec $\rho : \boldsymbol{x} \mapsto \boldsymbol{x}^2$ on retrouve les moindres carrés classiques.

Plus que la fonction ρ, c'est sa dérivée, appelée *influence* qui est importante car les méthodes de minimisation non linéaires que nous utilisons sont variationnelles. La figure 5.7 qui compare trois estimateurs différents constitue un bon exemple.

Afin de pouvoir réutiliser les approches classiques de moindres carrés, on utilise une approche aux moindres carrés itérativement repondérés :

$$\boldsymbol{x}_k = \arg\min C(\boldsymbol{x}) = \arg\min \sum_{i=1}^{m} \omega(f_i(\boldsymbol{x}_{k-1})) \cdot f_i(\boldsymbol{x}), \tag{2.114}$$

où le poids ω, constant au sein de chaque itération est défini comme suit :

$$\omega : \quad x \mapsto \frac{1}{x} \underbrace{\frac{d\rho(x)}{dx}}_{\text{influence}}. \tag{2.115}$$

Annexe 2.A Représentations d'une rotation

Selon le contexte, plusieurs représentations équivalentes d'une rotation autour de l'origine, dans un repère euclidien, peuvent être utilisées. La plus utilisée est la représentation matricielle : une rotation est en effet une application linéaire.

2.A.1 Rotation 2D

Une rotation 2D est définie par un seul paramètre : l'angle de rotation. Pour un angle θ une rotation 2D s'exprime :

$$\begin{aligned} \mathbf{R}(\theta) : \quad & \mathbb{R}^2 \to \mathbb{R}^2 \\ & \boldsymbol{q} \mapsto \mathbf{R}(\theta)\boldsymbol{q} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \boldsymbol{q}. \end{aligned} \tag{2.116}$$

2.A.2 Rotations 3D de base

On définit trois rotations de base, autour de chacun des axes du repère euclidien. Elles sont équivalentes à des rotations 2D dans le plan orhogonal à chaque axe, c'est pourquoi on reconnait des formes similaires à l'équation (2.116) :

$$\mathbf{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \tag{2.117}$$

$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.118}$$

$$\mathbf{R}_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.119}$$

2.A.3 Angles d'Euler

Toute rotation 3D peut être représentée par une combinaison des rotations de base :

$$\mathbf{R}_{zyx}(\alpha, \beta, \gamma) = \mathbf{R}_x(\gamma), \mathbf{R}_y(\beta), \mathbf{R}_z(\alpha) \tag{2.120}$$

où α, β et γ sont les trois angles d'Euler, appelés respectivement *lacet*, *roulis*, et *tangage*. L'avantage de cette représentation est qu'elle est minimale, c'est à dire qu'on ne peut pas utiliser moins de paramètres : une rotation 3D possède trois degrés de liberté. Cette propriété est désirable pour les problèmes d'optimisation où la convergence est d'autant plus rapide que le nombre de paramètres est faible.

Cette représentation est liée à l'ordre choisi de la composition des rotations de base, et les différentes conventions ne sont pas équivalentes, ce qui peut être une source de confusion. On peut choisir par exemple :

$$\mathbf{R}_{yxz}(\alpha, \beta, \gamma) = \mathbf{R}_z(\gamma), \mathbf{R}_x(\beta), \mathbf{R}_y(\alpha). \tag{2.121}$$

Mais le principal problème de ces représentations est qu'elles ne sont pas uniques. D'abord, il s'agit d'angles donc définis modulo 2π même s'ils peuvent être restreints à certains intervalles pour réduire les ambiguïtés. Surtout, dans certaines configurations (on parle de *blocage de cardan*), deux axes des rotations élémentaires sont

alignés et suppriment effectivement un degré de liberté. Par exemple, en fixant $\beta = \frac{\pi}{2}$ dans l'équation 2.120, on obtient :

$$\begin{aligned}\mathbf{R}_{zyx}(\alpha, \frac{\pi}{2}, \gamma) &= \mathbf{R}_x(\gamma), \mathbf{R}_y(0), \mathbf{R}_z(\alpha) \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 1 \\ \sin\alpha + \gamma & \cos\alpha + \gamma & 0 \\ -\cos\alpha + \gamma & \sin\alpha + \gamma & \end{pmatrix}\end{aligned} \tag{2.122}$$

qui dépend de la somme $\alpha + \gamma$ et possède donc une infinité de couples (α, γ) équivalents.

2.A.4 Représentation angle-axe

Une rotation 3D peut être paramétrée par un couple $(\theta, \boldsymbol{u})$ où $\boldsymbol{u}$, unitaire, définit l'axe de rotation et θ l'angle. Pour une optimisation plus aisée, on manipule souvent le vecteur $\boldsymbol{x} = \theta\boldsymbol{u}$, à trois coordonnées non contraintes. Le passage entre les représentations matrice et angle-axe découle de la théorie de l'algèbre de Lie (voir Kanatani, 1990), nous ne reproduirons ici que les résultats utilisés dans ce mémoire.

Matrice de rotation vers angle-axe. Soit $\mathbf{R}$ une matrice de rotation et $(\theta, \boldsymbol{u})$ sa représentation angle-axe, on a :

$$\theta = \arccos \frac{\text{trace}(\mathbf{R}) - 1}{2} \tag{2.123}$$

$$\boldsymbol{u} = \frac{1}{2\sin\theta} \begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix}. \tag{2.124}$$

Angle-axe vers matrice de rotation. Le passage d'une représentation angle-axe $(\theta, \boldsymbol{u})$ vers une représentation matricielle se fait au moyen de la formule dite de Rodrigues :

$$\mathbf{R} = \mathbf{I}\cos\theta + [\boldsymbol{u}]_\times \sin\theta + (1 - \cos\theta)\boldsymbol{k}\boldsymbol{k}^\top. \tag{2.125}$$

2.A.5 Autres représentations

D'autres représentations sont également couramment utilisées, notamment les quaternions (Schmidt et Niemann, 2001). Ceux-ci sont composés de quatre coefficients et permettent de combiner des rotations de manière plus efficace que les produits de matrice. Cependant ils doivent vérifier la contrainte d'être unitaires ce qui rend l'optimisation plus complexe. Dans ce mémoire nous utiliserons uniquement les représentations matricielle et angle-axe.

Chapitre 3

État de l'art des solutions de réalité augmentée

Comme évoqué en introduction (chapitre 1), trois composants principaux peuvent être identifiés dans un système de réalité augmentée :

- *reconstruction* de tout ou une partie de l'environnement afin de pouvoir placer les augmentations,
- *localisation* à six degrés de libertés au sein de l'environnement reconstruit pour connaître le point de vue à adopter,
- *affichage* des augmentations alignées avec la réalité.

Dans la suite nous présentons un résumé de l'état de l'art des solutions à ces trois problèmes. Cela permettra de positionner les contributions de cette thèse pour les applications visées en identifiant des axes d'amélioration. Il s'agit de trouver un compromis entre trois objectifs aujourd'hui incompatibles :

- la *précision*, c'est à dire la minimisation du décalage spatial entre la position désirée et la position apparente de l'augmentation, c'est le critère principal pour une augmentation réaliste ;
- la *latence du système* : le décalage temporel entre l'image affichée et la réalité dû aux délais d'acquisition, de traitement et d'affichage ;
- la *robustesse* et *tolérance aux pannes* qui décrivent les risques d'échec d'un des blocs et les conséquences éventuelles, cela est plus ou moins important selon la criticité de l'application.

3.1 Reconstruction 3D

L'objectif de l'étape de reconstruction est de construire un modèle 3D de l'environnement à partir des images (appelées *vues*) issues d'une ou plusieurs caméras perspectives dont les paramètres extrinsèques et intrinsèques sont connus. Nous considérons pour commencer une scène rigide, c'est à dire qui ne change pas entre les vues (il peut s'agir d'une scène dynamique à condition que toutes les vues soient

prises au même instant).

3.1.1 Reconstruction 3D éparse

Une reconstruction éparse consiste simplement en la triangulation (voir section 2.5.2) de correspondances éparses de primitives géométriques (voir section 2.6). Les primitives éparses reconstruites sont en nombre limité mais fiables.

La gestion de plus de deux vues peut se faire indépendamment pour chaque primitive : après une triangulation initiale, il est possible d'effectuer un raffinement non linéaire minimisant l'erreur de reprojection (voir section 2.5.2) dans toutes les images où la primitive est visible.

3.1.2 Reconstruction 3D dense

Une reconstruction dense consiste également en la triangulation de correspondances, denses cette fois (voir section 2.7). Le résultat est représenté par une *carte de profondeur*. Lorsque plus de deux vues sont disponibles, les correspondances sont généralement trop ambigües pour utiliser l'erreur de reprojection. C'est donc un critère de similarité d'apparence (voir section 2.6.2) qui est minimisé sur chaque image (Graber, 2011; Newcombe *et al.*, 2011b). Les points de vue doivent par conséquent être proches.

La gestion de points de vue éloignés nécessite la génération de plusieurs cartes de profondeur : les images disponibles sont regroupées par points de vue similaires et une carte de profondeur est estimée pour chaque groupe. Ensuite, il est possible de garder les cartes de profondeur séparées sous la forme d'images-clefs constituées : d'une image, d'une carte de profondeur et des paramètres extrinsèques et intrinsèques associés. Cette représentation suffit pour une localisation dense (voir section 3.2.2) et est utilisée par Newcombe *et al.* (2011b); Meilland et Comport (2013). Ces cartes de profondeur peuvent aussi être projetées dans l'espace 3D sous forme de nuages de points, principalement pour visualisation. Une représentation plus utile mais plus difficile à estimer est la surface 3D correspondant à la scène observée. La plupart des méthodes de l'état de l'art (Newcombe *et al.*, 2011a; Steinbruecker *et al.*, 2014) utilisent une approche volumétrique où chaque profondeur de chaque carte vote pour la position de la surface au sein d'un volume défini et séparé en voxels.

3.1.3 Capteurs actifs de profondeur

L'estimation de cartes de profondeur reste complexe et peu robuste. C'est pourquoi on observe un intérêt croissant pour les capteurs dits actifs, qui estiment directement

la profondeur en émettant une information lumineuse. Ces capteurs peuvent être séparés en deux familles : les capteurs temps de vol et les capteurs à lumière structurée. Les capteurs temps de vols émettent un signal lumineux et mesurent le temps mis pour que le signal réfléchi revienne au capteur. La vitesse de la lumière étant connue, cela permet d'estimer la profondeur. Les caméras à temps de vol font l'acquisition d'une carte de profondeur en une seule fois alors que les lidars (*light radars*) font une acquisition point par point, 1D ou 2D, grâce à un laser monté sur un support mobile. Les capteurs à lumière structurée illuminent la scène avec un motif prédéfini (bandes noires et blanches, tâches aléatoires…). L'observation de la déformation de ce motif par une caméra permet d'en extraire la géométrie.

Tous ces dispositifs opèrent souvent dans les infrarouges pour être invisibles à l'œil humain. Des développements grand public récents ont permis de réduire grandement les coûts : en particulier la première version Kinect de Microsoft, vendue à plusieurs dizaines de millions d'exemplaires, utilise une approche basée lumière structurée ; la deuxième version est basée temps de vol. Quand un capteur de profondeur est adjoint à une caméra couleur de telle manière que la carte de profondeur et l'image couleur puissent être alignées on parle de caméra *RGB-D* (*RGB* pour la couleur et *Depth* pour la profondeur).

Les capteurs actifs simplifient grandement l'estimation de cartes de profondeurs mais ne sont pas adaptés à toutes les situations. Leur principale limitation est que la source de lumière est sujette à l'atténuation dûe à la distance, affectant la portée maximale, et peut être indiscernable en présence d'une source tierce puissante (soleil). Plusieurs capteurs actifs utilisés en même temps peuvent également interférer et dégrader les résultats (Butler *et al.*, 2012a). Enfin, les contraintes d'espace, de coût et de consommation d'énergie limitent souvent la résolution des cartes de profondeur et rendent l'intégration dans des systèmes embarqués parfois complexe.

3.1.4 Reconstruction 3D non rigide

Nous avons jusqu'ici considéré des reconstructions de scènes rigides. En réalité dans beaucoup de situations les scènes observées sont au moins partiellement dynamiques. Il y a deux manières d'effectuer une reconstruction non rigide.

Flot de scène

Le flot de scène est une reconstruction qui associe à chaque primitive reconstruite un déplacement 3D (assimilable à une vitesse). Dans le cadre de l'aide à la navigation cette information est cruciale car elle permet de prédire la reconstruction future de la scène et anticiper par exemple une collision. Pour calculer ce flot de scène on peut

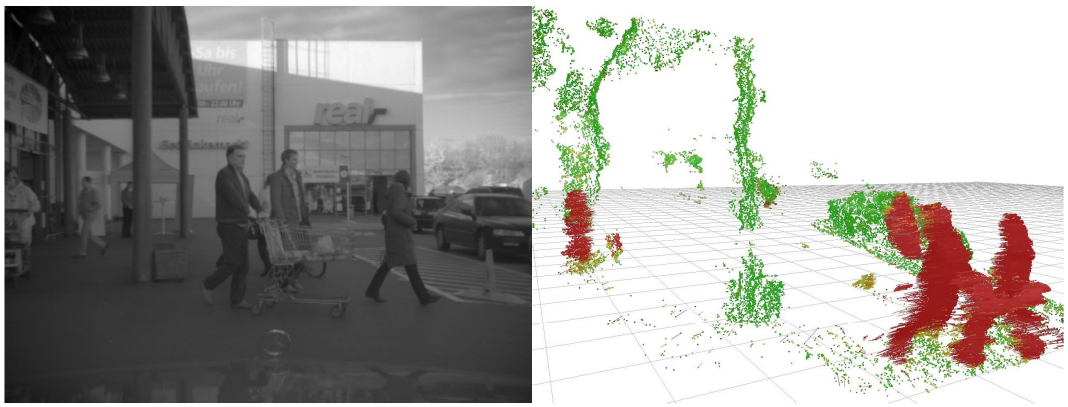


FIGURE 3.1 – Exemple de flot de scène extrait de Rabe *et al.* (2010). La couleur indique la vélocité, du minimum en vert au maximum en rouge. Les points de vue sont différents mais les piétons, mobiles, se détachent clairement et une éventuelle collision peut être évitée.

utiliser une caméra stéréo générant des cartes de profondeurs associée à un calcul de flot optique temporel pour lier les reconstructions successives (Rabe *et al.*, 2010). La figure 3.1 représente une telle reconstruction dans le cadre de l'anticipation de collision intégrée à un véhicule.

Modèle de déformation

Dans certain cas il existe un *a priori* fort sur les déformations admissibles qui permet de contraindre la reconstruction 3D. Le cas le plus développé est celui de la reconstruction 3D monoculaire de surfaces déformables. Par exemple, Bartoli *et al.* (2012) contraignent la reconstruction à partir d'une seule image et d'un modèle 3D texturé de la surface, associé à des contraintes supplémentaires, d'inextensibilité par exemple. Torresani *et al.* (2008); Garg *et al.* (2013a) utilisent des trajectoires (obtenues par le suivi de primitives 2D d'une surface quelconque) extraites d'une séquence vidéo et estiment la déformation associée la plus « simple » (contrainte sur le rang de la matrice correspondante).

3.2 Localisation

Le processus de localisation vise à obtenir la pose 3D (rotation et translation) d'une caméra perspective aux paramètres intrinsèques connus, à partir d'une image. Le tableau 3.1 compare les méthodes proposées.

Tableau 3.1 – Comparaison des méthodes de localisation. Les avantages sont colorés en vert, les inconvénients en rouge, et les propriétés qui dépendent des paramètres du problème sont indiquées en jaune.

Méthode	Précision	Latence	Robustesse
Cible	zone de l'image utilisée limitée	faible complexité	la cible doit rester visible
Modèle géométrique pur	dépend du modèle (taille et nombre de primitives)	faible complexité	dépend de l'initialisation
Modèle photo-géométrique	dépend du modèle (taille et nombre de primitives)	faible complexité	dépend de la taille du modèle
Modèle texturé (scène complète)	la localisation dense utilise toute l'image	calculs lourds	robustesse au flou mais convergence non garantie
Relocalisation	pose approximative, dépend de l'apprentissage	dépend de la taille de la base d'apprentissage	le point de vue doit être proche de l'apprentissage
3D-3D	dépend du nombre de points	itérations selon initialisation et complexité de modèle	convergence non garantie

3.2.1 Localisation par cible

Les systèmes de réalité augmentée sont souvent embarqués et disposent de capacités de calcul limitées. L'utilisation d'une localisation par cible (Kato et Billinghurst, 1999) permet de s'affranchir de l'étape de reconstruction. Du point de vue du système, l'environnement est réduit à un marqueur connu (la cible) constitué la plupart du temps d'une image plane, par exemple une carte comme dans la figure 1.1a. Il suffit alors de détecter ce marqueur dans l'image pour estimer la pose de la caméra par rapport à la cible (en calculant des homographies, voir section 2.5.3). De plus, la cible peut contenir des informations supplémentaires comme le choix de l'augmentation afficher, on parle alors de *cible codée*. Ces approches sont donc simples et efficaces mais présentent l'inconvénient de n'utiliser qu'une partie de l'image pour la localisation. Des instabilités peuvent ainsi apparaître lorsque l'utilisateur s'éloigne de la cible. Si cette dernière n'est pas visible, toute information de localisation est perdue.

3.2.2 Localisation basée modèle

Si un modèle 3D d'une partie de l'environnement est connu, il est possible d'utiliser ce dernier pour faire office de cible.

Modèles géométriques

Il existe de nombreuses techniques de localisation basées sur l'association 2D-3D de primitives de l'image (Lepetit et Fua, 2005). Les modèles utilisés pour ces associations sont séparés en deux catégories :

les modèles géométriques purs sont constitués uniquement d'un ensemble de primitives 3D : arrêtes, plans, maillage… ils nécessitent d'effectuer les associations par suivi 2D et sont donc dépendants d'une initialisation précise ;

les modèles photo-géométriques sont constitués d'un ensemble de descripteurs 2D associés à des primitives 3D, souvent des points, et ne permettent de bonnes associations que dans des conditions proches de celles de l'acquisition du modèle (en particulier luminosité et point de vue) pour préserver les descripteurs.

De plus une instabilité est possible pour les mêmes raisons que la localisation par cible.

Modèles texturés

Les modèles texturés permettent une localisation dense. Ils consistent en un maillage 3D associé à un ensemble de textures et permettent d'effectuer des rendus graphiques simulés réalistes. Ils peuvent être vus comme une généralisation des modèles photométriques et peuvent d'ailleurs servir à leur élaboration. Mais contrairement à la localisation basée modèle, la localisation dense n'extraie pas de primitives géométriques. Elle vise à trouver le point de vue permettant le rendu simulé le plus similaire à l'image observée. Les modèles utilisés couvrent souvent la scène entière ce qui permet d'utiliser toute l'information de l'image pour une précision maximale. Par contre, seule des méthodes variationnelles locales permettent cette optimisation qui est donc dépendante d'une initialisation correcte. C'est une localisation de ce type qui est utilisée par Newcombe *et al.* (2011b); Steinbrucker *et al.* (2011).

3.2.3 (Re)localisation par association d'images

La relocalisation est utilisée pour obtenir une localisation grossière mais absolue dans un environnement connu. Elle utilise pour cela une base d'images annotées avec les paramètres extrinsèques associés au point de vue utilisé. Le processus de relocalisation consiste à trouver l'image de cette base la plus similaire à l'image à localiser et de récupérer la pose correspondante. Des méthodes de recherche efficaces (Nister et Stewenius, 2006) existent pour cela, adaptées à de grandes bases de données de plusieurs dizaines de millier d'images. Il est ensuite possible de raffiner cette estimation imprécise avec une localisation basée modèle (Platonov *et al.*, 2006).

3.2.4 Localisation 3D-3D

Il est possible de calculer une localisation à partir d'informations purement 3D. Il s'agit en fait d'estimer la transformation entre deux reconstructions de la même scène, le plus souvent sous forme de nuage de points 3D. La majorité des méthodes consiste en variantes de ICP (*Iterative Closest Point*, Besl et McKay, 1992; Chen et Medioni, 1992) qui alternent deux étapes : associations 3D-3D par plus proche voisin (purement géométrique), puis estimation de la transformation minimisant les distances entre points associés. La convergence n'est pas garantie et pour des modèles 3D complexes, une initialisation proche de la solution est donc nécessaire. Newcombe *et al.* (2011a) utilisent une telle approche pour la localisation.

Tableau 3.2 – Comparaison des méthodes de localisation et reconstruction simultanées.

Méthode	Précision	Latence	Robustesse
SLAM images-clefs	dérive rapide en translation, rotation et facteur d'échelle	ajustement de faisceaux seulement aux images-clefs	dépend des primitives utilisées
SLAM contraint	pas de dérive grâce au modèle	même ordre de grandeur que le SLAM	dépend des primitives utilisées
SLAM dense	meilleure précision que SLAM mais toujours dérive	calculs très lourds	robustesse au flou

3.3 Localisation et reconstruction simultanées

La localisation en milieu inconnu est un problème très courant. Dans ce cas, les étapes de reconstruction et localisation sont inter-dépendantes : on parle alors de localisation et reconstruction simultanées (*Simultaneous Localization And Mapping*, *SLAM*). Chaque image est d'abord localisée par rapport à la partie reconstruite de l'environnement, puis elle est utilisée pour étendre et mettre à jour cette reconstruction. Le lecteur intéressé pourra se référer à Mouragnon (2007) pour plus détails.

Il ne suffit pourtant pas d'alterner ces deux étapes : les méthodes itératives sont sujettes à l'accumulation d'erreurs (*a minima* les erreurs numériques de calcul) et susceptibles de diverger : on parle de phénomène de *dérive*. Une étape d'optimisation globale ou semi globale est donc nécessaire pour contraindre et raffiner conjointement la localisation et la reconstruction. En effet les différentes caméras localisées ne sont pas indépendantes : les primitives 3D reconstruites sont par définition observées par au moins deux caméras qui sont donc liées par des contraintes géométriques. Nous évoquons ici quelques approches, comparées de manière synthètique dans le tableau 3.2.

3.3.1 Filtrage

Les premières méthodes de localisation et reconstruction simultanées étaient présentées comme un problème de filtrage non linéaire (variantes du filtre de Kalman (1960) étendu). Les publications de Davison *et al.* (2007) et Lemaire *et al.* (2007) en sont de bons exemples. Des méthodes similaires sont toujours utilisées, principalement dans le domaine robotique car elles permettent d'intégrer aisément

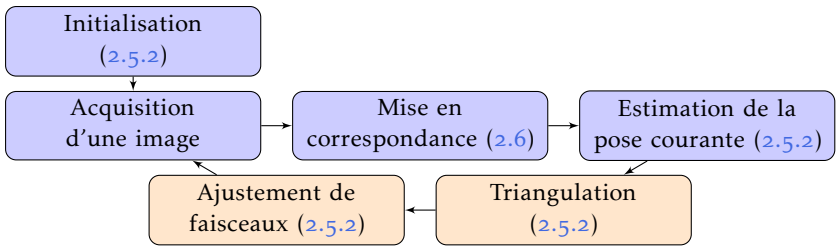


FIGURE 3.2 – Vue d'ensemble du processus SLAM épars monoculaire basé images-clefs. Les étapes colorées en orange ne sont effectuées que pour les images-clefs.

des informations provenant de capteurs différents caractérisés par des incertitudes propres. L'inconvénient de ces méthodes est qu'elles doivent maintenir à jour une matrice de covariance qui croît avec le nombre de primitives 3D reconstruites. Cela restreint l'application de ces méthodes à des environnements réduits (une pièce) en limitant le nombre de primitives 3D considérées (au détriment de la précision et de la robustesse de la localisation).

3.3.2 Images-clefs

Il a été démontré (Strasdat *et al.*, 2010) que les approches dites *basées images-clefs* demandent moins de calcul et permettent généralement d'obtenir de meilleurs résultats que le filtrage. Après la localisation et la reconstruction, un ensemble d'images appelées images-clefs est sélectionné pour un ajustement de faisceaux (section 2.5.2). Idéalement toutes les images seraient prises en compte dans l'optimisation mais cela augmenterait la complexité sans gain significatif (Mouragnon *et al.*, 2006). Différents critères (Mouragnon *et al.*, 2006; Klein et Murray, 2007) peuvent être utilisés pour la sélection des images-clefs comme les distances euclidienne et temporelle ou le nombre de correspondances 2D avec la précédente image-clef. À titre d'exemple la figure 3.2 résume le processus monoculaire de localisation et reconstruction éparses.

Pour une exécution temps réel il est nécessaire de limiter le nombre d'images-clefs optimisées à l'aide d'une fenêtre glissante temporelle et/ou spatiale (Strasdat *et al.*, 2011). Cependant, plus la fenêtre est réduite, plus la méthode est sujette au phénomène de dérive évoqué plus haut.

3.3.3 SLAM contraint

Comme on l'a vu précédemment les méthodes basées modèle géométrique, et donc associations 2D-3D ne sont pas concernées par le phénomène du dérive du SLAM épars. Par contre elles nécessitent que le modèle occupe une grande partie de

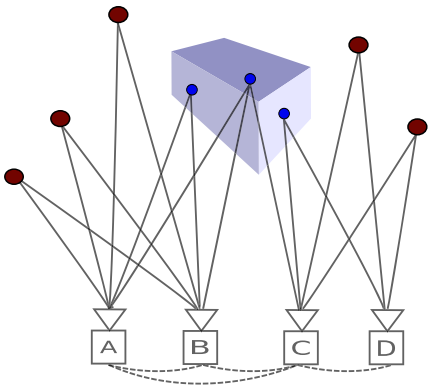


Figure 3.3 – Principe du SLAM contraint : les points rouges appartiennent à l'environnement et sont optimisés comme dans le SLAM classique basé images-clefs, les points bleus sont associés au modèle et contraignent l'optimisation. Figure reproduite avec l'aimable autorisation de Tamaazousti (2013).

l'image et ne prennent pas du tout en compte la partie inconnue de la scène observée. Cette réflexion a inspiré Tamaazousti *et al.* (2011) dans la mise au point du *SLAM contraint*. Ce dernier consiste en un SLAM monoculaire basé images-clefs dont l'ajustement de faisceaux prend en compte des contraintes 2D-3D basées modèle (voir figure 3.3). Cela permet de combiner les avantages des deux approches : la localisation est possible et continue que le modèle soit visible ou non, et lorsqu'il est visible il empêche toute sorte de dérive en constituant une référence absolue. L'inconvénient est que le modèle doit rester immobile dans l'environnement.

3.3.4 SLAM dense

Les méthodes de SLAM denses consistent elles aussi en l'alternance d'une étape de reconstruction et de localisation, mais denses. La reconstruction est effectuée avec des méthodes passives (section 3.1.2) ou actives (section 3.1.3). La localisation utilise soit l'approche photométrique décrite en section 3.2.2 soit l'approche géométrique de la section 3.2.4. Ces méthodes sont relativement récentes et la majorité des efforts est concentrée sur les approches utilisant des capteurs RGB-D (Newcombe *et al.*, 2011a; Steinbruecker *et al.*, 2014). Les approches monoculaires (Newcombe *et al.*, 2011b; Pradeep *et al.*, 2013) sont encore rares et limitées, la génération de cartes de profondeur avec une seule caméra restant difficile et peu fiable.

Les méthodes de localisation sont incrémentales : elles nécessitent d'être initialisées près de la solution et ne peuvent calculer la pose qu'entre images peu éloignées. Un des problèmes majeurs est donc la détection des fermetures de boucle, c'est à

dire le passage dans une zone déjà observée par une caméra précédente. Pour cette raison il peut être utile de détecter et décrire des primitives au cours du processus : Whelan *et al.* (2013), par exemple, utilisent des descripteurs binaires en plus de la localisation et reconstruction dense.

3.4 Affichage

Les systèmes de réalité augmentée classiques dits *video see-through* (VST) superposent des éléments virtuels à un flux vidéo et l'affichent sur un écran. Le faible coût du matériel nécessaire (une tablette numérique suffit) et la simplicité de la technique (voir figure 3.4, gauche) ont permis un essor important de cette technologie (Juniper Research, 2012). Cependant, ces approches n'affichent l'image qu'après l'avoir traitée, ce qui introduit une latence (acquisition, traitement et rendu graphique). De plus les caméras actuelles ont une résolution, un taux de rafraîchissement et un contraste très limités par rapport aux capacités visuelles humaines. Enfin et surtout, de tels systèmes peuvent couper totalement l'utilisateur de la réalité en cas de défaillances matérielles ou logicielles.

Ces problèmes, peu gênants pour les applications ludiques, sont inacceptables pour les applications critiques évoquées en début de chapitre telles que l'assistance aux opérations chirurgicales ou l'aide à la conduite, qui ne peuvent tolérer aucune indirection entre la réalité et l'utilisateur. Des systèmes utilisant un affichage semi transparent sont plus adaptés : même si l'augmentation elle-même peut présenter des défauts, la réalité observée par transparence est toujours inaltérée. On parle alors d'*optical see-through* (OST). Le principe de fonctionnement de tels systèmes est schématisé figure 3.4, montrant clairement la complexité ajoutée par rapport aux systèmes classiques *video see-through*. En particulier l'affichage dépend alors de la pose de la scène observée et de la position de l'utilisateur *par rapport à l'écran*.

La comparaison entre les deux méthodes d'affichage est résumée dans le tableau 3.3.

3.4.1 Écrans physiques et virtuels

Tous les systèmes d'affichages semi transparents peuvent être assimilés à un écran mais avec une distinction très importante. Les écrans que nous appelons *physiques* (voir figure 3.5) utilisent les technologies LCD (affichage des augmentations par absorption de la lumière) ou OLED (affichage par émission). À l'inverse, les écrans que nous appelons *virtuels* sont produits par des systèmes optiques comprenant une source (projecteur, laser) passant par un système optique de lentilles avant d'être superposée à la réalité par le biais d'une surface semi réfléchissante. Du point de

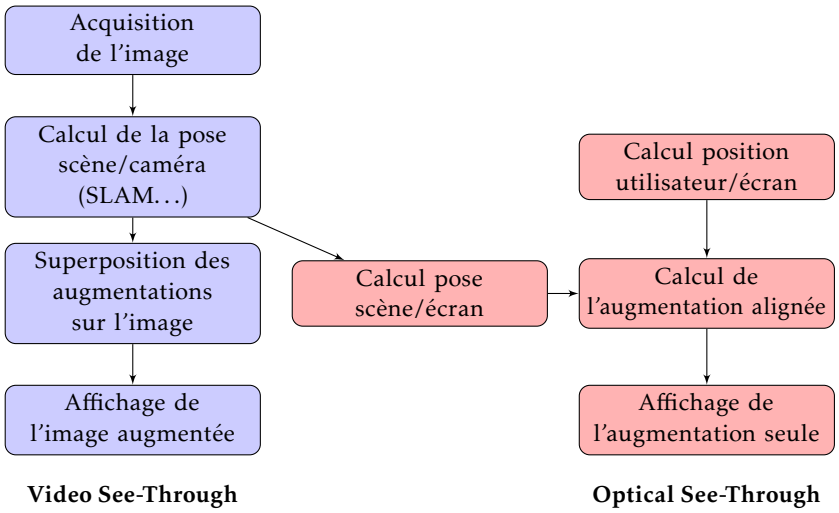
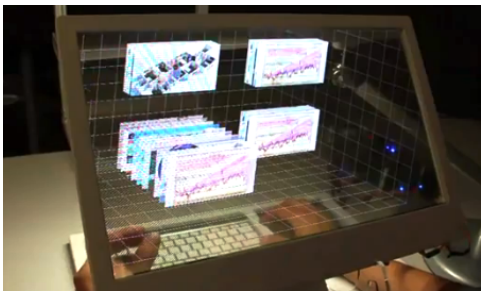


FIGURE 3.4 – Étapes principales des processus *video see-through* (en bleu) et *optical see-through* (en rouge). On observe que ce dernier est beaucoup plus complexe et nécessite notamment les poses relatives des dispositifs de localisation par rapport à l'écran (en vert). Celles-ci sont fixes pendant l'exécution.

TABLEAU 3.3 – Comparaison des méthodes d'affichage.

Méthode	Précision	Latence	Tolérance aux pannes
Video See-Through	augmentation directement sur l'image	affichage seulement après traitement	en cas de défaillance, utilisateur « aveugle »
Optical See-Through	nécessite un bon étalonnage et une faible latence	la latence affecte seulement les augmentations	seules les augmentations sont affectées en cas de défaillance

(a) LCD : concept *Smart Window* de Samsung



(b) OLED : démonstration *Behind the Screen Overlay Interactions* de Microsoft

FIGURE 3.5 – Technologies d'écrans physiques semi transparents. Les écrans LCD (a) agissent par absorption de lumière et nécessitent donc une scène lumineuse, les écrans OLED (b) agissent par émission de lumière et nécessitent donc une scène sombre.

vue de l'utilisateur ces systèmes sont presque équivalents à un écran physique, avec l'avantage de pouvoir placer l'écran virtuel à une distance arbitraire en modifiant le système optique, y compris à l'infini. On parle alors de source « collimatée », c'est à dire dont tous les rayons lumineux sont parallèles. Par contre il est compliqué de changer cette distance une fois réglée. De plus, à cause du système optique l'affichage n'est visible que depuis une zone restreinte appelée *boîte de l'œil*.

3.4.2 Nécessité de connaître la position de l'utilisateur

Dans le cas général, comme on peut le voir sur la figure 3.6, l'affichage de l'augmentation sur un écran semi transparent dépend de la position de l'œil de l'utilisateur. L'utilisateur doit donc être localisé pour pouvoir calculer à chaque instant le point de vue (ou les deux points de vue pour un affichage stéréoscopique) à adopter.

Cependant, cet effet devient négligeable si la distance entre l'écran (physique ou virtuel) et la zone de la scène à augmenter est faible par rapport à la distance entre l'utilisateur et l'écran (voir la figure 4.12 pour une démonstration mathématique). Une configuration classique est donc de placer un écran virtuel à l'infini ce qui permet de gérer des augmentations entre quelques dizaines de mètres et l'infini.

