Numéro d'ordre : 40180

THÈSE

Présentée et soutenue publiquement par

Yara BACHALANY

le 18/12/2009

pour obtenir le grade de

Docteur de l'Université Lille 1, Sciences et Technologies

en *Automatique, Génie Informatique, Traitement du Signal et des Images*

Estimation du mouvement 3D d'une sphère de surface réfléchissante

P. Bonton	Rapporteur	Professeur à l'Université Blaise Pascal, Clermont-Ferrand
M. Rudko	Rapporteur	Professeur à Union College, Schenectady, NY, USA
O. Colot	Examinateur	Professeur à l'Université de Lille1, Villeneuve d'Ascq Cedex
P. Sayd	Examinateur	Chargé de recherche au CEA, SACLAY
S. Ambellouis	Co-directeur	Chargé de recherche à l'INRETS, Villeneuve d'Ascq Cedex
F. Cabestaing	Co-directeur	Professeur à l'Université de Lille1, Villeneuve d'Ascq Cedex

Thèse préparée au **L**aboratoire d'**A**utomatique, **G**énie **I**nformatique & **S**ignal

LAGIS - UMR CNRS 8146

Remerciements

Le travail présenté dans ce mémoire a été réalisé au Laboratoire d'Automatique, Génie Informatique et Signal de l'Université des Sciences et Technologies de Lille dirigé par Monsieur François Cabestaing, Professeur à l'Université des Sciences et Technologies de Lille et Monsieur Sébastien Ambellouis, Chargé de Recherche à l'INRETS. Je les remercie pour leurs conseils, leurs encouragements et leur grande disponibilité qui m'ont permis de mener à bien ce travail.

Je suis très honorée que Monsieur Pierre Bonton, Professeur à l'Université Blaise Pascal et Monsieur Mickael Rudko, Professeur à Union College, Schenectady, NY aient accepté de rapporter mon travail. Je les remercie pour la rapidité avec laquelle ils ont lu mon manuscrit et l'intérêt qu'ils ont porté à mon travail.

Je tiens aussi à assurer de ma reconnaissance Monsieur Patrick Sayd, Chargé de recherche au CEA, Saclay qui a accepté de juger mon mémoire et participer à mon jury.

Je voudrais tout particulièrement exprimer ma gratitude à Monsieur Olivier Colot, Professeur à l'Université des Sciences et Technologies de Lille qui m'a fait l'honneur de présider ce jury.

Je remercie enfin toute l'équipe de chercheurs du LAGIS pour leur présence amicale et le soutien qu'ils m'ont apporté.

Table des matières

Table des figures

Introduction

I.1 Positionnement du problème

La recherche décrite dans cette thèse concerne l'estimation des paramètres 3D du mouvement d'un objet non déformable dans une séquence d'images 2D. Cette problématique est primordiale dans de nombreux contextes applicatifs et est fortement étudiée par la communauté scientifique depuis plusieurs dizaines d'années. Parmi les applications les plus courantes, nous pouvons citer le suivi de la trajectoire et de la pose d'un objet en mouvement dans une scène et plus indirectement la compression des séquences d'images.

La grande majorité des méthodes d'estimation proposées dans la littérature supposent que, localement, les variations spatio-temporelles de la luminosité entre deux images successives d'une séquence dépendent directement du mouvement de l'objet dans la scène filmée. Les méthodes les plus usuelles estiment les paramètres 3D des deux manières suivantes :

- soit en une seule étape grâce à la minimisation d'une fonction de coût exprimée à partir des variations spatio-temporelles de la luminosité ;
- soit en deux étapes, en analysant la déformation, au cours du temps, du champ apparent des vecteurs vitesse préalablement calculés.

Toutefois, ces méthodes sont soumises aux difficultés suivantes.

Les images d'un objet perçu par une caméra sont produites après projection de la scène sur la surface photo-sensible : un point de l'objet repéré par trois coordonnées dans la scène réelle se trouve alors associé à un pixel de l'image repéré par deux coordonnées. Ceci constitue une perte de l'information de profondeur et certains mouvements de rotation et de translation deviennent alors ambigüs.

La plupart du temps, les surfaces des objets sont supposées lambertiennes et

la luminosité de l'objet projeté dans les images ne varie donc pas lors de son déplacement. Cette propriété permet de définir ce que nous appelons l'équation de contrainte du "flot optique". Par ailleurs, parce que ces méthodes proposent une analyse locale des propriétés spatio-temporelles de la luminosité, seule la composante dans la direction au gradient de l'intensité lumineuse peut être estimée. Cette difficulté est connue sous le nom de "problème d'ouverture".

Afin de pallier ces difficultés, plusieurs solutions ont été étudiées. Certaines agissent en ajoutant des contraintes supplémentaires sur la nature du champ apparent des vecteurs vitesse ou du mouvement 3D de l'objet et d'autres proposent d'augmenter la taille du voisinage sur lequel l'analyse des variations spatio-temporelle de l'intensité lumineuse est effectuée. Malheureusement, un voisinage trop grand et de forme inadaptée nécessite une contrainte plus complexe sur la nature du champ apparent des vecteurs vitesse ou du mouvement 3D et s'avère contre-performant en matière de temps de calcul.

Dans ce mémoire, nous décrivons une méthode hybride d'estimation fondée sur la connaissance d'un modèle global de l'objet composé d'un modèle géométrique 3D et de son modèle de texture et d'une mise en correspondance 2D/2D issue des travaux de Lucas et Kanade. Le principe est d'estimer les paramètres 3D du mouvement en mettant en correspondance les images de synthèse calculées à partir de ce modèle et chaque image de la séquence. L'estimation finale est déterminée par la transformation appliquée au modèle afin d'obtenir une image synthétique semblable à chaque image de la séquence.

La première originalité est que nous avons associé cette méthode à la méthode classique de Lucas-Kanade afin d'éviter de produire systématiquement une image de synthèse et de réduire les temps de calcul. La seconde originalité de notre travail est d'avoir pris en compte des objets de "'surface réfléchissante"'. Dans ce cas, pour un même mouvement 3D de l'objet le champ apparent des vecteurs vitesse est différent selon que la texture de l'objet est celle d'une réflexion spéculaire ou celle d'une réflexion diffuse. Par exemple, pour un objet sphérique de surface réfléchissante en rotation autour de l'un de ses axes, alors que la caméra et la source d'éclairage sont fixes, le reflet apparaît statique. Dans ce cas, un algorithme ne tenant pas

compte de cette particularité aboutit à une estimation erronée des paramètres du mouvement.

Cette recherche se place dans le cadre du développement d'une application de suivi de la posture et de la trajectoire d'une libellule lors de la capture d'une proie. Au moins trois étapes composent son vol. La première est la décision de décollage, la deuxième est la navigation vers une trajectoire d'interception et la troisième est la coordination des mouvements des pattes dans l'espace et le temps afin d'agripper sa proie. Les biologistes étudient ce processus en émettant des hypothèses qu'ils tentent de vérifier grâce à l'observation de séquences vidéo. A ce jour, ce travail est effectué de manière manuelle, ce qui en fait un travail fastidieux et sujet à certaines imprécisions. L'équipe VI du laboratoire LAGIS a donc proposé une automatisation de l'analyse des séquences d'images et tout particulièrement de l'estimation des paramètres 3D du mouvement de l'insecte. Le lecteur pourra se référer au tome 2 de ce mémoire et tout particulièrement :

- au chapitre 1 qui présente respectivement quelques éléments sur le contexte expérimental, sur la méthode d'analyse manuelle et sur les premières déductions faites par les biologistes ;
- au chapitre 2 qui détaille les conclusions d'une analyse préliminaire des séquences vidéo des vols de la libellule. Elle nous a permis de définir les défis que pose le problème de l'estimation du mouvement 3D de la tête ainsi que des autres membres de la libellule et de proposer notre méthode hybride d'estimation automatique du mouvement.

Pour résumer, nous avons proposé notre méthode hybride afin de répondre aux deux difficultés suivantes :

- la variation temporelle de l'intensité lumineuse de la libellule en tout point de l'image, variations dues à la nature réfléchissante de la surface et au phénomène de pseudopupille.
- un déplacement de grande amplitude (plus de dix pixels) entre deux images successives.

Face à la difficulté d'obtenir un modèle géométrique, un modèle cinématique et un modèle dynamique complets de la libellule, nous nous sommes concentrés

sur le suivi de la posture et de la trajectoire de la tête de l'insecte. Nous avons assimilé cette tête à une sphère munie de certaines propriétés de réflexion. Tous les résultats présentés dans ce mémoire ont donc été obtenus sur une séquence d'images d'une sphère réfléchissante en mouvement.

I.2 Organisation du mémoire

Dans le premier chapitre, nous présentons un aperçu des méthodes d'estimation du mouvement. Ces méthodes sont divisées en deux grandes familles : les méthodes exploitant uniquement les images et les méthodes exploitant un modèle. D'une part, les méthodes exploitant uniquement les images comportent les méthodes d'estimation du flot optique dont la méthode de Lucas/Kanade fait partie. Ces méthodes exploitent une information de texture lors de l'estimation du mouvement. D'autre part, les méthodes exploitant un modèle sont divisées en deux parties : celles exploitant un modèle 3D et celles exploitant un modèle 2D. Ces modèles sont généralement déduits des contours externes de l'objet à suivre.

Dans le deuxième chapitre, nous détaillons les deux méthodes de la littérature dont notre approche s'inspire : la méthode de mise en correspondance 3D/2D et la méthode de Lucas/Kanade. Ensuite, à l'aide d'exemples, nous présentons les limites de ces deux méthodes lors de l'estimation du mouvement 3D d'une sphère de surface réfléchissante.

Dans le troisième chapitre, nous présentons notre approche. Nous reprenons les exemples considérés au chapitre 2 afin de montrer que la propriété réfléchissante de la surface n'est plus considérée comme un obstacle mais que, au contraire, cette propriété procure une information supplémentaire sur le mouvement recherché.

Dans le dernier chapitre, nous avons présenté des résultats expérimentaux obtenus sur des séquences d'images synthétiques. Nous commençons par une étude comparative de notre approche avec les deux méthodes de la littérature détaillées dans le chapitre 2 (celle de Lucas-Kanade et la mise en correspondance 3D/2D) pour différentes méthodes de choix de patchs. Ensuite, nous appliquons notre technique à des paires d'images successives, dans des situations où le mouvement est élémentaire mais d'amplitude croissante afin de déterminer les marges de bon fonction-

nement de notre algorithme. Enfin, nous présentons quelques résultats d'analyse de séquences synthétiques contenant plusieurs dizaines d'images afin de vérifier la stabilité de notre méthode.

Ce mémoire se conclura par une synthèse de cette recherche et par quelques perspectives.

Chapitre 1

État de l'art : estimation du mouvement 3D

1.1 Introduction

L'estimation des six paramètres du mouvement 3D d'un objet rigide par analyse d'une séquence d'images demeure un problème fortement exploré par les chercheurs en vision par ordinateur. Parmi les applications les plus courantes de cette technique, citons entre autres : la télédétection, l'amélioration de la sécurité par vidéo-surveillance, la surveillance de la circulation routière, l'incrustation d'images en temps-réel pour les effets spéciaux en cinéma numérique, ou encore l'imagerie médicale interventionnelle.

Nombreux sont les défis que doivent relever ces méthodes d'analyse, qui disposent à la base des images fournies par une ou plusieurs caméras fixes ou mobiles. Selon l'application, de nombreuses caractéristiques de la scène observée peuvent varier du tout au tout : le nombre d'objets suivis, leurs propriétés géométriques, leurs couleurs et textures, l'information disponible *a priori* sur l'environnement et les conditions d'acquisition (éclairage contrôlé ou non, caractéristiques intrinsèques et extrinsèques des caméras, etc.).

Les caractéristiques du système assurant l'analyse sont également très variables : cadence d'acquisition des images, puissance de calcul disponible pour réaliser les traitements, précision requise sur les résultats, etc. De nouvelles techniques émergent donc continuellement pour tenter de résoudre ce problème.

Dans ce chapitre, nous décrivons les principales méthodes de reconstruction du mouvement, soit dans l'image, soit dans la scène, à partir d'une séquence monoculaire. Dans une première partie, nous détaillerons les méthodes qui exploitent

uniquement le contenu de l'image. La seconde partie est consacrée aux méthodes qui tirent parti de la connaissance *a priori* d'un modèle pour analyser plus finement le mouvement des objets. Enfin, nous concluons en constatant que très peu de méthodes permettent de traiter le cas d'images d'objets dont la surface est réfléchissante.

1.2 Quelques notations

Une image peut en premier lieu être considérée comme une fonction scalaire de trois variables réelles, les deux premières représentant les coordonnées d'un *point* du plan image, la troisième représentant le temps :

$$\mathcal{I}(x, y, t), \text{ avec } x, y, t \in \mathbb{R} \ .$$

Physiquement, la valeur de cette fonction scalaire mesure en général l'éclairement reçu par le point du capteur à l'instant t. Comme la plupart des auteurs, nous appellerons cette valeur *intensité* du point à l'instant t.