3.4.3 Lunettes augmentées et affichages tête haute

On parle de *lunettes augmentées* pour un système de réalité augmentée composé d'un ou deux affichages semi transparents, rigidement fixés aux yeux. Pour ces systèmes, l'utilisateur peut être considéré fixe par rapport à le ou les écrans, ce qui

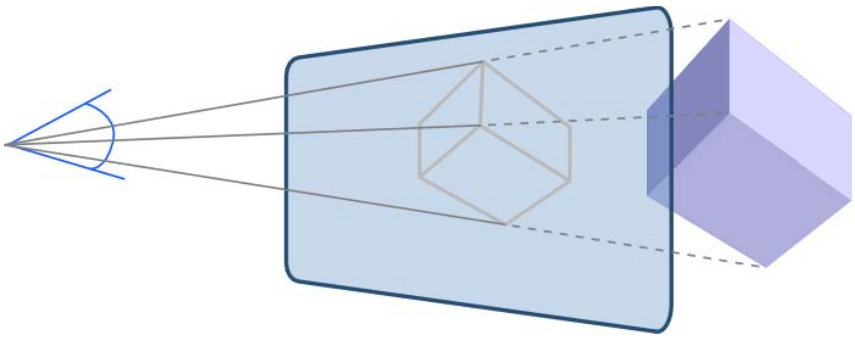


FIGURE 3.6 – Nécessité du suivi de l'utilisateur pour l'affichage sur écran semi transparent.

TABLEAU 3.4 – État actuel et à venir des technologies de lunettes augmentées. De haut en bas : champ de vue diagonal, résolution de l'affichage, poids, résolution des caméras en mégapixels, date de sortie, prix.

Google Glass	Vuzix STAR 1200XLD	Atheer One
15°	35°	65°
640 × 360	2 × 852 × 480 (stéréo)	2 × 1048 × 784 (stéréo)
50g	>100g	75g
caméra 0.9MP	caméra 2.1MP	×2 caméras 8MP
avril 2013 (pré-version)	août 2013	prévues été 2015
1500$	4999$	500$

simplifie le processus d'augmentation (voir figure 3.4). Malgré le très fort intérêt et les innovations rapides (voir tableau 3.4), les lunettes augmentées grand public restent encombrantes avec un champ de vue limité. Elles sont de plus sujettes aux rapides mouvements de la tête qui compliquent la localisation. Enfin, la résolution limitée, la faible luminosité des affichages et les problèmes de latence empêchent les augmentations de se fondre dans la réalité, et elles sont ressenties comme une gêne lors de périodes d'utilisation prolongées.

Les *affichages tête haute*, tels que présents dans les avions de chasse (voir figure 3.7) ne sont pas fixés rigidement à l'utilisateur. Ils suivent un paradigme de *fenêtre augmentée* avec une zone d'augmentation bien délimitée ce qui est plus confortable. Pour que les augmentations restent alignées, elles sont affichées à l'infini.

Figure 3.7 – Affichage tête haute d'un avion de chasse F/A-18C.

Cette approche n'est donc pas adaptée aux applications que l'on vise. De plus les technologies employées sont souvent encombrantes, lourdes et chères.

3.5 Positionnement

Les sections précédentes nous amènent à notre positionnement. Rappelons pour commencer les applications visées en début de chapitre : aide à la vente, aide à la maintenance, aide à la navigation et aide à la chirurgie. Le tableau 3.5 résume les choix envisagés et la portée de nos contributions. Le problème de localisation pour la chirurgie est très spécifique à l'application, dans un environnement contrôlé donc pas nécessairement basé vision et hors de notre sujet. Pour les autres applications, nous considérons que le SLAM contraint de Tamaazousti *et al.* (2011) (section 3.3.3) convient le mieux comme solution de localisation pour la réalité augmentée : il est précis, sans dérive, tout en restant temps réel. Une implémentation mature a d'ores et déjà donné lieu à la commercialisation d'applications pour l'aide à la vente.

Le SLAM contraint est toutefois dépendant d'un modèle géométrique. Or les modèles nécessaires ne sont pas toujours disponibles à l'achat et les processus de modélisation existants restent contraignants (nécessité de nombreuses prises de vue, fonctionnement uniquement en intérieur pour la plupart des capteurs RGB-D). L'étape de reconstruction la plus importante est la mise en correspondance. Nous proposons en chapitre 5 une nouvelle méthode de mise en correspondance qui combine les avantages des méthodes denses et éparses qui pourra servir de base à

TABLEAU 3.5 – Positionnement de nos travaux. Le tableau présente les configurations envisagées en fonction des applications. Les travaux sur les correspondances denses (chapitre 5) entre images participent à l'amélioration des cellules bleues. Ceux sur l'étalonnage de systèmes *optical see-through* (chapitre 4) contribuent aux cellules jaunes.

	Reconstruction	Localisation	Affichage
Aide à la vente	Fusion de cartes de profondeurs	SLAM contraint	VST
Aide à la maintenance	Fusion de cartes de profondeurs	SLAM contraint	OST
Aide à la navigation	Flot de scène	SLAM contraint	OST
Aide à la chirurgie	Non-rigide*	Spécifique	OST

* Les reconstructions rigides sont actuellement plus développées car plus matures et applicables aux structures solides (os...) et à certains organes selon le contexte. Il est donc envisageable d'utiliser la fusion de cartes de profondeurs proposée également pour la chirurgie.

une reconstruction dense rapide et robuste. L'application principale visée est donc la génération de modèle pour le SLAM contraint, mais la généricité de la méthode permet de l'envisager pour calculer un flot de scène dense ou reconstruire une surface déformable.

En ce qui concerne l'affichage, nous présentons en chapitre 4 un nouveau système *optical see-through* plus souple que les approches existantes, ainsi que le processus d'étalonnage associé.

Chapitre 4

Étalonnage d'un système de réalité augmentée sur écran semi transparent

Ce chapitre a fait l'objet d'une publication internationale : conférence 3DIMPVT (2012) et deux publications nationales : congrès ORASIS (2013a) et journal Traitement du Signal (2014a).

4.1 Introduction

Comme expliqué en section 3.4.3, les systèmes actuels de réalité augmentée sur affichage semi transparent ne répondent pas aux attentes d'applications critiques telles que l'aide à la maintenance ou à la chirurgie. En effet, les lunettes augmentées sont encore trop limitées, difficiles à localiser en présence de mouvements de tête rapides et désagréables pour l'utilisateur. Les affichage tête haute sont plus confortables (fenêtre d'augmentation limitée) mais peu mobiles et coûteux en raison du système optique nécessaire, et le plus souvent adaptés à des scènes lointaines.

Nous nous intéressons ici à une nouvelle catégorie de systèmes que nous appellons *tablette augmentée*. Un tel système est constitué d'un écran semi transparent sur lequel sont fixés un dispositif de suivi de l'utilisateur et un dispositif de localisation de l'écran dans la scène (figure 4.1). Cela permet de s'affranchir de la plupart des inconvénients évoqués : le système est mobile et n'est pas soumis à des mouvements aussi rapides qu'avec des lunettes. La contrepartie est une complexité accrue du système : les dispositifs de localisation opérants dans leurs repères respectifs, l'estimation de leurs poses par rapport à l'écran est nécessaire pour aligner l'affichage de l'information virtuelle avec la réalité (voir figure 4.2).

4.2 Travaux connexes

L'étalonnage extrinsèque d'un tel système consiste à estimer les poses des deux dispositifs de localisation par rapport à l'écran. À notre connaissance, les publica-

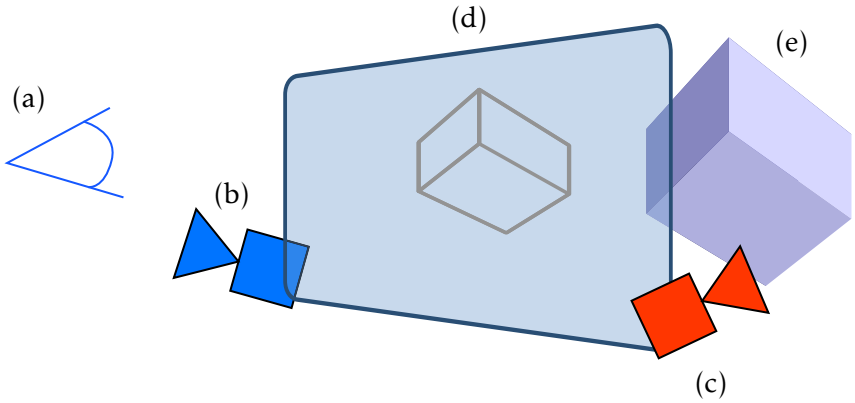


FIGURE 4.1 – Représentation schématique du système : (a) utilisateur, (b) dispositif de suivi de l'utilisateur, (c) dispositif de localisation de l'écran dans la scène, (d) écran semi transparent, (e) scène.

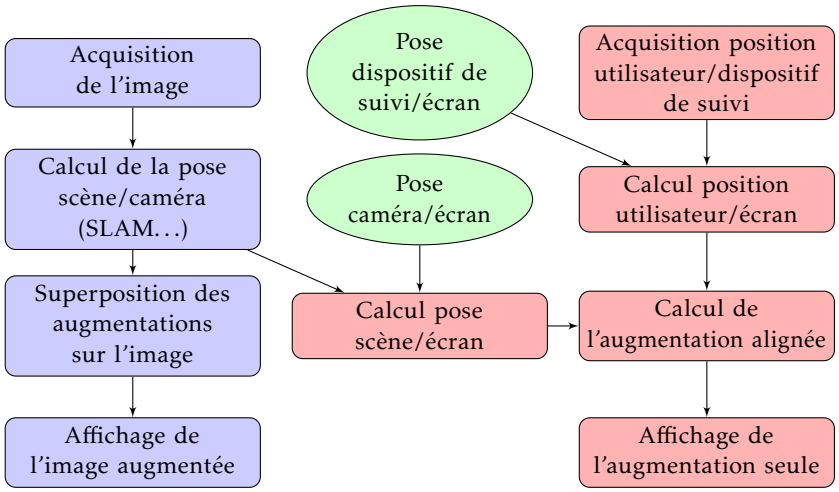


FIGURE 4.2 – Étapes principales des processus *video see-through* (en bleu) et *optical see-through* (en rouge). On observe que ce dernier est beaucoup plus complexe et nécessite notamment les poses relatives des dispositifs de localisation par rapport à l'écran (en vert). Celles-ci sont fixes pendant l'exécution.

tions liées à ce problème concernent toutes le cas où les dispositifs de localisation sont des caméras. Nous résumons ici ces méthodes, qu'elles soient directement applicables dans notre contexte ou non.

4.2.1 Étalonnage extrinsèque de caméras à champs recouvrants

Commençons par considérer le sous-problème de l'étalonnage extrinsèque d'un système à deux caméras. Si elles ont un champ recouvrant, il peut suffir d'utiliser des associations 2D-3D communes, si disponibles, et effectuer un étalonnage de chaque caméra (section 2.4.3). Les paramètres extrinsèques des caméras sont alors exprimés dans le même repère ce qui rend trivial le calcul des poses relatives (voir section 2.5.2). Des méthodes n'utilisant que des associations 2D-2D existent aussi : par exemple Svoboda *et al.* (2005) proposent un étalonnage à partir du suivi simultané dans les différentes caméras d'un point lumineux.

Cependant, dans notre contexte, nous ne voulons pas faire d'hypothèse forte sur la configuration adoptée et le cas de champs recouvrants n'a pas de sens pour le système considéré (figure 4.1).

4.2.2 Étalonnage extrinsèque de caméras à champs disjoints

L'étalonnage extrinsèque de deux caméras rigidement liées à champs disjoints a été abondamment traité dans la littérature. Il est possible d'utiliser ici aussi des associations 2D-3D et un ajustement de faisceaux, mais il faut pour cela déplacer les caméras de telle sorte qu'elles observent successivement les même primitives d'une scène rigide, par exemple en faisant une rotation à 360° (Carrera *et al.*, 2011). Cependant, ces méthodes dépendent d'associations de primitives entre caméras différentes à des temps différents, qui peuvent manquer de fiabilité à cause de variations de point de vue, d'illumination…

Les méthodes basées trajectoire (Caspi et Irani, 2002; Esquivel *et al.*, 2007; Lébraly *et al.*, 2011) calculent une localisation indépendante mais synchronisée de chaque caméra, puis exploitent la contrainte de rigidité entre les caméras, et donc entre les trajectoires, pour estimer les paramètres extrinsèques.

Rahimi *et al.* (2004); Anjum *et al.* (2007) proposent également d'étalonner un système de caméras fixes grâce à un objet connu en mouvement, suivi en 3D lors de ses passages dans le champ de chaque caméra. L'estimation des paramètres extrinsèques constitue alors un problème mal posé puisque l'objet n'est jamais visible dans les deux caméras en même temps. Une résolution est toutefois possible en appliquant un modèle de mouvement prédéfini (vitesse constante, filtre de Kalman…) à l'objet suivi. Enfin, pour Lebraly *et al.* (2010), un miroir plan est utilisé

pour que les deux caméras, bien qu'à champs disjoints, observent simultanément un objet de référence.

Toutes les méthodes précédentes permettent d'obtenir les poses relatives des deux caméras mais ne permettent aucunement d'estimer la pose de l'écran, comme l'illustre la figure 4.3.

4.2.3 Calcul de la pose d'un objet hors champ

Sturm et Bonfort (2006); Kumar *et al.* (2008); Rodrigues *et al.* (2010); Takahashi *et al.* (2012) utilisent un miroir pour calculer la pose d'un objet hors du champ de vision d'une caméra. Soit un point 3D $\boldsymbol{Q}$ de la scène, son symmétrique $\boldsymbol{Q}_\Pi$ par rapport à un miroir plan Π d'équation $(\boldsymbol{n}, d)^\top$ s'exprime :

$$\boldsymbol{Q}_\Pi = \boldsymbol{Q} + 2(d - \boldsymbol{n}^\top \boldsymbol{Q})\boldsymbol{n}, \tag{4.1}$$

ou encore, en écriture matricielle :

$$\dot{\boldsymbol{Q}}_\Pi = \underbrace{\begin{pmatrix} 1 - 2\boldsymbol{n}\boldsymbol{n}^\top & 2d\boldsymbol{n} \\ 0_3{}^\top & 1 \end{pmatrix}}_{\mathbf{S}} \dot{\boldsymbol{Q}} = \mathbf{S}\dot{\boldsymbol{Q}}. \tag{4.2}$$

Une caméra $\mathscr{C}$ de paramètres extrinsèques $\mathbf{R}$, $\boldsymbol{T}$ observant une scène à travers le miroir Π est équivalente à une caméra virtuelle $\mathscr{C}_\Pi$ (voir figure 4.4) de paramètres extrinsèques :

$$\begin{pmatrix} \mathbf{R}_\Pi & \boldsymbol{T}_\Pi \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \boldsymbol{T} \end{pmatrix} \mathbf{S} \tag{4.3}$$

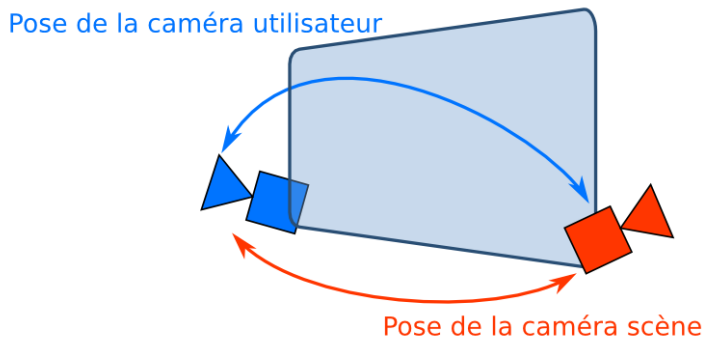
ou encore :

$$\mathbf{R}_\Pi = \mathbf{R}\left(\mathbf{I} - 2\boldsymbol{n}\boldsymbol{n}^\top\right) \qquad \boldsymbol{T}_\Pi = \boldsymbol{T} - 2d\mathbf{R}\boldsymbol{n}. \tag{4.4}$$

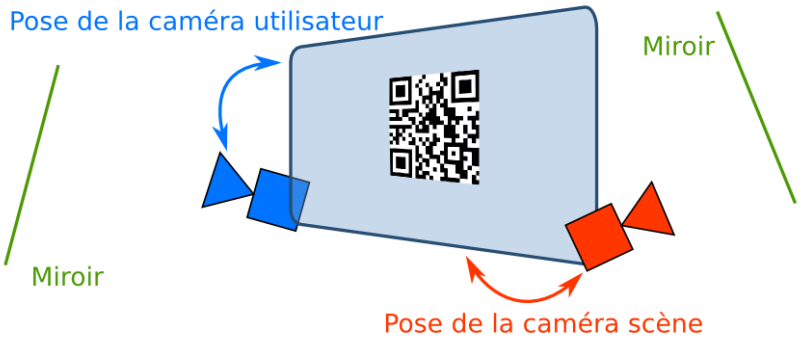
À noter que la matrice $\mathbf{R}_\Pi$ n'est pas une matrice de rotation valide car elle intègre une symétrie et son déterminant est donc négatif (voir section 2.4.1). Pour pouvoir appliquer les calculs de pose classiques, on peut par exemple prendre l'opposée de toutes les coordonnées 3D lors de l'étalonnage et estimer alors :

$$\mathbf{R}'_\Pi = -\mathbf{R}_\Pi = \mathbf{R}\left(2\boldsymbol{n}\boldsymbol{n}^\top - \mathbf{I}\right) \qquad \boldsymbol{T}'_\Pi = -\boldsymbol{T}_\Pi = 2d\mathbf{R}\boldsymbol{n} - \boldsymbol{T} \tag{4.5}$$

qui constituent des paramètres extrinsèques valides. Si la pose du miroir est connue, l'étalonnage de la caméra virtuelle permet d'estimer directement les paramètres extrinsèques de la caméra réelle. Dans le cas contraire il faut répéter l'opération



(a) Étalonnage extrinsèque de caméras à champs disjoints : la pose de l'écran est manquante.



(b) Méthodes utilisant un miroir : deux étalonnages caméra/écran indépendants sont nécessaires.

FIGURE 4.3 – Illustration des méthodes d'étalonnage de l'état de l'art et leurs limites.

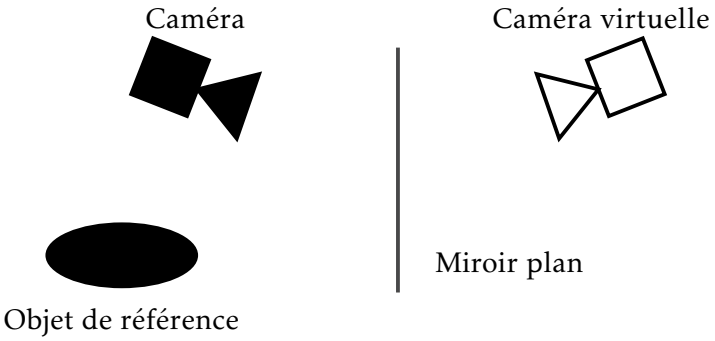


FIGURE 4.4 – Estimation de la pose d'une caméra par rapport à un objet hors champ grâce à un miroir. L'ensemble caméra/miroir est équivalent (à une symmétrie près) à une caméra virtuelle dont le champ de vision contient l'objet de référence.

avec différentes poses du miroir. Soit n le nombre de caméras virtuelles étalonnées, il y a $6 + 3n$ inconnues : les six paramètres extrinsèques de la caméra réelle et trois paramètres pour chaque plan défini par une position du miroir. Pour chaque caméra virtuelle, l'équation (4.5) fournit six équations linéaires indépendantes. Il faut donc un minimum de trois caméra virtuelles pour résoudre le système (15 inconnues pour 18 équations).

L'inconvénient majeur de ces méthodes est le fait que le miroir introduit une perte de précision due aux phénomènes de réfraction (Lebraly *et al.*, 2010) et au fait qu'il est impossible de garantir qu'un miroir soit parfaitement plat. Néanmoins, en appliquant une telle méthode pour estimer la pose de chaque caméra par rapport à l'écran, il est théoriquement possible d'étalonner complètement notre système. Cependant les deux caméras seraient alors étalonnées totalement indépendamment rendant le processus plus lourd à mettre en œuvre et moins précis. De plus ces méthodes minimisent une erreur de reprojection dans le plan image des caméras, alors que pour l'utilisateur l'important est de réduire l'*erreur d'alignement* (définie par la suite).

4.2.4 Étalonnage caméra / écran et erreur d'alignement

Le critère principal d'évaluation d'un système de réalité augmentée est le réalisme des augmentations, c'est à dire à quel point elles apparaissent telles qu'elles le seraient physiquement. L'erreur d'alignement permet d'optimiser l'aspect géométrique de ce critère : elle consiste en la distance dans l'écran (représentée sur la figure 4.5) entre l'affichage 2D et la projection optimale de l'augmentation 3D. Il s'agit d'ailleurs de l'objectif de la plupart des méthodes d'étalonnage des systèmes de

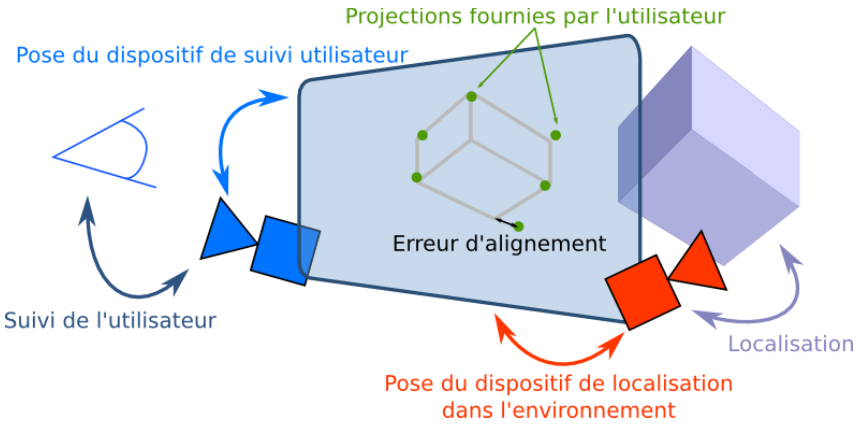


FIGURE 4.5 – Représentation du problème d'étalonnage : les poses des deux dispositifs de localisation par rapport à l'écran sont les inconnues, l'erreur d'alignement est l'énergie à minimiser et les autres informations sont les données en entrées. A visualiser de préférence en couleur.

type lunettes augmentées, en alignant un motif affiché avec une mire de calibration (Tang, 2003).

4.3 Formulation du problème et positionnement

Nous proposons dans ce chapitre une nouvelle méthode qui minimise directement l'erreur d'alignement à partir des observations 2D indiquées par l'utilisateur (figure 4.5), dans le contexte du scénario d'étalonnage suivant :

— un objet connu est placé derrière l'écran, les positions de plusieurs points-clefs de l'objet sont connues dans le repère du dispositif de localisation dans la scène ;
— l'utilisateur indique la projection apparente de ces points sur l'écran pendant que sa position est suivie dans le repère du deuxième dispositif.

Ce processus est pour le moment effectué en faisant cliquer l'utilisateur sur les projections dans un ordre prédéfini et depuis différents points de vue.

En considérant le système dans son ensemble, l'approche est indépendante des capteurs et algorithmes utilisés pour le suivi. Il est par exemple possible d'utiliser une caméra stéréoscopique [1] ou un système électromagnétique pour le suivi de l'utilisateur. De plus, à la différence des méthodes précédentes considérant chaque

1. La solution retenue pour le prototype est une caméra stéréoscopique bas coût associée à un algorithme de détection de pupille, voir section 4.6.1.

caméra indépendamment, les erreurs des capteurs ou de modélisation peuvent être en partie compensées.

4.3.1 Notations et conventions

Le système représenté dans la figure 4.1 est composé de trois composants fixés rigidement : l'écran semi transparent, le dispositif de suivi utilisateur $\mathscr{C}_u$ et et le dispositif de localisation dans la scène $\mathscr{C}_o$. L'écran transparent est utilisé comme référence et définit le repère monde $\mathscr{W}$ où les axes sont illustrés en figure 4.6 et où l'origine est un point arbitraire de l'écran. La pose de $\mathscr{C}_u$ dans $\mathscr{W}$ s'exprime par la transformation

$$\mathbf{M}_{\mathscr{C}_u \to \mathscr{W}} = \begin{pmatrix} \mathbf{R}_{\mathscr{C}_u \to \mathscr{W}} & \boldsymbol{T}_{\mathscr{C}_u \to \mathscr{W}} \end{pmatrix} \tag{4.6}$$

où $\mathbf{R}_{\mathscr{C}_u \to \mathscr{W}} \in \mathbb{R}^{3\times3}$ est la matrice de rotation et $\boldsymbol{T}_{\mathscr{C}_u \to \mathscr{W}} \in \mathbb{R}^3$ le vecteur translation. De même, $\mathbf{M}_{\mathscr{C}_o \to \mathscr{W}}$, $\mathbf{R}_{\mathscr{C}_o \to \mathscr{W}}$, et $\boldsymbol{T}_{\mathscr{C}_o \to \mathscr{W}}$ décrivent la pose de $\mathscr{C}_o$ dans $\mathscr{W}$.

Soient n points 3D de référence choisis dans la scène dont les coordonnées $\boldsymbol{O}^{(\mathscr{C}_o)}_{j=1\ldots n}$ dans $\mathscr{C}_o$ sont supposées connues. Depuis m positions distinctes, notées $\boldsymbol{U}^{(\mathscr{C}_u)}_{i=1\ldots m}$ dans $\mathscr{C}_u$, l'utilisateur indique la position des observations 2D $\boldsymbol{c}_{ij}$ de ces points de référence dans l'écran. Pour une position utilisateur $\boldsymbol{U}^{(\mathscr{C}_u)}_i$ et un point de référence $\boldsymbol{O}^{(\mathscr{C}_o)}_j$, $\boldsymbol{c}_{ij}$ est l'intersection du rayon $\left(\boldsymbol{U}^{(\mathscr{W})}_i, \boldsymbol{O}^{(\mathscr{W})}_j\right)$ avec le plan $z = 0$ (l'écran) dans $\mathscr{W}$:

$$\begin{aligned} \boldsymbol{c}_{ij} &= f_{\text{int}}(\boldsymbol{U}^{(\mathscr{W})}_i, \boldsymbol{O}^{(\mathscr{W})}_j) \\ &= f_{\text{int}}(\mathbf{M}_{\mathscr{C}_u \to \mathscr{W}} \boldsymbol{U}^{(\mathscr{C}_u)}_i, \mathbf{M}_{\mathscr{C}_o \to \mathscr{W}} \boldsymbol{O}^{(\mathscr{C}_o)}_j) \end{aligned} \tag{4.7}$$

où f_{int} est la fonction d'intersection définie pour $z_1 > 0$ et $z_2 < 0$ par :

$$f_{\text{int}}\left((x_1, y_1, z_1)^\top, (x_2, y_2, z_2)^\top\right) = \begin{pmatrix} x_1 - z_1 \times \frac{x_2 - x_1}{z_2 - z_1} \\ y_1 - z_1 \times \frac{y_2 - y_1}{z_2 - z_1} \end{pmatrix}. \tag{4.8}$$

4.3.2 Modèle de bruit

En pratique, l'estimation des positions de l'utilisateur et des points 3D de référence est sujette à du bruit. Les observations 2D fournies par l'utilisateur sont également imprécises. En effet, quelle que soit la méthode employée, ces observations restent soumises à des facteurs technologiques (flou introduit par l'écran, précision du dispositif d'acquisition…) et humains (défauts de vision, tremblements…). Ces bruits sont modélisés par des variables gaussiennes. Les versions

bruitées des variables d'entrée du système sont notées surmontées d'un tilde. Soit $\mathcal{N}(\mu, \Sigma)$ une variable aléatoire suivant la loi normale de moyenne μ et de matrice de covariance Σ, on a :

$$\widetilde{\boldsymbol{U}}_i^{(\mathscr{C}_u)} = \boldsymbol{U}_i^{(\mathscr{C}_u)} + \mathcal{N}(0_3, \Sigma_{\boldsymbol{U}}) \tag{4.9}$$

$$\widetilde{\boldsymbol{O}}_j^{(\mathscr{C}_o)} = \boldsymbol{O}_j^{(\mathscr{C}_o)} + \mathcal{N}(0_3, \Sigma_{\boldsymbol{O}}) \tag{4.10}$$

$$\widetilde{\boldsymbol{c}}_{ij} = \boldsymbol{c}_{ij} + \mathcal{N}(0_2, \Sigma_{\boldsymbol{c}}) \tag{4.11}$$

pour tout i entre 1 et m et j entre 1 et n. 0_2 et 0_3 sont les vecteurs nuls de dimensions respectives 2 et 3. Les bruits varient peu pour une configuration donnée du système et il est possible d'estimer leurs covariances (voir section 4.6.1 pour un exemple), et ainsi d'améliorer la précision de l'étalonnage. Il est nécessaire de distinguer les paramètres estimés de leurs véritables valeurs. Les estimations sont notées avec un chapeau : $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$, $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$, $\widehat{\boldsymbol{U}}_{i=1\ldots m}^{(\mathscr{C}_u)}$, $\widehat{\boldsymbol{O}}_{j=1\ldots n}^{(\mathscr{C}_o)}$.

4.3.3 Problème d'étalonnage

L'objectif de la méthode proposée est de trouver les meilleures estimations $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ et $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$ sachant $\widetilde{\boldsymbol{U}}_{i=1\ldots m}^{(\mathscr{C}_u)}$, $\widetilde{\boldsymbol{O}}_{j=1\ldots n}^{(\mathscr{C}_o)}$ et les $\widetilde{\boldsymbol{c}}_{ij}$ correspondants, grâce à l'équation (4.7). En considérant des bruits gaussiens, la solution optimale minimise la fonction de coût suivante :

$$\begin{aligned} C(\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}, \widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}, \widehat{\boldsymbol{U}}_{i=1\ldots m}^{(\mathscr{C}_u)}, \widehat{\boldsymbol{O}}_{j=1\ldots n}^{(\mathscr{C}_o)}) = \\ \sum_{\substack{i=1\ldots m \\ j=1\ldots n}} \sum \left\| \widetilde{\boldsymbol{c}}_{ij} - f_{\text{int}}(\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}} \widehat{\boldsymbol{U}}_i^{(\mathscr{C}_u)}, \widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}} \widehat{\boldsymbol{O}}_j^{(\mathscr{C}_o)}) \right\|_{\Sigma_{\boldsymbol{c}}}^2 \\ + \sum_{i=1}^{m} \left\| \widetilde{\boldsymbol{U}}_i^{(\mathscr{C}_u)} - \widehat{\boldsymbol{U}}_i^{(\mathscr{C}_u)} \right\|_{\Sigma_{\boldsymbol{U}}}^2 \\ + \sum_{j=1}^{n} \left\| \widetilde{\boldsymbol{O}}_j^{(\mathscr{C}_o)} - \widehat{\boldsymbol{O}}_j^{(\mathscr{C}_o)} \right\|_{\Sigma_{\boldsymbol{O}}}^2 . \end{aligned} \tag{4.12}$$

Il s'agit d'un problème d'optimisation non linéaire aux moindres carrés généralisés (voir section 2.8.3). On reconnaît dans le premier terme l'erreur d'alignement 2D, non convexe, alors que les autres termes sont les erreurs 3D sur les positions des points de référence et de l'utilisateur. La section suivante présente une méthode d'initialisation convexe.

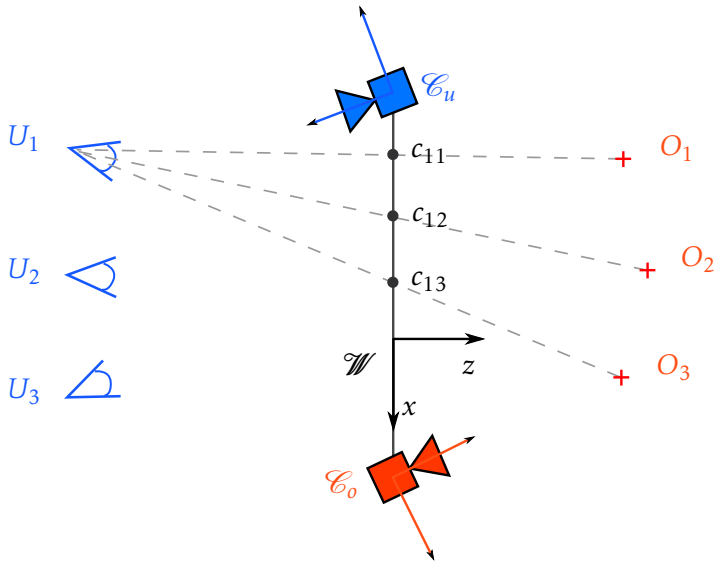


FIGURE 4.6 – Notations (les éléments de même couleur sont exprimés dans le même repère).

4.4 Étalonnage et caméras virtuelles

4.4.1 Définition des caméras virtuelles

L'erreur de reprojection de l'équation (4.12) est semblable à l'erreur de reprojection utilisée dans l'ajustement de faisceaux pour l'étalonnage de caméras (voir section 2.5.2). C'est pourquoi nous proposons l'introduction de caméras virtuelles permettant de décomposer le problème en plusieurs sous-problèmes classiques d'étalonnage. Soient m caméras virtuelles $\mathcal{V}_{i=1\ldots m}$ centrées sur les positions utilisateurs et partageant toutes le plan de l'écran ($z = 0$) comme plan focal, leurs points principaux étant confondus avec l'origine de $\mathscr{W}$ (voir figure 4.7). Cette définition donne à ces caméras des propriétés intéressantes : tous leurs axes optiques sont parallèles et pointent dans la direction de l'axe z dans $\mathscr{W}$. Les poses des caméras virtuelles dans $\mathscr{W}$ sont donc définies par :

$$\forall i \in [\![1, m]\!] \quad \mathbf{M}_i = \begin{pmatrix} \mathbf{I} & \boldsymbol{U}_i^{(\mathscr{W})} \end{pmatrix} \tag{4.13}$$

où $\mathbf{I}$ est la matrice identité 3×3 et $\boldsymbol{U}_i^{(\mathscr{W})} = \mathbf{M}_{\mathscr{C}_u \to \mathscr{W}} \boldsymbol{U}_i^{(\mathscr{C}_u)}$ est la position $\boldsymbol{U}_i^{(\mathscr{C}_u)}$ exprimée dans $\mathscr{W}$. Comme les observations $\boldsymbol{c}_{ij}$ sont exprimées en unités de $\mathscr{W}$, les « pixels virtuels » des caméras virtuelles sont parfaitement carrés : les longueurs focales en x et y sont égales ($f_{x_i} = f_{y_i} = f_i$), l'obliquité est nulle ($s_i = 0$) et il n'y a aucune distortion. De plus, comme les caméras sont définies par leur plan focal et point

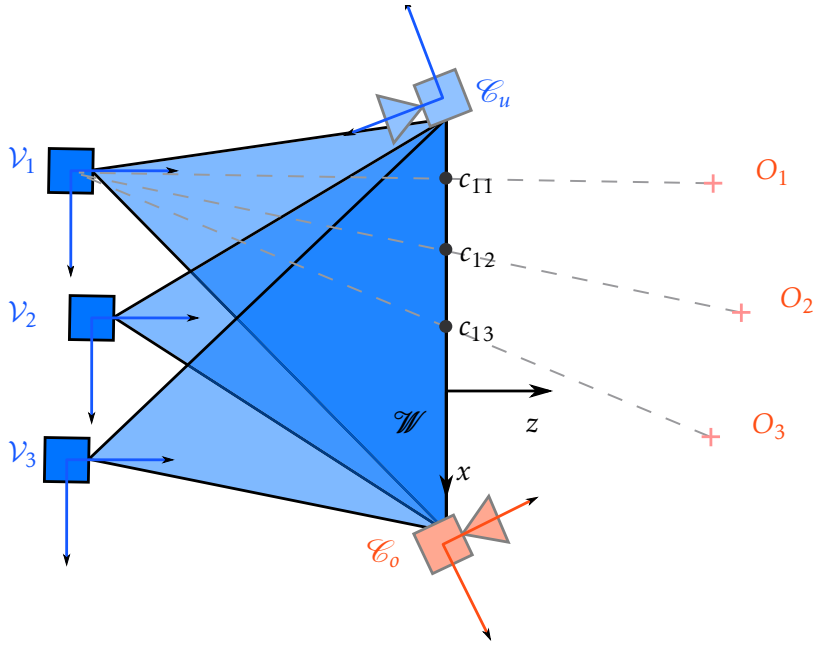


FIGURE 4.7 – Caméras virtuelles utilisateur : les centres optiques correspondent aux positions de l'utilisateur et les plans focaux sont confondus avec l'écran (voir texte, section 4.4.1).

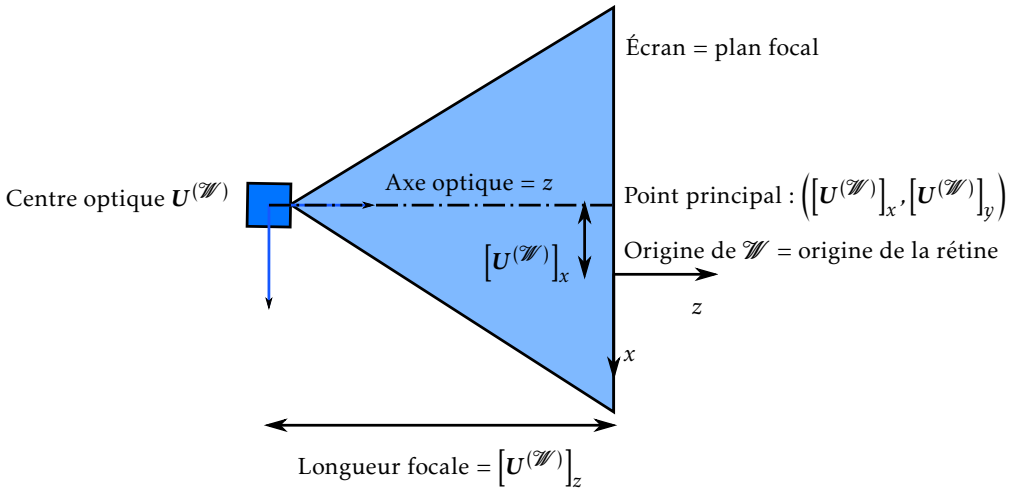


FIGURE 4.8 – Lien entre les paramètres extrinsèques et intrinsèques des caméras virtuelles.

principal dans $\mathscr{W}$, leurs paramètres intrinsèques (point principal (u_i, v_i) et focale f_i) sont liés à la position de leur centre optique :

$$\boldsymbol{U}_i^{(\mathscr{W})} = \left(\left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_x, \left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_y, \left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_z\right)^T. \tag{4.14}$$

La matrice des paramètres intrinsèques s'écrit alors de la manière suivante (voir figure 4.8) :

$$\begin{aligned}\mathbf{K}_i &= \begin{pmatrix} f_{xi} & s_i & u_i \\ 0 & f_{y_i} & v_i \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} f_i & 0 & u_i \\ 0 & f_i & v_i \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} -\left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_z & 0 & \left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_x \\ 0 & -\left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_z & \left[\boldsymbol{U}_i^{(\mathscr{W})}\right]_y \\ 0 & 0 & 1 \end{pmatrix}.\end{aligned} \tag{4.15}$$

Les observations 2D $\boldsymbol{c}_{ij}$ sont les projections des points 3D de référence dans $\mathcal{V}_i$, donc :

$$\boldsymbol{c}_{ij} = \pi\left(\mathbf{K}_i \mathbf{M}_i^{-1} \mathbf{M}_{\mathscr{C}_o \to \mathscr{W}} \dot{\boldsymbol{O}}_j^{(\mathscr{C}_o)}\right) \tag{4.16}$$

$$\boldsymbol{c}_{ij} = \pi\left(\mathbf{H}_i \dot{\boldsymbol{O}}_j^{(\mathscr{C}_o)}\right) \tag{4.17}$$

où

$$\mathbf{H}_i = \mathbf{K}_i \mathbf{M}'_i \tag{4.18}$$

$$\mathbf{M}'_i = \mathbf{M}_i^{-1} \mathbf{M}_{\mathscr{C}_o \to \mathscr{W}} = \left[\mathbf{R}_{\mathscr{C}_o \to \mathscr{W}} | \boldsymbol{T}_{\mathscr{C}_o \to \mathscr{W}} - \boldsymbol{U}_i^{(\mathscr{W})} \right]. \tag{4.19}$$

4.4.2 Étalonnage des caméras virtuelles

Nous verrons que le calcul des paramètres extrinsèques et intrinsèques d'une caméra virtuelle $\mathcal{V}_i$ permet d'estimer la pose $\mathbf{M}_{\mathscr{C}_o \to \mathscr{W}}$ dans le repère monde (de l'écran) du dispositif de localisation $\mathscr{C}_o$ dans la scène. La première étape consiste à estimer la matrice $\mathbf{H}_i$ à partir des associations 3D-2D. Nous considérons dans ce chapitre l'utilisation d'un objet de référence 3D qui permet d'exploiter la *Direct Linear Transform* (DLT, voir 2.4.3). L'étape suivante est d'extraire les paramètres intrinsèques $\mathbf{K}_i$ et extrinsèques $\mathbf{M}'_i$ vérifiant l'équation (4.18) : $\mathbf{H}_i = \mathbf{K}_i \mathbf{M}'_i$ grâce à une décomposition telle que présentée dans la section 2.4.3.

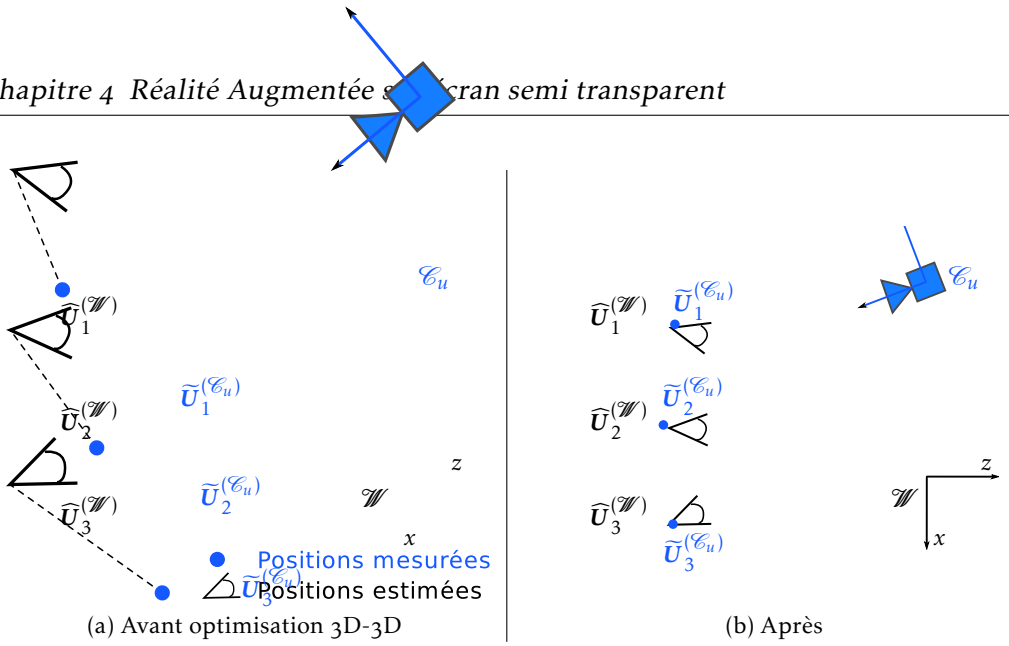
4.4.3 Extraction de la pose du dispositif $\mathscr{C}_o$

Il est maintenant possible d'extraire $\widehat{\boldsymbol{U}}_i^{(\mathscr{W})}$ à partir de $\widehat{\mathbf{K}}_i$ grâce à l'équation (4.15). Nous construisons ensuite $\widehat{\mathbf{M}}_i$ à partir de $\widehat{\boldsymbol{U}}_i^{(\mathscr{W})}$ en utilisant l'équation (4.13). Enfin $\widehat{\mathbf{M}}^{(i)}_{\mathscr{C}_o \to \mathscr{W}}$ est extrait de $\widehat{\mathbf{M}}'_i$ avec l'équation (4.19). Notons que ces estimations sont effectuées indépendamment pour chaque caméra virtuelle, d'où la présence de l'indice (i).

Toutes ces estimations $\widehat{\mathbf{M}}^{(i)}_{\mathscr{C}_o \to \mathscr{W}}$ doivent être agrégées en un seul $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$ pour initialiser l'ajustement de faisceaux. Dans un premier temps les échecs manifestes sont éliminés : caméra virtuelles pour lesquelles le raffinement non linéaire n'a pas convergé ou dont les matrices de paramètres intrinsèques sont invalides. Ensuite, nous avons choisi de sélectionner l'estimation dont l'erreur de reprojection après raffinement est la plus faible. C'est l'heuristique qui donne les meilleurs résultats selon notre expérience. Le lecteur intéressé pourra se référer à la section 4.4.6 pour une discussion à propos d'une solution optimale.

4.4.4 Extraction de la pose du dispositif $\mathscr{C}_u$

Une fois calculé $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$, nous proposons deux manières d'obtenir une estimation $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ de la pose du dispositif $\mathscr{C}_u$.



(a) Avant optimisation 3D-3D (b) Après

FIGURE 4.9 – Étalonnage de $\mathscr{C}_u$ par alignement 3D-3D.

Alignement 3D-3D

Les caméras virtuelles sont centrées sur les positions de l'utilisateur et l'étalonnage de chaque caméra virtuelle $\mathcal{V}_i$ donne une estimation $\widehat{\boldsymbol{U}}_i^{(\mathscr{W})}$ de $\boldsymbol{U}_i^{(\mathscr{W})} = \mathbf{M}_{\mathscr{C}_u \to \mathscr{W}} \boldsymbol{U}_i^{(\mathscr{C}_u)}$. Le dispositif de localisation fournit une mesure bruitée $\widetilde{\boldsymbol{U}}_i^{(\mathscr{C}_u)}$. Il est ensuite possible d'estimer $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ en résolvant un problème d'alignement 3D-3D (Horn, 1987) comme illustré figure 4.9.

Caméras virtuelles centrées sur les points de référence

Il est aussi possible de répéter la même approche d'étalonnage avec des caméras virtuelles centrées sur les points 3D de référence de la scène. Le problème est symétrique avec quelques changements. Comme leurs axes optiques sont dans la direction $-z$, les poses des caméras virtuelles sont définies par $\mathbf{M}_j^{O\,(\mathscr{C}_o)} = [\mathbf{R}^O | O_j^{(\mathscr{W})}]$ où $\mathbf{R}^O = \mathrm{diag}(-1, 1, -1)$ et $\boldsymbol{O}_j^{(\mathscr{W})} = \mathbf{M}_{\mathscr{C}_o \to \mathscr{W}} \boldsymbol{O}_j^{(\mathscr{C}_o)}$. L'équation (4.15) devient :

$$\mathbf{K}_j^O = \begin{pmatrix} \left[\boldsymbol{O}_j^{(\mathscr{W})}\right]_z & 0 & -\left[\boldsymbol{O}_j^{(\mathscr{W})}\right]_x \\ 0 & \left[\boldsymbol{O}_j^{(\mathscr{W})}\right]_z & \left[\boldsymbol{O}_j^{(\mathscr{W})}\right]_y \\ 0 & 0 & 1 \end{pmatrix} \tag{4.20}$$

et les projections des points $\boldsymbol{U}^{(\mathscr{C}_u)}_{i=1\ldots m}$ dans la caméra virtuelle j sont

$$\boldsymbol{c}^{O}_{ij} = (-(\boldsymbol{c}_{ij})_x, (\boldsymbol{c}_{ij})_y)^T. \tag{4.21}$$

Le reste du processus de résolution est identique et permet d'obtenir une estimation $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ de la pose de $\mathscr{C}_u$.

Choix de stratégie

Nous proposons donc trois stratégies :

Symétrique : estimer $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$ avec des caméras virtuelles centrées sur l'utilisateur et $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ avec des caméras virtuelles centrées sur les points 3D de référence de la scène.

Caméras centrées sur l'utilisateur seulement : estimer $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$ avec des caméras virtuelles centrées sur l'utilisateur et $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ par alignement 3D-3D.

Caméras centrées sur les points de référence seulement : estimer $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ avec des caméras virtuelles centrées sur les points 3D de référence de la scène et $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$ par alignement 3D-3D.

L'étape la plus délicate de l'étalonnage est la DLT. Elle est sensible au bruit sur les observations et non sur la position du centre des caméras virtuelles. Il est donc préférable de choisir comme centres les données les plus bruitées (positions utilisateurs ou points 3D de référence de la scène). En pratique, il est plus difficile de suivre l'utilisateur que de localiser un objet connu. Le bon conditionnement de la DLT est également lié à la contrainte de non-planarité des points d'étalonnage (voir la section 4.4.6), difficilement applicable pour les positions utilisateurs limitées par le champ de vision à travers l'écran. Ces deux arguments montrent que centrer les caméras virtuelles sur l'utilisateur seulement est *a priori* la meilleure stratégie. Cependant pour obtenir la meilleure estimation possible indépendamment de la configuration du problème, l'erreur de reprojection est estimée avec chaque approche et le meilleur résultat est conservé.

4.4.5 Ajustement de faisceaux

Les estimations $\widehat{\mathbf{M}}_{\mathscr{C}_u \to \mathscr{W}}$ et $\widehat{\mathbf{M}}_{\mathscr{C}_o \to \mathscr{W}}$ et les mesures bruitées $\widetilde{\boldsymbol{U}}^{(\mathscr{C}_u)}_{i=1\ldots m}$ et $\widetilde{\boldsymbol{O}}^{(\mathscr{C}_o)}_{j=1\ldots n}$ permettent d'initialiser l'ajustement de faisceaux global avec covariances qui minimise la fonction (4.12) en utilisant l'algorithme de Levenberg-Marquardt (voir section 2.8.2). Si l'algorithme converge (initialisation proche de la solution) et que les covariances des bruits sont correctement estimées, la solution obtenue est optimale du point de vue de l'utilisateur : l'erreur d'alignement est minimisée.

4.4.6 Discussions

Contraintes et cas dégénérés

Les contraintes géométriques viennent de l'étape DLT (section 4.4.2). Nous avons déjà établi que le nombre de points 3D de référence devait respecter la contrainte $m \geq 6$ pour obtenir une solution unique. De plus, certaines configurations des points 3D de référence peuvent amener à des cas dégénérés, traités par Hartley et Zisserman (2004). Le cas le plus notable est le cas où les points sont coplanaires. L'objet de référence ne peut donc en particulier pas être plat ni trop éloigné de $\mathscr{C}_o$.

Optimisation multi-vue

Le problème de l'agrégation des résultats (section 4.4.3) provient du fait qu'aucune contrainte multi-vue n'est utilisée. Le problème (4.17) peut être reformulé avec une matrice d'observation de la manière suivante :

$$\begin{pmatrix} \lambda_{11}\boldsymbol{c}_{11} & \cdots & \lambda_{1n}\boldsymbol{c}_{1n} \\ \vdots & & \vdots \\ \lambda_{m1}\boldsymbol{c}_{m1} & \cdots & \lambda_{mn}\boldsymbol{c}_{mn} \end{pmatrix} = \begin{pmatrix} \mathbf{K}_1[\mathbf{R}_{\mathscr{C}_o\to\mathscr{W}}|\boldsymbol{T}_{\mathscr{C}_o\to\mathscr{W}} - \boldsymbol{T}_1] \\ \vdots \\ \mathbf{K}_m[\mathbf{R}_{\mathscr{C}_o\to\mathscr{W}}|\boldsymbol{T}_{\mathscr{C}_o\to\mathscr{W}} - \boldsymbol{T}_m] \end{pmatrix} \left(\boldsymbol{O}_1^{(\mathscr{C}_o)} \cdots \boldsymbol{O}_n^{(\mathscr{C}_o)}\right) \tag{4.22}$$

où les λ_{ij} sont les profondeurs projectives, calculables à partir des matrices fondamentales ou par des méthodes itératives (Triggs, 1996; Oliensis et Hartley, 2005). Dans notre contexte, nous observons que l'approche DLT ignore trois contraintes : le fait que $\mathbf{R}_{\mathscr{C}_o\to\mathscr{W}}$ et $\boldsymbol{T}_{\mathscr{C}_o\to\mathscr{W}}$ sont partagés par toutes les vues et que les $\boldsymbol{U}_i^{(\mathscr{W})}$ sont liés aux $\mathbf{K}_i$ par l'équation (4.15). À notre connaissance, il n'existe aucune méthode permettant de calculer une estimation directe utilisant ces contraintes. Notons par ailleurs que notre approche a l'avantage de rendre le processus aisément parallélisable pour une résolution plus rapide. Enfin, l'estimation directe n'est utilisée ici que pour une estimation initiale, qui doit seulement permettre la convergence de l'ajustement de faisceaux pour garantir l'optimalité de la solution.

4.5 Scénarios envisageables et applications

La méthode a été introduite dans le cadre d'un système particulier mais nous nous sommes efforcé de ne pas utiliser d'information *a priori*, ce qui permet plusieurs généralisations. Les applications envisagées peuvent être classées en trois catégories.

4.5.1 Lunette augmentée

Dans le cas des lunettes augmentées ou *Head-Mounted Displays* (HMD), l'écran est rigidement fixé à l'utilisateur.

Avantages : Il n'y a pas besoin de suivi de l'utilisateur $\mathscr{C}_u$. Il est aisé d'afficher un flux vidéo adapté à chaque œil pour que l'affichage soit aligné pour les deux yeux. Le système est mobile et ne restreint pas les mouvements de l'utilisateur.

Inconvénients : La localisation par vision dans la scène est complexe à cause des mouvements rapides de rotation de la tête et des contraintes fortes sur la latence. Les solutions éprouvées sont basées sur une localisation externe (systèmes électromagnétiques ou capture du mouvement avec marqueurs par exemple). En l'état actuel des technologies, les HMD induisent une fatigue due aux incohérences perçues par le cerveau (principalement la latence).

Étalonnage : La position utilisateur $\boldsymbol{U}^{(\mathscr{W})}$ est constante. L'étalonnage peut être effectué en utilisant une seule caméra virtuelle centrée sur l'utilisateur. Ce dernier devra connecter un matériel spécifique (souris…) pour entrer la position des observations 2D des points 3D de référence. Alternativement, les points de référence peuvent être organisés en un motif spécifique et l'utilisateur devra se déplacer pour faire coïncider le motif s'affichant sur son écran avec le motif réel ; cela permet d'effectuer l'étalonnage en une seule étape.

4.5.2 Vitrine augmentée

Dans la configuration de *vitrine augmentée* (voir figure 4.10), le système est fixe dans la scène.

Avantages : Il n'y a dans ce cas pas besoin du dispositif de localisation dans la scène $\mathscr{C}_o$. L'étalonnage fournit toutes les informations requises pour ajouter des informations en réalité augmentée sur l'objet de référence. De plus, en restreignant la réalité augmentée à une fenêtre bien définie dans l'espace, l'inconfort pour l'utilisateur est réduit car une grande partie de son champ de vision n'est pas affecté, contrairement aux lunettes augmentées.

Inconvénients : En l'absense de dispositif de localisation dans la scène, l'étalonnage doit être actualisé à chaque déplacement du système ou modification de la scène.

Étalonnage : Le repère $\mathscr{C}_o$ est remplacé par le repère local de l'objet. L'étalonnage fournit la pose $\mathbf{M}_{\mathscr{C}_u \to \mathscr{W}}$ permettant de localiser l'utilisateur par rapport à l'écran ainsi que la pose $\mathbf{M}_{\mathscr{C}_o \to \mathscr{W}}$ qui est ici directement la pose de l'objet dans $\mathscr{W}$.

FIGURE 4.10 – Vitrine composée d'un écran transparent présentée par Samsung. Le concept de vitrine augmentée présentée en section 4.5.2 utiliserait un tel dispositif pour afficher des augmentations alignées sur les objets exposés.

Il s'agit de la configuration choisie pour l'évaluation présentée en section 4.6.3 car elle permet de s'affranchir d'un algorithme de localisation dans la scène.

4.5.3 Tablette augmentée

Les vitrines virtuelles sont limitées aux systèmes statiques, et en l'état actuel des technologies les HMD restent limités et trop intrusifs. C'est pourquoi nous croyons au paradigme de la *tablette augmentée*, utilisé tout au long de ce chapitre. Cette approche utilise un écran semi transparent, un dispositif de localisation dans la scène (pour que le système puisse être déplacé sans nouvel étalonnage) et un dispositif de suivi de l'utilisateur (pour que l'utilisateur soit libre de ses mouvements par rapport au système). Un tel système combine les avantages des deux approches précédentes. Il existe des solutions pour garder libres les mains de l'utilisateur (les applications visées sont l'aide à la chirurgie et à la maintenance de systèmes complexes). Par exemple un bras articulé, déjà courant dans les environnements médicaux, pourrait maintenir la tablette en place.

L'inconvénient majeur, précédemment évoqué, de cette approche est qu'elle est la plus complexe à étalonner. Cependant, le cadre de travail introduit dans ce chapitre contribue à atténuer ce problème.

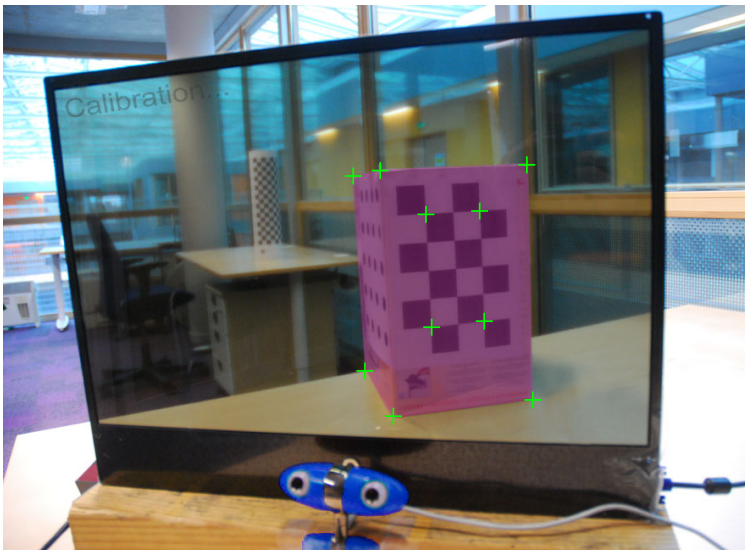


FIGURE 4.11 – Prototype utilisé pour les expérimentations. L'objet de référence est coloré en mauve, et les points de référence utilisés sont indiqués par une croix verte. Le dispositif de localisation de l'utilisateur est une caméra stéréo (en bleu).

4.6 Évaluation

Cette section est dédiée à la démonstration de la robustesse de notre méthode et de sa précision. L'algorithme a été implémenté en Python et les temps de calcul sont compris entre 1 et 3 secondes par étalonnage.

4.6.1 Détails du prototype et estimation du bruit

Les étapes de raffinement non linéaire et d'ajustement de faisceaux nécessitent une estimation de la variance du bruit sur les positions de l'utilisateur Σ_U, les points 3D de référence de la scène Σ_O et les observations 2D de l'utilisateur Σ_c. Nous définissons ici la configuration du prototype (figure 4.11) utilisé pour l'évaluation.

Observations 2D de l'utilisateur

L'écran transparent est un écran SAMSUNG 22 pouces de référence LTI220MT02. La surface active est de 473.6×296.1 mm^2 avec une résolution de 1680×1050 pixels. La taille d'un pixel (carré) est donc $s_{px} = 0.282$ mm/px.

L'utilisateur indique la position des observations 2D des points de référence dans l'écran par le biais de clics avec la souris. Notons qu'en pratique ces clics

sont imprécis à cause de plusieurs facteurs : flou introduit par l'écran transparent, précision de la souris (un pixel), forme du curseur, tremblements de la main, défauts de vision et autres facteurs humains.

Une étude simple a été menée pour évaluer cette erreur dans la configuration choisie : une croix est affichée à une position aléatoire sur l'écran et l'utilisateur devait cliquer au plus proche de la croix depuis un point de vue aléatoire à l'intérieur de la zone définie par l'équation (4.29). Après avoir répété l'opération 100 fois, l'erreur moyenne observée est d'environ 3 pixels donc on établit :

$$\begin{aligned} \Sigma_c &= \mathrm{diag}(3^2, 3^2) && \text{en pixels} \\ &= \mathrm{diag}(0.846^2, 0.846^2) && \text{en millimètres} \end{aligned} \tag{4.23}$$

Il est à noter que l'imprécision de la position de l'utilisateur se traduit par une imprécision supplémentaire sur les données 2D. L'imprécision totale est calculée dans l'équation (4.28).

Points 3D de référence

Nous utilisons comme objet de référence une boîte rectangulaire de dimensions $372 \times 305 \times 229$ mm, placée à environ 1 mètre de l'écran. 10 points de référence sont choisis : les 6 coins visibles ainsi que 4 points additionnels sur les côtés. L'objet est statique (configuration *vitrine augmentée* section 4.5) et le repère $\mathscr{C}_o$ est défini comme les coordonnées locales de l'objet. Il n'y a donc pas de dispositif de localisation dans la scène et le bruit sur les points de référence est donc uniquement imputable aux erreurs du modèle 3D. En pratique ces dernières sont très faibles, on fixe expérimentalement :

$$\Sigma_O = \mathrm{diag}(1^2, 1^2, 1^2) \text{ en millimètres.} \tag{4.24}$$

Positions de l'utilisateur

Tout type de méthode de suivi de l'utilisateur peut être utilisée selon les contraintes de l'application. Il est par exemple possible d'utiliser un suivi électromagnétique ou une solution commerciale monoculaire telle que `faceAPI` [2]. Nous utilisons dans nos expérimentations une caméra stéréoscopique USB Minoru (deux images de 640×480 pixels). La pupille gauche de l'utilisateur est détectée dans chaque caméra (Viola et Jones, 2001), puis un algorithme Mean-Shift (Comaniciu *et al.*, 2000) permet de trouver le centre de la pupille. Le principal avantage de cette méthode est d'être basée détection et donc plus robuste que les méthodes nécessitant un suivi dans le

2. http://www.seeingmachines.com/product/faceapi/

temps. Cependant, il est difficile de localiser précisément la pupille avec une telle caméra et la position obtenue est donc très bruitée. Pour modéliser le bruit, nous avons observé le comportement de l'estimation pour des utilisateurs immobiles pendant plusieurs centaines d'images. La variance observée est :

$$\Sigma_{\boldsymbol{U}} = \text{diag}(5^2, 5^2, 20^2) \text{ en millimètres} \tag{4.25}$$

Lors de la mise en œuvre de l'étalonnage, l'utilisateur doit théoriquement rester parfaitement immobile pendant chaque série de clics, ce qui est impossible. Pour limiter la perte de précision, sa position est mesurée à chaque clic et la moyenne est calculée pour chaque série. Pour une séquence de n clics, la variance de la position utilisateur est donc divisée par n : $\Sigma_{\boldsymbol{U}_{\text{average}}} = \frac{1}{n}\Sigma_{\boldsymbol{U}}$. Nous avons 10 clics par séquence donc :

$$\begin{aligned} \Sigma^{(n=10)}_{\boldsymbol{U}_{\text{average}}} &= \text{diag}(5^2/n, 5^2/n, 20^2/n) \\ &= \text{diag}(1.6^2, 1.6^2, 6.3^2) \qquad \text{en millimètres} \end{aligned} \tag{4.26}$$

Cependant, les mouvements de l'utilisateur autour de cette position moyenne induisent une erreur sur les observations 2D. On peut effectuer une approximation de cette erreur 2D supplémentaire grâce à la figure 4.12. Pour un utilisateur et un objet à distances moyennes respectives par rapport à l'écran de $d_U = 700$ mm et $d_O = 1000$ mm, l'erreur 2D est :

$$\begin{aligned} \Sigma_{\boldsymbol{c}_{\text{mouv.}}} &= \left(\frac{d_O}{d_U + d_O}\right)^2 \cdot \text{diag}(\Sigma_{U11}, \Sigma_{U22}) \\ &\approx \text{diag}(2, 2) \text{ en millimètres} \\ &\approx \text{diag}(7.3^2, 7.3^2) \text{ en pixels.} \end{aligned} \tag{4.27}$$

L'erreur totale des observations 2D est alors, en intégrant l'équation (4.23) :

$$\Sigma_{\boldsymbol{c}_{\text{total}}} = \Sigma_{\boldsymbol{c}} + \Sigma_{\boldsymbol{c}_{\text{mouv.}}} = \text{diag}(10.3^2, 10.3^2) \text{ en pixels.} \tag{4.28}$$

La caméra a un champ de vision restreint qui empêche le suivi de l'utilisateur hors de la zone définie par :

$$\boldsymbol{U}_i^{(\mathscr{C}_u)} \in [-300, 300] \times [-150, 150] \times [400, 1000] \text{ en mm} \tag{4.29}$$

Cette zone est encore réduite par la contrainte que tous les points 3D de référence doivent être visibles à travers l'écran. Le nombre de points de vue différents pour l'évaluation est fixé à 20.

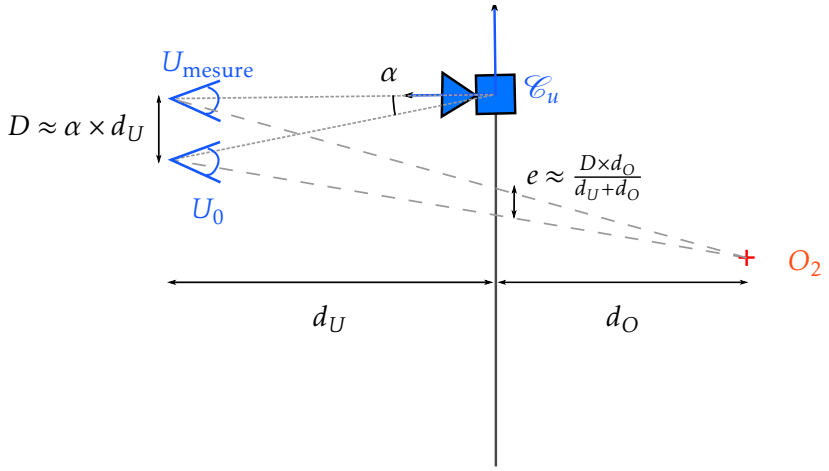


FIGURE 4.12 – Étude de l'impact d'une erreur angulaire (α) ou en translation (D) sur l'erreur d'alignement. Ces relations sont uniquement valables pour de faibles valeurs de l'erreur angulaire α mais permettent d'estimer l'erreur d'alignement minimale e.

4.6.2 Évaluation sur données synthétiques

Pour étudier le comportement de notre étape d'estimation directe, une scène synthétique est générée avec une géométrie et des niveaux de bruit similaires au cas réel présenté en section 4.6.1. À partir de cette référence réaliste, l'influence des différents paramètres est mesurée sur 50 tirages aléatoires avant de calculer la moyenne et l'écart-type de l'erreur. L'erreur 3D de l'étalonnage en rotation et translation est utilisée plutôt que l'erreur 2D de reprojection dans les caméras virtuelles qui n'est pas un bon indicateur de la qualité de l'initialisation. Les lignes 1 et 2 de la figure 4.13 montrent que le bruit sur les positions utilisateur n'affecte pas les caméras centrées sur ces dernières, et le bruit sur les points 3D de référence n'affecte pas les caméras centrées sur les points de référence. Le résultat le plus intéressant de ces expériences est l'importance cruciale de la géométrie du problème (théoriquement justifié en section 4.4.6). En effet, elle est mise en évidence par les courbes sur l'échelle de l'objet (ligne 3) : l'erreur en translation de la pose du dispositif de localisation dans la scène est par exemple triplée lorsque la taille de l'objet est divisée par deux.