Dans certains cas, pour simplifier les notations, nous désignerons les deux variables d'espace par un seul vecteur $\mathbf{x}$ dont les coordonnées sont x et y, la fonction image devenant :

$$\mathcal{I}(\mathbf{x}, t) \ .$$

Quand l'image comporte plusieurs composantes, elle est représentée par une fonction vectorielle des trois mêmes variables réelles :

$$\mathbf{I}(x, y, t) = (\mathcal{I}^1(x, y, t) \ \cdots \ \mathcal{I}^c(x, y, t))^T \ ,$$

ou du vecteur $\mathbf{x}$ et du temps :

$$\mathbf{I}(\mathbf{x}, t) = (\mathcal{I}^1(\mathbf{x}, t) \ \cdots \ \mathcal{I}^c(\mathbf{x}, t))^T \ ,$$

dans lesquelles c désigne le nombre de composantes, par exemple 3 pour une image trichromatique standard. Physiquement, chaque composante mesure l'éclairement

reçu par le point du capteur à l'instant t, pondéré par la sensibilité spectrale du capteur acquérant cette composante.

Lorsque la coordonnée temporelle est discrétisée, par le biais d'un échantillonnage de période Δt, la fonction image scalaire à l'instant $n \cdot \Delta t$ sera notée :

$$I_n(x, y) = \mathcal{I}(x, y, n \cdot \Delta t), \text{ avec } n \in \mathbb{Z} \ ,$$

ou encore

$$I_n(\mathbf{x}) = \mathcal{I}(\mathbf{x}, n \cdot \Delta t) \ ,$$

quand les variables d'espace sont désignées par un vecteur.

Enfin, quand les coordonnées spatiales sont également discrétisées, selon une grille rectangulaire dont les pas d'échantillonnage sont Δx et Δy, le *niveau de gris* du *pixel* de coordonnées (i, j) est donné par la valeur de la fonction image scalaire au point de coordonnées $(i \cdot \Delta x, j \cdot \Delta y)$:

$$G_n(i, j) = I_n(i \cdot \Delta x, j \cdot \Delta y), \text{ avec } i, j \in \mathbb{Z} \ .$$

1.3 Méthodes exploitant uniquement les images

Les méthodes d'estimation du mouvement basées sur l'apparence (texture, intensité, couleur, etc.) sont nombreuses. Parmi les plus utilisées nous distinguons les méthodes d'estimation du flot optique. Dans [R3], le lecteur peut trouver une description d'un grand nombre des méthodes visant à estimer le flot optique. Dans [A5, C14, C20], sont rapportées des études portant sur les performances de quelques unes de ces approches sur un ensemble de séquences réelles et synthétiques vis à vis de la précision, de la robustesse, de la densité du champ de vecteurs vitesse calculé ainsi que la rapidité de calcul.

Le problème de l'estimation du flot optique fût largement exploré dès 1981 avec l'algorithme de Horn et Schunck [A20]. Baron *et coll.* ont classé les méthodes d'estimation du flot optique en quatre catégories [A5] : les méthodes d'appariement de blocs, les méthodes différentielles, enfin les méthodes de filtrage spatio-temporel elles-mêmes divisées en deux branches suivant que l'énergie ou la phase de la sor-

tie des filtres est analysée. C'est en suivant cette classification des approches que nous présenterons l'état de l'art.

Il faut souligner que la plupart des méthodes basées uniquement sur l'image visent principalement à estimer le mouvement apparent. Certaines reconstituent par la suite le mouvement 3D en analysant le champ des vitesses déterminé dans la première phase. En revanche, quelques méthodes introduisent directement des contraintes liées au mouvement 3D dans le processus d'estimation du mouvement.

Dans une première section, nous présentons les méthodes qui déterminent le mouvement directement dans le domaine spatio-temporel, c'est à dire sur la base des informations contenues dans la séquence d'images. Dans une deuxième section, nous décrivons les méthodes qui nécessitent de changer l'espace dans lequel les données image sont représentées, notamment par le biais d'une transformation temps-fréquence ou espace-fréquence comme la transformée de Fourier.

1.3.1 Flot optique et champ des déplacements

Le mouvement 3D des objets qui se déplacent dans la scène observée par une caméra entraîne la plupart du temps une modification de l'image de cette scène. Cette modification du contenu de l'image au cours du temps est appelée *mouvement apparent*. Pour caractériser ou quantifier le mouvement apparent dans une séquence d'images, on utilise les notions de *flot optique* et de *champ des déplacements*.

Dans le cadre d'une étude réalisée pour l'US Air Force, James Jerome Gibson tenta de décrire quels étaient les indices visuels qu'utilisaient les pilotes pour se guider lors de l'atterrissage. Il formula ces indices en termes de gradients optiques produits par la projection sur une surface 2D de configurations particulières d'objets mobiles. Il s'est penché sur un gradient en particulier : celui de la déformation apparente (sur la rétine) de la scène durant le mouvement de l'observateur. Il baptisa ce gradient *flot optique* (optical flow) [L3].

En vision par ordinateur, le flot optique est donc le champ dense des vitesses apparentes, donc dans le plan image, résultant du déplacement relatif de la caméra et de l'objet. Plus précisément, le flot optique associé à une image $\mathcal{I}(x, y, t)$ est par définition la projection dans le plan image du champ des vecteurs vitesse de chaque

point de la scène visible à l'instant t.

Si on tient compte du déplacement des objets plutôt que de leur vitesse, on aboutit à la notion de champ des déplacements. Par définition, le champ des déplacements associé à une paire d'images $\mathcal{I}(x, y, t_1)$ et $\mathcal{I}(x, y, t_2)$ est la projection dans le plan image du champ des vecteurs déplacement reliant les positions aux instants t_1 et t_2 de chaque point visible de la scène.

1.3.2 Approche intuitive : estimation par block-matching

La mise en correspondance de blocs, ou block-matching [T1, R1, A40, C29], est une technique d'estimation du mouvement apparent qui s'est surtout imposée dans les standards de compression de séquences d'images comme le MPEG [C13, A26].

Pour estimer le vecteur de mouvement apparent associé à un point de coordonnées (x, y) de l'image à l'instant t, dont l'intensité est $\mathcal{I}(x, y, t)$, on doit rechercher le point de coordonnées (x', y') dans l'image à l'instant t' dont l'intensité $\mathcal{I}(x', y', t')$ est la plus proche de $\mathcal{I}(x, y, t)$. Ce point est appelé correspondant du point initial, d'où l'appellation *mise en correspondance*. Le vecteur déplacement $\boldsymbol{\delta} = (\delta_x \ \delta_y)^T$, relie le point initial à son correspondant.

En pratique, une méthode aussi simple est rarement fiable, car de nombreux points ont des intensités très similaires dans la deuxième image. Il faut donc faire porter la comparaison des deux images non pas sur l'intensité d'un seul point, mais sur les intensités de tous les points contenus dans un voisinage du point considéré. Lorsque le voisinage est un rectangle centré sur le point initial, on parle alors de mise en correspondance de blocs [A42].

Ainsi, le déplacement associé au point de coordonnées (x, y), centre du bloc de dimension $Bx \times By$ dans l'image à l'instant t, est déterminé par la position (x', y') à l'instant t' du centre du bloc de même dimension, qui *ressemble* le plus au bloc initial. Une fois la correspondance trouvée, le déplacement est calculé par une simple différence des positions des centres des deux blocs : $\boldsymbol{\delta} = (x - x' \quad y - y')^T$.

La ressemblance est évaluée par une fonction de similarité, ou de dissimilarité, que nous détaillerons par la suite. La recherche du bloc le plus similaire est en général réalisée dans une zone située à proximité du bloc initial afin de limiter les

calculs. Cette zone est appelée *fenêtre de recherche*. Les coordonnées des déplacements pouvant être estimés sont donc souvent bornées par des valeurs liées à la dimension de la fenêtre de recherche [A41].

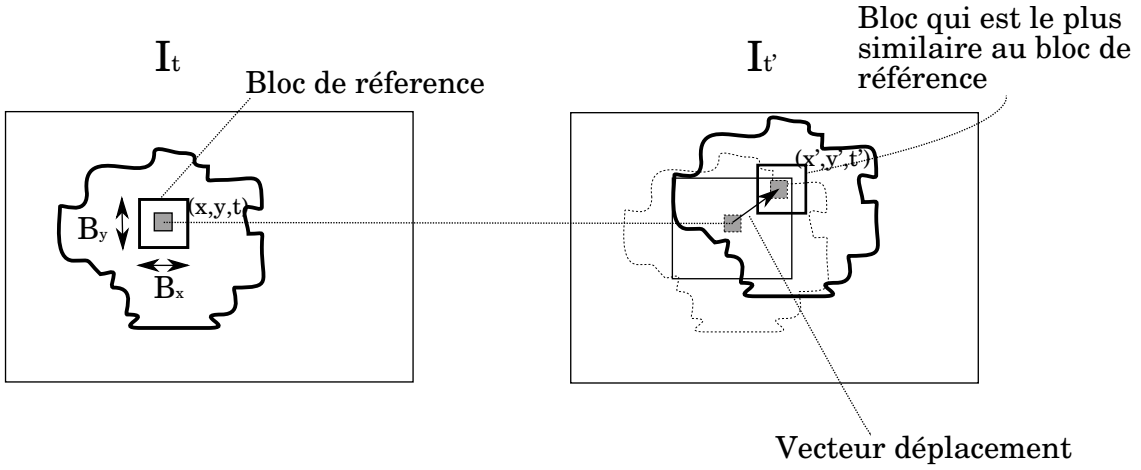


FIGURE 1.1 : Principe de la mise en correspondance de blocs

Les hypothèses principales autorisant la mise en correspondance de blocs sont au nombre de deux : l'invariance spatiale et l'invariance temporelle du mouvement apparent. La première impose la constance du mouvement apparent à l'intérieur du bloc 2D utilisé, et la deuxième la constance dans le temps de ce mouvement. En d'autres termes, le mouvement apparent doit globalement être de type translation uniforme. Quand ces hypothèses ne sont pas parfaitement satisfaites, le déplacement estimé par cette technique de mise en correspondance de blocs est entaché d'erreur.

1.3.3 Estimation dans le domaine spatio-temporel

Les hypothèses exposées ci-dessus, qui permettent la mise en correspondance de blocs, sont très restrictives. En premier lieu, elles ne considèrent que le problème de l'estimation du mouvement apparent, et non du mouvement 3D dans la scène. De plus, le mouvement apparent doit être une translation uniforme. Ces deux limitations peuvent être éliminées en reformulant le problème, tout en conservant le principe de comparaison du contenu de deux voisinages.

Afin de tenir compte des différents degrés de liberté associés au mouvement 3D, donc de lever la contrainte du mouvement apparent de type translation uniforme,

il faut considérer que le voisinage dans l'image à l'instant t' n'est pas simplement une translation du voisinage à l'instant t. On considère alors que chaque point $\mathbf{x} = (x\ y)^T$ du voisinage initial correspond à un point $\mathbf{x}' = (x'\ y')^T$ du nouveau voisinage, dont les coordonnées sont fonction de $\mathbf{x}$ et d'un certain nombre de paramètres du mouvement 3D. Dans la formulation générale exposée ci-dessous, les paramètres du mouvement sont désignés de façon implicite par un vecteur $\mathbf{p}$.

Afin de localiser, dans la nouvelle image, le voisinage le plus similaire au voisinage initial, il s'agit de trouver le minimum de la fonction de dissimilarité définie par l'expression suivante :

$$E(\mathbf{p}, \Omega) = \int_{\mathbf{x}\in\Omega} [\mathcal{I}(\mathbf{x}, t) - \mathcal{I}(\mathbf{W}(\mathbf{x}, \mathbf{p}), t')]^2 d\mathbf{x} \ . \tag{1.1}$$

Dans cette équation, la fonction $\mathbf{W}(\mathbf{x}, \mathbf{p})$ calcule les coordonnées du point sur lequel se projette, à l'instant t', le point de la scène qui se projetait sur $\mathbf{x}$ à l'instant t. Cette fonction peut être considérée comme une transformation paramétrique du plan image, ou encore comme une déformation (warping en anglais, d'où la notation $\mathbf{W}$), qui tient compte des paramètres du mouvement 3D, lesquels sont représentés par le vecteur $\mathbf{p}$.

Cette modélisation du mouvement apparent est celle utilisée par Lucas et Kanade [T4], dont les travaux seront détaillés par la suite. Il faut souligner qu'elle ne permet pas de représenter tous les mouvements possibles de l'objet dans l'espace 3D, car elle suppose que tous les points visibles à l'instant t le sont également à l'instant t'. A notre connaissance, il n'existe pas dans la littérature de formulation plus générale qui permettrait de lever cette restriction.

Dans l'équation (1.1), Ω désigne le voisinage sur lequel la fonction de dissimilarité est calculée. Ce voisinage n'est pas forcément rectangulaire comme l'est le bloc utilisé dans la méthode basique de mise en correspondance. Il faut noter que Ω est défini dans l'image à l'instant t et que le voisinage transformé dans l'image à l'instant t' n'a généralement pas la même forme que Ω.

Il n'est pas possible d'exploiter directement l'équation (1.1), du fait qu'elle est construite pour des images définies sur un espace continu, alors que les images disponibles dans un système de vision sont discrètes. Cependant, de nombreuses

méthodes décrites dans la littérature peuvent être considérées comme dérivées de cette formulation générale. Elles diffèrent selon : 1) la façon dont le mouvement 3D est modélisé, donc dans le calcul de la déformation du voisinage Ω ; 2) la façon dont le minimum de la fonction $E(\mathbf{p}, \Omega)$ est recherché.