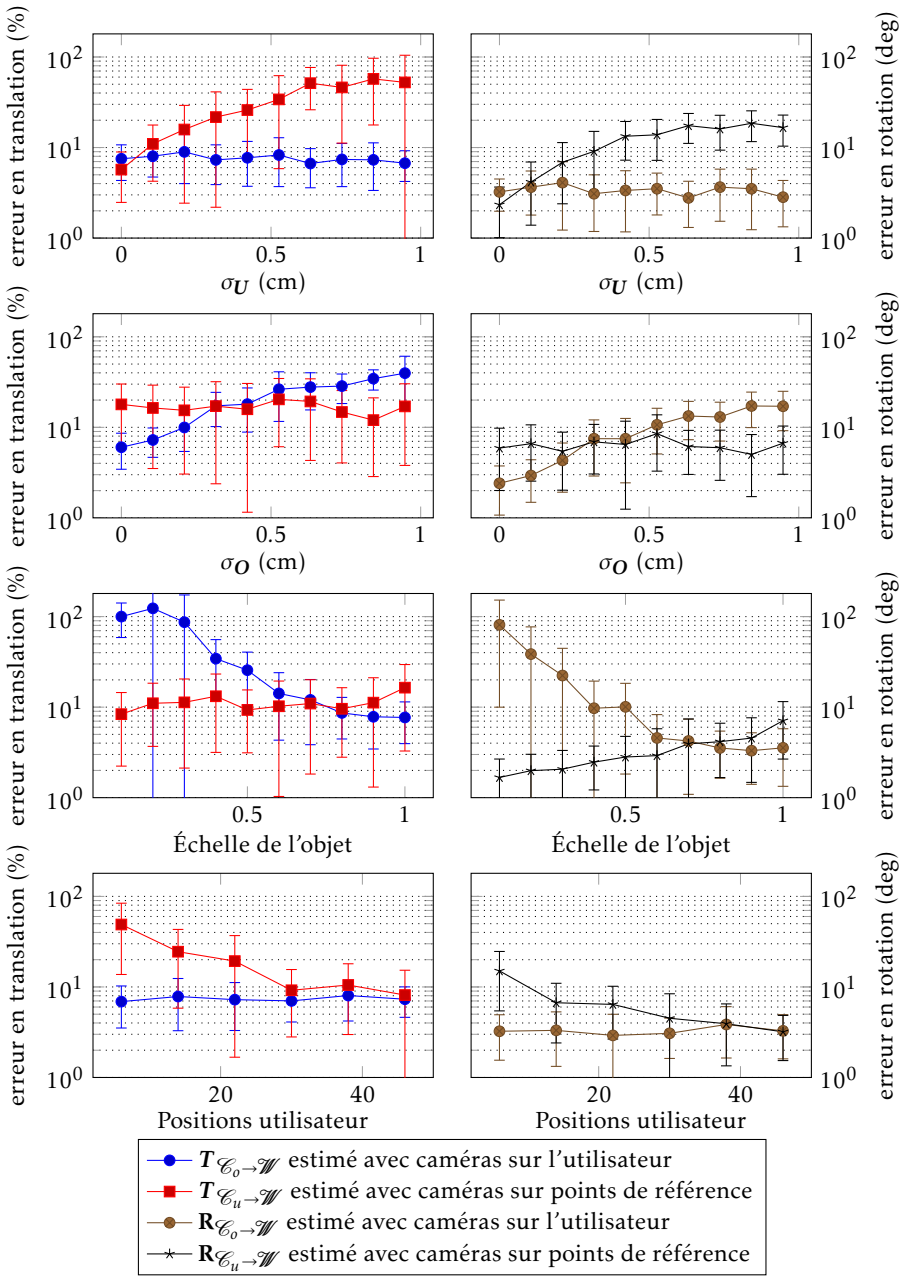


FIGURE 4.13 – Comparaison de la robustesse au bruit sur les positions utilisateur, positions des points 3D de référence de la scène, échelle de l'objet et nombre de positions utilisateur. Les points correspondent à la valeur moyenne sur 50 échantillons et les barres correspondent à l'écart-type. L'échelle en y est logarithmique.

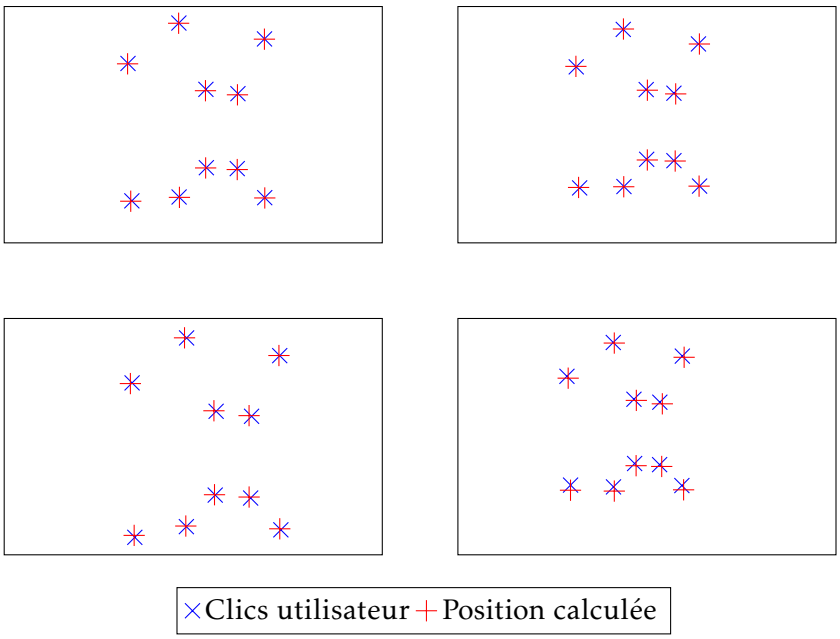


Figure 4.14 – Exemples d'alignement sur les données de validation croisée.

4.6.3 Évaluation sur données réelles

Le processus d'étalonnage est ensuite mis à l'épreuve sur le prototype réel visible sur la figure 4.11. Un utilisateur clique à la souris sur 10 points précédemment repérés sur la boîte et répète l'opération depuis 20 points de vue différents. Qualitativement, l'erreur d'alignement est presque imperceptible comme le montrent les exemples de reprojections de la figure 4.14. Des résultats quantitatifs sont produits au moyen d'une validation croisée : les poses de la caméra et de l'objet sont estimées en utilisant 19 positions et l'erreur d'alignement est calculée du point de vue de la position restante. Les résultats sont répertoriés sur la figure 4.15. L'erreur de reprojection moyenne, à 10.56 pixels, est sensiblement égale à notre modélisation : 10.3 pixels dans l'équation (4.28), et est donc optimale. Cela correspond à moins de 3 mm à l'écran.

4.6.4 Comparaison avec travaux similaires

Il est intéressant de comparer ce résultat d'étalonnage aux méthodes utilisant un miroir telles que Rodrigues *et al.* (2010); Takahashi *et al.* (2012). Ces méthodes permettent d'estimer la pose d'une caméra par rapport à un objet hors champ. Elles estiment en fait les paramètres extrinsèques de caméras virtuelles, images de la

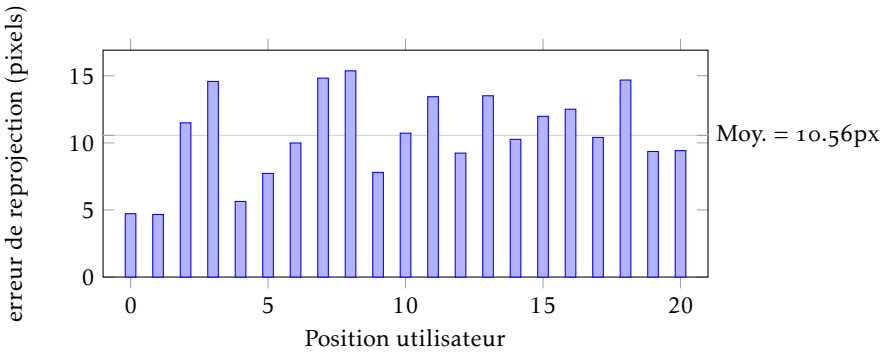


FIGURE 4.15 – Validation croisée.

caméra réelle dans le miroir (voir figure 4.4). Les deux publications présentent des résultats en conditions réelles (voir figure 4.16) et rapportent respectivement une erreur de 0.7° = 0.012 rad (Rodrigues *et al.*, 2010) et 3.26° = 0.057 rad (Takahashi *et al.*, 2012).

La figure 4.12 montre que dans le cas d'une erreur angulaire α faible, pour un utilisateur à une distance d_U de l'écran et un point 3D à une distance d_O de l'écran, l'erreur de reprojection minimale résultante est $e = \frac{\alpha d_U d_O}{d_U + d_O}$. Dans le cas d'un utilisateur à $d_U = 0.7$ mètres de la caméra (position moyenne dans notre configuration) et d'un point de référence à $d_O = 1$ mètre de l'écran, une erreur angulaire de 0.012 rad se traduit par une erreur d'alignement de $\frac{0.012 \times 0.7 \times 1}{0.7+1} \approx 0.005$ mètres (5 mm). Cette borne inférieure de l'erreur, en conditions optimale (12 prises de vue), sans prendre en compte l'erreur en translation et les erreurs introduites par les méthodes de localisation, est presque deux fois supérieure à nos résultats expérimentaux (3 mm). On peut supposer une erreur similaire pour la calibration de la caméra observant la scène, amenant à une erreur totale de l'odre du centimètre. Cette erreur est beaucoup trop importante pour des applications de réalité augmentée. De plus, comme précisé en introduction, ces méthodes ne sont applicables qu'à des caméras alors que notre méthode est indépendante des dispositifs de localisation utilisés.

4.7 Conclusion

Ce chapitre présente une solution à l'étalonnage des systèmes de réalité augmentée utilisant un affichage semi transparent et deux dispositifs de localisation (utilisateur et scène). Notre première contribution consiste en la formalisation du problème de calibration d'un tel système, menant à la formulation d'un ajustement de faisceaux pour obtenir la solution optimale. Notre deuxième contribution

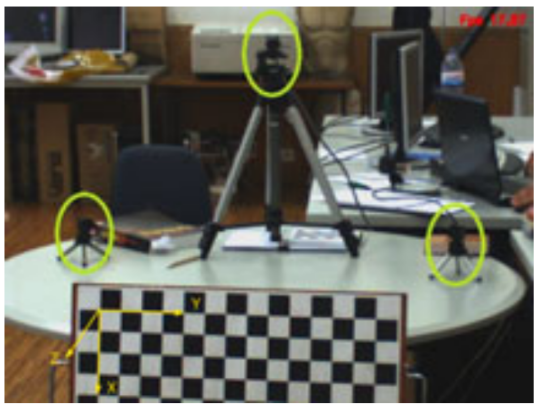


FIGURE 4.16 – Image extraite de la publication de Rodrigues *et al.* (2010) décrivant l'évaluation de leur méthode en conditions réelles. Les observations de la cible par trois caméras à travers un miroir permettent de calculer leurs paramètres extrinsèques. La cible est ici dans le champ des trois caméras pour pouvoir calculer une vérité terrain sans miroir.

consiste en l'introduction d'un nouveau formalisme autour de caméras virtuelles afin d'obtenir une initialisation convexe à cet ajustement de faisceaux. Nos expérimentations ont démontré la précision et la robustesse de l'approche, ainsi que sa généralisation à d'autres configurations. Nous envisageons par la suite d'étudier différents types de solutions de localisation afin de mettre en place une chaîne complète de réalité augmentée. Il est également envisageable d'adapter la méthode à des surfaces non planes telles que les pare-brise de voiture. Il serait également avantageux de trouver un moyen de compenser l'erreur d'alignement induite par les mouvements involontaires de l'utilisateur durant l'étalonnage. Comme détaillé en annexe 4.A, l'exploitation des progrès théoriques proposés reste limitée par le matériel disponible actuellement. Cette situation pourrait toutefois changer à court terme car les technologies requises existent.

Annexe 4.A Défis et verrous matériels

L'approche développée ici apporte une réponse théoriquement optimale à la réalité augmentée sur affichage semi transparent, cependant le confort de l'expérience utilisateur reste limitée par des contraintes matérielles. Nous listons ici les défis et verrous matériels identifiés ainsi que les solutions ou contournements envisageables grâce aux technologies actuelles ou à venir.

4.A.1 Transparence et luminosité de l'affichage

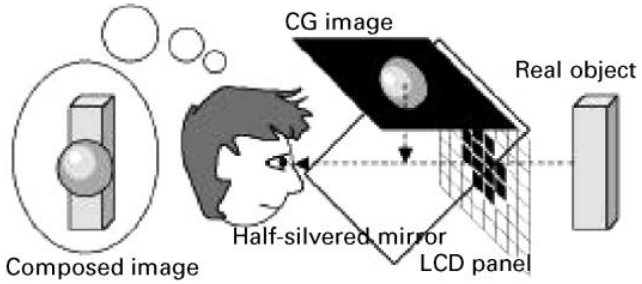
On peut classer les technologies d'affichage semi transparent en affichages additifs (par émission) et soustractifs (par absorption), comme les technologies LCD et OLED et présentées sur la figure 3.5. Les affichages additifs superposent les augmentations à la réalité en ajoutant une information lumineuse, et sont donc peu visibles dans un environnement très lumineux (à l'extérieur en plein jour par exemple), ce qui est un obstacle au développement de lunettes de réalité augmentée. De plus, les augmentations ne peuvent totalement occulter la scène réelle qui reste toujours visible par transparence. À l'opposé, les affichages soustractifs atténuent sélectivement la lumière entrante. Ils fonctionnent donc de manière optimale quand la scène observée est lumineuse, claire et homogène (un ciel nuageux par exemple). C'est pour cette raison que de tels écrans sont pour le moment utilisés en tant que fenêtre (figure 3.5a) ou vitrine (figure 4.10).

Des solutions ont été proposées, par exemple par Kiyokawa *et al.* (2001), pour combiner les avantages des deux technologies : l'affichage est effectué par une technologie additive, mais les zones occultées de la scène réelle sont cachées grâce à un écran LCD (voir figure 4.17). Ainsi, la scène réelle n'est pas visible à travers les augmentations pour un meilleur confort d'utilisation et plus de réalisme. Ces approches sont encore peu mises en œuvres car peu compactes.

4.A.2 Stéréo-vision

Comme expliqué en section 3.4.3, le rendu des augmentations dépend du point de vue de l'utilisateur. On peut préciser cette propriété en disant que le rendu des augmentations dépend de *la position de l'œil de l'utilisateur*. Par conséquent, si la distance entre les yeux n'est pas négligeable devant la distance à l'écran, il faut effectuer un rendu pour chaque œil sous peine d'observer un décalage entre les observations des deux yeux. La solution la plus simple à ce problème, utilisée lors des expériences précédentes, est de n'utiliser qu'un seul œil, en fermant ou cachant l'autre. Cela permet une évaluation correcte du système mais n'est pas envisageable pour un système final. Une autre solution est d'utiliser une approche de type « lunettes » où chaque œil possède un écran dédié, mais en perdant les avantages de notre approche *tablette augmentée*. Nous envisageons plutôt une troisième solution à base de stéréoscopie. L'idéal serait une solution auto-stéréoscopique, où l'ajout d'une couche spéciale à l'écran permet de diriger l'image vers chaque œil alternativement. Deux technologies envisageables sont décrites en figure 4.18. Elles ne sont malheureusement pas encore disponibles pour des écrans transparents. À plus court terme, nous prévoyons d'employer des technologies de stéréoscopie active où des lunettes spéciales occultent alternativement chaque œil de manière synchronisée

(a) Problème : la scène réelle apparaît en transparence (image fantôme).



(b) Solution : un écran LCD cache les zones occultées de la scène réelle pour éviter qu'elles n'apparaissent en transparence.

FIGURE 4.17 – Gestion des occultations mutuelles par Kiyokawa *et al.* (2001). Images extraites de leur publication.

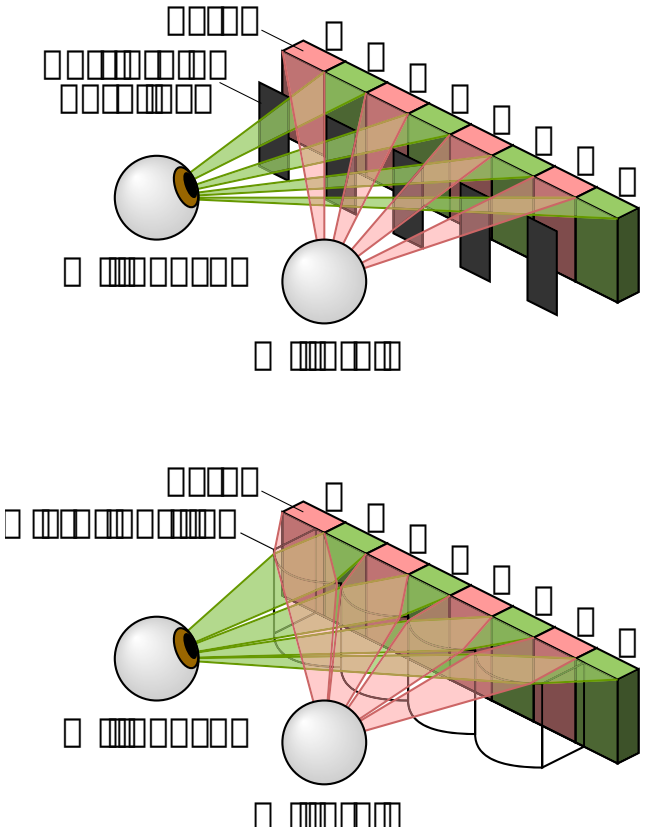


FIGURE 4.18 – Technologies d'affichage auto-stéréoscopique. *Source : Wikipedia (Autostereoscopy).*

avec l'affichage. Cette approche serait intéressante malgré l'emploi de lunettes : elles sont beaucoup plus légères et confortables que les lunettes de réalité augmentée et gênent peu la vision.

4.A.3 Focalisation

Les systèmes d'affichage additifs sont constitués de pixels qui émettent chacun dans plusieurs directions (voir figure 4.19). Ainsi, si l'œil de l'utilisateur n'est pas focalisé sur l'écran, les augmentations apparaissent floues. Si la distance de la scène observée est connue et fixe, la meilleure solution est d'utiliser un système optique pour placer un écran virtuel proche de la position 3D des augmentations. Une extension de cette méthode utilise un support vibrant pour effectuer un rendu 3D : les vibrations sont traduites par un changement de profondeur de l'écran virtuel,

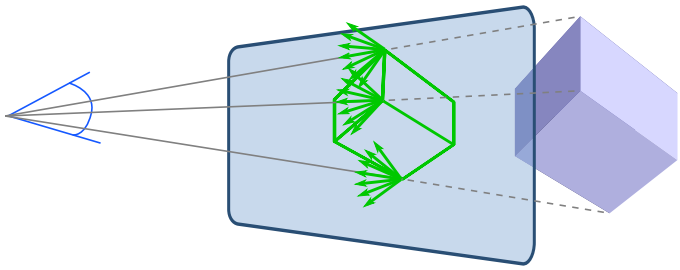
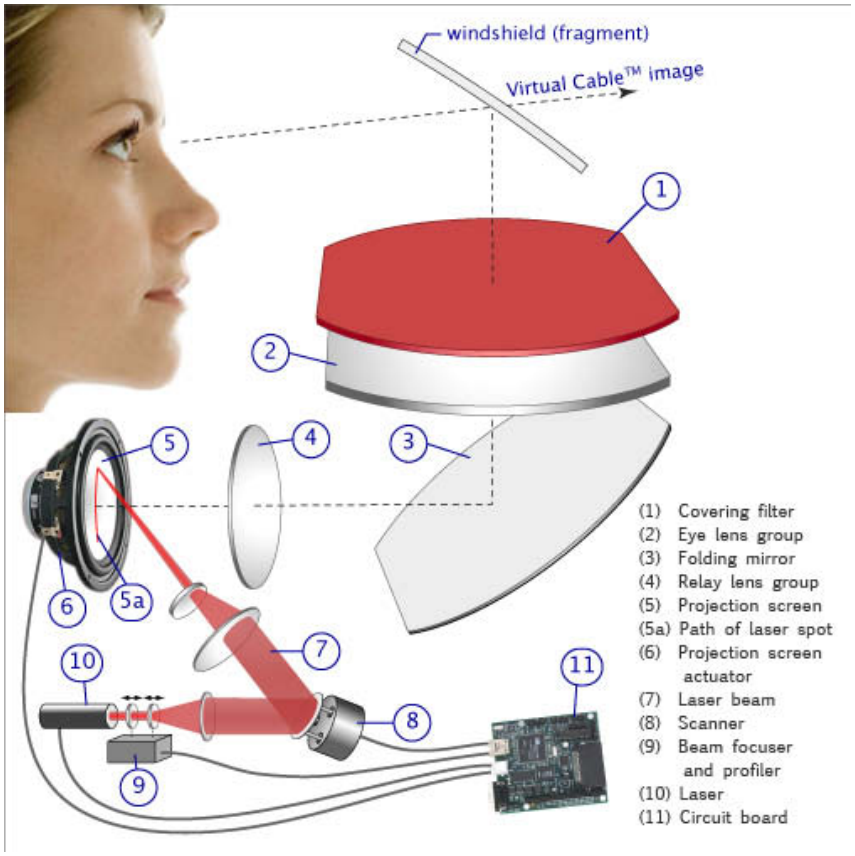
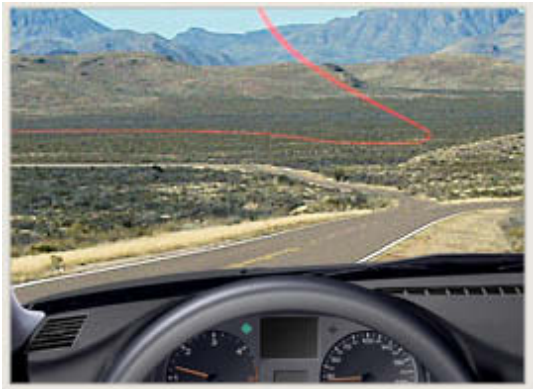


FIGURE 4.19 – Problème de focalisation lors de l'utilisation d'un système d'affichage par émission : les pixels émettent dans toutes les directions et apparaissent flous si l'augmentation 3D est éloignée du plan de l'écran.

il faut ensuite synchroniser le rendu pour que chaque partie de l'augmentation soit rendue à la bonne profondeur. La vibration doit être rapide pour ne pas être perceptible ce qui rend la synchronisation difficile. À notre connaissance, le système de câble virtuel présenté en figure 4.20 est la seule application commerciale de cette technique. À cause des contraintes citées précédemment il est limité à l'affichage d'un seul câble virtuel. Enfin, des technologies affichant les images directement sur la rétine à l'aide d'un laser sont théoriquement capables de répondre à ce problème en modulant la focalisation de chaque pixel (Tidwell *et al.*, 1995) mais de telles solutions ne sont pas encore développées.



(a) Principe : les vibrations du dispositif ⑥, amplifiées par le système optique ②③④, font varier la profondeur apparente du câble symbolisé par le laser. Le dispositif ⑧ fait varier la position latérale du laser.



(b) Résultat

FIGURE 4.20 – Système *Virtual Cable* de la société *Making Virtual Solid*.

Chapitre 5

Utilisation de correspondances éparses pour l'alignement dense d'images

Ce chapitre a fait l'objet de deux publications internationales : conférences BMVC (2013b) et ICCV (2013c), et d'une publication nationale : congrès RFIA (2014b).

5.1 Introduction

Comme nous l'avons vu en section 2.5.2, la mise en correspondance dense entre deux images est une étape essentielle pour la reconstruction 3D (cartes de profondeurs) et le flot de scène. Dans ce chapitre, nous proposons une méthode pour élargir significativement le bassin de convergence de méthodes variationnelles d'estimation de champs de correspondances avec des applications au calcul de flot optique et à l'estimation de cartes de profondeurs par stéréovision, ainsi qu'à l'alignement de surfaces déformables.

Les techniques d'estimation de ces correspondances peuvent être classées en deux catégories largement indépendantes :

— correspondances denses (souvent par optimisation variationnelle, voir section 2.7) en présence de faibles déformations, souvent par optimisation variationnelle,

— correspondances de primitives (par recherche de voisins dans un espace de descripteurs, voir section 2.6), éparses mais plus discriminantes, pour les déformations importantes.

La dualité entre correspondances locales denses et correspondances globales éparses se retrouve dans toute les applications de mise en correspondance d'images. Pour la reconstruction par stéréovision, par exemple, il y a d'un côté l'estimation de cartes de profondeurs denses (Hirschmuller, 2005; Mei *et al.*, 2011; Ranftl *et al.*, 2012) adaptées aux faibles parallaxes, et de l'autre les reconstructions éparses par triangulation de points d'intérêt (*Structure from Motion*, Snavely *et al.*, 2006). De manière similaire, l'alignement de surfaces déformables est séparé en deux

problèmes distincts : le *suivi de surface* (Gay-Bellile *et al.*, 2010; Garg *et al.*, 2013b) qui consiste à estimer une déformation au long d'une séquence où les changements d'image à image sont assez faibles pour utiliser une approche variationnelle ; et la *détection de surface* (Pilet *et al.*, 2008; Pizarro et Bartoli, 2012; Tran *et al.*, 2012) consiste à estimer directement une déformation potentiellement importante.

Des techniques de chaque catégorie peuvent parfois être utilisées de manière complémentaire : initialisation par correspondances de primitives puis raffinement dense une fois proche de la solution. Par exemple, Furukawa et Ponce (2010) densifient une reconstruction rigide initiale en générant des surfaces autour des points reconstruits, puis en les filtrant et en les étendant grâce aux contraintes multi-vues (voir figure 5.1). Cependant le passage d'un champ de correspondance épars à un champ dense n'est pas trivial et sujet à des minimums locaux potentiellement irrécupérables. Par exemple, pour l'alignement de surfaces non rigides, (Pizarro et Bartoli, 2012) opèrent en deux étapes : ajustement d'un modèle de déformation paramétrique aux correspondances éparses (avec une étape préalable de filtrage), puis raffinement dense mais ce dernier ne peut éviter de mauvaise performances dans les zones peu texturées.

Notre objectif est de mettre fin à cette séparation pour bénéficier des avantages de chaque méthode et mitiger leurs inconvénients, ceci grâce à l'optimisation conjointe d'une combinaison de termes de donnée épars et denses (Wills *et al.*, 2006; Xu *et al.*, 2010; Brox et Malik, 2011; Leordeanu *et al.*, 2013). D'une manière similaire à Brox et Malik (2011), nous intégrons un coût basé sur des correspondances éparses au sein d'une méthode variationnelle dense, mais nous proposons un nouveau modèle plus flexible et robuste.

Il est à noter que quelques méthodes utilisent pourtant des descripteurs pour effectuer directement des correspondances denses entre images très dissimilaires (Liu *et al.*, 2011; Tola *et al.*, 2010). Cependant, elles utilisent des optimisations discrètes, dont le temps de calcul croît exponentiellement en fonction du nombre de déplacements considérés. C'est pour cette raison que SIFT-Flow (Liu *et al.*, 2011) sous-échantillonne fortement les images avant traitement, ce qui produit un champ de correspondance cohérent mais imprécis (voir figure 5.2). Le descripteur DAISY (Tola *et al.*, 2010) pour sa part n'est appliqué qu'à des problèmes de stéréovision unidimensionnels, réduisant ainsi grandement l'espace des solutions. Nous recherchons pour les applications de réalité augmentée considérées (voir chapitre 1) un cadre général permettant d'obtenir des correspondances denses 2D, à la fois rapides et précises, ce que ne permettent pas ces approches.

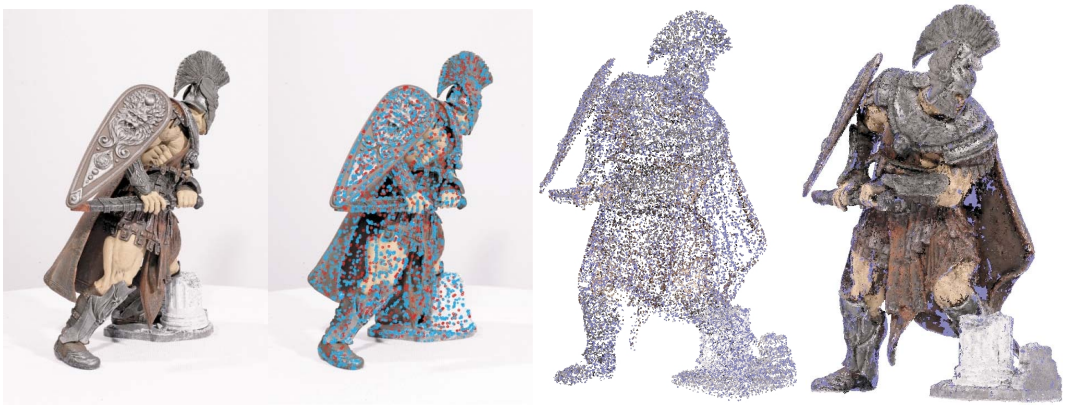


FIGURE 5.1 – Exemple de reconstruction par Furukawa et Ponce (2010). De gauche à droite : une des images en entrée, primitives détectées, reconstruction éparse, reconstruction dense.

(a) I_1 (b) I_2 (c) Champ de déplacements

FIGURE 5.2 – Exemple de sortie de SIFT-Flow (Liu *et al.*, 2011). L'estimation est robuste mais le champ de déplacement est grossier avec des blocs visibles.

Formulation du problème et notations

Soit une paire d'images I_1 et I_2 de domaines respectifs Ω_{I_1} et Ω_{I_2}. Les images sont modélisées par des fonctions continues à valeurs réelles (voir section 2.3) et sont normalisées pour appartenir à l'intervalle $[0,1]$. On cherche à estimer le champ de déplacement suivant :

$$\boldsymbol{u} = (u_x, u_y) : \Omega_0 \to \mathbb{R}^2$$

tel que :

$$\forall \boldsymbol{x} \in \Omega_1 : \qquad I_1(\boldsymbol{x}) \approx I_2(\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})), \tag{5.1}$$

ou de manière équivalente le champ de correspondance $\mathbf{a}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ tel que :

$$\forall \boldsymbol{x} \in \Omega_1 : \qquad I_1(\boldsymbol{x}) \approx I_2(\mathbf{a}(\boldsymbol{x})). \tag{5.2}$$

5.2 État de l'art de l'utilisation de primitives pour l'estimation de correspondances denses

Cette section résume les différentes approches de la littérature pour répondre à deux aspects majeurs de notre problème :

— la *densification* de la contrainte éparse basée primitive est abordée en section 5.2.1,

— la gestion des correspondances erronées est traitée en section 5.2.2.

5.2.1 Densification de la contrainte basée primitives

Pour pouvoir intégrer des correspondances de primitives dans une méthode d'estimation dense, il faut répartir l'influence des ces correspondances éparses sur toute l'image. Nous appelons un tel processus *densification*. Trois approches peuvent être identifiées : ajustement de modèle, optimisation discrète et densification multi-résolution.

Modèles de mouvement

Une manière naïve de densifier la contrainte éparse est d'interpoler les correspondances ce qui revient à faire l'hypothèse que le champ de mouvement observé est doux sans discontinuités. Cette hypothèse peut être valide pour les surfaces déformables (Pizarro et Bartoli, 2012; Tran *et al.*, 2012) en l'absence de pliure, mais rarement dans les autres cas. Un modèle de mouvement plus général, tel que affine

par morceaux, peut alors être préféré (Wills *et al.*, 2006; Leordeanu *et al.*, 2013). Cela permet de regrouper les correspondances en ensembles cohérents, ou *couches* dont les frontières sont estimées à partir de l'image (segmentation). Une fois les couches bien définies, il est possible d'effectuer un raffinement dense au sein de chaque ensemble. Les étapes de segmentation et raffinement peuvent être itérées pour de meilleurs résultats. Ces approches produisent des champs de déplacement avec des discontinuités bien nettes, elles sont cependant vulnérables aux erreurs de segmentations et dépendantes de la validité du modèle de mouvement utilisé.

Candidats pour optimisation discrète

Xu *et al.* (2010) utilisent les correspondances pour générer une liste globale de déplacements 2D, fusionnés par optimisation discrète (voir figure 5.3). Cette étape est intégrée à un traitement pyramidal, de basse à haute résolution, ainsi qu'à un raffinement variationnel. L'algorithme complet fournit parmi les meilleurs résultats de l'état de l'art, il est cependant très lent. De plus l'algorithme QPBO (Rother *et al.*, 2007), utilisé pour l'optimisation discrète, possède la propriété indésirable de dépendre de l'ordre d'intégration des candidats de déplacements. Enfin et surtout, en convertissant les correspondances en vecteurs de déplacement, cette méthode ignore toute information de localisation des primitives dans l'image. Elle est aussi incompatible avec des correspondances de primitives non ponctuelles telles que les segments de droite pour lesquelles on ne peut extraire de déplacement unique.

Densification multi-résolution

Brox et Malik (2011) ont proposé d'introduire des correspondances de primitives (points ou régions) au sein d'une méthode de flot optique variationnelle grâce à l'addition d'un nouveau terme épars dans la fonction de coût. Ce terme a pour but de favoriser les champs de déplacements qui sont en cohérence avec les correspondances. Ils ont mis en évidence le comportement intéressant, illustré figure 5.4 d'une approche multi-résolution pyramidale pour la densification du terme basé correspondances. L'évolution est similaire à un recuit simulé : à basse résolution l'image contient peu de variations (lissage implicite) ce qui rend le terme dense peu influent alors que les correspondances sont quasi denses et contraignent fortement l'optimisation ; à haute résolution l'équilibre est inversé et les correspondances ont une influence négligeable ce qui permet de préserver la précision du terme dense.

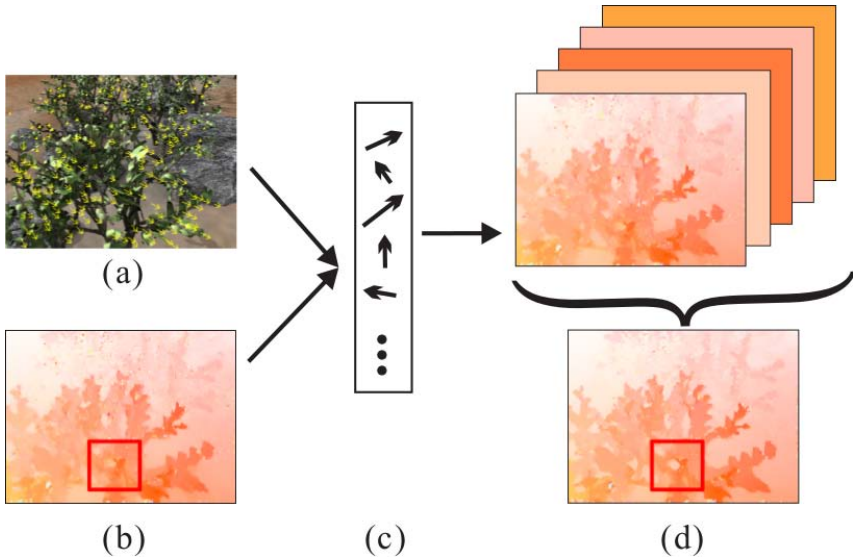


FIGURE 5.3 – Reproduction de Xu *et al.* (2010). Les correspondances de points (a) et le champ de déplacement (b) calculé à la résolution inférieure sont utilisés pour générer les vecteurs de déplacement candidats (c). Une optimisation discrète sélectionne le meilleur candidat pour chaque pixel dans le nouveau champ de déplacement (d).