Nous décrivons ci-après les trois approches les plus connues, à savoir : 1) les techniques de mise en correspondance de blocs, en nous appuyant cette fois sur le formalisme général exposé ci-dessus ; les méthodes différentielles d'estimation du flot optique suivant une approche 2) globale ou 3) locale.

1.3.3.1 Retour au block-matching

Dans les méthodes de mise en correspondance de blocs, l'équation (1.1) peut être modifiée ou adaptée de différentes façons afin d'estimer le mouvement apparent. (il n'existe pas à notre connaissance de technique de ce genre recherchant explicitement le mouvement 3D).

Pour tenir compte de l'échantillonnage temporel de la séquence d'images, l'équation (1.1) est en premier lieu modifiée afin de comparer deux images successives, acquises aux instants $n \cdot \Delta t$ et $(n+1) \cdot \Delta t$:

$$E(\mathbf{p}, \Omega) = \int_{\mathbf{x} \in \Omega} [I_n(\mathbf{x}) - I_{n+1}(\mathbf{W}(\mathbf{x}, \mathbf{p}))]^2 d\mathbf{x} \, . \tag{1.2}$$

Ensuite, il s'agit de tenir compte de l'échantillonnage spatial des images de la séquence. Le voisinage Ω peut alors être considéré comme un rectangle discret du plan image défini à partir d'un sous-ensemble de $\mathbb{Z}^2$, noté Ω_{i_0,j_0} lorsqu'il est centré sur (i_0, j_0), défini par :

$$\Omega_{i_0,j_0} = \{(i_0 + i, j_0 + j) \mid -B_x/2 \leq i \leq B_x/2 \, , -B_y/2 \leq j \leq B_y/2\} \, . \tag{1.3}$$

En utilisant le voisinage Ω_{i_0,j_0} pour calculer la dissimilarité, l'équation (1.2) devient :

$$E(\mathbf{p}, \Omega_{i_0,j_0}) = \sum_{(i,j) \in \Omega_{i_0,j_0}} [G_n(i,j) - I_{n+1}(\mathbf{W}(i \cdot \Delta x, j \cdot \Delta y, \mathbf{p}))]^2 \, . \tag{1.4}$$

Dans cette équation, $G_n(i,j)$ désigne le niveau de gris d'un *pixel* de l'image à

l'instant $n \cdot \Delta t$, mais le deuxième terme reste fonction d'un *point* $\mathbf{x}' = \mathbf{W}(i \cdot \Delta x, j \cdot \Delta y, \mathbf{p})$ de l'image à l'instant $(n+1) \cdot \Delta t$, point dont les coordonnées ne sont pas forcément des multiples entiers des pas d'échantillonnage spatial.

Pour pallier ce problème, il faut utiliser des techniques d'estimation de l'intensité d'un point de l'image à partir des niveaux de gris des pixels situés dans son voisinage immédiat. Par la suite, nous noterons $\hat{G}_n(\mathbf{x})$ l'estimation de cette intensité. On peut par exemple estimer $\hat{G}_n(\mathbf{x})$ par le niveau de gris du pixel de l'image discrète qui est le plus proche du point $\mathbf{x}$:

$$\hat{G}_n(\mathbf{x}) = \hat{G}_n(x, y) = G_n(\text{ent}(x/\Delta x + 1/2), \text{ent}(y/\Delta y + 1/2)) \ , \tag{1.5}$$

dans laquelle la fonction ent$(\cdot)$ détermine la partie entière d'un nombre. Des techniques d'interpolation plus élaborées sont souvent utilisées afin d'obtenir une estimation plus précise de l'intensité du point image [A49].

La version la plus basique de la technique de mise en correspondance de blocs, décrite dans la section 1.3.2, consiste à rechercher un vecteur déplacement décrivant le mouvement apparent. Dans ce cas, le vecteur paramètre $\mathbf{p}$ peut coder directement la translation, ses deux coordonnées correspondant à celles du déplacement recherché $\mathbf{p} = \boldsymbol{\delta} = (\delta_x \ \delta_y)^T$, et la déformation $\mathbf{W}(\mathbf{x}, \mathbf{p})$ devient :

$$\mathbf{W}(\mathbf{x}, \mathbf{p}) = \mathbf{x} + \mathbf{p} = \mathbf{x} + \boldsymbol{\delta} \ . \tag{1.6}$$

Le vecteur translation $(\delta_x \ \delta_y)^T$, décrivant le mouvement apparent pour le pixel de coordonnées (i_0, j_0), est alors obtenu par minimisation, selon δ_x et δ_y, de la fonction de dissimilarité suivante :

$$E(\delta_x, \delta_y, \Omega_{i_0,j_0}) = \sum_{(i,j)\in\Omega_{i_0,j_0}} [G_n(i,j) - \hat{G}_{n+1}(i \cdot \Delta x + \delta_x, i \cdot \Delta y + \delta_y)]^2 \ . \tag{1.7}$$

Souvent, la translation minimisant cette fonction est recherchée en balayant de façon exhaustive l'ensemble des décalages contenus dans la fenêtre de recherche, dont les coordonnées sont également discrétisées : $(\delta_x \ \delta_y)^T = (k \cdot \Delta x \ \ l \cdot \Delta y)^T$. De ce fait, il n'est plus nécessaire d'avoir recours à une interpolation afin d'estimer

l'intensité du point dans la deuxième image, car il correspond alors directement à un pixel. La fonction de dissimilarité devient ainsi :

$$E(k,l,\Omega_{i_0,j_0}) = \sum_{(i,j)\in\Omega_{i_0,j_0}} [G_n(i,j) - G_{n+1}(i+k,j+l)]^2 \ , \tag{1.8}$$

qu'il s'agit de minimiser en calculant sa valeur pour tous les décalages k et l contenus dans la fenêtre de recherche.

La fonction de dissimilarité définie par l'équation (1.8) est qualifiée de SSD, pour « Sum of Squared Differences ». D'autres expressions ont été proposées dans la littérature. Par exemple, on peut calculer la somme des valeurs absolues des différences (SAD : Sum of Absolute Differences) [C28] ou étendre le calcul de dissimilarité en tenant compte des trois composantes dans le cas d'une séquence d'images couleur.

Quand le minimum a été localisé pour un vecteur de translation dont les coordonnées sont discrètes, il est possible dans un deuxième temps de raffiner le résultat en recherchant à proximité un autre minimum pour un point dont les coordonnées sont non entières (sub-pixel refinement, [A10]).

1.3.3.2 Méthodes différentielles

Supposons que le voisinage Ω soit réduit à un seul point et que le mouvement apparent recherché soit une translation pure. Dans ce cas, la valeur minimale de la fonction de similarité (1.1) est nulle, pour le vecteur de translation $\boldsymbol{\delta} = (\delta_x \ \delta_y)^T$ vérifiant :

$$E(\boldsymbol{\delta}) = 0 = \mathcal{I}(\mathbf{x},t) - \mathcal{I}(\mathbf{x}+\boldsymbol{\delta}, t+\delta_t) \ ,$$

soit :

$$\mathcal{I}(\mathbf{x},t) = \mathcal{I}(\mathbf{x}+\boldsymbol{\delta}, t+\delta_t) \ , \tag{1.9}$$

δ_t désignant l'intervalle de temps séparant l'acquisition des images ($\delta_t = t' - t$).

La vitesse apparente associée au point $\mathbf{x}$ à l'instant t peut être définie comme le rapport du vecteur de translation $\boldsymbol{\delta}$ par l'intervalle de temps séparant l'acquisition des deux images. Quand cet intervalle de temps tend vers zéro, le rapport donne le

vecteur vitesse apparente, ou plus précisément le flot optique associé au point x à l'instant t :

$$\mathbf{v}(\mathbf{x},t) = (u(\mathbf{x},t) \;\; v(\mathbf{x},t))^T = \lim_{\delta_t \to 0} \; (\frac{\delta_x}{\delta_t} \;\; \frac{\delta_y}{\delta_t})^T \; . \tag{1.10}$$

Pour déterminer le flot optique, les méthodes différentielles exploitent une approximation en série de Taylor de l'intensité de l'image exprimée en fonction de ses dérivées partielles par rapport aux dimensions spatiales et temporelles. Plus précisément, le terme de droite de l'équation (1.9) est approché par :

$$\mathcal{I}(x+\delta_x, y+\delta_y, t+\delta_t) \approx \mathcal{I}(x,y,t) + \frac{\partial \mathcal{I}(x,y,t)}{\partial x} \cdot \delta_x + \frac{\partial \mathcal{I}(x,y,t)}{\partial y} \cdot \delta_y + \frac{\partial \mathcal{I}(x,y,t)}{\partial t} \cdot \delta_t \; . \tag{1.11}$$

En combinant les équations (1.9), (1.10) et (1.11), on obtient :

$$\frac{\partial \mathcal{I}(x,y,t)}{\partial t} + \frac{\partial \mathcal{I}(x,y,t)}{\partial x} \cdot u(\mathbf{x},t) + \frac{\partial \mathcal{I}(x,y,t)}{\partial y} \cdot v(\mathbf{x},t) = 0 \; ,$$

ou encore :

$$\frac{\partial \mathcal{I}(\mathbf{x},t)}{\partial t} + \nabla \mathcal{I}(\mathbf{x},t) \cdot \mathbf{v}(\mathbf{x},t) = 0 \; , \tag{1.12}$$

connue sous le nom d'*équation de contrainte du mouvement apparent* (ECMA).

Dans ce qui suit, par souci de simplification, nous notons par I_x, $\frac{\partial \mathcal{I}(x,y,t)}{\partial x}$ par I_y, $\frac{\partial \mathcal{I}(x,y,t)}{\partial y}$ et par I_t, $\frac{\partial \mathcal{I}(x,y,t)}{\partial t}$. Nous ne disposons donc que d'une seule équation à deux inconnues pour résoudre le problème, car en chaque pixel de l'image nous n'avons qu'une seule contrainte scalaire pour déterminer le vecteur $\mathbf{v}(\mathbf{x},t)$. Cette non-unicité de la solution est connue sous le nom de *problème d'ouverture*. Par conséquent, afin de retrouver le flot optique $\mathbf{v}(\mathbf{x},t)$, il faut introduire de nouvelles contraintes qui résultent en un ensemble d'équations additionnelles. Le type de contrainte utilisé permet de répartir les méthodes dans deux catégories : les approches globales et les approches locales.

1.3.3.2.1 Approches Locales Les approches locales formulent des hypothèses sur le mouvement applicables localement, c'est à dire sur des fenêtres de petites tailles centrées sur le point analysé [A14, A25, A46, C27, T5, C4, C4].

Afin d'estimer le flot optique [C21], Bruce Lucas et Takeo Kanade ont introduit

une méthode de calcul minimisant une fonction de dissimilarité équivalente à celle de l'équation (1.7). Dans cette première version de leur algorithme, le mouvement apparent recherché est une translation uniforme, supposée constante dans le voisinage utilisé pour calculer la similarité.

L'originalité de leur approche repose sur le fait que le minimum n'est pas recherché en balayant de façon systématique tous les décalages possibles dans une fenêtre de recherche, mais par une méthode directe qu'ils ont appelée *méthode des différences*.

Lorsque le mouvement apparent est considéré constant égal à $\mathbf{v}(\mathbf{x},t)$ sur n points voisins, les n ECMA fournissent le système d'équations :

$$\begin{aligned} I_{x_1} \cdot u(\mathbf{x},t) + I_{y_1} v(\mathbf{x},t) &= -I_{t_1} \\ I_{x_2} \cdot u(\mathbf{x},t) + I_{y_2} v(\mathbf{x},t) &= -I_{t_2} \\ &\vdots \\ I_{x_n} \cdot u(\mathbf{x},t) + I_{y_n} v(\mathbf{x},t) &= -I_{t_n} \ , \end{aligned}$$

où I_{x_i} correspond à $\frac{\partial I}{\partial x}(x_i,y_i,t)$, I_{y_i} à $\frac{\partial I}{\partial y}(x_i,y_i,t)$ et I_{t_i} à $\frac{\partial I}{\partial t}(x_i,y_i,t)$. Cette méthode fournit n équations pour les images en niveaux de gris et $3 \times n$ équations pour les images couleur à trois composantes. En appliquant la méthode des moindres carrés pour résoudre ce système sur-dimensionné, nous obtenons le vecteur vitesse comme solution du système de deux équations à deux inconnues :

$$\begin{pmatrix} \sum_{i=1}^{n} I_{x_i} \cdot I_{x_i} & \sum_{i=1}^{n} I_{x_i} \cdot I_{y_i} \\ \sum_{i=1}^{n} I_{x_i} \cdot I_{y_i} & \sum_{i=1}^{n} I_{y_i} \cdot I_{y_i} \end{pmatrix} \mathbf{v}(\mathbf{x},t) = -\begin{pmatrix} \sum_{i=1}^{n} I_{x_i} \cdot I_{t_i} \\ \sum_{i=1}^{n} I_{y_i} \cdot I_{t_i} \end{pmatrix}$$

Il est important de voir que ce système n'a pas toujours une solution unique lorsque les équations initiales sont toutes équivalentes. C'est le cas lorsque les points considérés dans le voisinage n'apportent pas d'information supplémentaire en terme d'estimation du mouvement. Si l'apport du voisinage est peu significatif, la matrice est mal conditionnée et la solution est elle-même peu significative. Ces

cas peuvent aisément être détectés lors de la résolution du système, le vecteur vitesse n'étant pas estimé pour ce voisinage. La *densité* du champ estimé dépend du nombre de vecteurs correctement estimés, à partir de matrices bien conditionnées.