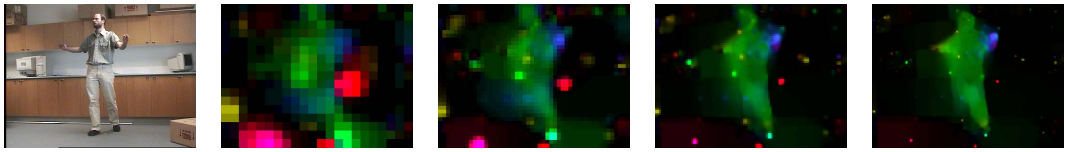
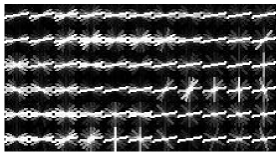


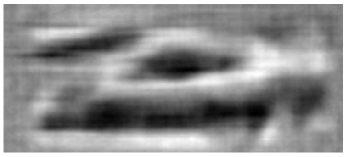
FIGURE 5.4 – Densification multi-résolution par Brox et Malik (2011). L'effet des correspondances est ici surtout visible grâce aux correspondances erronées qui génèrent de larges tâches incohérentes à basse résolution (à gauche), alors que leur influence est négligeable à haute résolution (à droite).

(a) Image et fausse détection

(b) Gros plan



(c) Descripteurs HOG



(d) Représentation HOGgles

FIGURE 5.5 – Images extraites de Vondrick *et al.* (2013). La représentation inverse HOGgles montre l'ambiguïté au niveau de la description HOG et explique la fausse détection d'une voiture.

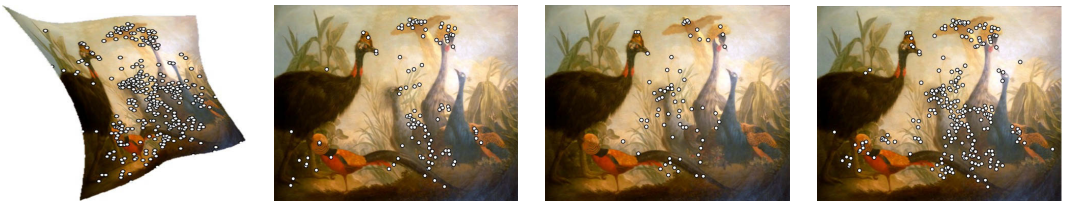
5.2.2 Gestion des correspondances erronées

Les correspondances de primitives considérées en entrée sont susceptibles de contenir des correspondances erronées. Il existe deux types d'approche pour supprimer leur influence : les filtrages explicite et implicite détaillés ci-après.

Filtrage explicite

La majorité des méthodes paramétriques d'alignement d'images sont basées sur l'ajustement d'un modèle de déformation (voir section 2.7.3) par optimisation aux moindre carrés. Les correspondances erronées ont un impact très fort avec un coût quadratique, la plupart des méthodes utilisent donc une étape de filtrage explicite où les correspondances erronées sont détectées et supprimées. Les distances entre primitives dans l'espace des descripteurs ne sont pas représentatives de la qualité des correspondances : une fausse correspondance peut très bien avoir une faible distance en cas de motifs répétitifs ou même de similitudes inattendues au niveau du descripteur (par exemple, avec le descripteur HOG Vondrick *et al.*, 2013, voir figure 5.5).

Un filtrage efficace nécessite une information supplémentaire, habituellement le modèle de déformation attendu : les correspondances erronées sont considérées



Cible	Modèle	FBDSD	RANSAC
331	111 vraies	43 faux pos.	143 faux pos.
correspondances	(< 2px)	86 faux neg.	2 faux neg.

FIGURE 5.6 – Comparaison des méthodes de filtrage explicite de correspondances FBDSD (Pizarro et Bartoli, 2012) et RANSAC (Tran *et al.*, 2012). Soit il manque de vraies correspondances, soit il en reste des fausses. Voir texte en section 5.2.2.

purement aléatoires et donc incohérentes avec le modèle. Une des méthodes les plus utilisées est RANSAC (*RANdom SAmple Consensus*, Fischler et Bolles, 1981) qui consiste en les étapes suivantes :

1. sélection aléatoire d'un nombre minimal de correspondances pour contraindre le modèle,
2. ajustement du modèle à ces correspondances,
3. énumération des correspondances cohérentes (*inliers*) avec la déformation estimée, retour à l'étape 1 si ce nombre est inférieur à un seuil prédéfini,
4. raffinement des paramètres du modèle avec tous les *inliers*.

Par exemple Tran *et al.* (2012) l'appliquent aux filtrages de correspondances pour l'alignement de surfaces déformables. D'autres méthodes plus spécifiques existent également comme Pizarro et Bartoli (2012).

La grande difficulté de ces méthodes est le réglage de la sensibilité des classificateurs pour trouver le bon équilibre entre faux positifs (correspondances erronées acceptées) et faux négatifs (correspondances correctes rejetées). Ce dilemne est illustré en figure 5.6 comparant deux méthodes de filtrage explicite sur un cas synthétique, avec des correspondances de points SIFT classifiées en tant que correctes si elles sont à moins de deux pixels de la vérité terrain.

Filtrage implicite

Le filtrage implicite consiste à diminuer l'influence des correspondances erronées au cours de l'optimisation de manière progressive, sans classification *inliers*/*outliers* binaire. Brox et Malik (2011, LDOF) pondèrent chaque correspondance par une

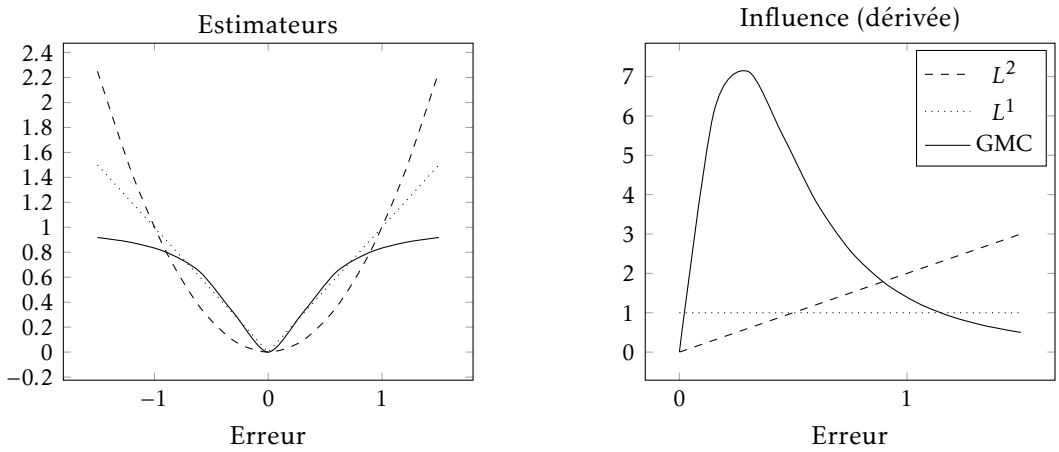


FIGURE 5.7 – Comparaison de trois estimateurs et de leur influence : les moindres carrés - - - donnent un poids plus important aux données erronées (erreur importante) qu'aux données correctes, la norme L^1 ····· donne la même influence à toutes les données, alors que l'estimateur redescendant de Geman McClure —— réduit l'influence des données erronées. On appelle ce processus filtrage implicite.

mesure de confiance basée sur la distance entre les deux plus proches voisins. Cependant, comme expliqué plus haut, il est insuffisant de considérer uniquement les descripteurs pour le filtrage. Un véritable filtrage implicite est habituellement basé sur l'utilisation de M-estimateurs, non convexes et *redescendants*, c'est à dire dont l'influence (la dérivée) croît puis décroît en fonction de l'erreur (voir figure 5.7). La pseudo-norme L^1 utilisée par LDOF, convexe, empêche les correspondances erronées d'avoir plus d'influence que les correspondances correctes mais ne leur en donne pas moins. C'est probablement une des raisons principales qui limitent LDOF à des images proches.

Pilet *et al.* (2008) utilisent un estimateur redescendant spécifique pour leur méthode d'alignement de surfaces déformables. Sa sélectivité est graduellement augmentée durant l'optimisation pour un comportement similaire au recuit simulé. Il est à noter la différence avec LDOF qui présente aussi un comportement similaire au recuit simulé (section 5.2.1) : Pilet *et al.* (2008) optimisent le filtrage des correspondances erronées alors que Brox et Malik (2011) optimisent la fusion entre les termes épars et dense. Les deux approches sont en fait complémentaires et il est avantageux d'utiliser les deux comme nous le verrons en section 5.3.2.

(a) I_1 (b) I_2

FIGURE 5.8 – Deux vues d'une scène rigide, issues du jeu de donnée de Tola *et al.* (2010), avec correspondances de segments de droite (Wang *et al.*, 2009a). Cette paire sera utilisée pour illustrer nos contributions au long de la section 5.3.

5.3 Méthode proposée

Nous proposons une approche pour améliorer la plupart des méthodes d'estimation de champ de déplacements qui minimisent une fonction de coût avec au moins un terme de données dense tel que défini par l'équation (2.80) :

$$C_{\text{base}}(\boldsymbol{u}, I_1, I_2) = \iint_{\Omega_{I_1}} D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\, d\boldsymbol{q}. \tag{5.3}$$

La cohérence spatiale du champ estimé doit être assurée, que ce soit par un modèle paramétrique (voir section 2.7.3) ou non paramétrique (voir section 2.7.2).

Deux contributions sont apportées pour permettre de grandement élargir le bassin de convergence. D'abord, pour que le coût dense fonctionne avec des images éloignées, nous proposons une gestion explicite et complète des occultations. Ensuite, nous introduisons un nouveau terme basé correspondances de primitives, compatible avec des primitives non ponctuelles (segments de droite), à coordonnées non entières, avec un filtrage implicite des correspondances erronées.

La paire d'images de la figure 5.8, avec des correspondances de segments, sera utilisée au long de cette section pour illustrer les différents aspects de la méthode proposée.

5.3.1 Occultations

L'estimation de déformations importantes nécessite la gestion des occultations, comme l'illustre la figure 5.9. Les trois types d'occultations (voir section 2.7.4) :

(a) Occultations (b) Sans (c) Avec

FIGURE 5.9 – Importance de la gestion des auto-occultations. En (a) : occultations calculées sur la paire d'image de la figure 5.8 (auto-occultations en vert et occultations externes en rouge). Comparaison de cartes de profondeurs calculées sans (b) et avec (c) gestion des occultations.

auto-occultations, occultations externes et bordure de l'image doivent être prises en compte. Pour cela le terme dense C_{base} de la fonction de coût est intégré dans un nouveau terme :

$$C^{\star}_{\text{base}}(\boldsymbol{u}, I_1, I_2) = \iint_{\Omega_{I_1}} D^{\star}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\, d\boldsymbol{q} \tag{5.4}$$

où

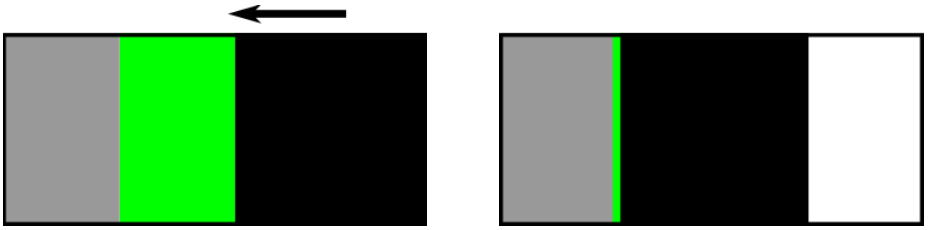
$$D^{\star}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) = \mathcal{P}_{\theta_s}(\boldsymbol{q}, \boldsymbol{u}) \cdot \delta_{\Omega_{I_2}}(\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q})) \cdot \min(\theta_e, D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)) \tag{5.5}$$

avec D le terme de données de base, $\mathcal{P}_{\theta_s}$ la probabilité de chaque pixel d'être auto-occulté, $\delta_{\Omega_{I_2}}$ la fonction indicatrice du domaine de l'image I_2 et θ_e un seuil pour diminuer la sensibilité aux occultations externes. Toutes ces additions seront détaillées par la suite.

Auto-occultations

Les pixels de I_2 occultés dans I_1 ne posent pas de problème : ils peuvent simplement rester sans correspondance. Nous considérons ici le cas inverse : des pixels de I_1 occultés dans I_2, plus gênant car le champ de correspondances est défini sur tout le domaine Ω_{I_1}.

Les auto-occultations sont caractérisées par le fait que tous les pixels occultants ont des correspondances. Les pixels occultés dans I_1 n'ont donc pas de correspondances mais leurs voisins si, ce qui contraint le champ de correspondances estimé sur la frontière d'occultation. Le phénomène est illustré schématiquement sur la figure 5.10. En d'autres termes, la dérivée du champ de correspondances



(a) I_1 (b) I_2

FIGURE 5.10 – Illustration d'une auto-occultation. Quand le carré noir se déplace vers la gauche, les zones grise et noire peuvent être mises en correspondances. Par conséquent, les correspondances des pixels de la zone verte occultée sont contraintes sur la frontière d'occultation, la dérivée du champ de correspondances y est donc nulle. La zone blanche reste sans correspondance.

s'annule [1] dans la direction orthogonale à la frontière d'occultation (Gay-Bellile *et al.*, 2010). Cette observation permet de développer un test fiable pour détecter les auto-occultations : nous présentons ci-dessous notre adaptation de la méthode de Gay-Bellile *et al.*.

Soit le champ de correspondances $\mathbf{a} : \boldsymbol{q} \mapsto \boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q})$, le pixel $\boldsymbol{q}$ est occulté par auto-occultation si et seulement si :

$$\begin{aligned} \exists \mathbf{d} \in \mathbb{R}^2, \|\mathbf{d}\| = 1 \quad \text{t.q.} \quad & \nabla_{\mathbf{d}} \mathbf{a}(\boldsymbol{x}) = 0 \quad \textit{la dérivée s'annule} \\ \Leftrightarrow \quad & \nabla_{\mathbf{d}} \boldsymbol{u}(\boldsymbol{q}) = -\mathbf{d} \end{aligned} \tag{5.6}$$

où $\nabla_{\mathbf{d}} \boldsymbol{u}(\boldsymbol{q}) \approx \frac{\boldsymbol{u}(\boldsymbol{q}+\mathbf{d}) - \boldsymbol{u}(\boldsymbol{q}-\mathbf{d})}{2}$ est la dérivée partielle, basée sur des différences centrales, dans la direction $\mathbf{d}$. La plus petite dérivée partielle au carré, σ_0 est liée à la jacobienne $\mathbf{J}$ du champ de correspondance par :

$$\sigma_0(\boldsymbol{q}, \boldsymbol{u}) = \min_{\|\mathbf{d}\|=1} \mathbf{d}^\top \mathbf{J}(\boldsymbol{q}, \boldsymbol{u})^\top \mathbf{J}(\boldsymbol{q}, \boldsymbol{u}) \mathbf{d} \tag{5.7}$$

où

$$\mathbf{J}(\boldsymbol{q}, \boldsymbol{u}) = \begin{pmatrix} \frac{\partial a_x(\boldsymbol{q})}{\partial x} & \frac{\partial a_x(\boldsymbol{q})}{\partial y} \\ \frac{\partial a_y(\boldsymbol{q})}{\partial x} & \frac{\partial a_y(\boldsymbol{q})}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial u_x(\boldsymbol{q})}{\partial x} + 1 & \frac{\partial u_x(\boldsymbol{q})}{\partial y} \\ \frac{\partial u_y(\boldsymbol{q})}{\partial x} & \frac{\partial u_y(\boldsymbol{q})}{\partial y} + 1 \end{pmatrix}. \tag{5.8}$$

1. Cette propriété n'est vraie que sous l'hypothèse que le champ de correspondance est continu dans la zone occultée. Une régularisation au premier ordre telle que la variation totale (voir section 2.7.2) peut s'opposer à ce phénomène en générant un champ de déplacement « en escalier ». Toutefois, les implémentations mises en œuvres en section 5.4 et 5.5 ne sont pas affectées par ce problème.

Il vient donc naturellement que σ_0 est la plus petite valeur singulière de $\mathbf{J}(\boldsymbol{q}, \mathbf{a})$, c'est à dire la plus petite valeur propre de $\mathbf{O}(\boldsymbol{q}, \boldsymbol{u}) = \mathbf{J}(\boldsymbol{q}, \boldsymbol{u})^\top \mathbf{J}(\boldsymbol{q}, \boldsymbol{u})$. Les valeurs propres sont les racines du polynôme caractéristique :

$$\begin{aligned} \sigma \text{ valeur propre} \quad &\Leftrightarrow \quad \det\left(\mathbf{O}(\boldsymbol{q}, \boldsymbol{u}) - \sigma \mathbf{I}\right) = 0 \\ &\Leftrightarrow \quad \sigma^2 - \sigma(O_{11} + O_{22}) + O_{11}O_{22} - O_{12}^2 \quad = 0 \end{aligned} \tag{5.9}$$

ayant pour solutions :

$$\sigma = \frac{1}{2}\left(O_{11} + O22 \pm \sqrt{(O_{11} - O_{22})^2 + 4O_{12}^2}\right) \tag{5.10}$$

où les O_{ij} sont les coefficients de $\mathbf{O}(\boldsymbol{q}, \boldsymbol{u})$ avec la dépendance en $\boldsymbol{q}$ et $\boldsymbol{u}$ cachée pour une meilleure lisibilité. La matrice $\mathbf{O}(\boldsymbol{q}, \boldsymbol{u})$ est symétrique définie positive donc ses valeurs propres sont positives et la plus petite valeur propre vaut :

$$\sigma_0(\boldsymbol{q}, \boldsymbol{u}) = \frac{1}{2}\left(O_{11} + O_{22} - \sqrt{(O_{11} - O_{22})^2 + 4O_{12}^2}\right). \tag{5.11}$$

Un seuil θ_s et une fonction « en S » d'interpolation cubique (voir figure 5.11) :

$$\mathcal{S}(x) = \begin{cases} 3x^2 - 2x^3 & \text{si } 0 \leq x \leq 1 \\ 1 & \text{sinon} \end{cases} \tag{5.12}$$

permettent d'associer σ_0 à une probabilité de non occultation :

$$\mathcal{P}_{\theta_s}(\boldsymbol{x}, \boldsymbol{u}) = \mathcal{P}_{\theta_s}(\boldsymbol{x} \text{ non occulté} \mid \boldsymbol{u}) = \mathcal{S}\left(\frac{\sigma_0(\boldsymbol{x}, \boldsymbol{u})}{\theta_s}\right). \tag{5.13}$$

Occultations externes

Les occultations externes sont causées par la présence d'un élément occultant dans une seule des deux images. Contrairement aux auto-occultations, aucune hypothèse ne peut être faite sur le comportement du champ de correspondances dans ces zones. Nous les considérons donc comme données aberrantes, comme d'autres défauts éventuels tels qu'un niveau important de bruit. Nous appliquons un seuillage de la fonction de coût, de manière similaire à Ochs *et al.* (2014) pour réduire l'influence de ces données aberrantes. Dans nos expériences, la présence du seuil améliore la robustesse mais sa valeur θ_e, au delà de laquelle les données sont considérées aberrantes, est peu sensible : nous fixons expérimentalement $\theta_e = 0.5$.

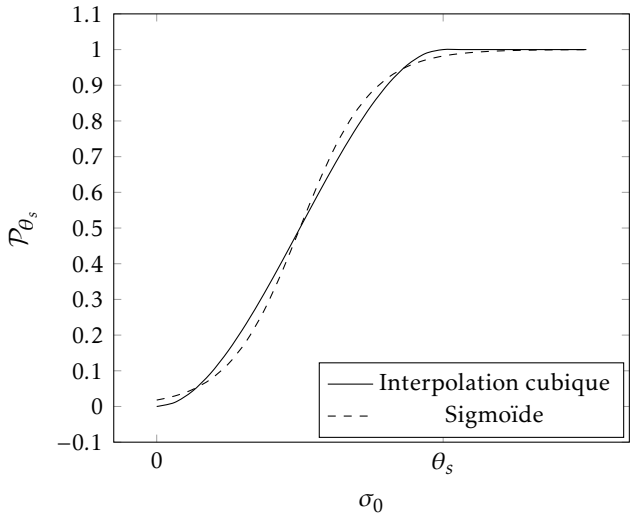


FIGURE 5.11 – Fonction d'interpolation pour exprimer la probabilité de non auto-occultation en fonction de la plus petite valeur singulière σ_0 de la matrice jacobienne du champ de correspondances. Notre interpolation cubique ne possède qu'un seul paramètre alors que la sigmoïde utilisée par Gay-Bellile *et al.* (2010), d'équation $\frac{\exp(2k(\sigma_0-r))}{1+\exp(2k(\sigma_0-r))}$, en nécessite deux ($k$ et r). Voir section 5.3.1 pour les détails.

Bordure de l'image

Le champ de déplacements ne peut être estimé pour les pixels de I_1 dont les correspondances sont « en dehors » du domaine de I_2. Pour de petits déplacements, il peut suffire de considérer que tous les pixels à l'extérieur de Ω_{I_2} sont de valeur nulle (Wedel *et al.*, 2009), mais cette approximation peut introduire des erreurs. Une meilleure solution est d'ignorer le terme de données pour ces points, ce qui est effectué grâce à la fonction indicatrice suivante :

$$\delta_{\Omega_{I_2}}(\boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q})) = \begin{cases} 1 & \text{si } \boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q}) \in \Omega_{I_2} \\ 0 & \text{sinon} \end{cases}. \tag{5.14}$$

Pour les termes denses considérant un voisinage (avec une distance Census par exemple, voir section 2.6.2), la fonction indicatrice vaut zéro dès qu'un des points du voisinage sort du domaine Ω_{I_2}. Nous préférons toutefois laisser ce détail hors de la formulation (5.14) pour une meilleure lisibilité.

5.3.2 Terme basé correspondances proposé

D'après les observations de l'état de l'art présentées en section 5.2, nous dérivons un nouveau terme basé correspondances éparses. Nous adoptons la méthode de densification par multi-résolution (section 5.2.1) et un filtrage implicite des correspondances erronées par l'estimateur de Geman-McClure $\Psi_\sigma(x) = \frac{x^2}{x^2+\sigma}$. Nous proposons également des contributions supplémentaires pour gérer les primitives non ponctuelles et à cordonnées non entières (telles que les segments).

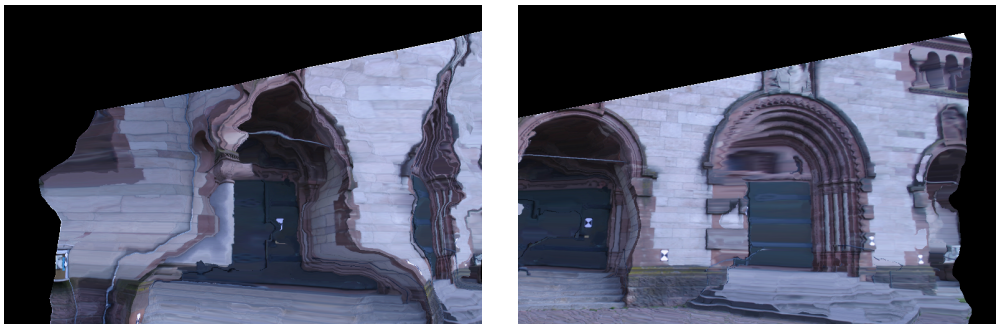
Soit un ensemble $\mathcal{F} = \left\{(\mathbf{f}_1^{(1)}, \mathbf{f}_2^{(1)}), \dots (\mathbf{f}_1^{(n)}, \mathbf{f}_2^{(n)})\right\}$ de n correspondances de primitives, notre terme basé correspondance est :

$$C_{\text{corr.}}(\boldsymbol{u}, \mathcal{F}) = \iint_{\Omega_{I_1}} \sum_{(\mathbf{f}_1, \mathbf{f}_2) \in \mathcal{F}} F(\boldsymbol{q}, \boldsymbol{u}, \mathbf{f}_1, \mathbf{f}_2)\, d\boldsymbol{q} \tag{5.15}$$

où

$$F(\boldsymbol{q}, \boldsymbol{u}, \mathbf{f}_0, \mathbf{f}_1) = \rho(\boldsymbol{q}, \mathbf{f}_0) \Psi_\sigma(\Delta(\boldsymbol{q}, \boldsymbol{u}(\boldsymbol{q}), \mathbf{f}_1)) \tag{5.16}$$

est le terme de données par correspondance. La fonction d'influence ρ et la distance Δ sont détaillérs dans les paragraphes suivants. La figure 5.12 illustre la nécessité des correspondances de primitives pour que l'estimation du champ de déplacements converge en présence de déformations importantes.



(a) sans correspondances éparses (b) avec correspondances éparses

FIGURE 5.12 – Alignement de la paire d'image de la figure 5.8 : image I_2 alignée avec I_1 en appliquant le champ de déplacements calculé : sans (a) et avec (b) le terme basé correspondances (voir section 5.3.2).

Distance point-primitive

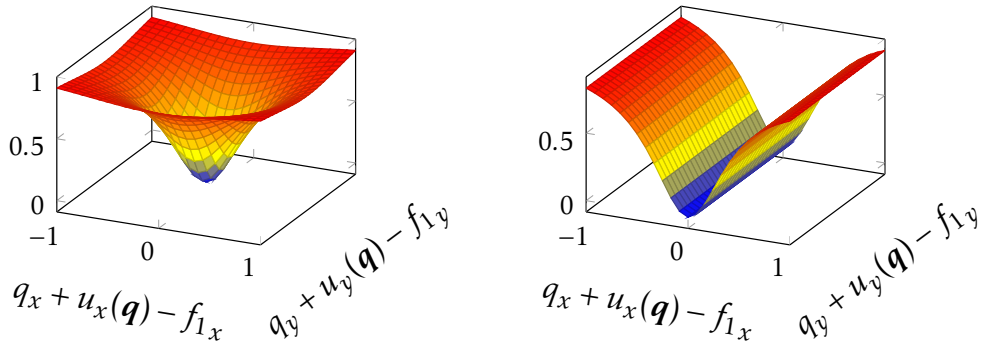
Un problème non abordé dans la littérature est la prise en charge de primitives non ponctuelles. En effet, soit une primitive $\mathbf{f}_1$ de I_1 correspondant à la primitive $\mathbf{f}_2$ de I_2, la plupart des méthodes (Brox et Malik, 2011; Xu *et al.*, 2010) commencent par extraire le déplacement associé : $\mathbf{f}_1 - \mathbf{f}_2$. Le problème de cette formulation est qu'elle empêche de considérer des primitives non ponctuelles, qui ne peuvent être associées à un mouvement unique. C'est pourquoi nous adoptons une approche plus générique en considérant directement la distance point-primitive $\Delta(\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q}), \mathbf{f}_2)$ entre les correspondances $\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q})$ des pixels affectés par $\mathbf{f}_1$ et la primitive $\mathbf{f}_2$ correspondante.

Nous dérivons ici deux distances, pour les points et les segments, représentées sur la figure 5.13. La distance appropriée pour les points est la distance euclidienne classique. Soit un point $\mathbf{f} = \boldsymbol{q}_f$:

$$\Delta_{\text{point}}(\boldsymbol{q}, \mathbf{f}) = \|\boldsymbol{q} - \boldsymbol{q}_f\|. \tag{5.17}$$

Les segments, pour leur part, doivent correspondre à la même droite mais aucune correspondance ponctuelle (de leurs extrémités par exemple) n'est garantie. D'ailleurs, en raison de la présence d'occultations et de déformations perspectives c'est rarement le cas. La distance appropriée est donc la distance orthogonale point-à-ligne, qui ne contraint qu'une seule dimension. Soit un segment défini par ses extrémités : $\mathbf{f} = (\boldsymbol{q}_{f_b}, \boldsymbol{q}_{f_e})$

$$\Delta_{\text{segment}}(\boldsymbol{q}, \mathbf{f}) = \frac{\|(\boldsymbol{q}_{f_e} - \boldsymbol{q}_{f_b}) \times (\boldsymbol{q} - \boldsymbol{q}_{f_b})\|}{\|\boldsymbol{q}_{f_e} - \boldsymbol{q}_{f_b}\|}. \tag{5.18}$$



(a) Points (b) Segments de droites

FIGURE 5.13 – Distances point-primitive avec l'estimateur de Geman McClure : $\Psi_\sigma \circ \Delta$. Les graphes représentent les distances pour un pixel $\boldsymbol{q} = (q_x, q_y) \in \Omega_{I_1}$, associé à un déplacement $\boldsymbol{u}(\boldsymbol{q}) = (u_x(\boldsymbol{q}), u_y(\boldsymbol{q}))$ et deux primitives en correspondance : $(\mathbf{f}_0, \mathbf{f}_1)$.

Fonction d'influence

La plupart des primitives ont une précision de localisation inférieure au pixel. La fonction d'influence ρ traduit cette propriété grâce à une interpolation bilinéaire. Soit une primitive $\mathbf{f}$ située en $\boldsymbol{q}_\mathbf{f} = \boldsymbol{q}_{0_\mathbf{f}} + \boldsymbol{dq}$ avec $\boldsymbol{q}_{0_\mathbf{f}} = E(\boldsymbol{q}_f)$ la partie entière de $\boldsymbol{q}_f$. On définit $\overline{\boldsymbol{dq}} = (1,1)^T - \boldsymbol{dq}$ et la fonction d'influence intermédiaire ρ'_i pour les quatre pixels voisins :

$$\begin{aligned} \rho'\left(\boldsymbol{q}_{0_\mathbf{f}}, \mathbf{f}\right) &= \overline{\boldsymbol{dq}}_x \overline{\boldsymbol{dq}}_y & \rho'\left(\boldsymbol{q}_{0_\mathbf{f}} + (0,1)^T, \mathbf{f}\right) &= \overline{\boldsymbol{dq}}_x \boldsymbol{dq} \\ \rho'\left(\boldsymbol{q}_{0_\mathbf{f}} + (1,0)^T, \mathbf{f}\right) &= \boldsymbol{dq}_x \overline{\boldsymbol{dq}}_y & \rho'\left(\boldsymbol{q}_{0_\mathbf{f}} + (1,1)^T, \mathbf{f}\right) &= \boldsymbol{dq}_x \boldsymbol{dq}_y. \end{aligned} \tag{5.19}$$

Les segments de droite sont d'abord discrétisés en un ensemble de points espacés de 1 pixel (voir figure 5.15), et l'influence de chaque point est calculée comme ci-dessus pour produire une représentation lissée du segment. Cependant, cela donne un poids disproportionné aux longs segments, alors qu'ils ne sont pas nécessairement plus fiables. Pour résoudre ce problème, la fonction d'influence est normalisée par la longueur de telle sorte que la somme des poids soit unitaire pour toutes les primitives. Soit $l(\mathbf{f})$ la longueur de la primitive $\mathbf{f}$ en pixels :

$$\rho(\boldsymbol{q}, \mathbf{f}) = \frac{\rho'(\boldsymbol{q}, \mathbf{f})}{l(\mathbf{f})} \quad \forall \boldsymbol{q} \in \Omega_{I_1}. \tag{5.20}$$

Les primitives ponctuelles peuvent être considérées comme ayant une longueur de 1 pixel pour généraliser cette formulation de la fonction d'influence. Des correspon-

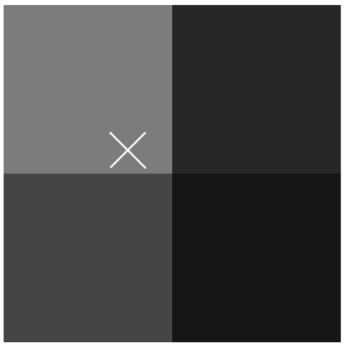


FIGURE 5.14 – Exemple de fonction d'influence ρ. La croix représente la position d'une primitive ponctuelle. Les niveaux de gris sont proportionnels à l'influence exercée sur chaque pixel.

dances de segments, quelles que soient leur longueurs peuvent donc être mélangées avec des correspondances ponctuelles avec la même influence.

Comparaison avec l'état de l'art

Dans cette section, nous mettons en évidence, les différences et nos contributions par rapport aux deux méthodes de l'état de l'art qui ont des aspects similaires à notre approche.

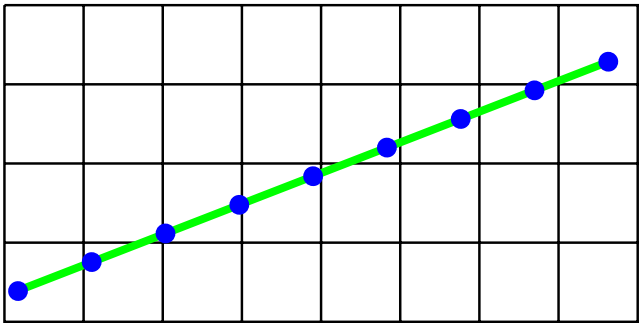


FIGURE 5.15 – Discrétisation des segments pour la fonction d'influence : ils sont considérés comme des points régulièrement espacés.

Pilet *et al.* (2008) effectuent eux aussi un filtrage implicite des correspondances erronées, dans le contexte de l'alignement de surfaces déformables. Leur approche est purement basée correspondances éparses, ce qui la rend plus rapide, mais néglige une grande quantité d'information. Dans notre approche, le terme basé correspondances éparses ne sert qu'à guider le terme dense jusqu'à convergence, ce qui permet d'utiliser toute l'information de l'image pour une précision optimale. De plus, l'estimateur qu'ils utilisent a une dérivée nulle au delà d'un certain seuil [2] ce qui peut empêcher la convergence dans le cas de conditions initiales défavorables. Enfin, l'optimisation multi-résolution nous permet d'utiliser un M-estimateur avec une sélectivité constante alors que Pilet *et al.* (2008) doivent la modifier manuellement au cours du processus, rajoutant un paramètre à l'algorithme.