Dans une deuxième version de leur algorithme [C21], les mêmes auteurs ont modélisé le mouvement apparent par une transformation affine, laquelle peut être reliée aux paramètres du mouvement 3D. Dans cette version, la fonction de déformation est définie par :

$$\mathbf{W}(\mathbf{x},\mathbf{p}) = \begin{pmatrix} (1+p_1) & p_3 \\ p_2 & (1+p_4) \end{pmatrix} \cdot \mathbf{x} + \begin{pmatrix} p_5 \\ p_6 \end{pmatrix} , \tag{1.13}$$

avec $\mathbf{p} = (p_1\ p_2\ p_3\ p_4\ p_5\ p_6)^T$.

Dans ce cas, le vecteur p ne peut pas être déterminé directement par la méthode des différences, mais de façon itérative. Ainsi, Baker *et coll.* décrivent les méthodes numériques d'optimisation de la fonction d'erreur [R2]. Ces techniques visent à améliorer le temps de calcul tout en conservant une bonne précision du résultat.

La fonction de dissimilarité peut être modifiée afin de pondérer les pixels intervenant dans son calcul. Des poids différents peuvent être appliqués aux différents pixels de la fenêtre Ω pour que les pixels proches du pixel central aient plus d'influence que ceux situés à la périphérie [C27, T5] :

$$E(\mathbf{p},\Omega) = \int_{\mathbf{x}\in\Omega} \alpha(\mathbf{x})^2 \cdot [I_n(\mathbf{x}) - I_{n+1}(\mathbf{W}(\mathbf{x},\mathbf{p}))]^2 d\mathbf{x} \ . \tag{1.14}$$

1.3.3.2.2 Approches globales Dans [A20, A34, A35, A3, A2, C2], les auteurs proposent d'ajouter une étape de régularisation qui correspond implicitement à ajouter des équations. Ils supposent que le champ doit varier de façon régulière d'un point à son voisin dans le cas du mouvement d'objets rigides.

Dans leurs travaux, Horn et Schunck [A20] ont ajouté un terme de régularisation à l'équation de contrainte du flot optique qui se traduit par la minimisation de la norme au carré du gradient du flot optique : $\frac{\partial u}{\partial x}^2 + \frac{\partial u}{\partial y}^2$ et $\frac{\partial v}{\partial x}^2 + \frac{\partial v}{\partial y}^2$. Ainsi, ils ont défini une fonctionnelle, qu'il s'agit de minimiser sur une zone de l'image, qui

s'exprime sous la forme générale :

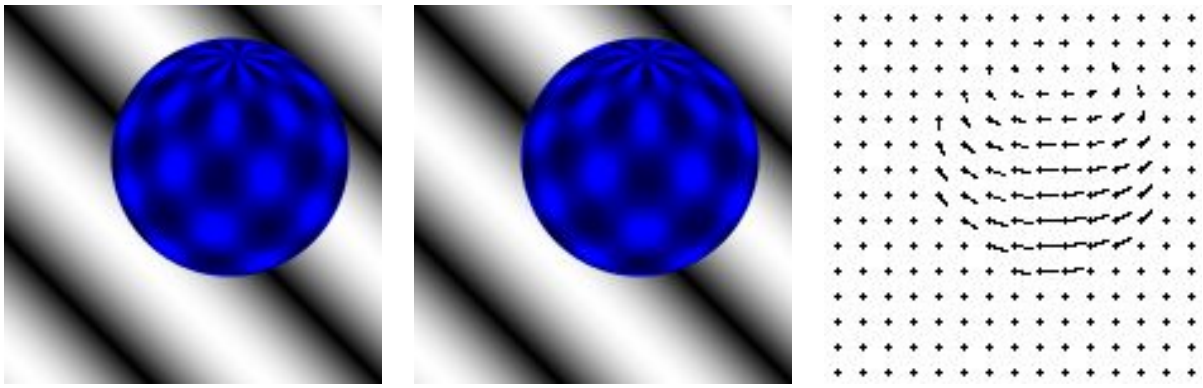
$$F(u,v) = \int_{image} (I_x u + I_y v + I_t)^2 + \alpha^2[(u_x)^2 + (u_y)^2 + (v_x)^2 + (v_y)^2]d\mathbf{x}, \quad (1.15)$$

où α représente le poids appliqué au terme de régularisation afin de remédier aux erreurs de quantification et aux bruits. En effet, ce terme de régularisation doit prendre une valeur d'autant plus importante que le niveau de bruit présent dans l'image est élevé.

Puisque la variation de mouvement n'est pas toujours lisse le terme de régularisation est remplacé par une contrainte de lissage modifiée :

$$F(u,v) = \int_{image} [(ECMA)^2 + \alpha[\varphi(|\delta u|) + \varphi(|\delta v|)] \quad (1.16)$$

dans laquelle $\varphi(.)$ est une fonction scalaire positive. La fonction, $\varphi(s) = s^2$ donne la solution de Horn et Schunck (figure 1.2).



(a) Image à l'instant t (b) Image à l'instant t' (c) Flot optique mesuré

FIGURE 1.2 : Estimation du flot optique par la méthode de Horn et Schunck [C14]

Nagel [A34, A35] fut le premier à utiliser des dérivées du second ordre pour l'estimation du flot optique dans le but est de pallier le problème des occultations. Il propose une contrainte de lissage orientée pour laquelle la variation du flot optique n'est pas obligatoirement lisse dans les zones de fortes variations d'intensité (par exemple les contours). Ces travaux furent également à la base de ceux de Alvarez *et coll.* [A3].

Les travaux de [A5] ont montré que les approches locales sont plus robustes au bruit que les approches globales. Par contre, ces dernières produisent des champs

de vecteurs mouvement plus denses, du fait que les matrices des systèmes d'équations fournissant le vecteur vitesse sont souvent mieux conditionnées. Bruhn *et coll.* [A8] tentent de profiter des avantages de chacune de ces méthodes en les combinant.

Une autre approche, l'approche multi-contraintes [T2] consiste à augmenter le nombre d'équations significatives en ajoutant des contraintes du même ordre que celle sur l'intensité, mais sur les composantes de couleur, ou sur l'entropie... On obtient un système soluble mais très dépendant des contraintes additionnelles et de leur validité.

1.3.4 Estimation dans le domaine fréquentiel

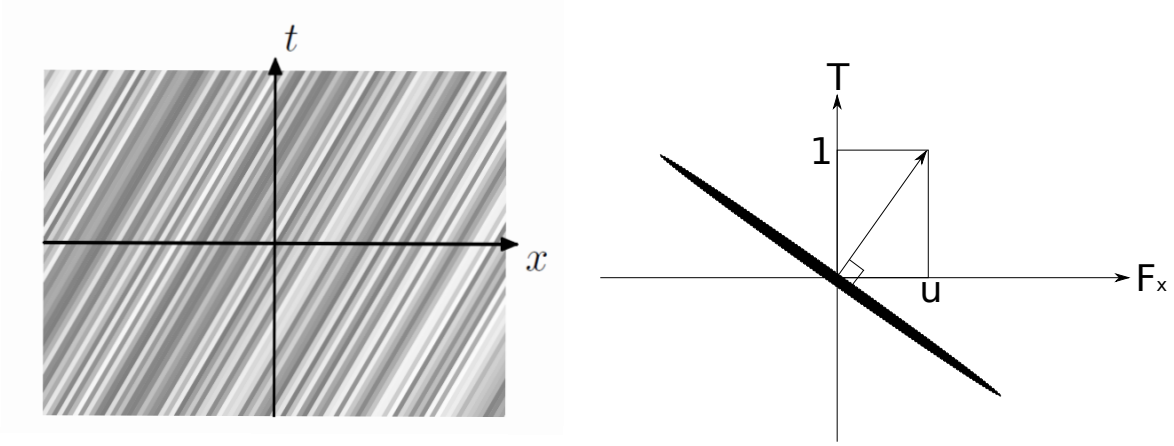
Le principe des méthodes fréquentielles est de mettre en évidence dans le domaine fréquentiel des propriétés d'une séquence d'images relatives à la présence d'un mouvement dans le domaine spatio-temporel. Les approches fréquentielles doivent leur nom à l'utilisation de filtres spatio-temporels accordés à la vitesse (en anglais *velocity tuned filters*). Ces méthodes d'analyse fréquentielle locales supposent également qu'une hypothèse est satisfaite, à savoir que le flot optique est constant sur toute l'étendue spatiale du support des filtres. On distingue deux sous-approches, exploitant soit l'énergie, soit la phase dans l'espace transformé.

1.3.4.1 Approches basées sur l'énergie

Cette technique a pour origine des recherches concernant la vision des mammifères, qui avaient mis en évidence la présence de cellules nerveuses se comportant comme des filtres passe-bande spatio-temporels. La réponse de ces cellules dépend de la vitesse d'un objet en mouvement ainsi que de sa forme et de ses caractéristiques fréquentielles [A19].

Quand l'image complète est affectée d'un mouvement de translation uniforme, la transformée de Fourier de la séquence a une propriété bien particulière : toute l'énergie est concentrée dans un hyperplan unique de l'espace transformé. Cette propriété est illustrée sur des images 1D par la figure 1.3, qui représente une séquence d'images monodimensionnelles (a) et le module de sa transformée de Fou-

rier (b). Dans le domaine spatial, la présence de droites toutes inclinées dans la même direction est caractéristique d'un mouvement de translation uniforme. Dans le domaine fréquentiel, toute l'énergie est alors concentrée dans une seule droite (représentée en grisé dans (b)). Le vecteur orthogonal à cette droite permet de déterminer la vitesse de translation, à savoir en 1D la composante u.



(a) Séquence en (x, t) (b) Spectre de la séquence

FIGURE 1.3 : Illustration du principe de base de la mesure du flot optique par filtrage spatio-temporel

Pour des images 2D, donc des séquences d'images représentées dans un espace 3D, l'espace des fréquences est également tridimensionnel. En présence d'un mouvement de translation, l'énergie est concentrée dans un plan unique. Le vecteur orthogonal à ce plan permet également de déterminer le vecteur vitesse.

Quand la totalité de l'image n'est pas en translation, mais seulement une zone limitée, une partie de l'énergie reste cependant concentrée dans un plan dans l'espace des fréquences. On peut ainsi, en repérant des plans particuliers dans l'espace des fréquences, déterminer les mouvements affectant l'image dans le domaine spatio-temporel. Les méthodes décrites dans la littérature utilisent une famille de filtres spatio-temporels accordés sur certaines fréquences (souvent des filtres de Gabor) et sensibles au mouvement [C1, C5, A6, T3].

Le problème majeur de ces techniques est qu'elles utilisent un grand nombre d'images consécutives [C9]. Heeger [A17, C16] utilise une famille de 12 filtres de Gabor de résolutions spatiales différentes afin d'extraire l'information requise sur le mouvement.

1.3.4.2 Approches basées sur la phase

Dans le domaine de Fourier, la phase porte également une information relative au mouvement dans le domaine spatio-temporel [C19]. Cette propriété est à la base des recherches menées notamment par Jepson [A21, L4, A50, A22, A4]. Comme les méthodes basées sur l'énergie, ces méthodes utilisent des familles de filtres ajustés sur différentes vitesses afin d'extraire l'information de la représentation de la séquence d'images dans le domaine fréquentiel. L'estimation du flot optique est assurée par la recherche de ruptures dans la phase des données en sortie des filtres. La détection de ces « contours » dans la phase est généralement plus fiable que dans l'énergie quand l'amplitude de la translation est faible.

Certaines de ces méthodes considèrent le cas où des discontinuités sont présentes dans le calcul du flot optique [A4]. Plus récemment, Weber et Malik [C25] ont proposé une méthode différentielle filtrée. L'équation du flot optique est convoluée avec des filtres sélectifs présentant différentes orientations. On obtient ainsi en un endroit donné autant d'équations que de filtres différents ayant répondu. Toutefois, cette méthode présuppose également que le flot optique est localement uniforme.

1.3.5 Conclusion

De nombreuses études comparatives ont été menées sur les méthodes décrites précédemment, afin de déterminer si une approche est plus efficace que les autres. Par exemple, Baron *et coll.* [A5] ont montré que les méthodes de Lucas-Kanade [C21, T4] et celles de Fleet et Jepson [A21, L4] sont les meilleures en terme de précision du vecteur estimé.

Dans notre cas, l'approche de Lucas-Kanade est la plus intéressante, car elle permet d'introduire explicitement un modèle de mouvement autre que la simple translation. Cela nous permettra d'introduire directement dans les équations le modèle du mouvement 3D recherché, via la transformation de coordonnées $\mathbf{W}(\mathbf{x}, \mathbf{p})$.

1.4 Méthodes exploitant un modèle

Lorsque l'utilisation de marqueurs telles des diodes électroluminescentes ou des pastilles sur les objets suivis n'est pas possible, une solution consiste à se baser sur des caractéristiques naturelles de l'objet qu'on doit retrouver dans les images. La fiabilité de la mise en correspondance est directement liée à la complexité des éléments caractéristiques : plus cette information est riche plus l'estimation du mouvement s'avère aisée. Par contre, la précision de cette information est directement liée au temps consacré au calcul.

Les méthodes de mise en correspondance modèle/images sont basées sur l'étude de l'évolution temporelle des caractéristiques les plus pertinentes de l'objet. Une première phase obligatoire consiste en la détermination de ces caractéristiques. La projection visible sur l'image d'une caractéristique particulière est appelée *élément structurel*. La mise en correspondance des éléments structurels extraits de deux images permet d'estimer le déplacement, donc le mouvement des objets.

Ces méthodes ont fait l'objet de maintes études [A28]. Elles peuvent être divisées en deux grandes catégories : celles à base d'un modèle 2D décrivant le contenu de l'image, et celles à base d'un modèle 3D décrivant le contenu de la scène.

1.4.1 Modèles 2D

Les modèles 2D peuvent être de simples éléments repérés dans l'image, comme des points caractéristiques, des éléments de contours (un contour étant considéré ici comme un changement brusque du niveau de gris), ou des descripteurs plus complexes, souvent invariants aux changements d'échelle et aux rotations, comme les descripteurs SIFT (Scale Invariant Feature Transform) par exemple.

1.4.1.1 Modèles à base de points caractéristiques

Les points caractéristiques peuvent être des coins, des points anguleux, des points de courbure maximale, des points isolés, des extrémités de lignes, etc. Plus généralement, ils sont définis par une expression mathématique permettant de décrire une caractéristique de l'objet suivi. Cette expression se doit d'être robuste aux

variations d'illumination ainsi qu'aux transformations affines résultant du mouvement de l'objet dans la scène. L'étape de sélection des points caractéristiques est le plus souvent automatisée. Une étude bibliographique de ces méthodes de sélection est disponible dans [A12, A39].

Les méthodes les plus fréquemment utilisées pour la détection de coins sont les opérateurs de Moravec [C23], de Forstner [C11], le détecteur Harris-Stephen / Plessey [C18], le détecteur de Shi-Tomasi [C31], le détecteur de Trajkovic et Hedley [A43]. Le lecteur peut trouver une évaluation des performances de plusieurs détecteurs de coins dans [A32].

Moravec fut l'un des premiers chercheurs à proposer un algorithme de détection de points caractéristiques. Cependant, bien qu'il permette un calcul rapide, cet algorithme présente deux inconvénients majeurs : 1) il est anisotrope et donc sensible aux rotations des objets ; 2) il détecte de faux points d'intérêts sur les lignes de contours.

L'algorithme de Harris-Stephen lève les limitations de l'algorithme de Moravec. Ces chercheurs ont introduit la matrice qu'ils ont nommée matrice d'autocorrélation :

$$\begin{pmatrix} \sum I_x^2 & \sum I_x.I_y \\ \sum I_y.I_x & \sum I_y^2 \end{pmatrix},$$

dont les valeurs propres indiquent :

- la présence de coins si elles sont toutes les deux positives ;
- l'absence de coins si elles sont toutes les deux nulles ;
- la présence de contours si l'une est positive et l'autre nulle.

En 1999, David Lowe [C22] a proposé l'algorithme SIFT pour la détection et le suivi des éléments structurels. Comme son nom l'indique cet algorithme est caractérisé par sa robustesse vis à vis du changement d'échelle. Il est également robuste à la rotation des éléments structurels, au bruit superposé à l'image, ainsi qu'aux changements d'illumination. L'algorithme SIFT exploite des histogrammes calculés selon différentes orientations, ce qui lui donne sa faculté de tolérer les déformations locales. Mikolajczyk *et coll.* ont montré dans [A33] que ce descripteur était le plus efficace de ceux connus à la date de leur étude.

L'estimation du mouvement est ensuite réalisée en mettant en correspondance les $\mathbf{x}_{i,t}$ points caractéristiques extraits de l'image à l'instant t et les $\mathbf{x}_{j,t'}$ points caractéristiques extraits de l'image à l'instant t'. Une méthode classique [A51] consiste à parcourir, dans l'image $I_{t'}$, le voisinage spatial de chaque point $\mathbf{x}_{i,t}$ à la recherche d'éventuels points caractéristiques $\mathbf{x}_{j,t'}$. La recherche se base sur la maximisation de la similarité de fenêtres locales centrées sur les points caractéristiques.

1.4.1.2 Modèles à base de contours

De nombreuses méthodes de suivi exploitent un modèle de contour décrit par des primitives géométriques telles des segments de droites ou des portions de courbes. Les contours constituent un bon compromis entre la complexité du modèle et la facilité d'implémentation, l'efficacité et la rapidité d'exécution des algorithmes. De plus, ces méthodes s'avèrent robustes au changement d'éclairage et à d'autres phénomènes intervenant lors de la formation de l'image, comme les reflets sur des surfaces réfléchissantes.

Nous pouvons distinguer trois catégories. La première catégorie regroupe les méthodes qui repèrent de façon exhaustive tous les contours dans les deux images, puis les mettent en correspondance afin d'estimer le mouvement. Dans la deuxième catégorie de méthodes, les contours sont détectés localement dans la nouvelle image, à proximité de ceux qui ont été marqués dans l'image précédente. Enfin, la troisième catégorie est constituée de toutes les méthodes basées sur le modèle de contour actif proposé en 1988 par Kass, Witkin et Terzopoulos [C33, A27].

1.4.1.2.1 Extraction globale des contours Cette approche consiste à mettre en correspondance des modèles de primitives extraites de façon exhaustive des deux images. Par exemple, dans le cas de la figure 1.4, les primitives sont des segments de droites décrivant la projection dans l'image des arêtes d'une description polyhédrique de la voiture [A23]. Les primitives rectilignes extraites de l'image sont mises en correspondance avec des segments de droite qui constituent le modèle. Les modèles peuvent être des ensembles de segments de courbes paramétriques

plus complexes.

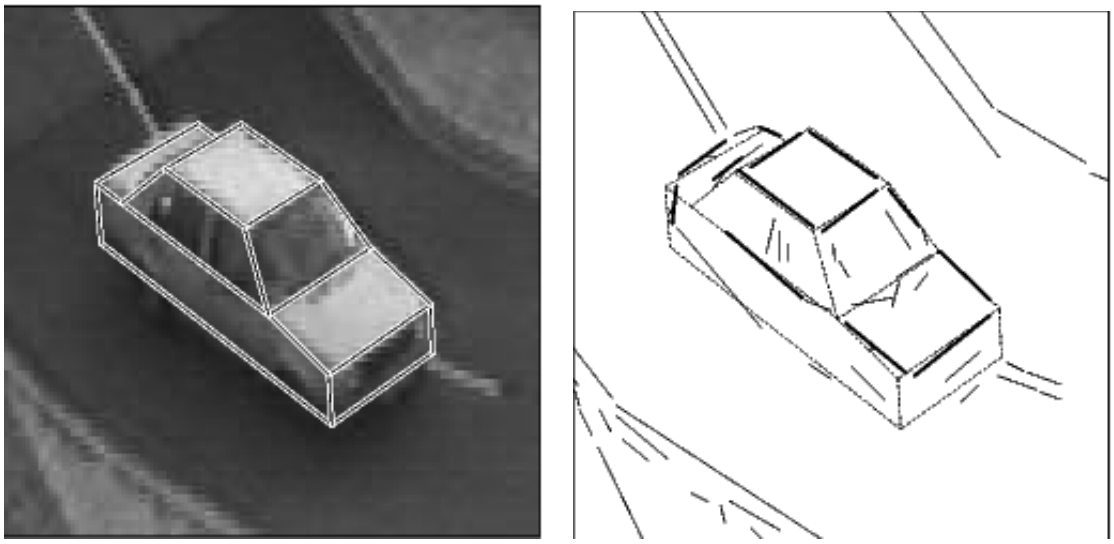


FIGURE 1.4 : Exemple de primitives en segments de droite utilisées dans [A23]
.

Dans l'exemple présenté, des segments de droites sont extraits de l'image analysée alors que le modèle de segments est projeté suivant une position et orientation prédites. Dans ce cas, la mise en correspondance se fait par minimisation d'une distance de Mahalanobis. Dans [C35] une transformée de Hough est utilisée pour l'extraction de segments de droite. Lowe [A30] réalise l'extraction de contours de segments de droites grâce au détecteur Marr et Hildreth.

1.4.1.2.2 Extraction locale des contours L'extraction des contours consiste souvent à rechercher les régions de l'image où le module du gradient, déterminé à partir des dérivées partielles directionnelles I_x et I_y, est élevé. Le calcul du gradient et la recherche de ses maxima locaux est une étape qui peut s'avérer gourmande en temps de calcul. C'est pourquoi certains auteurs ont proposé de limiter cette recherche aux alentours d'une première approximation de la position des contours provenant de la projection du modèle précédent dans la nouvelle image.

Harris *et coll.* [C15] ont présenté l'algorithme RaPiD "Real-time Attitude and Position Determination". Comme son nom l'indique, cet algorithme se distingue par sa rapidité de calcul. Il fût l'un des premiers algorithmes de suivi 3D à fonctionner en temps réel. Depuis, plusieurs modifications ont été apportées afin d'améliorer la performance de cet algorithme. L'idée de base est de considérer plusieurs points

de contrôle appartenant aux contours, puis d'estimer le mouvement 3D de l'objet grâce au déplacement de ces points de contrôle. Pour cela, pour chacune des images une initialisation de la position et de l'orientation initiale du modèle donne la position sur l'image de chacun des points de contrôle. Ensuite, ces points de contrôle sont mis en correspondance avec les maxima du gradient calculés uniquement localement.

Certains auteurs utilisent une approche relevant de l'*asservissement visuel* afin de mettre en correspondance le modèle 2D avec les images sans qu'il soit nécessaire d'extraire les contours, mais simplement de calculer le module du gradient [A11, A31, C34]. Dans ces méthodes, les paramètres du modèle sont ajustés afin que sa projection dans l'image s'approche au mieux des maxima locaux du gradient.

1.4.1.2.3 Modèles de contours actifs En 1988, Kass, Witkin et Terzopoulos introduisent les contours actifs appelés snakes [C33, A27]. Les snakes tiennent leur nom de leur aptitude à se déformer comme des serpents durant les itérations nécessaires à la convergence de l'algorithme. Un contour actif [L1, C7, C26, C24, C10] est une courbe plane paramétrique $\mathbf{c} = \{\mathrm{x}(s) = (x(s)\ y(s))^T |\ s \in [0,1]\}$, les fonctions $x(s)$ et $y(s)$ étant dérivables au moins jusqu'à l'ordre deux (*cf.* figure 1.5). s désigne l'abscisse curviligne d'un point de la courbe $\mathbf{c}$. L'ensemble des modèles possibles est ici de dimension infinie.

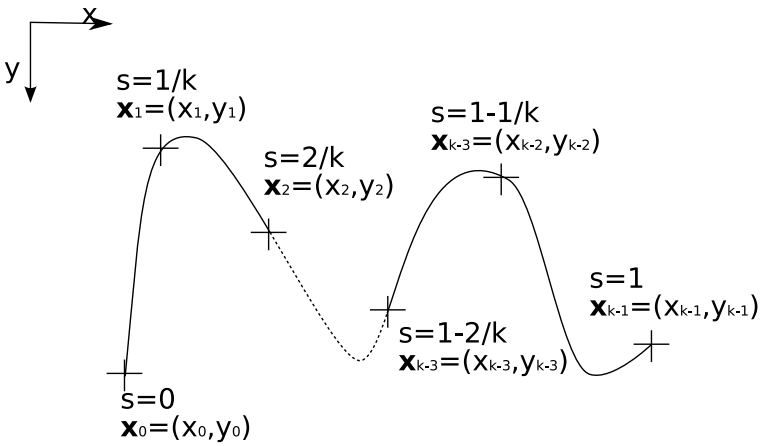


FIGURE 1.5 : Exemple d'abscisse curviligne et de courbe paramétrique.

Afin que la courbe puisse modéliser correctement le contenu de l'image, tout en conservant un aspect régulier, il s'agit d'introduire deux contraintes qui sont

parfois antagonistes. Kass *et coll.* ont proposé d'introduire ces contraintes sous la forme de deux termes d'une énergie associée au modèle de contour actif. Le premier terme, d'adéquation aux données ou *énergie externe*, prend une valeur minimale lorsque la courbe représente fidèlement le contenu de l'image. Le deuxième terme, de régularité ou *énergie interne*, prend une valeur minimale lorsque la forme de la courbe est régulière. Le processus de modélisation consiste à minimiser l'énergie totale, qui est la somme de ces deux termes : $E_{totale}(\mathbf{c}) = E_{int}(\mathbf{c}) + E_{ext}(\mathbf{c})$.

Dans [C33], les auteurs proposent l'expression suivante pour calculer l'énergie interne E_{int} du snake :

$$E_{int}(\mathbf{c}) = \int_0^1 \left(\alpha(s) \left| \frac{\partial \mathbf{x}}{\partial s}(s) \right|^2 + \beta(s) \left| \frac{\partial^2 \mathbf{x}}{\partial s^2}(s) \right|^2 \right) ds \ , \tag{1.17}$$

dans laquelle les fonctions $\alpha(s)$ et $\beta(s)$ permettent de pondérer l'influence de chaque point de la courbe. Quand la fonction $\alpha(s)$ est positive, le premier terme permet de pénaliser les courbes de longueur importante, une valeur minimale de zéro étant obtenue pour une courbe dégénérant en un point unique. Le deuxième terme pénalise les courbes en fonction de leur courbure quand la fonction $\beta(s)$ est positive, le minimum étant obtenu pour une courbe de forme circulaire lorsque $\beta(s)$ est constante.