La publication de Brox et Malik (2011, LDOF) nous a inspiré l'exploitation de l'optimisation multi-résolution pour « densifier » le terme épars. Cependant, notre approche lève de nombreuses restrictions présentes dans LDOF. Premièrement, l'emploi d'un estimateur robuste permet un filtrage implicite sans besoin de calculer un score de confiance pour chaque correspondance. Cela est préférable pour deux raisons : ces scores sont peu fiables en pratique, et de plus parfois indisponibles par les implémentations publiques de descripteurs de primitives. Notre approche assure une compatibilité maximale. Par ailleurs, LDOF ne prévoit pas la gestion de primitives à coordonnées non entières, contrairement à la fonction d'influence bilinéaire que nous proposons.

Enfin, aucune publication de la littérature n'a jusqu'ici, à notre connaissance, abordé le sujet des primitives non ponctuelles. Notre formulation en terme d'une distance point-primitive permet d'adapter aisément la méthode à différentes primitives, ce qui est démontré par la prise en charge de segments de droite. Rien n'empêche *a priori* la généralisation à d'autres primitives (régions, courbes…) si ce n'est la disponibilité de descripteurs adaptés.

La validité et la flexibilité de notre approche est démontrée par sa mise en œuvre au sein de deux méthodes : une suivant un modèle paramétrique et une suivant un modèle non paramétrique.

Algorithme 1 : Résumé de l'implémentation avec modèle non paramétrique.

Données : Images I_1 et I_2, correspondances de primitives $\mathcal{F}$, terme de données dense D_{base}
Résultat : Champ de déplacement $\boldsymbol{u} : \Omega \mapsto \mathbb{R}^2$

pour chaque *niveau de multi-résolution* **faire**
  pour *w itérations* **faire** alignement de I_2 vers I_1
    pour chaque *pixel* $\boldsymbol{q} \in \Omega_{I_1}$ **faire**
      /* occlusions externes (section 5.3.1) */
      $D^{\star}_{\text{base}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \leftarrow \min(D_{\text{base}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2), \theta_e)$
      /* bordure de l'image (section 5.3.1) */
      $D^{\star}_{\text{base}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \leftarrow 0$ si $\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \notin \Omega_{I_2}$
      /* gestion des auto-occultations (section 5.3.1) */
      $D^{\star}_{\text{base}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \leftarrow \mathcal{P}_{\theta_s}(\boldsymbol{q}, \boldsymbol{u}) \cdot D^{\star}_{\text{base}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)$
      /* ajout du terme basé correspondances (section 5.3.2) */
      $D_{\text{tout}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2, \mathcal{F}) \leftarrow D^{\star}_{\text{base}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) + \sum_{(\mathbf{f}_1, \mathbf{f}_2) \in \mathcal{F}} F(\boldsymbol{q}, \boldsymbol{u}, \mathbf{f}_1, \mathbf{f}_2)$
      linéarisation de D_{tout} par rapport à $\boldsymbol{u}$
    fin
    pour *i itérations* **faire** optimisation convexe et régularisation TGV^2
    passage à la résolution supérieure
  fin
fin

5.4 Implémentation et résultats avec un modèle non paramétrique

Notre implémentation non paramétrique (voir algorithme 1), conçue pour être la plus générique possible, est basée sur un terme de données dense de type Census (voir section 2.6.2) et une régularisation variation totale généralisée (voir section 2.7.2) au deuxième ordre :

$$C_{\text{non-param}}(\boldsymbol{u}, I_1, I_2) = \lambda \iint_{\Omega_{I_1}} D_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\, d\boldsymbol{q} + \mu C_{\text{corr.}}(\boldsymbol{u}, \mathcal{F}) + R_{\text{TGV}^2}(\boldsymbol{u}, \alpha_0, \alpha_1), \quad (5.21)$$

où $\lambda \in \mathbb{R}$ et $\mu \in \mathbb{R}$ sont les poids respectifs des termes de données dense et basé correspondances éparses.

2. L'estimateur de Pilet *et al.* (2008) est défini par : $\Psi_r(x) = \begin{cases} \frac{3(r^2 - x^2)}{4r^3} & \text{si } x^2 < r^2 \\ 0 & \text{sinon} \end{cases}$.

5.4.1 Détails d'implémentation

Optimisation

L'algorithme de Chambolle et Pock, utilisant une approche primale-duale (Chambolle et Pock, 2011) avec préconditionnement (Pock et Chambolle, 2011), est utilisé pour l'optimisation de la fonction de coût (5.21). L'algorithme est résumé en annexe 5.B et les détails pratiques de l'implémentation sont référencés dans les publications de Bredies (2012) et Ranftl *et al.* (2012). Les opérateurs différentiels discrétisés sont repris en annexe 5.A. À noter qu'il s'agit d'un algorithme itératif et que le critère d'arrêt adopté est un nombre fixe d'itérations, noté i.

Linéarisation

L'algorithme de Chambolle et Pock est dédié à l'optimisation d'une fonction de coût convexe. Or, le terme de données Census utilisé, de même que tous les autres termes de données dense définis en section 2.6.2 dépendent de l'image et ne peuvent pas être supposés convexes. Pour de très faibles déplacements, de l'ordre du pixel, une approximation convexe locale est toutefois possible. Concrètement, le terme de données est linéarisé au début de w itérations appelées *alignements*, (*warps* en anglais). Cette approximation n'est valide que proche du point de linéarisation, nous restreignons donc le champ de déplacements optimisé à un rayon r autour du point de linéarisation. De manière similaire à Werlberger (2012), le rayon est initialisé à une valeur de un pixel, et divisé par un facteur 1.2 à chaque alignement pour limiter les oscillations dûes aux non-convexités.

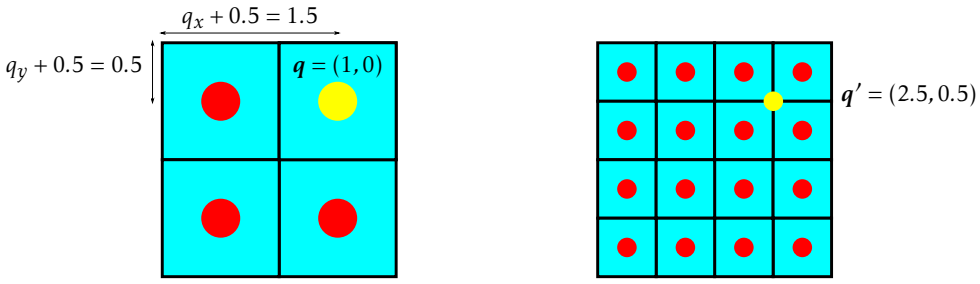
Multi-résolution

Les w alignements décrits ci-dessus sont répétés pour chaque niveau d'une pyramide d'image, de la plus basse à la plus haute résolution. Les images originales sont redimensionnées par interpolation linéaire d'un facteur $s \in [0.5, 1[$. Nous utilisons le redimensionnement anisotropique suggéré par Sun *et al.* (2014) (voir figure 5.16) : le facteur de redimensionnement est réduit pour la dimension la plus petite de l'image de telle sorte que le dernier niveau de la pyramide soit un carré de 4 par 4 pixels.

Les redimensionnements, que ce soit pour réduire la résolution lors de la construction de la pyramide d'images ou pour passer le champ de déplacements à la résolution supérieure durant l'optimisation, sont basés sur une interpolation linéaire. Pour une image originale I, l'image redimensionnée I_{redim} est définie par :

$$\forall \boldsymbol{q} \in \Omega_{I_{\text{redim}}} : \quad I_{\text{redim}}(\boldsymbol{q}) = I(\boldsymbol{q}') \tag{5.22}$$

FIGURE 5.16 – Redimensionnement anisotropique : l'image à la plus basse résolution est carrée.



(a) Image redimensionnée (b) Image originale

FIGURE 5.17 – Redimensionnement bilinéaire pour un facteur $s_x = s_y = 2$. L'image originale est considérée comme à valeurs continues par interpolation bilinéaire. Les coordonnées du point $\boldsymbol{q}'$ sont calculées d'après l'équation (5.23). Les coordonnées entières sont considérées comme le centre de pixels carrés, d'où la compensation de 0.5 pixels.

où les nouvelles coordonnées sont (voir figure 5.17) :

$$q'_x = s_x \cdot (q_x + 0.5) - 0.5 \qquad q'_y = s_y \cdot (q_y + 0.5) - 0.5 \tag{5.23}$$

avec s_x et s_y les facteurs de redimensionnement pour les dimensions horizontale et verticale respectivement.

Jeu de paramètres

La mise au point de cette méthode a toujours été soumise à un objectif de généricité et robustesse. Le même soin a été pris lors du choix des paramètres, listés dans le tableau 5.1. Même s'ils sont nombreux, seulement deux paramètres : λ et θ_s, sont modifiés au cours de toutes les expériences.

Le nombre d'itérations w et i ainsi que le facteur de redimensionnement sont repris de Ranftl *et al.* (2012). Le terme de données Census 3×3 est préféré pour une meilleure robustesse aux distorsions (voir section 2.6.2). Ensuite, on optimise d'abord les paramètres α_0, λ, θ_e et θ_s, sans utiliser de correspondances éparses, puis uniquement les paramètres σ et μ avec des correspondances SIFT. Chaque

Tableau 5.1 – Paramètres utilisés pour la méthode non paramétrique de mise en correspondance.

	Petits déplacements	Stéréo	Non-rigide
Régularisation	TGV^2	—	—
α_0	1	—	—
α_1	1	—	—
Terme dense	Census 3×3	—	—
λ	6	30	1
θ_e	0.5	—	—
θ_s	0.2	0.5	—
Primitives	SIFT/ASIFT	Segments	SURF
μ	1	—	—
σ	0.2	—	—
Optimisation	Chambolle et Pock	—	—
s	0.8	—	—
w	20	—	—
i	40	—	—

Le symbole — est utilisé lorsque les paramètres sont identiques à ceux de la première colonne.

sous-ensemble des paramètres est optimisé selon le processus suivant, répété jusqu'à obtention d'un jeu de paramètres satisfaisants :

1. optimisation globale de l'erreur moyenne par essaims particulaires (Eberhart et Shi, 2000) sur les 20 premières images possédant une vérité terrain du jeu de données KITTI, redimensionnées par un facteur 0.3 pour une évaluation plus rapide,
2. arrondissement des paramètres à un seul chiffre significatif pour assurer une robustesse au surapprentissage,
3. test sur les différentes séquences avec vérité terrain, à pleine résolution.

La figure 5.7 permet de valider le choix de la sélectivité $\sigma = 0.2$ de l'estimateur robuste : les erreurs supérieures à un pixel ont une influence très limitée.

5.4.2 Expériences

Diverses expériences ont été réalisées, d'abord sur des jeux de données à faible déplacements pour évaluer quantitativement l'apport de chacune de nos contributions, puis sur des jeux de données plus difficiles pour démontrer l'élargissement

considérable du bassin de convergence de la méthode.

Faibles déplacements

Nous appelons faibles déplacements ceux qui peuvent être estimés de manière relativement fiable par des méthodes variationnelles classiques. Cette définition est plus large que celle employée dans Brox et Malik (2011, LDOF). En effet, même si LDOF reste une référence en terme de robustesse, les méthodes variationnelles actuelles obtiennent des performances similaires ou supérieures dans la plupart des cas.

Nos contributions ont pour but d'élargir le bassin de convergence et non d'augmenter la précision dans les cas où les méthodes variationnelles classiques convergent. Cependant, les paires d'images à faibles déplacements sont tout de même utiles pour valider le choix du terme de données dense et de la régularisation adoptés, ainsi que pour vérifier que l'ajout du terme basé correspondances éparses ne dégrade pas les résultats.

Trois jeux de données, dont une partie de la vérité terrain est gardée non publique pour une comparaison équitable, sont dominants pour l'évaluation des algorithmes de flot optique :

— Middlebury (Baker *et al.*, 2011) est composé de paires d'images avec de très faibles déplacements, constants par morceaux pour la plupart, et dans des conditions d'illuminations contrôlées et favorables ; il est au moment de la rédaction de ce mémoire encore utilisé pour tester certaines approches d'estimations de mouvement basées segmentation, mais est sinon remplacé par des jeux de données plus réalistes ;

— KITTI (Geiger *et al.*, 2012) est un jeu de données large (flot optique, stéréo, odométrie…) dédié à la vision pour les véhicules : les images sont acquises en conditions réelles grâce à une voiture instrumentée, dans des contextes différents (milieu urbain, route de campagne…) avec une grande variété des amplitudes des déplacements ainsi que des conditions d'illumination ;

— Sintel (Butler *et al.*, 2012b) est un jeu de données synthétique mais réaliste (issu du film d'animation libre Sintel [3]), comportant des perturbations importantes telles que : des mouvement rapides avec flou cinématique, du brouillard, de la fumée et des poussières.

Les figures 5.18 et 5.19 présentent une évaluation détaillée des différents composants mis en œuvre dans l'algorithme proposé. L'erreur moyenne en pixels est mesurée sur tout le jeu de données d'entrainement (avec vérité terrain) de Middlebury, et sur les 40 premières paires du jeu de données KITTI. Le jeu de donnée Sintel

3. http://www.sintel.org/

est composé de 23 séquences contenant 70 images en moyenne ; pour obtenir un sous-ensemble représentatif, deux images consécutives sont choisies aléatoirement dans chaque séquence. Pour chaque jeu de données, les régularisations variation totale (TV) et variation totale généralisée (TGV^2) sont comparées (voir section 2.7.2), ainsi que les différences absolues et Census pour le terme de données dense (voir section 2.6.2). La gestion des occultations proposée est également évaluée : seuil pour les occultations externes (section 5.3.1) et pondération pour les auto-occultations (section 5.3.1). Les images couleurs permettent enfin d'évaluer les différents espaces de couleurs (voir section 2.6.2).

Le jeu de données Middlebury comporte des déplacements majoritairement constants par morceaux, donc *a priori* plus adaptés à la régularisation par variation totale. Cependant, les résultats de la figure 5.18 peuvent paraître à première vue incohérents. Cela est dû au fait que dans ce jeu de données les déplacements sont de faible amplitude et les erreurs sont donc peu significatives, mais également au fait que la majorité des erreurs proviennent des contours d'occultations [4], qui ne sont pas pris en compte explicitement dans notre modèle. La régularisation TGV^2 a tendance a adoucir ces contours et produire des résultats « moyens » quel que soit le terme de données utilisé. La régularisation TV, elle, produit des contours très nets, mais est dépendante d'un terme de données discriminant pour qu'ils soient bien localisés. C'est pourquoi les meilleurs résultats sur ce jeu de données sont obtenus avec le couple TV + Census et les moins bons résultats avec le couple TV + AD. La figure 5.20 donne une illustration de ce phénomène.

Pour les jeux de données plus réalistes, où les champs de déplacements ne sont pas constants par morceaux, on observe que la combinaison TGV^2 et Census avec seuillage produit les meilleurs résultats. La grande influence du seuillage sur le jeu de donnée Sintel avec Census est dûe à la présence des nombreuses perturbations (brouillard, fumée, flou…) : la nature discrète du descripteur Census peut entraîner un fort gradient dans les régions perturbées et dégrader la carte de déplacements calculée. Le seuillage permet de traiter ces données comme aberrantes. Pour les images en couleurs la décomposition L*a*b* est, sans surprise, la plus adaptée.

Notre terme de données basé correspondances éparses est également évalué. Les correspondances SIFT ne permettent pas d'améliorer la précision sur le jeu de données Middlebury car les déplacements à estimer sont très faibles avec peu de minimums locaux. Par contre, le jeu de données KITTI, plus réaliste, est sujet à de tels minimums locaux et à ce titre, bénéficie d'une amélioration. Le jeu de données Sintel est lui aussi sujet à des minimums locaux lors de l'estimation, mais

4. On appelle contours d'occultations les discontinuités du champ de déplacements. De telles discontinuités sont en effet généralement observées quand un élément de l'image occulte un autre élément ayant un mouvement différent.

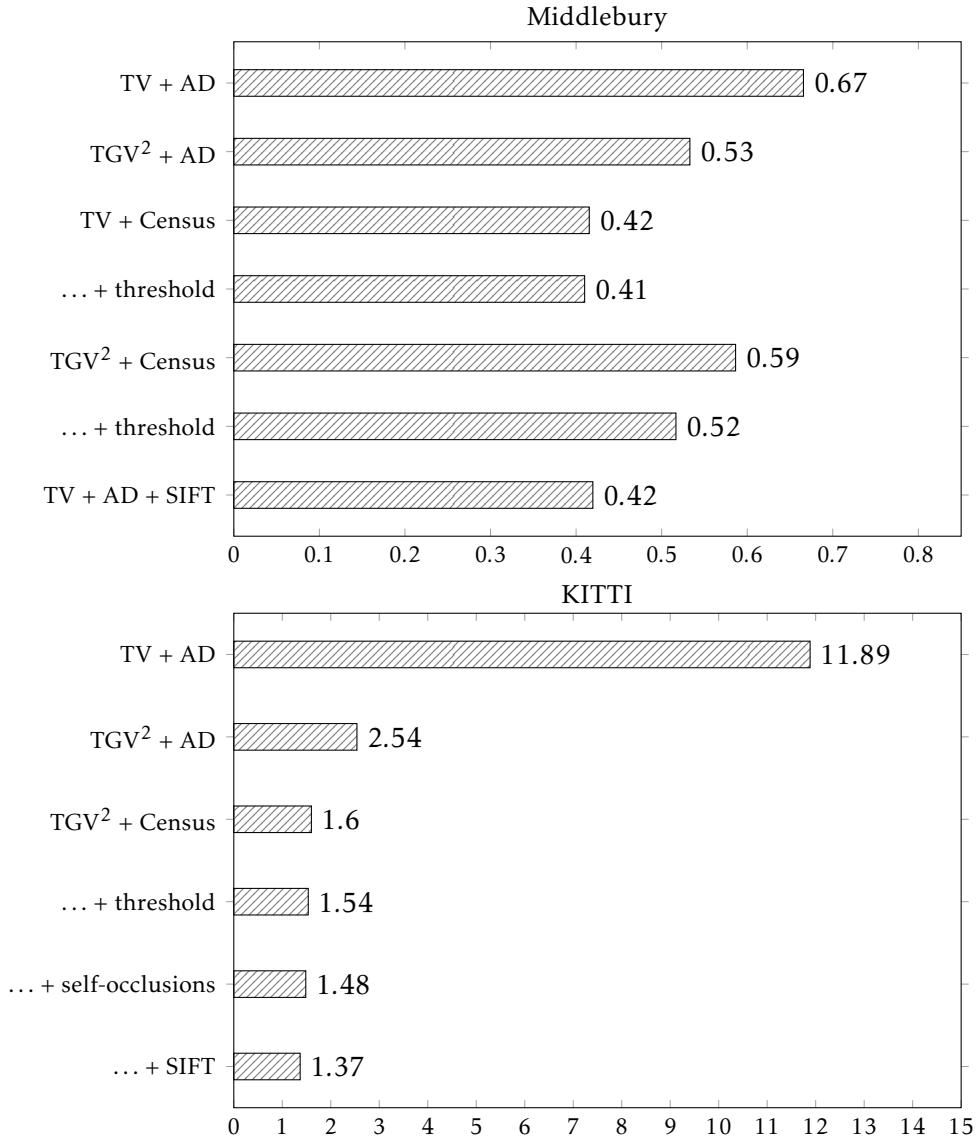


FIGURE 5.18 – Erreur moyenne en pixels de notre méthode sur les jeux de données Middlebury et KITTI avec différents termes de données : Différences Absolues (AD) et Census 3×3, et différentes régularisations : Variation Totale (TV) et Variation Totale Généralisée (TGV).

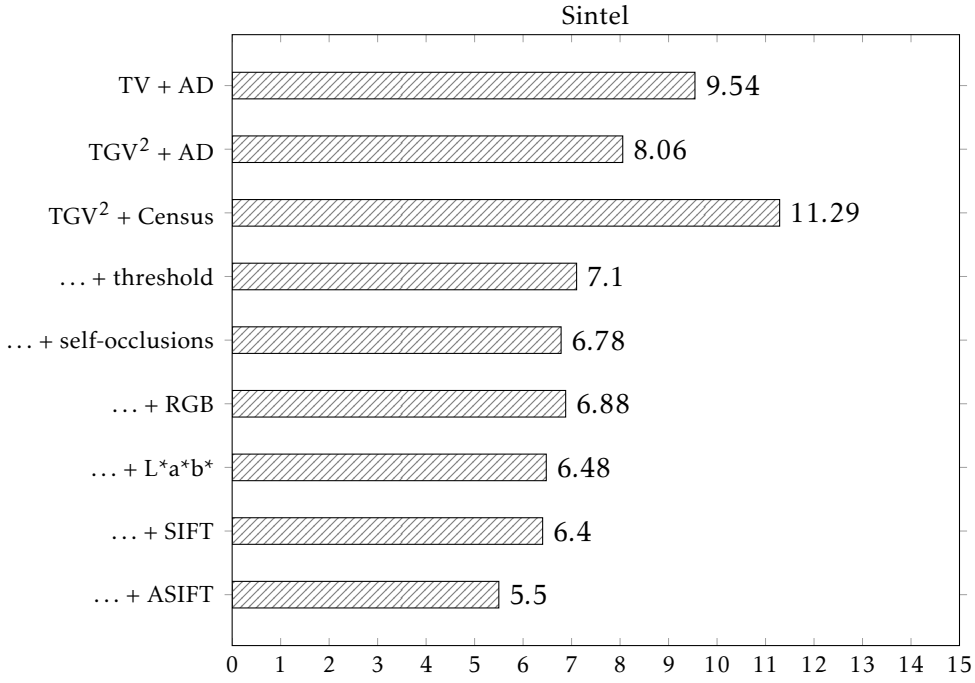
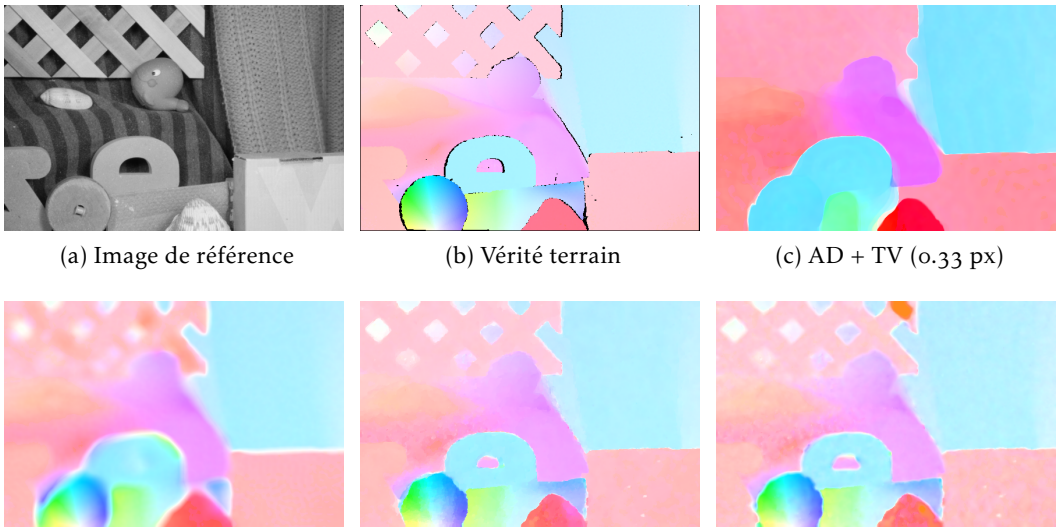


Figure 5.19 – Erreur moyenne en pixels de notre méthode avec différents termes de données et régularisations sur le jeu de données Sintel.



(a) Image de référence (b) Vérité terrain (c) AD + TV (0.33 px)

(d) AD + TGV (0.22 px) (e) Census + TV (0.11 px) (f) Census + TGV (0.13 px)

FIGURE 5.20 – Résultats qualitatifs et erreur moyenne en pixels sur la paire d'image RubberWhale du jeu de données Middlebury avec différents termes de données et régularisations. La régularisation Total Variation est plus adaptée à ces images, à condition que le terme de données utilisé soit assez discriminant pour bien estimer les contours d'occultations.

Paire d'images | 3447 corresp. ASIFT

vérité terrain | sans correspondances | avec correspondances

erreur moyenne : | 40.8px | 20.15px

FIGURE 5.21 – Démonstration de l'amélioration apportée par notre terme basé correspondances éparses sur une paire d'image de la séquence `market_5` du jeu de données Sintel. La grande amplitude du mouvement ainsi que l'effet important de flou de bougé rendent l'estimation difficile. Il est à noter que les correspondances ASIFT erronées, bien visibles dans l'image, ne perturbent pas l'estimation.

les conditions difficiles rendent les correspondances SIFT trop peu nombreuses et trop peu fiables. Nous utilisons donc pour ce jeu de données des correspondances ASIFT, plus robustes. Les gains sont très significatifs, comme le montrent les résultats quantitatifs et l'illustration de la figure 5.21.

Enfin, nous comparons dans le tableau 5.2 notre méthode à l'état de l'art par le biais des évaluations publiques KITTI (avec des correspondances SIFT) et Sintel (avec des correspondances ASIFT). Les résultats obtenus sont compétitifs avec l'état de l'art. Il est à noter qu'au moment de la rédaction de ce manuscrit, les 6 premiers résultats de l'évaluation KITTI utilisent des informations supplémentaires : paires stéréo, contraintes épipolaires ou multi-vues, et ne peuvent donc pas être comparées avec notre approche.

L'approche DeepFlow de Weinzaepfel *et al.* (2013) est la seule méthode non anonyme ayant de meilleures performance que notre approche sur le jeu de données Sintel. Ils utilisent eux aussi une approche inspirée de LDOF avec de nouvelles correspondances semi denses de grande qualité, ce qui valide notre idée de départ, mais ils ne règlent aucune des restrictions de LDOF (occultations, correspondances erronées). De plus ils optimisent leurs paramètres pour chaque jeu de données alors que les nôtres restent inchangé pour les différentes évaluations. Même avec ce surapprentissage, sur le jeu de données KITTI les images moins texturées et les

Tableau 5.2 – Rangs sur les évaluations publiques de méthodes de flot optique.

	Sintel		KITTI	
Méthode	final[a]	clean[a]	pur[b]	temps d'exécution[c]
EpicFlow (anon.)	1	1	6	15s
TriFlowFused (anon.)	2	2	18	350s
DeepFlow (Weinzaepfel *et al.*, 2013)	3	3	7	17s
IVANN (anon.)	4	4	14	1073s
Notre méthode *(AnyFlow)*	7	8	3	**10+5s**

[a] L'évaluation Sintel porte sur deux versions du jeu de données : avec (*final*) et sans (*clean*) les perturbations telles que flou cinématique, brouillard, fumée...
[b] Au moment de la rédaction de ce manuscrit, les 6 premiers résultats de l'évaluation KITTI utilisent des informations supplémentaires : paires stéréo, contraintes épipolaires et multi-vues. Les autres méthodes calculent un flot optique « pur » comme notre méthode.
[c] Notre temps d'exécution est séparé entre mises en correspondance éparse et dense.

limites de la régularisation au premier ordre qu'ils utilisent inversent le classement par rapport à notre méthode.

Terme basé correspondances éparses

Il convient de vérifier expérimentalement les propriétés du terme basé correspondances éparses introduit en section 5.3.2. Pour cela nous générons une paire d'images (figure 5.22) par une déformation synthétique mettant en défaut toutes les approches denses multi-résolution : une rotation de 180 degrés. Nous utilisons, pour cette expérience, des correspondances parfaites artificielles sur une grille régulière.

Pour commencer, en figure 5.23, on mesure l'erreur moyenne en fonction du poids μ affecté au terme basé sur 256 correspondances parfaites. On observe une transition de phase qui confirme que les correspondances éparses sont essentielles à la convergence pour cette paire d'images. Le fait que cette transition soit très nette valide l'hypothèse que les correspondances éparses permettent de guider l'estimation hors des minimums locaux mais laissent ensuite le terme de données dense maximiser la précision. L'influence du nombre de correspondances éparses est évaluée en figure 5.24 : le poids μ minimal pour atteindre la transition est inversement proportionnel au nombre de correspondances. Cela signifie que l'influence globale des correspondances est proportionnelle à la surface couverte, en pixels, et explique donc le phénomène de densification par multi-résolution (section 5.2.1).

Enfin, le pouvoir filtrant de l'estimateur de Geman McClure est mesuré en figure 5.25. L'erreur moyenne est mesurée en ajoutant graduellement des corres-

FIGURE 5.22 – La paire d'images « Lena ». La rotation de 180 degrés met en défaut les méthodes multi-résolution variationnelles.

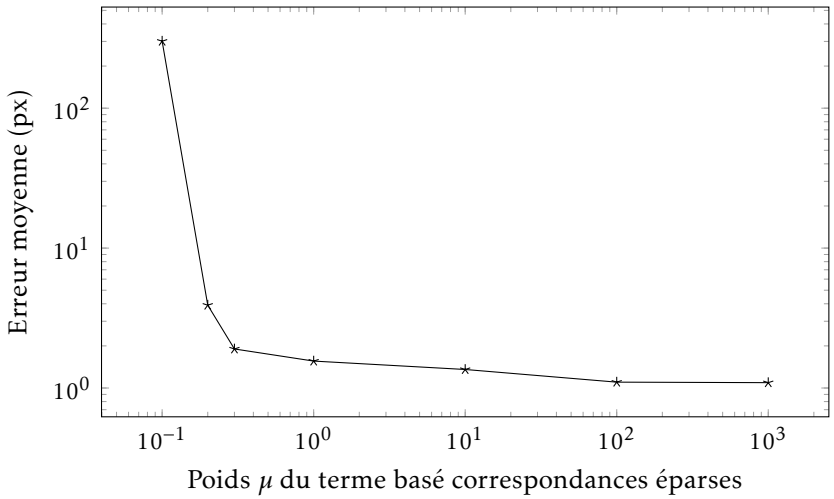


FIGURE 5.23 – Influence du poids du terme basé correspondances éparses sur la paire d'images Lena avec une grille régulière de 256 correspondances parfaites. On observe une transition de phase nette.

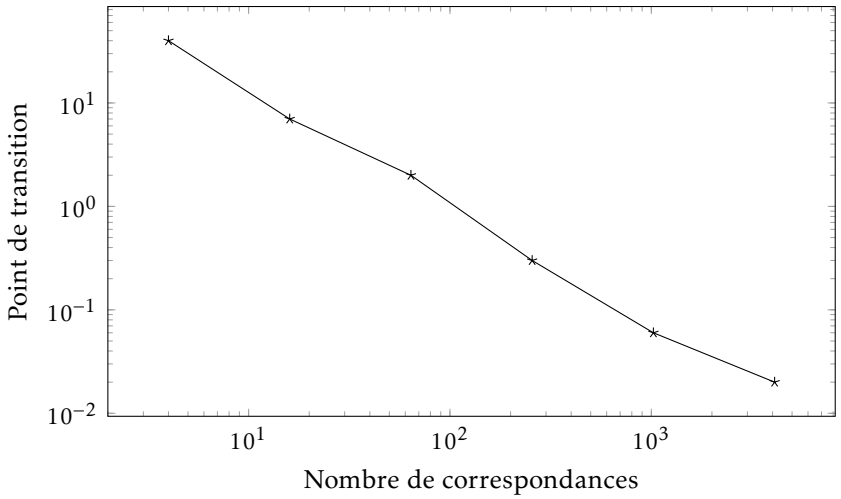


FIGURE 5.24 – Évolution du point de transition en fonction du nombre de correspondances. Le point de transition est la valeur minimale du poids μ du terme basé correspondances éparses pour atteindre la convergence, sur la paire d'images Lena. Les deux axes utilisent une échelle logarithmique.

pondances erronées à 256 correspondances parfaites de points. On observe que l'estimateur est capable de filtrer sans dégradation environ 200 correspondances erronées, c'est à dire un taux de 44%.

Stéréo-vision à large parallaxe

Nous nous intéressons maintenant à la véritable valeur ajoutée de notre approche : l'élargissement du bassin de convergence permet d'étendre considérablement le domaine d'applicabilité des méthodes variationnelles. En particulier, nous décrivons ici l'application à la stéréo-vision avec large parallaxe.

La mise en correspondance dense entre images stéréos est une étape importante de la reconstruction 3D (voir section 3.1.2), et les larges parallaxes sont préférables pour deux raisons. D'abord, plus la parallaxe est large, plus l'incertitude en profondeur est faible (voir section 2.5.2) ; de plus une large parallaxe permet d'effectuer une reconstruction équivalente avec moins d'images qu'une technique pour faibles parallaxes, ce qui permet des gains en temps de calcul.

Pourtant, la mise en correspondance est souvent beaucoup plus difficile avec des points de vue éloignés. En effet, la présence de distorsions perspectives et de variations de luminosité sont susceptibles de mettre en défaut les descripteurs photométriques (section 2.6.2). De plus, de larges zones sont souvent impossibles

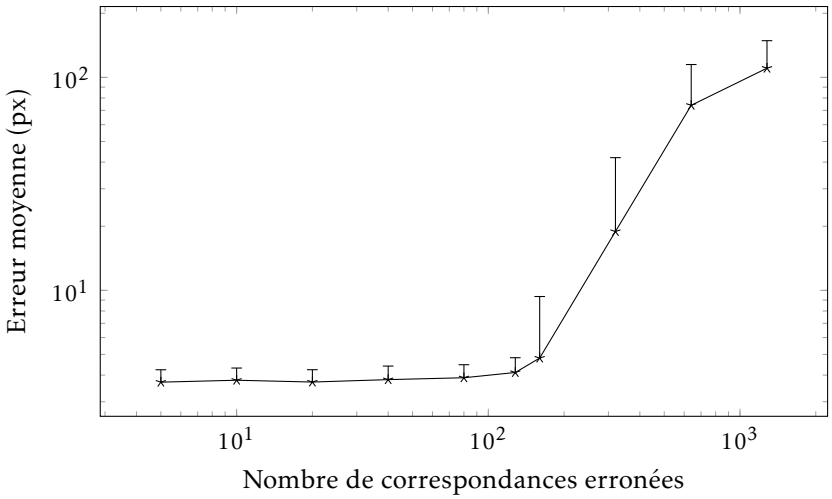


FIGURE 5.25 – Influence des correspondances erronées sur l'estimation de la transformation entre les images de la paire Lena. Des correspondances erronées aléatoires sont ajoutées à 256 correspondances parfaites sur une grille régulière. Les points de la courbe représentent la moyenne de l'erreur sur 100 tirages et les barres d'erreur la déviation standard des résultats.