L'énergie externe du contour actif, qui dérive d'un potentiel $V(\mathrm{x})$ défini en chaque point $\mathrm{x} = (x\ y)^T$ de l'image, est déterminée par l'expression :

$$E_{ext}(\mathbf{c}) = \int_0^1 V(\mathbf{x}(s)) ds \ . \tag{1.18}$$

En pratique, le potentiel est défini en fonction du type d'élément qu'on cherche à modéliser dans l'image. Dans le cas d'une image caractérisée par son niveau de gris $\mathcal{I}(\mathrm{x})$, le contour actif décrit des zones sombres quand $V(\mathrm{x})$ est égal à $\mathcal{I}(\mathrm{x})$. Quand le modèle décrit des contours, le potentiel $V(\mathrm{x})$ est une fonction qui passe par un minimum local quand le point x est situé sur un contour, par exemple l'opposé du module au carré du gradient de la fonction niveau de gris :

$$V(\mathbf{x}) = -(\nabla I(\mathbf{x}))^2 \ , \tag{1.19}$$

ou dans l'exemple des contours géodésiques [A9] :

$$V(\mathbf{x}) = g(\nabla I(\mathbf{x})) \ , \tag{1.20}$$

avec $g(r) = \frac{1}{1+r^m}$, et $m = 1$ ou 2.

Le contour actif représentant au mieux le contenu de l'image est obtenu en recherchant un minimum local de l'énergie. Pour ce faire, le contour est discrétisé et décrit par une ensemble fini de points appelés *snaxels*. Modéliser le contenu de l'image revient à rechercher de façon itérative l'ensemble de snaxels qui minimise l'énergie associée au snake.

Les snakes ont été largement utilisés pour suivre des objets dans une séquence d'images [A13, A24, A44, A29, A36, A45]. Une fois que le snake a convergé sur l'image courante de la séquence, il est utilisé comme position initiale du modèle recherché dans l'image suivante. Cela suppose souvent que le déplacement de l'objet reste faible entre les deux images. Pour tenter de lever cette contrainte, certains auteurs ont proposé d'introduire une étape de prédiction de la position initiale, faisant intervenir plusieurs modèles successifs de l'objet [A29, A45].

La principale limitation des contours actifs réside dans l'initialisation du modèle, qui le plus souvent est réalisée manuellement. Peu d'auteurs se sont attachés à proposer une procédure d'initialisation automatique de la position initiale du snake.

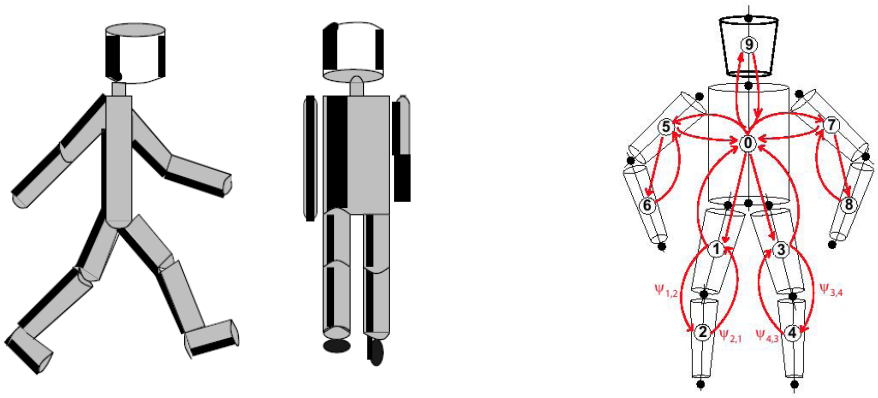
1.4.2 Modèles 3D

L'inconvénient majeur des modèles 2D consiste en leur sensibilité vis à vis de l'angle de la caméra, défi que les modèles 3D peuvent relever. En effet, les modèles 2D ne décrivent que le contenu de l'image, pas celui de la scène. Un avantage notable des modèles 3D est leur capacité à gérer les occultations partielles. Pour aboutir à des résultats précis, le modèle doit être complexe et donc reposer sur un nombre élevé de paramètres. Cependant, un modèle complexe requiert un temps de calcul plus important durant la phase de mise en correspondance.

Les recherches qui ont été menées pour la modélisation 3D se concentrent pour la plupart sur le problème de suivi de personnes [A47, A15, A1], où le modèle 3D

consiste en un ensemble de volumes primitifs tels : des cylindres, des ellipsoides, des cônes (figure 1.6), des sphères...

Des articulations relient chacun des membres ou des modèles géométriques et des modèles cinématiques [C6, C8], dynamiques [C30] ou comportementaux [C3] doivent caractériser ces articulations.



(a) Modèle à base de cylindres [A18] (a) Modèle à base de cones [A48]

FIGURE 1.6 : Exemple de modèles 3D du corps humain.

Dans [A38] l'auteur utilise 14 ellipsoides afin de modéliser le corps humain. L'origine est fixée sur le centre du torse. Wachter et Nagel [A48] utilisent un modèle de cônes et se basent sur un filtre de Kalman itératif. Ils utilisent des informations sur les contours et sur les régions afin de déterminer les degrés de liberté des articulations ainsi que les orientations par rapport à la caméra.

Lorsque différentes images visualisant l'objet suivant plusieurs points de vues sont disponibles, il existe des méthodes [L2] qui reconstruisent des modèles 3D plus précis par "space carving" (littéralement "sculpture de l'espace"), à partir de silhouette ou par coloriage de voxels.

Une approche intéressante consiste à combiner une méthode de suivi de contour avec une méthode basée sur un modèle 3D [C34, A31]. Polat *et coll.* [A37] utilisent un suivi basé sur des hypothèses multiples (multiple hypothesis tracking-MHT) combiné à une mesure de Hausdorff pour l'analyse du mouvement d'objets multiples dans la scène. Cette approche, qui gère bien les occultations, est principalement utilisée pour le suivi de plusieurs objets de la scène. Cependant, comme pour les méthodes se basant uniquement sur la géométrie, l'information reste incom-

plète pour certaines applications.

Quand on dispose d'un modèle complet de l'objet suivi (modèle de la texture et de la forme géométrique) ainsi que des paramètres de la caméra et de la source d'éclairage, il est possible de calculer un rendu complet de la scène. Le but est de retrouver le vecteur paramètre qui, une fois appliqué au modèle, décrit au mieux l'objet dans la scène. Comme nous avons besoin de reconstruire le mouvement 3D, la mise en correspondance modèle 3D/image semble la méthode qui permet le plus de précision surtout dans la détermination des angles de rotation. Cette méthode est détaillée dans le chapitre suivant.

1.5 Discussion et conclusion

Dans ce chapitre bibliographique, nous avons passé en revue les différentes méthodes d'estimation du mouvement 3D à partir d'une séquence d'images. A notre connaissance, aucune méthode n'a été proposée pour traiter spécifiquement le cas des objets dont la surface est réfléchissante. Pour ce type d'objets, l'approche la plus fréquemment utilisée consiste à limiter l'influence des reflets en se basant sur la forme plutôt que sur l'aspect ou en effectuant l'analyse sur des patchs (zones de surface limitée) présentant une information suffisante tout en ne provenant que de la composante diffuse.

Certains auteurs ont néanmoins tenu compte des propriétés des sources lumineuses dans le processus d'analyse du mouvement. Ces méthodes récentes visent à reconstituer le mouvement 3D lorsque l'illuminant est non constant ([C17, A16, A7, C12]). Il faut noter que la plupart de ces méthodes [C17, A7] ne considèrent que le cas des surfaces lambertiennes. Yang *et coll.*, visent à estimer le mouvement de l'objet, de la source lumineuse et le modèle de texture [C17]. Basri et Jacobs [A7], cherchent à prouver que sous un éclairage arbitraire et complexe, un sous-espace linéaire à neuf dimensions est suffisant pour estimer la fonction d'illumination à l'aide d'harmoniques sphériques. Freedman et Turek [C12] tentent d'estimer un flot optique invariant au changement d'illumination en se basant sur des coupures de graphes. Hager and Belhumeur [A16] analysent le problème du suivi quand des complications telles des occultations, des changements d'éclairage ou des déplace-

ments de la caméra interviennent.

Les méthodes qui se basent sur un modèle géométrique, prenant par exemple en compte la position dans l'image des arêtes ou des contours des objets, sont robustes face à la présence de réflexions spéculaires. Cependant, dans certains cas cette information s'avère insuffisante pour l'estimation du mouvement 3D. Dans le cas de la sphère réfléchissante qui fait l'objet de notre travail, un mouvement de rotation pure autour d'un des axes n'est pas perceptible si la méthode exploite le mouvement apparent du contour extérieur de l'objet.

D'autre part, les méthodes qui se basent sur l'apparence et surtout sur la texture, c'est à dire principalement sur la projection de la composante diffuse dans le plan image, risquent d'aboutir à des résultats erronés lorsqu'elles traitent des régions de l'image correspondant principalement aux reflets. Le mouvement apparent de la composante diffuse est parfois très différent de celui de la composante spéculaire. Toujours dans le cas d'une sphère en rotation pure, l'image du reflet reste statique alors que l'image de la composante diffuse permet d'estimer le mouvement.

Les réflexions spéculaires constituent donc une source d'information, qui une fois prise en compte et traitée, apporte plus de précision que les méthodes traditionnelles concernant le mouvement 3D. L'un des apports de notre travail consiste à le prouver. Dans le chapitre qui suit, nous détaillons les méthodes de mise en correspondance 2D/2D et 3D/2D sur lesquelles notre méthode hybride est fondée.

Chapitre 2

Mise en correspondance 3D/2D vs. 2D/2D

2.1 Introduction

Notre but est de reconstituer le mouvement 3D d'un objet aux différents moments d'acquisition d'une séquence d'images enregistrée par une caméra statique.

Dans ce chapitre, nous présentons les deux méthodes principales de la littérature s'attachant à résoudre ce problème. La première, intitulée mise en correspondance 3D/2D, propose d'utiliser un modèle 3D de l'objet suivi (modèle géométrique et de texture), modèle dont les paramètres de position et d'orientation sont ajustés au moyen d'une optimisation afin que l'image synthétisée et l'image réelle *se ressemblent* le plus. Après convergence, on considère que les paramètres obtenus sont une estimation de ceux qui caractérisent l'objet dans la scène observée. La seconde, intitulée mise en correspondance 2D/2D, se base sur l'apparence de l'objet et plus particulièrement sur la déformation de la texture entre deux images successives. Elle procède à l'alignement des deux images $\mathbf{I}_{n-1}$ et $\mathbf{I}_n$ à la recherche des paramètres 3D décrivant le mouvement de l'objet entre les instants $(n-1)\cdot\Delta t$ et $n\cdot\Delta t$. Tout comme la méthode précédente, l'estimation des paramètres est calculée grâce à un processus d'optimisation.

La dernière partie de ce chapitre fait état des résultats obtenus lorsque nous appliquons ces deux approches dans le cas du suivi d'une sphère dont la surface est réfléchissante. L'objectif est de mettre en évidence les avantages et les inconvénients de chacune d'entre elles face à la présence de réflexions diffuses et spéculaires.

2.2 Mise en correspondance 3D/2D

2.2.1 Introduction

L'objectif est d'analyser la fonction vectorielle image $\mathbf{I}_n(\mathbf{x})$, d'un objet afin de retrouver les paramètres de position et d'orientation de celui-ci au moment de l'acquisition. Cette méthode suppose qu'un modèle 3D de l'objet est disponible. La méthode d'estimation consiste alors à déterminer les paramètres de position et d'orientation qui, une fois appliqués au modèle 3D, permettent de générer une image de synthèse qui *ressemble le plus* à l'image réelle analysée $\mathbf{I}_n(\mathbf{x})$. Elle est récursive et considère que les paramètres du mouvement de l'objet exprimés dans un repère donné, repère dont la définition sera explicitée ultérieurement, ont été retrouvés jusqu'à l'instant $(n-1)\cdot\Delta t$.

Plusieurs équipes ont étudié cette approche dans le passé et ont proposé plusieurs solutions algorithmiques. La principale différence entre ces travaux réside dans le choix des éléments caractéristiques mis en correspondance entre l'image de synthèse et l'image réelle :

- les points caractéristiques tels que les coins ;
- les contours ;
- les niveaux de gris ;
- les disparités calculées lors d'une étape de mise en correspondance stéréoscopique (cas d'un système de perception multi-caméras) ;

Le modèle 3D d'un objet est une manière informatique de représenter l'objet afin de produire son image synthétique lors d'une étape dite de *rendu 3D*. Afin d'être le plus complet possible, le modèle peut comporter plusieurs niveaux de représentation : un modèle géométrique (relatif à sa forme) et un modèle d'apparence (relatif à sa texture et à sa couleur qui dépendent directement des propriétés physiques de sa surface et des illuminants de la scène). Le modèle géométrique est composé d'un ensemble de points 3D reliés entre eux par des formes géométriques tels que des polygones (dont les triangles sont les plus couramment utilisés) ou des surfaces splines. Dans le cas idéal, c'est-à-dire lorsque le modèle est une représentation parfaite de l'objet réel, l'image calculée est identique à celle fournie par une

caméra vidéo observant l'objet considéré.