à mettre en correspondance car visibles dans une seule des images. Enfin, plus les images sont éloignées dans le temps, plus il est probable que l'hypothèse de scène rigide soit mise en défaut.

Peu de méthodes ont ainsi été proposées dans la littérature, et les meilleurs résultats sont ceux de Tola *et al.* (2010) avec leur descripteur DAISY. Celui-ci, bien que photométrique, est robuste à des distorsions importantes (voir section 2.6.2) et prévu pour être calculé de manière dense. Une mise en correspondance par optimisation discrète (avec classification explicite par pixel des occultations) permet d'obtenir des cartes de profondeurs.

Nous essayons de reproduire leurs résultats sur le jeu de données herz jesu en utilisant des correspondances de segments de droite (Wang *et al.*, 2009a), robustes aux déformations perspectives et aux changements de luminosité. Nous gardons le terme de données dense Census mais augmentons le seuil $\theta_s = 0.5$ (voir tableau 5.1) afin de surestimer les auto-occultations. Cela a pour effet de réduire l'influence du terme de données sur les surfaces présentant un angle important avec le plan de l'image (presque parallèles à l'axe optique), où il aurait été peu fiable. Cela permet de lui assigner un poids plus important $\lambda = 30$ car les scène observées sont bien texturées avec peu de perturbations. Nous n'avons pas développé d'algorithme spécifique à l'estimation stéréo mais projetons simplement le champ de déplacements

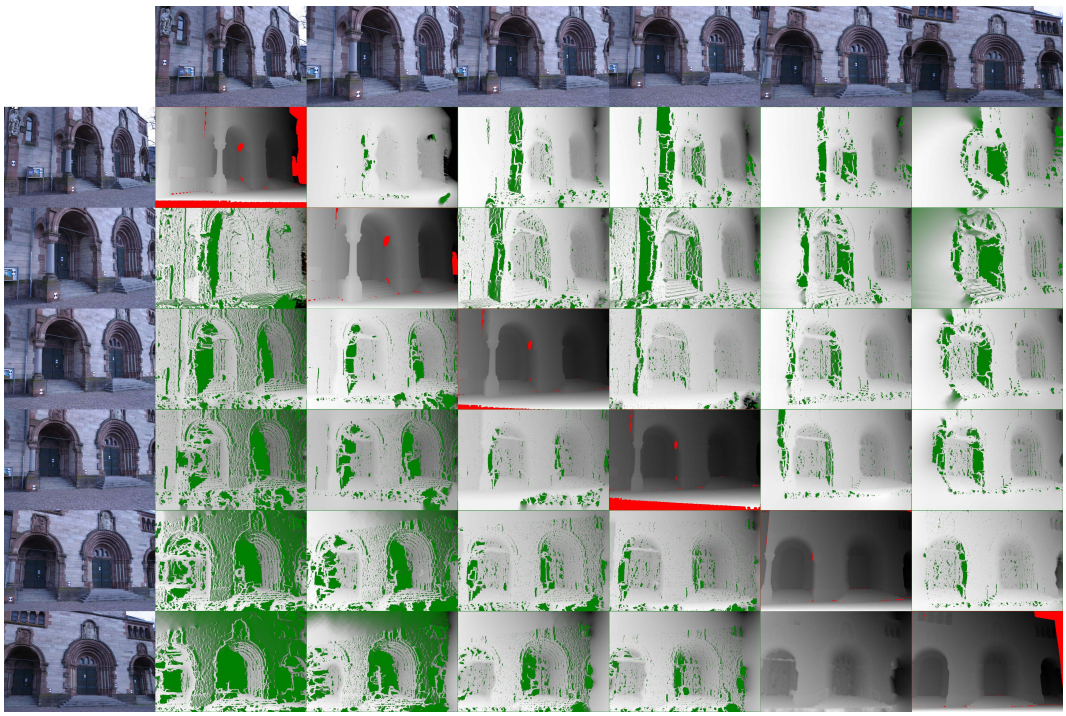


FIGURE 5.26 – Nos résultats sur le jeu de données herzjesu. La probabilité d'auto-occultation est représentée en vert.

sur les lignes épipolaires à chaque itération.

La figure 5.26 présente la vérité terrain ainsi que les cartes de profondeurs estimées via notre méthode sur le jeu de données herzjesu en utilisant la même forme matricielle que Tola *et al.* (2010) pour faciliter la comparaison. La probabilité d'auto-occultation $\mathcal{P}_{\theta_s}$ est dessinée en vert. On peut voir que les auto-occultations sont effectivement surestimées et réduisent l'influence du terme de données sur les surfaces inclinées. La figure 5.27 contient une comparaison quantitative entre la méthode dite « de base », c'est à dire sans terme basé correspondances éparses, la méthode proposée, et les résultats annoncés par Tola *et al.* (2010) avec la méthode DAISY. La première conclusion à tirer est que l'ajout de notre terme basé correspondances de segments élargit grandement le bassin de convergence et divise l'erreur moyenne par presque 5. La deuxième conclusion est que nos résultats étant à moins de 2% de ceux de DAISY, notre approche permet de réutiliser des méthodes génériques pour des problèmes spécifiques.

Deux exemples qualitatifs, sans vérité terrain, viennent illustrer le bon fonctionnement de la méthode avec des correspondances de segments en figure 5.28.

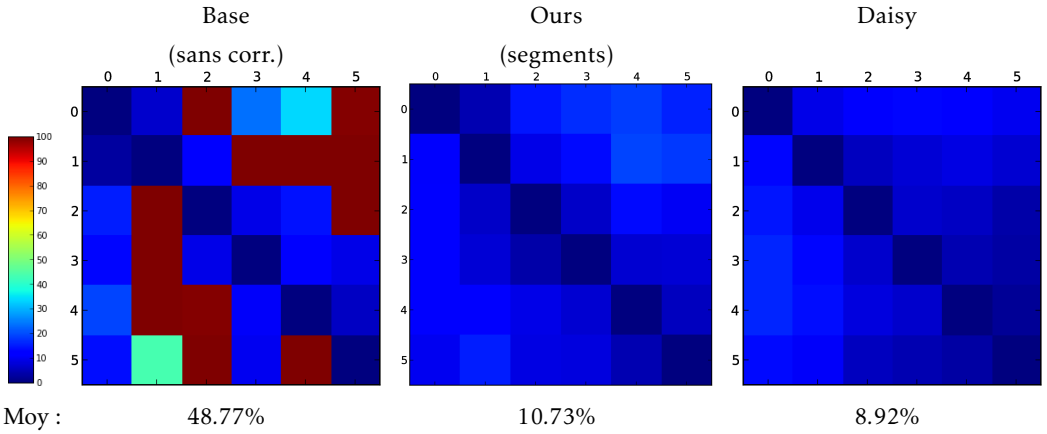


FIGURE 5.27 – Proportion de profondeurs erronées sur le jeu de données herzjesu. Comme Tola *et al.* (2010), nous considérons qu'une profondeur est erronée si l'erreur par rapport à la vérité terrain est supérieure à 5% de la variation de profondeur sur l'image (différence entre profondeurs maximale et minimale). La première ligne est une représentation en couleur de l'erreur pour chaque couple d'images, la deuxième ligne affiche l'erreur moyenne sur tous les couples

Alignement de surfaces déformables

Si des méthodes paramétriques sont plus adaptées au problème d'alignement de surfaces déformables (voir section 5.5), nous vérifions la généricité de l'approche non paramétrique par quelques résultats qualitatifs en figure 5.29. Nous comparons nos résultats à la méthode FBDSD (Pizarro et Bartoli, 2012), constituant l'état de l'art des méthodes basées correspondances de primitives. Nous utilisons l'implémentation publique C++ de Alcantarilla et Bartoli (2012) et utilisons les mêmes correspondances SURF (Bay *et al.*, 2006) en entrée de notre algorithme.

La déformation est estimée en utilisant le modèle plan comme image de référence I_1 car le champ de déplacements est ainsi défini sur toute l'image ce qui facilite l'estimation. Le champ de déplacements est ensuite inversé et appliqué à une grille colorée qui est superposée à l'image de la surface déformée. Les importantes déformations sont susceptibles d'introduire de nombreux minimums locaux pour le terme dense, nous réduisons donc son influence avec $\lambda = 1$ (voir tableau 5.1). Pour évaluer l'apport du terme dense, nous effectuons également une estimation en supprimant totalement son influence avec $\lambda = 0$.

Les résultats montrent que notre terme basé correspondances de primitives permet de rendre les approches variationnelles classiques viables pour de tels pro-

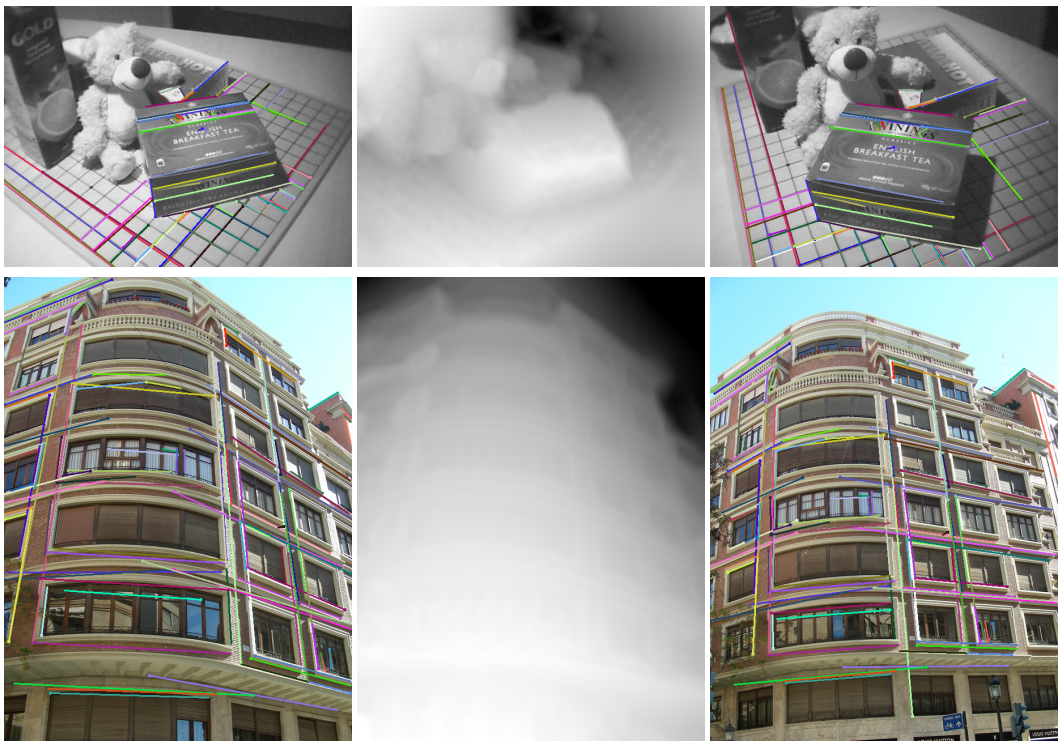


Figure 5.28 – Exemples d'estimation de profondeur par stéréo-vision. De gauche à droite : image de référence, carte de profondeur, deuxième image. Respectivement 81 et 111 correspondances de segments ont été utilisées.

blèmes. Les déformations sont bien estimées et les auto-occultations correctement gérées. Même sans le terme dense, nos résultats sont meilleurs que FBDSD pour les deux premières paires, ce qui prouve la supériorité du filtrage implicite. Le terme dense apporte toutefois des gains significatifs. Quelques défauts sont toujours présents mais surtout dûs à la régularisation. La section 5.5 présente des résultats quantitatifs avec un modèle paramétrique et une régularisation spécifique plus adaptée.

5.5 Implémentation avec modèle paramétrique pour surfaces déformables

La généricité de notre approche est démontrée par l'implémentation de notre terme basé correspondance au sein de la méthode paramétrique d'alignement de surfaces déformables de Gay-Bellile *et al.* (2010). Le champ de correspondances utilise un modèle de déformation *Free-Form Deformation* (voir section 2.7.3) paramétrisé par les déplacements D de points de contrôle (voir section 2.7.3) :

$$\boldsymbol{u} : \quad (\boldsymbol{q}, D) \mapsto FFD(\boldsymbol{q}, D) - \boldsymbol{q} \tag{5.24}$$

où FFD est défini par l'équation (2.91).

La fonction de coût à minimiser est :

$$C_{\text{GB}}(\mathcal{D}, I_1, I_2) = \underbrace{\lambda \iint_{\Omega_{I_1}} C^{\star}_{\text{AD}}(\boldsymbol{u}(\boldsymbol{q}, D), I_1, I_2)\, d\boldsymbol{q} + \lambda_s C_{\text{shrinker}}{}^{\ddagger}(D) + R_{\text{courb.}}(D)}_{\text{Gay-Bellile } et\ al. \text{ (2010)}} + \mu\, C_{\text{corr.}}(\boldsymbol{u}_D(\boldsymbol{q}), \mathbf{f}) \tag{5.25}$$

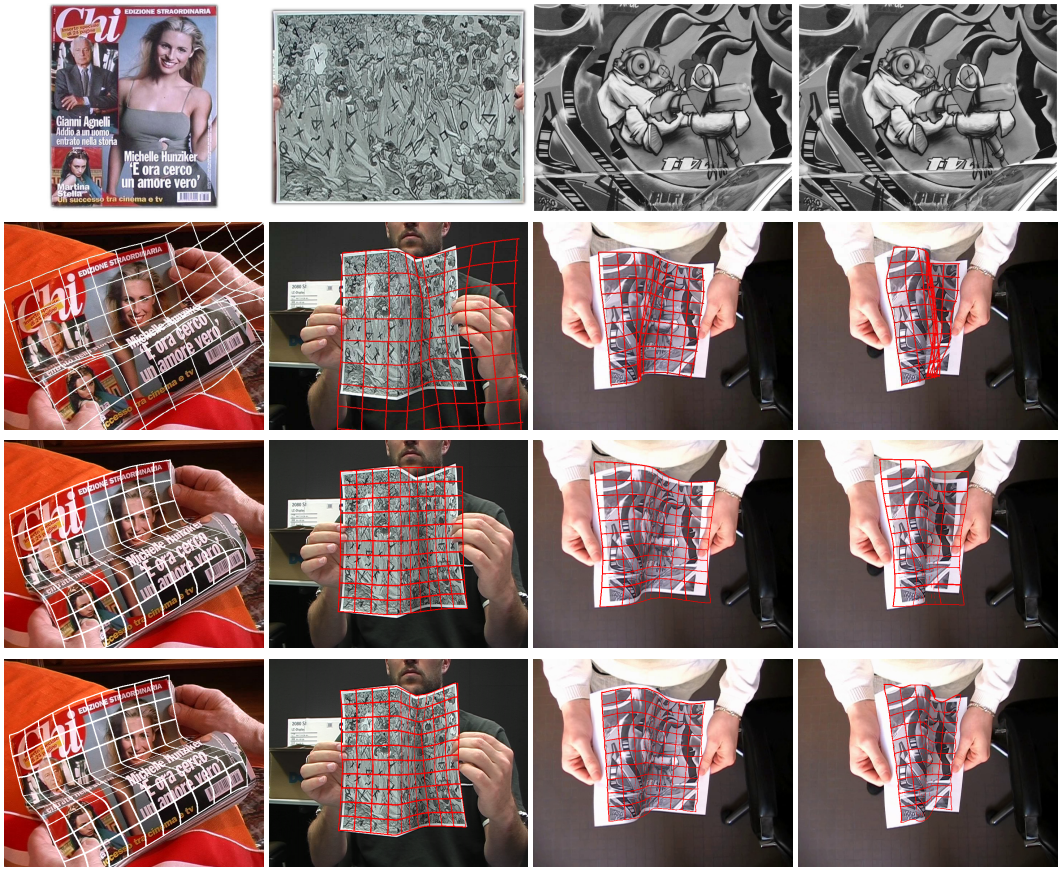
$R_{\text{courb.}}$ est l'énergie de courbure du champ de déplacement, définie par l'équation (2.88), calculable directement à partir du champ D (Prasad et Fitzgibbon, 2006).

5.5.1 Détails d'implémentation

Notre terme de données est intégré à l'implémentation publique Matlab de Gay-Bellile *et al.* (2010) utilisant :

— une optimisation Gauss-Newton (voir section 2.8.2),

‡. Le modèle de déformation FFD peut adopter des configurations aberrantes en 2D sous la forme de *replis*. Gay-Bellile *et al.* (2010) ajoutent un terme (le *shrinker*) dédié à la pénalisation de telles configurations.



156 174 148 106

FIGURE 5.29 – Exemples d'alignement de surfaces déformables avec la méthode non paramétrique. De haut en bas, modèles plans, résultats de FBDSD (Pizarro et Bartoli, 2012), résultats de notre méthode sans terme dense ($\lambda = 0$), résultats de notre méthode avec $\lambda = 1$, nombre de correspondances SURF. De gauche à droite, les paires d'images proviennent des publications suivantes : Ferrari *et al.* (2004); Salzmann *et al.* (2007); Gay-Bellile *et al.* (2010).

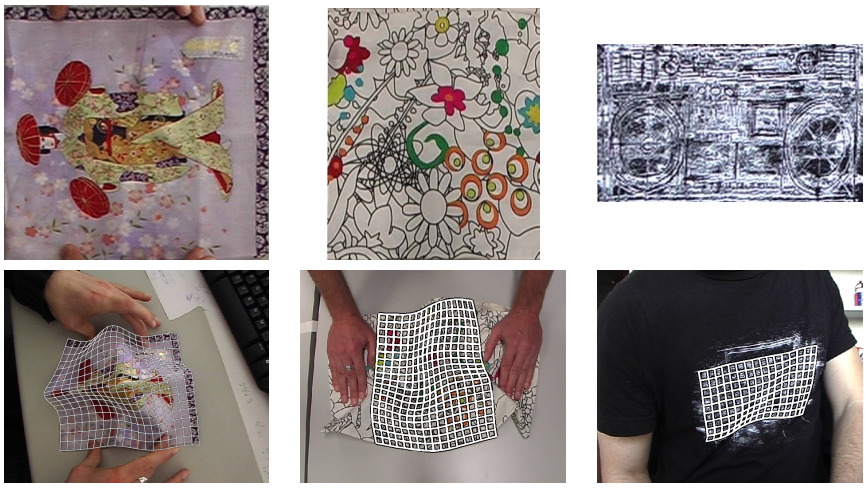


FIGURE 5.30 – Résultats qualitatifs sur le jeu de données de Salzmann *et al.* (2007); Salzmann et Fua (2009). Première ligne : surface plane, deuxième ligne : estimation de la déformation.

— 6 niveaux de résolution avec un facteur de subdivision (isotropique) $s = 0.5$,
— une grille de points de contrôle avec un pas $\varepsilon = 5$ pixels,
— les poids suivants : $\lambda = 2 \cdot 10^{-4}$, $\lambda_s = 20$ et $\mu = 0.16$.

5.5.2 Expériences

La méthode modifiée avec notre terme de données basé correspondances éparses est comparée à la méthode originale de Gay-Bellile *et al.* (2010), ainsi qu'à deux méthodes basées filtrage explicite de correspondances (Pizarro et Bartoli, 2012; Tran *et al.*, 2012). Ces méthodes utilisent un modèle pour supprimer explicitement les correspondances considérées aberrantes. Un modèle de déformation FFD est ensuite ajusté aux correspondances restantes.

Pour commencer, les résultats qualitatifs des figures 5.30 et 5.31 montrent un aperçu du potentiel de notre approche.

5.5.3 Séquence réelle

Il n'existe pas de jeu de données avec vérité terrain contenant d'importantes déformations non rigides. Par conséquent, nous utilisons la séquence `Graffiti` de Gay-Bellile *et al.* (2010), et les résultats de leur méthode de suivi comme référence. Ensuite, à partir de correspondances SIFT, nous comparons notre approche avec les méthodes basées correspondances de Tran *et al.* (2012, RANSAC), et de Pizarro et

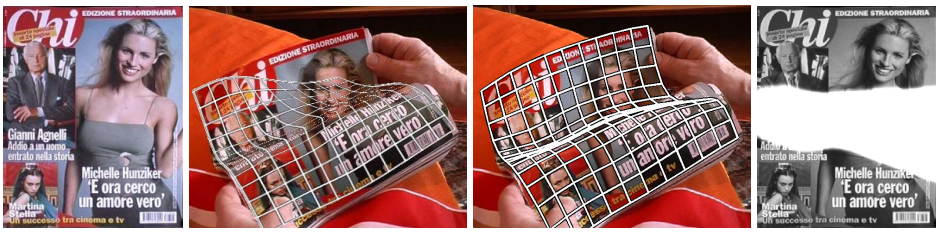


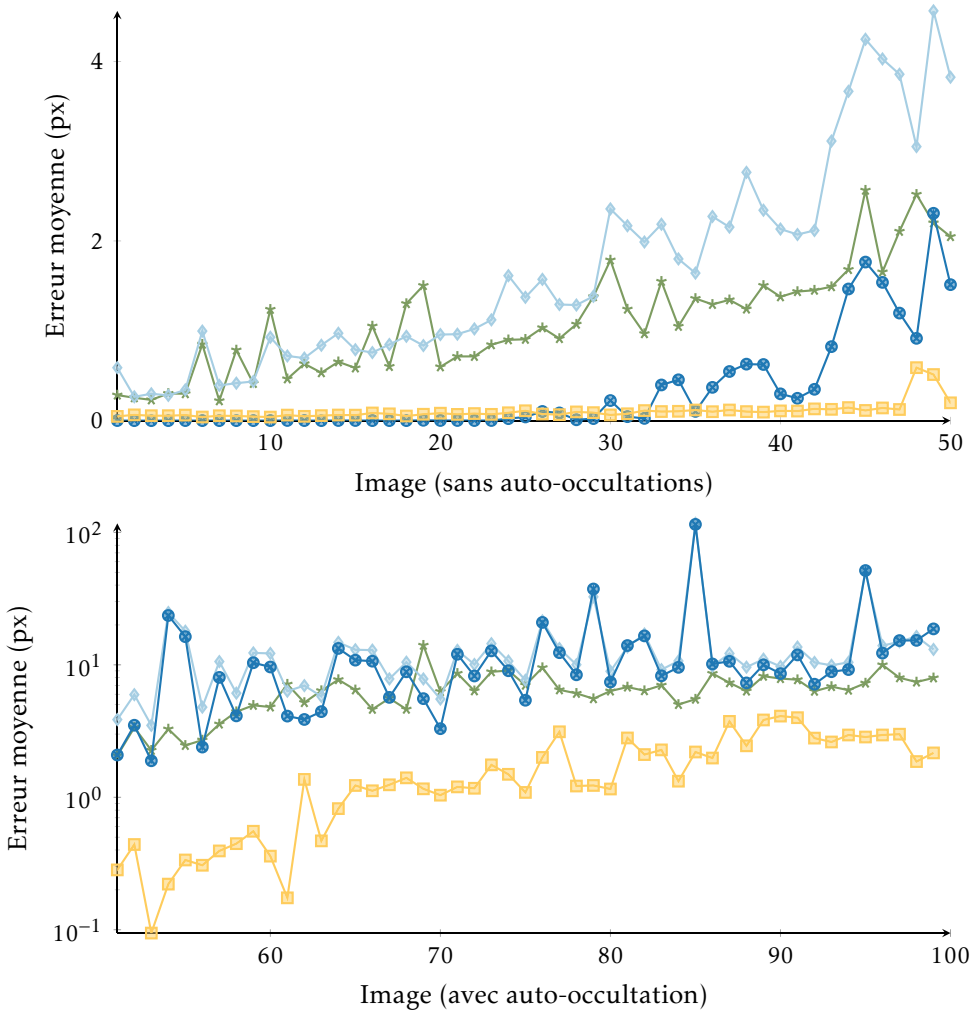
FIGURE 5.31 – Résultat qualitatif sur une paire particulièrement difficile du jeu de données toys (Ferrari *et al.*, 2004). De gauche à droite : surface plane, déformation estimée avec FBDSD (Pizarro et Bartoli, 2012), déformation estimée avec notre méthode et la probabilité d'auto-occultation $\mathcal{P}_{\theta_s}$ (en blanc).

Bartoli (2012, FBDSD), ainsi que cette dernière avec raffinement dense en utilisant Gay-Bellile *et al.* (2010, FBDSD+P). Les estimations sont effectuées entre la première image de la séquence et chacune des images suivantes, sans aucun suivi (le champ de correspondances est initialisé à zéro pour chaque paire). Les résultats, visibles sur la figure 5.32, montrent que notre approche ne dégrade pas significativement la précision pour de faibles déformations et surpasse largement les autres en présence de larges déformations.

5.6 Conclusion

Nous avons présenté ici une approche générique pour augmenter le bassin de convergence de n'importe quelle approche variationnelle d'alignement d'images grâce à l'ajout d'un terme de données basé correspondances éparses et une gestion explicite des occultations. Un estimateur robuste permet de filtrer implicitement les correspondances et empêche les données aberrantes de dégrader le résultat final. Les correspondances apportent ainsi un gain de robustesse important en réduisant les minimums locaux, ceci tout en préservant la précision de la méthode variationnelle dense. À notre connaissance, aucune méthode n'est à ce jour capable d'obtenir des résultats parmi les meilleurs sur les évaluations de flot optique (faibles déplacements) tout en permettant, sans modification, d'effectuer des alignements denses entre images stéréo avec large parallaxe et surfaces déformables.

Chaque contribution a été conçue dans un soucis de généricité : la méthode variationnelle de base peut être paramétrique ou non, les primitives des correspondances éparses peuvent être des points ou des segments de droite (une extension à d'autres primitives ne nécessite que l'établissement d'une nouvelle distance point-primitive), et la gestion des auto-occultations a été validée dans des contextes de scènes rigides avec déformations perspectives ainsi que de surfaces déformables. De plus le



Méthode	erreur moyenne sans auto-occultations	avec auto-occultations
RANSAC (Tran *et al.*, 2012)	1.07	6.22
FBDSD (Pizarro et Bartoli, 2012)	1.59	14.06
FBDSD+P (Pizarro et Bartoli, 2012)	0.29	12.74
méthode proposée	0.10	1.60

FIGURE 5.32 – Évolution de l'erreur des points de contrôle FFD par rapport à la méthode basée suivi de Gay-Bellile *et al.* (2010).

couplage entre les composants a été gardé au minimum, ce qui permet de changer facilement la combinaison utilisée de termes de données, régularisation, primitives et technique d'optimisation. Nous avons d'ailleurs démontré notre approche dans deux implémentations différentes : une paramétrique et une non paramétrique.

Dans le cadre de la réalisation d'un système de réalité augmentée, il serait intéressant d'utiliser notre méthode pour générer des cartes de profondeurs (avec l'avantage d'une robustesse accrue et du support de larges parallaxes par rapport aux méthodes existantes), qui seraient ensuite fusionnées en une reconstruction volumétrique (Newcombe *et al.*, 2011a; Steinbruecker *et al.*, 2014).

Nous pensons aussi que nos contributions peuvent remettre en cause le développement de méthodes spécifiques à chaque catégorie de mise en correspondance dense. Les travaux futurs envisagés consistent à améliorer chaque composant de base au fur et à mesure des évolutions de l'état de l'art. En particulier, il serait intéressant de tester les correspondances semi denses de Weinzaepfel *et al.* (2013, *DeepFlow*), ainsi que des correspondances de courbes ou de surfaces. Le point le plus limitant reste la méthode variationnelle dense. Il faudrait surtout une meilleure gestion des contours d'occultations, éventuellement avec la détection explicite de Leordeanu *et al.* (2012). Des techniques de mise en correspondance semi aléatoires de type *PatchMatch* pourraient encore plus réduire l'influence de minimums locaux. On peut également envisager l'application à de nouveaux domaines comme la correspondance entre scènes différentes à la manière de *SIFT-Flow* (Liu *et al.*, 2011).

Annexe 5.A Opérateurs différentiels

Nous reproduisons ici les opérateurs différentiels définis par Bredies (2012) ainsi que leurs approximations par différences finies. Ils sont choisis afin de conserver les propriétés des opérateurs adjoints (voir Bredies, 2012).

Discrétisation du domaine image

Soit une image I définie sur un domaine Ω_I, on appelle $\Omega_I^{(h)}$ le domaine discret défini par la grille de pas h (usuellement de un pixel) et I_h la fonction associée :

$$\begin{aligned} I_h : \Omega_I^{(h)} &\to \mathbb{R} \\ (i,j) &\mapsto I_h[i,j] = I(h \cdot i, h \cdot j). \end{aligned} \tag{5.26}$$

De même, on appelle $\mathbb{U}^{(h)}$, $\mathbb{V}^{(h)}$ et $\mathbb{W}^{(h)}$ les restrictions des espaces fonctionnels $\mathbb{U}^{(h)}$, $\mathbb{V}^{(h)}$ et $\mathbb{W}^{(h)}$ de la section 2.3.3 à $\Omega_I^{(h)}$.

On définit également les dimensions n et m de l'image telles que :

$$\Omega_I^{(h)} = [\![1, n]\!] \times [\![1, m]\!]. \tag{5.27}$$

Différences finies sur $\mathbb{U}$

Deux types de différences finies, avant et arrière, sont définies comme suit, $\forall u \in \mathbb{U}^{(h)}$, $\forall (i,j) \in \Omega_I^{(h)}$:

$$\partial_x^+ u[i,j] = \begin{cases} 0 & \text{si } i = m \\ u[i+1,j] - u[i,j] & \text{sinon} \end{cases} \tag{5.28}$$

$$\partial_y^+ u[i,j] = \begin{cases} 0 & \text{si } j = n \\ u[i,j+1] - u[i,j] & \text{sinon} \end{cases} \tag{5.29}$$

$$\partial_x^- u[i,j] = \begin{cases} u[1,j] & \text{si } i = 1 \\ -u[m-1,j] & \text{si } i = m \\ u[i,j] - u[i-1,j] & \text{sinon} \end{cases} \tag{5.30}$$

$$\partial_y^- u[i,j] = \begin{cases} u[i,1] & \text{si } j = 1 \\ -u[i,n-1] & \text{si } j = n \\ u[i,j] - u[i,j-1] & \text{sinon.} \end{cases} \tag{5.31}$$

Gradients

On reprend la définition du gradient sur $\mathbb{U}$ de l'équation (2.17) :

$$\nabla : \mathbb{U} \to \mathbb{V} \tag{5.32}$$

$$u \mapsto \nabla u = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} \tag{5.33}$$

et en différences finies :

$$\nabla_h : \mathbb{U}^{(h)} \to \mathbb{V}^{(h)} \tag{5.34}$$

$$u \mapsto \nabla_h u = \begin{pmatrix} \partial_x^+ u \\ \partial_y^+ u \end{pmatrix}. \tag{5.35}$$

Sur $\mathbb{V}$ on définit le gradient symétrique Γ :

$$\Gamma : \quad \mathbb{V} \to \mathbb{W} \tag{5.36}$$

$$\boldsymbol{v} = (v_1, v_2) \mapsto \Gamma \boldsymbol{v} = \begin{pmatrix} \frac{\partial v_1}{\partial x} & \frac{\frac{\partial v_1}{\partial y} + \frac{\partial v_2}{\partial x}}{2} \\ \frac{\frac{\partial v_1}{\partial y} + \frac{\partial v_2}{\partial x}}{2} & \frac{\partial v_2}{\partial y} \end{pmatrix} \tag{5.37}$$

$$\Gamma_h : \quad \mathbb{V}^{(h)} \to \mathbb{W}^{(h)} \tag{5.38}$$

$$\boldsymbol{v} = (v_1, v_2) \mapsto \Gamma_h \boldsymbol{v} = \begin{pmatrix} \partial_x^- v_1 & \frac{\partial_y^- v_1 + \partial_x^- v_2}{2} \\ \frac{\partial_y^- v_1 + \partial_x^- v_2}{2} & \partial_y^- v_2 \end{pmatrix}. \tag{5.39}$$

Divergence

On définit également les opérateur divergence :

$$\mathrm{div} : \quad \mathbb{V} \to \mathbb{U} \tag{5.40}$$

$$\boldsymbol{v} = (v_1, v_2) \mapsto \mathrm{div}\, \boldsymbol{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} \tag{5.41}$$

$$\mathrm{div}_h : \quad \mathbb{V}^{(h)} \to \mathbb{U}^{(h)} \tag{5.42}$$

$$\boldsymbol{v} = (v_1, v_2) \mapsto \mathrm{div}_h \boldsymbol{v} = \partial_x^- v_1 + \partial_x^- v_2. \tag{5.43}$$

$$\mathrm{div} : \quad \mathbb{W} \to \mathbb{V} \tag{5.44}$$

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & w_{12} \\ w_{12} & w_{22} \end{pmatrix} \mapsto \mathrm{div}\, \boldsymbol{W} = \begin{pmatrix} \frac{\partial w_{11}}{\partial x} + \frac{\partial w_{12}}{\partial y} \\ \frac{\partial w_{12}}{\partial x} + \frac{\partial w_{22}}{\partial y} \end{pmatrix} \tag{5.45}$$

$$\mathrm{div}_h : \quad \mathbb{W}^{(h)} \to \mathbb{V}^{(h)} \tag{5.46}$$

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & w_{12} \\ w_{12} & w_{22} \end{pmatrix} \mapsto \mathrm{div}_h \boldsymbol{W} = \begin{pmatrix} \partial_x^+ w_{11} + \frac{\partial w_{12}}{\partial y} \\ \partial_x^+ w_{12} + \frac{\partial w_{22}}{\partial y} \end{pmatrix}. \tag{5.47}$$

Annexe 5.B Optimisation Chambolle et Pock avec régularisation TGV^2

Nous résumons ici les étapes de l'algorithme de Chambolle et Pock (2011) utiles pour la reproduction de nos résultats, extraites des publications Ranftl *et al.* (2012)

et surtout Bredies (2012). Pour l'estimation de champs de déplacements denses, l'algorithme est appliqué à une optimisation de la forme :

$$\hat{\boldsymbol{u}} = \arg\min \iint_{\Omega} C(\boldsymbol{x}, u(\boldsymbol{x}))\, d\boldsymbol{x} + \mathrm{TGV}^2(u, \alpha_0, \alpha_1) \tag{5.48}$$

où $u : \Omega \to \mathbb{U}$; $C : \Omega \to \mathbb{R}$ est une fonction de coût convexe, et TGV^2 est défini en section 2.7.2. Dans ce cas, l'optimisation consiste en l'itération des étapes suivantes, omettant la dépendance en $\boldsymbol{x}$ pour une meilleure lisibilité :

$$\begin{cases} \boldsymbol{p}_{n+1} & = \Pi_{\|\boldsymbol{p}\|_\infty \leq 1}(\boldsymbol{p}_n + \sigma_p \alpha_1 (\nabla \bar{u}_n - \bar{\boldsymbol{w}}_n)) \\ \boldsymbol{q}_{n+1} & = \Pi_{\|\boldsymbol{q}\|_\infty \leq 1}(\boldsymbol{q}_n + \sigma_q \alpha_0 (\Gamma \bar{\boldsymbol{w}}_n)) \\ u_{n+1} & = (I + \tau_u \partial C)^{-1}(u_n + \tau_u \alpha_1 \operatorname{div} \boldsymbol{p}_{n+1}) \\ \boldsymbol{w}_{n+1} & = \boldsymbol{w}_n + \tau_w (\alpha_1 \operatorname{div} \boldsymbol{q}_{n+1} + \alpha_0 \boldsymbol{p}_{n+1}) \\ \bar{u} & = 2u_{n+1} - u \\ \bar{\boldsymbol{w}} & = 2\boldsymbol{w}_{n+1} - \boldsymbol{w}. \end{cases} \tag{5.49}$$

où les champs $u : \Omega \to \mathbb{U}$ et $\boldsymbol{w} : \Omega \to \mathbb{V}$ sont les variables primales et les champs $\boldsymbol{p} : \Omega \to \mathbb{V}$ et $\boldsymbol{q} : \Omega \to \mathbb{W}$ leurs variables duales associées. Les fonctions de différentiations sont celles définies en section 5.A. Les fonctions de projections sont définies par pixel comme suit :

$$\Pi_{\|\boldsymbol{p}\|_\infty \leq 1}(\boldsymbol{p}) : \quad \boldsymbol{x} \mapsto \frac{\boldsymbol{p}(\boldsymbol{x})}{\max(\|\boldsymbol{p}(\boldsymbol{x})\|, 1)} \tag{5.50}$$

$$\Pi_{\|\boldsymbol{q}\|_\infty \leq 1}(\boldsymbol{q}) : \quad \boldsymbol{x} \mapsto \frac{\boldsymbol{q}(\boldsymbol{x})}{\max(\|\boldsymbol{q}(\boldsymbol{x})\|, 1)}. \tag{5.51}$$

Si C est linéaire :

$$C(u) = C(u_0) + (u - u_0)\frac{\partial C(u_0)}{\partial x}. \tag{5.52}$$

le résolvant de l'opérateur sous-différentiel ∂C : $(I + \tau_u \partial C)^{-1}$ est défini par (Ranftl *et al.*, 2012) :

$$(I + \tau_u \partial C)^{-1}(u) = u + \begin{cases} \tau_u \frac{\partial C}{\partial x} & \text{si } C < -\tau_u \left(\frac{\partial C}{\partial x}\right)^2 \\ -\tau_u \frac{\partial C}{\partial x} & \text{si } C > \tau_u \left(\frac{\partial C}{\partial x}\right)^2 \\ -C\left(\frac{\partial C}{\partial x}\right)^{-1} & \text{sinon} \end{cases}. \tag{5.53}$$

Les pas σ_p, σ_q, τ_u, τ_w sont calculés par la méthode de préconditionnement de Pock et Chambolle (2011) :

$$\sigma_p = \frac{1}{3\alpha_1} \qquad \tau_u = \frac{1}{4\alpha_1} \tag{5.54}$$

$$\sigma_q = \frac{1}{2\alpha_0} \qquad \tau_w = \frac{1}{4\alpha_0 + \alpha_1}. \tag{5.55}$$

Chapitre 6

Conclusion

La réalité augmentée est un terme largement employé, mais souvent mal défini et aux enjeux incompris. Le présent mémoire a permis d'identifier des applications à fort potentiel : aide à la vente, aide à la maintenance, aide à la navigation et aide à la chirurgie. Ces applications sont accessibles avec les technologies matérielles actuelles ou à court terme mais ont des exigences de robustesse, précision et sécurité beaucoup plus importantes que les applications ludiques, actuellement prédominantes. Une vision globale et transversale des systèmes de réalité augmentée a permis de cibler les contributions sur deux aspects limitants : l'affichage et la reconstruction 3D, avec à chaque fois le soucis de proposer des solutions robustes et flexibles pour faciliter leur intégration et industrialisation.

Le premier problème traité est la réalité augmentée sur affichage semi transparent dont les gains, en terme de sécurité surtout, s'accompagnent d'un besoin d'étalonnage adapté. Nous avons défini une nouvelle catégorie de systèmes : la tablette augmentée, constituée d'un écran semi-transparent et de deux dispositifs de localisation (de l'utilisateur et de la scène) par rapport à l'écran. Nous avons développé un processus d'étalonnage original qui, grâce à l'introduction de caméras virtuelles, permet de tirer parti des méthodes testées et éprouvées d'étalonnage de caméras classiques. Notre approche a montré des résultats très satisfaisants, proches de l'optimalité.

Les principaux obstacles à une industrialisation du concept sont maintenant matériels. Pour une expérience réaliste et immersive, il faudrait en effet : un écran semi-transparent stéréoscopique à la fois lumineux et occultant, ainsi que des systèmes d'affichage et de localisation induisant une latence minimale. Une extension théorique intéressante serait la prise en charge de surfaces d'affichage non planes pour, par exemple, un pare-brise de voiture augmenté.

Ensuite, dans l'optique d'une solution de reconstruction 3D plus simple, rapide et précise que l'existant, nous avons proposé une nouvelle formulation pour la mise en

correspondance dense d'images. Nous avons pour cela mis en place un processus de mise à niveau des méthodes variationnelles existantes. Ces dernières sont basées sur la minimisation d'une fonction de coût avec un terme dense de similitude et limitées à de petites déformations. La première modification proposée est l'intégration d'un nouveau terme dans la fonction de coût, basé sur des correspondances de primitives éparses. Ce terme permet d'éviter des minimums locaux pour un bassin de convergence largement accru, tout en étant robuste aux correspondances erronées et en supportant les primitives non ponctuelles telles que les segments de droite. La deuxième modification consiste à détecter et gérer explicitement les occultations dans le terme de similitude. Le découplage des différents composants de l'algorithme est démontré par sa mise en œuvre avec différentes primitives, différents termes de similitude, différentes régularisations. La généricité de l'approche est soulignée par la variété des applications où elle est compétitive avec l'état de l'art : flot optique, stéréo à large parallaxe, alignement de surfaces déformables. Nous pensons que ces résultats peuvent avoir de fortes implications sur le domaine de la mise en correspondance d'images en montrant que les méthodes variationnelles, qui ont fait l'objet de beaucoup de travaux, peuvent être généralisées à des problèmes complexes en réduisant le besoin de dériver des solutions spécifiques *ad hoc*.

Pour obtenir de meilleures performances encore, il serait intéressant d'améliorer chacun des composants tout en conservant le cadre proposé : régularisation avec meilleure gestion des contours d'occultations, terme de similitude plus robuste aux déformations telles que rotations et changements d'échelle, correspondances de nouvelles primitives géométriques (courbes, régions...). Enfin, il faudrait intégrer cette méthode de mise en correspondance dans une chaîne de reconstruction 3D, par fusion de cartes de profondeurs par exemple. Le large bassin de convergence de notre approche devrait permettre une reconstruction robuste et précise avec un nombre minimal d'images, ce qui la rendrait plus pratique et plus rapide que les approches existantes.

Au delà des améliorations de chaque contribution, la perspective générale de ces travaux est la mise en œuvre d'un système complet de réalité augmentée, répondant au besoin identifié. Nous sommes convaincu qu'un système utilisant une localisation par SLAM contraint, une reconstruction 3D basée sur notre mise en correspondance dense et un affichage semi-transparent de type « tablette augmentée » peut satisfaire les contraintes de toutes les applications envisagées, notamment les plus critiques (maintenance, navigation), et enfin valider le potentiel de la réalité augmentée comme outil utile et non plus restreint à des applications ludiques.

Figures et tableaux

Table des figures

Liste des tableaux

Bibliographie

Y. Abdel-Aziz et H. Karara : Direct linear transformation from comparator to object space coordinates in close-range photogrammetry. *In ASP Symposium on Close-Range Photogrammetry*, pages 1–18. American Society of Photogrammetry, 1971. *(cité p. 16)*

P. F. Alcantarilla et A. Bartoli : Deformable 3d reconstruction with an object database. *In British Machine Vision Conference (BMVC)*, pages 1–12, 2012. *(cité p. 135)*

L. Alvarez, R. Deriche, T. Papadopoulo et J. Sánchez : Symmetrical dense optical flow estimation with occlusions detection. *In European Conference on Computer Vision (ECCV)*, pages 721–735. Springer, 2002. *(cité p. 43)*

N. Anjum, M. Taj et A. Cavallaro : Relative position estimation of non-overlapping cameras. *In International Conference on Acoustics, Speech and Signal Processing (ICASP)*, volume 2, pages II–281–II–284. IEEE Signal Processing Society, 2007. *(cité p. 71)*

S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black et R. Szeliski : A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 2011. *(cité p. 37, 124 et 149)*

A. Bartoli, Y. Gérard, F. Chadebecq et T. Collins : On template-based reconstruction from a single view : Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. *In Computer Vision and Pattern Recognition (CVPR)*, pages 2026–2033. IEEE, 2012. *(cité p. 56)*

A. Bartoli, M. Perriollat et S. Chambon : Generalized thin-plate spline warps. *International Journal of Computer Vision*, 88(1):85–110, 2010. *(cité p. 40)*

H. Bay, T. Tuytelaars et L. Van Gool : SURF : Speeded up robust features. *In European Conference on Computer Vision (ECCV)*, 2006. *(cité p. 135)*

P. J. Besl et N. D. McKay : Method for registration of 3-d shapes. *In Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992. *(cité p. 59)*

F. L. Bookstein : Principal warps : Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11 (6):567–585, 1989. *(cité p. 40)*

G. Bradski : The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000. *(cité p. 34)*

J. Braux-Zin, A. Bartoli, R. Dupont et R. Vinciguerra : Calibrating an optical see-through rig with two non-overlapping cameras : The virtual camera framework. *In 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 308–315. IEEE, 2012. *(cité p. 3 et 69)*

J. Braux-Zin, A. Bartoli, R. Dupont, R. Vinciguerra *et al.* : Caméras virtuelles pour la calibration d'un système de réalité augmentée composé d'un écran transparent et deux caméras à champs disjoints. *In Orasis, Congrès des jeunes chercheurs en vision par ordinateur*, 2013a. *(cité p. 3 et 69)*

J. Braux-Zin, A. Bartoli, R. Dupont, R. Vinciguerra *et al.* : Caméras virtuelles pour l'étalonnage d'un système de réalité augmentée sur affichage semi-transparent. *Traitement du Signal*, 2014a. En cours d'édition. *(cité p. 3 et 69)*

J. Braux-Zin, R. Dupont et A. Bartoli : Combining features and intensity for wide-baseline non-rigid surface registration. *In British Machine Vision Conference (BMVC)*. BMVA, 2013b. *(cité p. 4 et 101)*

J. Braux-Zin, R. Dupont et A. Bartoli : A general dense image matching framework combining direct and feature-based costs. *In International Conference on Computer Vision (ICCV)*. IEEE, 2013c. *(cité p. 4 et 101)*

J. Braux-Zin, R. Dupont, A. Bartoli *et al.* : Cadre générique pour le recalage dense combinant un coût dense et et un coût basé sur des correspondances de primitives. *In Congrès national sur la Reconnaissance de Formes et l'Intelligence Artificielle (RFIA)*, 2014b. *(cité p. 4 et 101)*

K. Bredies : Recovering piecewise smooth multichannel images by minimization of convex functionals with total generalized variation penalty. *SFB Report*, 6, 2012. *(cité p. 121, 142 et 145)*

K. Bredies, K. Kunisch et T. Pock : Total generalized variation. *SIAM Journal on Imaging Sciences*, 2010. *(cité p. 39)*

T. Brox et J. Malik : Large displacement optical flow : descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. *(cité p. 102, 105, 106, 108, 109, 116, 119, 124 et 150)*

D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges et D. Kim : Shake'n'Sense : Reducing interference for overlapping structured light depth cameras. *In SIGCHI Conference on Human Factors in Computing Systems*, pages 1933–1936. ACM, 2012a. *(cité p. 55)*

D. J. Butler, J. Wulff, G. B. Stanley et M. J. Black : A naturalistic open source movie for optical flow evaluation. *In European Conference on Computer Vision (ECCV)*, pages 611–625, 2012b. *(cité p. 124)*

J. Canny : A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov 1986. *(cité p. 29)*

G. Carrera, A. Angeli et A. J. Davison : Slam-based automatic extrinsic calibration of a multi-camera rig. *In International Conference on Robotics and Automation (ICRA)*, pages 2652–2659. IEEE, 2011. *(cité p. 71)*

Y. Caspi et M. Irani : Aligning non-overlapping sequences. *International Journal of Computer Vision*, 48(1):39–51, 2002. *(cité p. 71)*

A. Chambolle et T. Pock : A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 2011. *(cité p. 121, 123 et 144)*

Y. Chen et G. Medioni : Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. *(cité p. 59)*

D. Comaniciu, V. Ramesh et P. Meer : Real-time tracking of non-rigid objects using mean shift. *In Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 142–149. IEEE Computer Society, 2000. *(cité p. 88)*

N. Dalal et B. Triggs : Histograms of oriented gradients for human detection. *In Computer Vision and Pattern Recognition (CVPR)*, 2005. *(cité p. 31)*

A. J. Davison, I. D. Reid, N. D. Molton et O. Stasse : Monoslam : Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (6):1052–1067, 2007. *(cité p. 60)*

P. Dérian, P. Héas, C. Herzet et E. Mémin : Wavelets and optical flow motion estimation. *Numerical Mathematics : Theory, Methods and Applications*, 6(1):116–137, January 2013. *(cité p. 41)*

R. C. Eberhart et Y. Shi : Comparing inertia weights and constriction factors in particle swarm optimization. *In Congress on Evolutionary Computation*, volume 1, pages 84–88. IEEE, 2000. *(cité p. 123)*

S. Esquivel, F. Woelk et R. Koch : Calibration of a multi-camera rig from non-overlapping views. *In* F. Hamprecht, C. Schnörr et B. Jähne, éditeurs : *Pattern Recognition*, volume 4713 de *Lecture Notes in Computer Science*, pages 82–91. Springer, Berlin / Heidelberg, 2007. ISBN 978-3-540-74933-2. *(cité p. 71)*

O. Faugeras : *Three-dimensional computer vision : a geometric viewpoint*. MIT press, 1993. *(cité p. 22 et 27)*

V. Ferrari, T. Tuytelaars et L. Van Gool : Simultaneous object recognition and segmentation by image exploration. *In European Conference on Computer Vision (ECCV)*, pages 40–54. Springer, 2004. *(cité p. 138, 140 et 151)*

M. A. Fischler et R. C. Bolles : Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. *(cité p. 49 et 108)*

Y. Furukawa et J. Ponce : Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010. *(cité p. 102, 103 et 150)*

R. Garg, A. Roussos et L. Agapito : Dense variational reconstruction of non-rigid surfaces from monocular video. *In Computer Vision and Pattern Recognition (CVPR)*, pages 1272–1279. IEEE, 2013a. *(cité p. 56)*

R. Garg, A. Roussos et L. Agapito : A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 2013b. *(cité p. 102)*

V. Gay-Bellile, A. Bartoli et P. Sayd : Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. *(cité p. 43, 102, 112, 114, 137, 138, 139, 140, 141 et 152)*

A. Geiger, P. Lenz et R. Urtasun : Are we ready for autonomous driving ? The KITTI vision benchmark suite. *In CVPR*, 2012. *(cité p. 124)*

G. Graber : Realtime 3D reconstruction. Mémoire de D.E.A., Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria, 2011. *(cité p. 54)*

D. Hafner, O. Demetz et J. Weickert : Why is the census transform good for robust optic flow computation ? *In Scale Space and Variational Methods in Computer Vision*, pages 210–221. Springer, 2013. *(cité p. 30)*

R. W. Hamming : Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950. *(cité p. 31)*

B. M. Haralick, C.-N. Lee, K. Ottenberg et M. Nölle : Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994. *(cité p. 23)*

C. Harris et C. Stennett : Rapid-a video rate object tracker. *In British Machine Vision Conference (BMVC)*, pages 1–6, 1990. *(cité p. 34)*

C. Harris et M. Stephens : A combined corner and edge detector. *In Alvey vision conference*, 1988. *(cité p. 28)*

R. I. Hartley et A. Zisserman : *Multiple View Geometry in Computer Vision*. Cambridge University Press, second édition, 2004. ISBN 0521540518. *(cité p. 15, 18, 19, 22 et 84)*

R. I. Hartley : In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997. *(cité p. 20 et 22)*

K. Hirose et H. Saito : Fast line description for line-based slam. *In British Machine Vision Conference (BMVC)*, pages 1–11, 2012. *(cité p. 36)*

H. Hirschmuller : Accurate and efficient stereo processing by semi-global matching and mutual information. *In Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 807–814. IEEE, 2005. *(cité p. 101)*

B. K. Horn : Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987. *(cité p. 82)*

B. K. P. Horn et B. G. Schunck : Determining optical flow. *Artificial Intelligence*, 1981. *(cité p. 38)*

Juniper Research : Over 2.5 billion mobile augmented reality apps to be installed per annum by 2017. Press Release, 2012. *(cité p. 63)*

R. E. Kalman : A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960. *(cité p. 60)*

K. Kanatani : *Group-theoretical methods in image understanding*, volume 2. springer-Verlag Berlin, 1990. *(cité p. 51)*

H. Kato et M. Billinghurst : Marker tracking and hmd calibration for a video-based augmented reality conferencing system. *In International Workshop on Augmented Reality*, pages 85–94. IEEE & ACM, 1999. *(cité p. 58)*

K. Kiyokawa, Y. Kurata et H. Ohno : An optical see-through display for mutual occlusion with a real-time stereovision system. *Computers & Graphics*, 25(5):765–779, 2001. *(cité p. 95, 96 et 150)*

G. Klein et D. Murray : Parallel Tracking And Mapping for small ar workspaces. *In Mixed and Augmented Reality (ISMAR)*, pages 225–234. IEEE, 2007. *(cité p. 61)*

R. K. Kumar, A. Ilie, J.-M. Frahm et M. Pollefeys : Simple calibration of non-overlapping cameras with a mirror. *In Computer Vision and Pattern Recognition (CVPR)*, pages 1–7. IEEE Computer Society, 2008. *(cité p. 72)*

J.-M. Lavest, M. Viala et M. Dhome : Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? *In European Conference on Computer Vision (ECCV)*, pages 158–174. Springer, 1998. *(cité p. 14)*

P. Lebraly, C. Deymier, O. Ait-Aider, E. Royer et M. Dhome : Flexible extrinsic calibration of non-overlapping cameras using a planar mirror : Application to vision-based robotics. *In International Conference on Intelligent Robots and Systems (IROS)*, pages 5640–5647. IEEE/RSJ, 2010. *(cité p. 71 et 74)*

P. Lébraly, E. Royer, O. Ait-Aider, C. Deymier et M. Dhome : Fast calibration of embedded non-overlapping cameras. *In International Conference on Robotics and Automation (ICRA)*, pages 221–227. IEEE Robotics & Automation Society, 2011. ISBN 978-1-61284-386-5. *(cité p. 71)*

S. Lee, G. Wolberg, K. Chwa et S. Shin : Image metamorphosis with scattered feature constraints. *Visualization and Computer Graphics*, 1996. *(cité p. 40)*

T. Lemaire, C. Berger, I.-K. Jung et S. Lacroix : Vision-based slam : Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364, 2007. *(cité p. 60)*

M. Leordeanu, R. Sukthankar et C. Sminchisescu : Efficient closed-form solution to generalized boundary detection. *In European Conference on Computer Vision (ECCV)*, pages 516–529. Springer, 2012. *(cité p. 142)*

M. Leordeanu, A. Zanfir et C. Sminchisescu : Locally affine sparse-to-dense matching for motion and occlusion estimation. *In International Conference on Computer Vision (ICCV)*, 2013. *(cité p. 41, 102 et 105)*

V. Lepetit et P. Fua : Monocular model-based 3d tracking of rigid objects : A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005. *(cité p. 58)*

V. Lepetit, F. Moreno-Noguer et P. Fua : Epnp : An accurate o (n) solution to the PnP problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. *(cité p. 23)*

C. Liu, J. Yuen et A. Torralba : SIFT flow : Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. *(cité p. 102, 103, 142 et 150)*

D. G. Lowe : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004. *(cité p. 29, 31 et 36)*

C.-P. Lu, G. D. Hager et E. Mjolsness : Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, 2000. *(cité p. 24)*

B. D. Lucas et T. Kanade : An iterative image registration technique with an application to stereo vision. *In DARPA Image Understanding Workshop*, 1981. *(cité p. 28)*

D. W. Marquardt : An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2):431–441, 1963. *(cité p. 47)*

D. Marr et E. Hildreth : Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980. *(cité p. 29)*

X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang et X. Zhang : On building an accurate stereo matching system on graphics hardware. *In Third ICCV Workshop on GPUs for Computer Vision*, 2011. *(cité p. 101)*

M. Meilland et A. Comport : On unifying key-frame and voxel-based dense visual slam at large scales. *In International Conference on Intelligent Robots and Systems (IROS)*, pages 3677–3683, 2013. *(cité p. 54)*

K. Mikolajczyk et C. Schmid : A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, Oct 2005. *(cité p. 32)*

K. Mikolajczyk et C. Schmid : Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004. *(cité p. 29)*

E. Mouragnon : *Reconstruction 3D et localisation similtanée de caméras mobiles : une approche temps-réel par ajustement de faisceaux local*. Thèse de doctorat, Clermont-Ferrand 2, 2007. Dirigée par Michel Dhome, Vision pour la Robotique. *(cité p. 60)*

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser et P. Sayd : Real time localization and 3D reconstruction. *In Computer Vision and Pattern Recognition (CVPR)*, 2006. *(cité p. 61)*

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges et A. Fitzgibbon : Kinectfusion : Real-time dense surface mapping and tracking. *In Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011a. *(cité p. 54, 59, 62 et 142)*

R. A. Newcombe, S. J. Lovegrove et A. J. Davison : DTAM : Dense tracking and mapping in real-time. *In International Conference on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011b. *(cité p. 54, 59 et 62)*

D. Nistér : An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. *(cité p. 20 et 22)*

D. Nister et H. Stewenius : Scalable recognition with a vocabulary tree. *In Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168. IEEE, 2006. *(cité p. 59)*

P. Ochs, Y. Chen, T. Brox et T. Pock : iPiano : Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences (SIIMS)*, 2014. Preprint. *(cité p. 113)*

T. Ojala, M. Pietikainen et T. Maenpaa : Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. *(cité p. 31)*

J. Oliensis et R. Hartley : Iterative extensions of the Sturm/Triggs algorithm : Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2217–2233, 2005. *(cité p. 84)*

J. Pilet, V. Lepetit et P. Fua : Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008. *(cité p. 102, 109, 118, 119 et 120)*

D. Pizarro et A. Bartoli : Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 2012. *(cité p. 102, 104, 108, 135, 138, 139, 140, 141 et 150)*

J. Platonov, H. Heibel, P. Meier et B. Grollmann : A mobile markerless ar system for maintenance and repair. *In Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 105–108. IEEE Computer Society, 2006. *(cité p. 59)*

T. Pock et A. Chambolle : Diagonal preconditioning for first order primal-dual algorithms in convex optimization. *In International Conference on Computer Vision (ICCV)*, pages 1762–1769, 2011. *(cité p. 121 et 146)*

V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer et S. Bathiche : Monofusion : Real-time 3d reconstruction of small scenes with a single web camera. *In Mixed and Augmented Reality (ISMAR)*, pages 83–88, 2013. *(cité p. 62)*

M. Prasad et A. Fitzgibbon : Single view reconstruction of curved surfaces. *In Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1345–1354. IEEE, 2006. *(cité p. 137)*

C. Rabe, T. Müller, A. Wedel et U. Franke : Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. *In European Conference on Computer Vision (ECCV)*, pages 582–595. Springer, 2010. *(cité p. 56 et 149)*

A. Rahimi, B. Dunagan et T. Darrell : Simultaneous calibration and tracking with a network of non-overlapping sensors. *In Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2004. *(cité p. 71)*

R. Ranftl, S. Gehrig, T. Pock et H. Bischof : Pushing the limits of stereo using variational stereo estimation. *In Intelligent Vehicles Symposium (IV)*, 2012. *(cité p. 30, 101, 121, 122, 144 et 145)*

R. Rodrigues, J. Barreto et U. Nunes : Camera pose estimation using images of planar mirror reflections. *In European Conference on Computer Vision (ECCV)*, pages 382–395. Springer, 2010. *(cité p. 72, 92, 93, 94 et 150)*

C. Rother, V. Kolmogorov, V. Lempitsky et M. Szummer : Optimizing binary MRFs via extended roof duality. *In Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. *(cité p. 105)*

L. Rudin, S. Osher et E. Fatemi : Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 1992. *(cité p. 38)*

M. Salzmann et P. Fua : Reconstructing sharply folding surfaces : A convex formulation. *In Computer Vision and Pattern Recognition (CVPR)*, pages 1054–1061. IEEE, 2009. *(cité p. 139 et 151)*

M. Salzmann, R. Hartley et P. Fua : Convex optimization for deformable surface 3-d tracking. *In International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007. *(cité p. 138, 139 et 151)*

J. Schmidt et H. Niemann : Using quaternions for parametrizing 3-d rotations in unconstrained nonlinear optimization. *In Vision Modeling and Visualization Conference*, volume 1, pages 399–406, 2001. *(cité p. 52)*

J. Shi et C. Tomasi : Good features to track. *In Computer Vision and Pattern Recognition (CVPR)*, pages 593–600. IEEE, 1994. *(cité p. 28)*

N. Snavely, S. M. Seitz et R. Szeliski : Photo tourism : Exploring photo collections in 3d. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3):835–846, 2006. *(cité p. 101)*

F. Steinbrucker, J. Sturm et D. Cremers : Real-time visual odometry from dense rgb-d images. *In ICCV Workshop on Live Dense Reconstruction with Moving Cameras*, pages 719–722. IEEE, 2011. *(cité p. 59)*

F. Steinbruecker, J. Sturm et D. Cremers : Volumetric 3d mapping in real-time on a cpu. *In International Conference on Robotics and Automation (ICRA)*, Hongkong, China, 2014. *(cité p. 54, 62 et 142)*

H. Strasdat, A. Davison, J. M. M. Montiel et K. Konolige : Double window optimisation for constant time visual slam. *In International Conference on Computer Vision (ICCV)*, pages 2352–2359, 2011. *(cité p. 61)*

H. Strasdat, J. M. M. Montiel et A. Davison : Real-time monocular slam : Why filter ? *In International Conference on Robotics and Automation (ICRA)*, pages 2657–2664, 2010. *(cité p. 61)*

P. Sturm et T. Bonfort : How to compute the pose of an object without a direct view ? *In Asian Conference on Computer Vision (ACCV)*, volume 2 de *Lecture Notes in Computer Science*, pages 21–31. Springer, 2006. *(cité p. 72)*

D. Sun, S. Roth et M. Black : A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014. *(cité p. 43 et 121)*

T. Svoboda, D. Martinec et T. Pajdla : A convenient multicamera self-calibration for virtual environments. *PRESENCE : teleoperators and virtual environments*, 14 (4):407–422, 2005. *(cité p. 71)*

K. Takahashi, S. Nobuhara et T. Matsuyama : A new mirror-based extrinsic camera calibration using an orthogonality constraint. *In Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2012. *(cité p. 72, 92 et 93)*

M. Tamaazousti, V. Gay-Bellile, S. Collette, S. Bourgeois et M. Dhome : Nonlinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment. *In Computer Vision and Pattern Recognition (CVPR)*, pages 3073–3080, 2011. *(cité p. 3, 62 et 67)*

M. Tamaazousti : *L'ajustement de faisceaux contraint comme cadre d'unification des méthodes de localisation : application à la réalité augmentée sur des objets 3D*. Thèse de doctorat, Clermont-Ferrand 2, 2013. Dirigée par Michel Dhome, Vision pour la Robotique. *(cité p. 62)*

A. Tang : Evaluation of calibration procedures for optical see-through head-mounted displays. *In Mixed and Augmented Reality (ISMAR)*, page 161. IEEE Computer Society, 2003. *(cité p. 75)*

J.-P. Tardif, Y. Pavlidis et K. Daniilidis : Monocular visual odometry in urban environments using an omnidirectional camera. *In International Conference on Intelligent Robots and Systems (IROS)*, pages 2531–2538. IEEE, 2008. *(cité p. 24)*

M. Tidwell, R. S. Johnston, D. Melville et T. A. Furness : The virtual retinal display-a retinal scanning imaging system. *In Virtual reality World*, volume 95, pages 325–333, 1995. *(cité p. 98)*

E. Tola, V. Lepetit et P. Fua : Daisy : An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. *(cité p. 32, 33, 42, 102, 110, 133, 134, 135 et 151)*

F. Tombari, A. Franchi et L. Di Stefano : BOLD features to detect texture-less objects. *In International Conference on Computer Vision (ICCV)*, 2013. *(cité p. 36 et 149)*

P. H. Torr et D. W. Murray : The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997. *(cité p. 20)*

L. Torresani, A. Hertzmann et C. Bregler : Nonrigid structure-from-motion : Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008. *(cité p. 56)*

Q.-H. Tran, T.-J. Chin, G. Carneiro, M. S. Brown et D. Suter : In defence of RANSAC for outlier rejection in deformable registration. *In European Conference on Computer Vision (ECCV)*, pages 274–287. Springer, 2012. *(cité p. 102, 104, 108, 139, 141 et 150)*

B. Triggs : Factorization methods for projective structure and motion. *In Computer Vision and Pattern Recognition (CVPR)*, pages 845–851. IEEE Computer Society, 1996. *(cité p. 84)*

M. Unger, M. Werlberger, T. Pock et H. Bischof : Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. *In Computer Vision and Pattern Recognition (CVPR)*, pages 1878–1885. IEEE, 2012. *(cité p. 41)*

K. E. van de Sande, T. Gevers et C. G. Snoek : Color descriptors for object category recognition. *In Conference on Colour in Graphics, Imaging, and Vision*, pages 378–381. Society for Imaging Science and Technology, 2008. *(cité p. 32)*

P. Viola et M. Jones : Rapid object detection using a boosted cascade of simple features. *In Computer Vision and Pattern Recognition (CVPR)*, pages I–511–I–518. IEEE Computer Society, 2001. *(cité p. 88)*

R. G. Von Gioi, J. Jakubowicz, J.-M. Morel et G. Randall : Lsd : a line segment detector. *Image Processing On Line*, 2012. *(cité p. 29)*

C. Vondrick, A. Khosla, T. Malisiewicz et A. Torralba : HOGgles : Visualizing object detection features. *International Conference on Computer Vision (ICCV)*, 2013. *(cité p. 107 et 150)*

L. Wang, U. Neumann et S. You : Wide-baseline image matching using line signatures. *In International Conference on Computer Vision (ICCV)*, 2009a. *(cité p. 36, 110, 133 et 151)*

L. Wang, S. You et U. Neumann : Supporting range and segment-based hysteresis thresholding in edge detection. *In International Conference on Image Processing (ICIP)*, pages 609–612. IEEE, 2008. *(cité p. 29)*

Z. Wang, F. Wu et Z. Hu : MSLD : A robust descriptor for line matching. *Pattern Recognition*, 42(5):941–953, 2009b. *(cité p. 36)*

A. Wedel, T. Pock, C. Zach, H. Bischof et D. Cremers : An improved algorithm for TV-L 1 optical flow. *In Statistical and Geometrical Approaches to Visual Motion Analysis*. Springer, 2009. *(cité p. 42 et 115)*

P. Weinzaepfel, J. Revaud, Z. Harchaoui et C. Schmid : DeepFlow : Large displacement optical flow with deep matching. *In International Conference on Computer Vision (ICCV)*, 2013. *(cité p. 129, 130 et 142)*

M. Werlberger : *Convex Approaches for High Performance Video Processing*. Thèse de doctorat, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria, June, 2012. *(cité p. 121)*

M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers et H. Bischof : Anisotropic Huber-L1 optical flow. *In British Machine Vision Conference (BMVC)*, 2009. *(cité p. 42)*

T. Whelan, M. Kaess, J. Leonard et J. McDonald : Deformation-based loop closure for large scale dense RGB-D SLAM. *In International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013. *(cité p. 63)*

J. Wills, S. Agarwal et S. Belongie : A feature-based approach for dense segmentation and estimation of large disparity motion. *International Journal of Computer Vision*, 2006. *(cité p. 41, 102 et 105)*

G. Wyszecki et W. S. Stiles : *Color science*. Wiley New York, 1982. *(cité p. 34)*

L. Xu, J. Jia et Y. Matsushita : Motion detail preserving optical flow estimation. *In Computer Vision and Pattern Recognition (CVPR)*, 2010. *(cité p. 102, 105, 106, 116 et 150)*

G. Yu et J.-M. Morel : ASIFT : An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 2011. *(cité p. 32)*

R. Zabih et J. Woodfill : Non-parametric local transforms for computing visual correspondence. *In European Conference on Computer Vision (ECCV)*, 1994. *(cité p. 30)*

Z. Zhang : A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. *(cité p. 16)*