Soit *rendu 3D* la fonction de rendu 3D que nous définissons de la manière suivante :

rendu 3D : $(\textbf{Modèle de scène}, \mathbf{p}) \longrightarrow \mathbf{R}(\mathbf{x}, \mathbf{p})$ où $\mathbf{p}$ est un vecteur des paramètres de position et d'orientation de l'objet et $\mathbf{R}(\mathbf{x}, \mathbf{p})$ est l'image de rendu. L'objectif de cette fonction est de produire une image de synthèse de l'objet placé dans son environnement. Le rendu 3D est donc appliqué sur un **Modèle de scène** composé non seulement du modèle 3D de l'objet mais également du modèle de chaque source d'illumination éclairant la scène et du modèle de la caméra vidéo observant cette scène. Dans ce mémoire, nous ne ferons pas d'état de l'art précis des méthodes de rendu 3D dont les plus communes sont la rastérisation, le tracé de rayons et le lancé de rayons.

L'image de synthèse obtenue est une image vectorielle de type RVBα :

$$\mathbf{R}(\mathbf{x}, \mathbf{p}) = (\mathcal{R}^r(\mathbf{x}, \mathbf{p})\ \mathcal{R}^v(\mathbf{x}, \mathbf{p})\ \mathcal{R}^b(\mathbf{x}, \mathbf{p})\ \mathcal{R}^\alpha(\mathbf{x}, \mathbf{p}))^T \ ,$$

où les termes $\mathcal{R}^r(\mathbf{x}, \mathbf{p})$, $\mathcal{R}^v(\mathbf{x}, \mathbf{p})$, $\mathcal{R}^b(\mathbf{x}, \mathbf{p})$ et $\mathcal{R}^\alpha(\mathbf{x}, \mathbf{p})$ désignent respectivement les composantes rouge, verte, et bleue et α de l'image synthétisée. Cette dernière composante représente le coefficient d'opacité. Afin de simplifier les notations, dans la suite du mémoire, elle sera notée :

$$\boldsymbol{\alpha}(\mathbf{x}, \mathbf{p}) = \mathcal{R}^\alpha(\mathbf{x}, \mathbf{p}) \ .$$

Ce coefficient d'opacité est une fonction qui indique si un point du modèle se projette ou non en ce point de l'image. Dans le cas d'une image définie sur un espace continu, le coefficient d'opacité est un indicateur binaire. Dans le cas des images définies sur un espace discret, le coefficient d'opacité est une valeur comprise entre 0 et 1, qui indique quelle proportion de la surface d'un pixel est occupée par la projection du modèle sur ce dernier. Quand le pixel appartient intégralement à la projection du modèle, le coefficient d'opacité associé vaut 1. Il vaut 0 pour les pixels du « fond », c'est à dire sur lesquels aucun point du modèle ne se projette. Enfin, pour les pixels situés au voisinage de la projection du contour extérieur du modèle

dans l'image, le coefficient d'opacité est une valeur comprise entre 0 et 1.

Dans la littérature, les algorithmes de mise en correspondance 3D/2D les plus courants estiment les paramètres de position et d'orientation de l'objet en minimisant (ou maximisant) une fonction d'erreur. Cette fonction est calculée à partir des éléments caractéristiques extraits des images réelles et de synthèse. Lorsque la méthode d'optimisation est itérative, un nouveau jeu des paramètres est déterminé à chaque itération et donne lieu à la production d'une nouvelle image de synthèse. Le processus atteint un extremum lorsque les deux images de l'objet sont semblables. L'algorithme a alors convergé vers une estimation des paramètres recherchés.

Dans la suite de cette section, nous décrivons plus précisément cette méthode.

2.2.2 Modélisation de la scène

Chaque élément du modèle de scène est placé dans un repère cartésien principal. Le repère le plus courant est un repère direct centré sur le point focal de la caméra et de même orientation que celle-ci (figure 2.1).

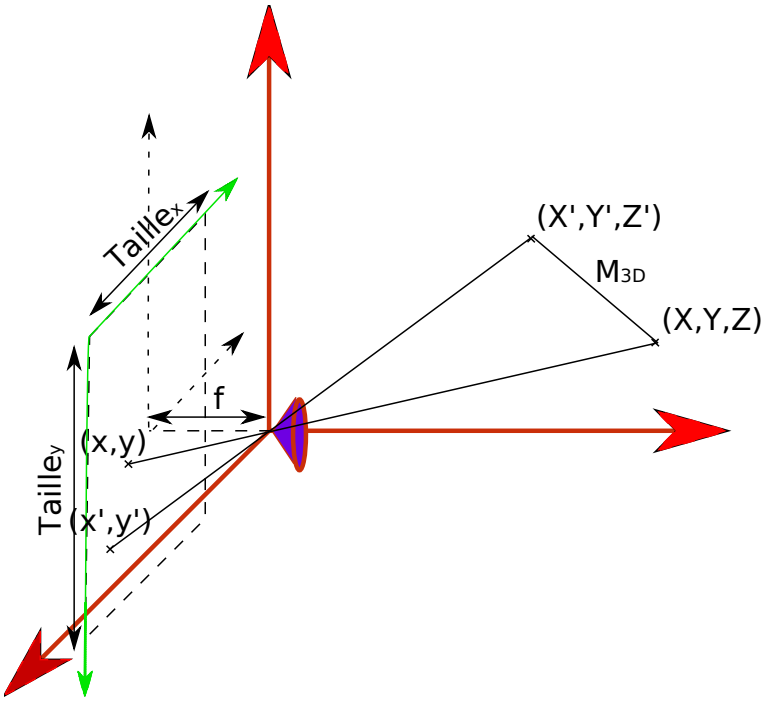


FIGURE 2.1 : Projection par rapport au repère caméra.

Nous faisons l'hypothèse d'un modèle de scène composé d'un seul modèle d'objet. Ce modèle est rigide et comporte une description géométrique et une description de la texture caractérisant la surface de l'objet. Dans le cas d'un modèle à facettes triangulaires, un modèle géométrique est représenté par le vecteur $\mathsf{S} = \left(v_1^T, ..., v_n^T\right)$ où les $v_{i=(1,...,n)} = (X_i, Y_i, Z_i)^T$ sont les sommets 3D du modèle définis dans le repère cartésien principal.

Soit $\mathbf{p} = (T_x, T_y, T_z, \theta_x, \theta_y, \theta_z)^T$ un vecteur représentant les paramètres de position (T_x, T_y, T_z) et d'orientation $(\theta_x, \theta_y, \theta_z)$ de l'objet dans le repère cartésien principal. Soit $M(\mathbf{p})$ la matrice globale de transformation calculée à partir de ces paramètres et permettant de positionner ce modèle 3D :

$$M(\mathbf{p}) = T(\mathbf{p}) \cdot R_y(\mathbf{p}) \cdot R_x(\mathbf{p}) \cdot R_z(\mathbf{p}) \quad , \tag{2.1}$$

avec

$$R_y(\mathbf{p}) = \begin{pmatrix} cos\theta_y & 0 & sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -sin\theta_y & 0 & cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad , \tag{2.2}$$

$$R_x(\mathbf{p}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta_x & -sin\theta_x & 0 \\ 0 & sin\theta_x & cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad , \tag{2.3}$$

$$R_z(\mathbf{p}) = \begin{pmatrix} cos\theta_z & -sin\theta_z & 0 & 0 \\ sin\theta_z & cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad , \tag{2.4}$$

$$T(\mathbf{p}) = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad . \tag{2.5}$$

Cette matrice de transformation est appliquée à chaque sommet v_i du modèle 3D supposé centré sur l'origine du repère caméra. Nous pouvons noter que l'ordre dans lequel les transformations élémentaires sont appliquées est primordial : les trois rotations sont appliquées au modèle avant la translation.

2.2.3 Fonction d'erreur et optimisation itérative

Soit $\mathbf{I}_n(\mathbf{x})$ l'image de l'objet acquise à l'instant $n \cdot \Delta t$ et soit $\mathbf{p}_{n,k}$ l'estimation des paramètres de position et d'orientation de l'objet perçu au même instant et calculée à l'itération k. A l'issue du processus d'optimisation, la suite $(\mathbf{p}_{n,k})_{k\in\mathbb{N}}$ converge vers le vecteur des valeurs réelles de chaque paramètre. Pour chaque nouvelle image $\mathbf{I}_n(\mathbf{x})$, l'initialisation du processus est assurée par la relation $\mathbf{p}_{n,0} = \mathbf{p}_{n-1,\infty}$ où $k = \infty$ pour l'itération finale. Cette relation n'est valable que sous l'hypothèse d'un faible mouvement de l'objet entre les instants $(n-1) \cdot \Delta t$ et $n \cdot \Delta t$. Par ailleurs, dans le cas de l'image $\mathbf{I}_0(\mathbf{x})$, $\mathbf{p}_{0,0}$ est initialisé manuellement (voir A.4).

A chaque itération k, la matrice de transformation $M(\mathbf{p}_{n,k})$ est calculée puis appliquée à chaque point du modèle de l'objet. A l'issue de cette transformation, nous calculons, $\mathbf{R}(\mathbf{x}, \mathbf{p}_{n,k})$, l'image du rendu 3D. Celle-ci est comparée à $\mathbf{I}_n(\mathbf{x})$ grâce à une fonction d'erreur qui permet de mesurer le degré de ressemblance entre $\mathbf{R}(\mathbf{x}, \mathbf{p}_{n,k})$ et $\mathbf{I}_n(\mathbf{x})$, sur l'ensemble des pixels compris dans un voisinage $\Omega = [0, x_{max}] \times [0, y_{max}]$:

$$\begin{aligned} E_{3D/2D}(\mathbf{p}_{n,k}) = (N \sum_{\mathbf{x};\boldsymbol{\alpha}(\mathbf{x},\mathbf{p}_{n,k})\neq 0} \boldsymbol{\alpha}(\mathbf{x}, \mathbf{p}_{n,k}))^{-1} \sum_{\mathbf{x};\boldsymbol{\alpha}(\mathbf{x},\mathbf{p}_{n,k})\neq 0} \boldsymbol{\alpha}(\mathbf{x}, \mathbf{p}_{n,k})[\\ (\mathcal{R}^r(\mathbf{x}, \mathbf{p}_{n,k}) - \mathcal{I}^r_n(\mathbf{x}))^2 \\ +(\mathcal{R}^v(\mathbf{x}, \mathbf{p}_{n,k}) - \mathcal{I}^v_n(\mathbf{x}))^2 \\ +(\mathcal{R}^b(\mathbf{x}, \mathbf{p}_{n,k}) - \mathcal{I}^b_n(\mathbf{x}))^2] \quad , \end{aligned} \tag{2.6}$$

où le premier terme, inverse de la somme de tous les coefficients d'opacité, correspond à la surface apparente de la projection du modèle dans l'image. Ainsi, sous couvert de modèles géométriques, de texture et d'apparence parfaits, les valeurs des composantes RVB de l'image réelle $\mathbf{I}_n(\mathbf{x})$ et celles de l'image synthétique $\mathbf{R}(\mathbf{x}, \mathbf{p}_{n,k})$ sont les mêmes sur tous les points $\mathbf{x} \in \Omega$ quand les deux images corres-

pondent. Dans ce cas l'erreur $E_{3D/2D}(\mathbf{p}_{n,k})$ est nulle.

Trouver l'extremum de cette fonction d'erreur revient ici à la minimiser. De nombreuses méthodes permettent d'y parvenir. Parmi elles, l'algorithme de descente du gradient est le plus classique. Il consiste à suivre, dans l'espace des paramètres, la ligne de plus grande pente pour atteindre le minimum de la fonction. Si l'on se place dans le cadre d'une fonction à plusieurs paramètres, la dérivée est le vecteur gradient de la fonction dont chaque élément est la dérivée partielle de la fonction suivant l'un des paramètres.

Cet algorithme nécessite une initialisation de l'estimation recherchée au plus proche de la valeur réelle afin, notamment, d'assurer une convergence en un nombre réduit d'itérations. Par ailleurs, il fait l'hypothèse que la fonction à minimiser est quadratique autour du minimum recherché. Lorsque ces deux pré-requis ne sont pas respectés, nous rencontrons des obstacles que nous détaillons dans l'annexe A.1.

Soit $E_{3D/2D}(\mathbf{p}_{n,0})$ une fonction (suffisamment dérivable) dont on recherche le minimum. La méthode du gradient construit une suite qui doit en principe s'approcher du minimum. Pour cela, on part de la valeur initiale $\mathbf{p}_{n+1,0}$ proche de la valeur réelle et l'on construit la suite :

$$\mathbf{p}_{n,k} = \mathbf{p}_{n,k-1} - \mu \nabla_{\mathbf{p}_n} E_{3D/2D}(\mathbf{p}_n) \mid_{\mathbf{p}_n = \mathbf{p}_{n,k-1}} \quad , \tag{2.7}$$

où :

- $\mathbf{p}_{n,k}$ est le vecteur paramètre calculé à l'instant $n \cdot \Delta t$ et à l'itération k ;
- $\mathbf{p}_{n,k-1}$ est le vecteur paramètre calculé à l'instant $n \cdot \Delta t$ et à l'itération $k-1$;
- μ, est le vecteur « pas de descente ». Les composantes de ce vecteur doivent être ajustées pour garantir la convergence de l'algorithme ;
- $\nabla_{\mathbf{p}_n} E_{3D/2D}(\mathbf{p}_n) \mid_{\mathbf{p}_n = \mathbf{p}_{n,k-1}}$ est le gradient de $E_{3D/2D}$ par rapport aux paramètres de transformation calculés à l'instant $n \cdot \Delta t$ et à l'itération $k-1$.

La figure 2.2 décrit le principe de la descente du gradient appliquée à la minimisation d'une fonction f quelconque à un seul paramètre x. On remarque que x_{k+1} est d'autant plus éloigné de x_k que la pente de la courbe est importante. On peut décider d'arrêter les itérations lorsque cette pente est suffisamment faible.

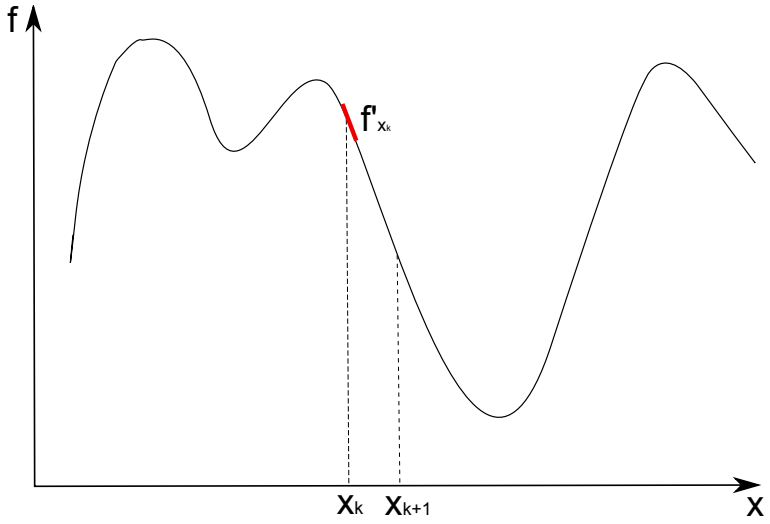


FIGURE 2.2 : Descente de gradient d'une fonction à un paramètre

2.3 Mise en correspondance 2D/2D

2.3.1 Introduction

L'objectif de cette méthode est d'aligner les deux fonctions $\mathbf{I}_n(\mathbf{x})$ et $\mathbf{I}_{n+1}(\mathbf{x})$ sur un voisinage $\Omega = [x_1, x_2] \times [y_1, y_2]$ appelé patch afin d'estimer les paramètres 3D du mouvement de l'objet entre les deux instants d'acquisition. Lorsqu'il est projeté sur une succession d'images, un mouvement 3D se traduit par une transformation locale de leur contenu, cette transformation dépendant directement des paramètres du mouvement recherché. Soit $\Delta\mathbf{p}_n$ le vecteur représentant les paramètres de ce mouvement entre les instants $n \cdot \Delta t$ et $(n+1) \cdot \Delta t$:

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \Delta\mathbf{p}_n \quad , \tag{2.8}$$

où $\mathbf{p}_n$ et $\mathbf{p}_{n+1}$ sont les vecteurs de position et d'orientation 3D de l'objet aux deux instants considérés.

Soit $\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}_n)$ la transformation locale 2D représentant un mouvement 3D $\Delta\mathbf{p}_n$ donné. Comme cela a été présenté dans le chapitre précédent, cette transformation est appelée *warping* (annexe 2.9) et s'exprime de la manière suivante :

$$\mathbf{x}' = \mathbf{W}(\mathbf{x}, \Delta\mathbf{p}_n) \quad , \tag{2.9}$$

où $\mathbf{x} = (x\ y)^T$ et $\mathbf{x}' = (x'\ y')^T$ sont respectivement les coordonnées d'un point de l'image $\mathbf{I}_n(\mathbf{x})$ et de son point homologue dans l'image $\mathbf{I}_{n+1}(\mathbf{x})$.

Nous nous intéressons particulièrement au cas où la transformation exacte 2.9 est approchée par une transformation linéaire 2D. Cette dernière peut s'exprimer à partir de coordonnées homogènes des points de la manière suivante :

$$(x'\ y'\ 1)^T = M_{2D}(\Delta\mathbf{p}_n) \cdot (x\ y\ 1)^T \quad . \tag{2.10}$$

Soient $(x\ y)$ les coordonnées d'un point quelconque d'une image acquise par une caméra, munie d'un objectif de longueur focale f, dont le capteur est de résolution $N_x \times N_y$ et de taille $Taille_x \times Taille_y$. Ce point est la projection d'un point $(X\ Y\ Z)^T$ dont les coordonnées sont exprimées dans le repère caméra. En coordonnées homogènes, la matrice de projection est donnée par :

$$\begin{pmatrix} sx \\ sy \\ sz \\ s \end{pmatrix} = T_{2D} \cdot H_{2D} \cdot P_{3D} \cdot \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad , \tag{2.11}$$

où s permet la normalisation du résultat. La matrice P_{3D} représente la projection du point 3D sur le plan image suivant le modèle de sténopé :

$$P_{3D} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} \quad , \tag{2.12}$$

H_{2D} est une homothétie qui permet un changement d'échelle :

$$H_{2D} = \begin{pmatrix} \frac{N_x}{Taille_x} & 0 & 0 & 0 \\ 0 & \frac{N_y}{Taille_y} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad , \tag{2.13}$$

et enfin T_{2D} est la matrice permettant le passage du repère centré sur le milieu de l'image au repère centré sur le coin supérieur gauche de l'image :

$$T_{2D} = \begin{pmatrix} 1 & 0 & 0 & N_x/2 \\ 0 & -1 & 0 & N_y/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad . \tag{2.14}$$

Considérons, qu'entre deux instants d'acquisition, le mouvement 3D $M_{3D}(\Delta\mathbf{p})$ de vecteur paramètre $\Delta\mathbf{p}$ transforme le point $(X'\ Y'\ Z')^T$ en un point $(X\ Y\ Z)^T$. Le mouvement 3D est ainsi décrit par la transformation :

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = M_{3D}(\Delta\mathbf{p}) \cdot \begin{pmatrix} X' \\ Y' \\ Z' \\ 1 \end{pmatrix} \quad . \tag{2.15}$$

En combinant (2.11) et (2.15), nous obtenons :

$$\begin{pmatrix} s'x' \\ s'y' \\ s' \\ 1 \end{pmatrix} = M_{2D}(\Delta\mathbf{p}) \cdot \begin{pmatrix} sx \\ sy \\ s \\ 1 \end{pmatrix} \quad , \tag{2.16}$$

où $M_{2D}(\Delta\mathbf{p})$ est la matrice de transformation suivante :

$$\begin{aligned} M_{2D}(\Delta\mathbf{p}) \quad = \quad & T_{2D} \cdot H_{2D} \cdot P_{3D} \cdot \\ & M_{3D}^{-1}(\Delta\mathbf{p}) \cdot \\ & (T_{2D} \cdot H_{2D} \cdot P_{3D})^{-1} \quad . \end{aligned} \tag{2.17}$$

Dans l'équation (2.16), quel que soit le point x, s et s' vérifient la condition suivante :

$$\frac{s}{s'} = \frac{Z}{Z'} \quad . \tag{2.18}$$

Généralement, cette méthode de mise en correspondance 2D/2D est appliquée lorsqu'aucune connaissance *a priori* sur la géométrie de l'objet n'est disponible. Elle permet de retrouver le mouvement de l'objet suivi entre les deux images successives qui sont analysées. Pour cela, une approximation est faite sur le patch : on suppose qu'il est la projection d'une portion d'un plan parallèle au plan image et situé à une profondeur Z connue. Ce paramètre Z peut être considéré soit comme une constante, soit comme une septième inconnue du problème. Plus le patch est grand, moins cette approximation est justifiée et donc moins l'estimation 2D/2D est précise.

2.3.2 Fonction d'erreur et optimisation

Comme les méthodes d'estimation par mise en correspondance 3D/2D décrites dans la section précédente, l'estimation des paramètres du mouvement est réalisée en recherchant l'extremum d'une fonction d'erreur dont le paramètre est $\Delta\mathbf{p}$.

Ainsi, pour un couple d'images $\mathbf{I}_n(\mathbf{x})/\mathbf{I}_{n+1}(\mathbf{x})$, l'optimisation itérative permet de construire une suite $(\mathbf{p}_{n,k})_{k\in\mathbb{N}}$. La limite $\mathbf{p}_{n,\infty}$ est obtenue en minimisant la fonction d'erreur suivante sur $\Omega = [x_1, x_2] \times [y_1, y_2]$:

$$\begin{aligned} E_{2D/2D}(\Delta\mathbf{p}_{n,k}) = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} [\quad & (\mathcal{I}_n^r(\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}_k) - \mathcal{I}_{n+1}^r(\mathbf{x}))^2 \\ + \quad & (\mathcal{I}_n^v(\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}_k) - \mathcal{I}_{n+1}^v(\mathbf{x}))^2 \\ + \quad & (\mathcal{I}_n^b(\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}_k) - \mathcal{I}_{n+1}^b(\mathbf{x}))^2] \ , \end{aligned} \tag{2.19}$$

où $|\Omega| = (x_2 - x_1 + 1).(y_2 - y_1 + 1)$ est le cardinal du voisinage Ω.

Diverses méthodes d'optimisation de la littérature permettent d'assurer la minimisation de cette fonction d'erreur. Dans leur article [C21], Lucas et Kanade utilisent l'algorithme de Gauss-Newton. Le lecteur pourra se référer à [R2] pour avoir une vision comparative des performances obtenues avec d'autres méthodes.

Dans la suite de ce mémoire, nous utilisons la méthode de descente de gradient tout comme le cas des méthodes par mise en correspondance 3D/2D et avec les mêmes conditions et limites d'utilisation : un état initial proche de l'état final recherché au voisinage duquel la fonction d'erreur est considérée quadratique.

Soit $\Delta\mathbf{p}_{n,0}$, la valeur initiale de la solution recherchée à l'itération 0. Nous la fixons à 0 car nous faisons l'hypothèse de petits déplacements.

Lorsque le but est de minimiser l'équation (2.19) par rapport à $\Delta\mathbf{p}_{n,k}$, l'algorithme de descente de gradient consiste à retrouver $\Delta\mathbf{p}_{n,k+1}$ de l'itération $k+1$ grâce à l'estimation de ces paramètres à l'itération k :

$$\Delta\mathbf{p}_{n,k+1} = \Delta\mathbf{p}_{n,k} - \mu\nabla_{\Delta\mathbf{p}_n} E_{2D/2D}(\Delta\mathbf{p}_n)\mid_{\Delta\mathbf{p}_n=\Delta\mathbf{p}_{n,k}} \quad , \tag{2.20}$$

où μ est le vecteur pas de descente, et $\nabla_{\Delta\mathbf{p}_n} E_{2D/2D}(\Delta\mathbf{p}_n)\mid_{\Delta\mathbf{p}_n=\Delta\mathbf{p}_{n,k}}$ représente le gradient de $E_{2D/2D}$ suivant les six paramètres de $\Delta\mathbf{p}_n$.

Avec une vue monoculaire, nous pouvons nous attendre à une ambiguïté lors de l'estimation de certains mouvements. En effet, le mouvement apparent de rotation pure autour de l'axe horizontal de l'objet (en considérant le vecteur paramètre $\Delta\mathbf{p} = [0,0,0,\alpha,0,0]^T$ avec $\alpha \neq 0$) sur un seul patch peut ressembler au mouvement de translation suivant l'axe vertical de celui-ci (en considérant le vecteur paramètre $\Delta\mathbf{p} = [0,b,0,0,0,0]^T$ avec $b \neq 0$). De même, une rotation pure autour de l'axe vertical de l'objet peut-être confondue avec une translation suivant son axe horizontal.

Dans la suite de ce chapitre, nous établissons un comparatif des résultats obtenus avec ces deux méthodes dans le cas du mouvement d'une sphère réfléchissante comportant une composante diffuse et spéculaire.

2.4 Etude comparative dans le cas d'une sphère réfléchissante

2.4.1 Introduction

Cette situation est en fait très complexe. En effet, dans ce cas :

- d'une part une méthode exploitant uniquement les propriétés géométriques de l'objet telles que son contour ne permet pas de détecter les rotations pures ;
- d'autre part les méthodes se basant sur la texture (manifestation d'une réflexion diffuse) risquent d'engendrer des erreurs d'estimation en présence de réflexions spéculaires des sources lumineuses (qualifiées par la suite de *reflets*).

Dans ce qui suit, nous présentons les avantages et les inconvénients des deux méthodes précédentes dans ce contexte d'application.

Nous disposons de séquences d'images montrant le mouvement d'une sphère texturée de rayon R connu et de surface réfléchissante. Sur $\mathbf{I}_0(\mathbf{x})$, la première image de la séquence, les paramètres de position et d'orientation de la sphère représentés par $\mathbf{p}_0$ sont supposés connus. L'objectif consiste à retrouver les valeurs successives de ces paramètres, par rapport au repère caméra, dans les différentes images de la séquence.

Nous considérons un éclairage de type Phong, assuré par un spot de puissance et de position connues. Les paramètres intrinsèques de la caméra, à savoir la taille du capteur ($Taille_x$ et $Taille_y$) et sa distance focale f, sont également connus (figure 2.3) : sa distance focale $f = 4.7 \cdot 10^{-3}$ m, les dimensions de son capteur sont respectivement $6.3 \cdot 10^{-3}$ m et $4.7 \cdot 10^{-3}$ m. Son point focal est supposé à l'origine du repère fixe considéré. La position de la sphère dans la première image est $(-2.5 \text{ m} \;\; 0 \;\; 6 \text{ m})$ et son rayon R vaut 1 m.

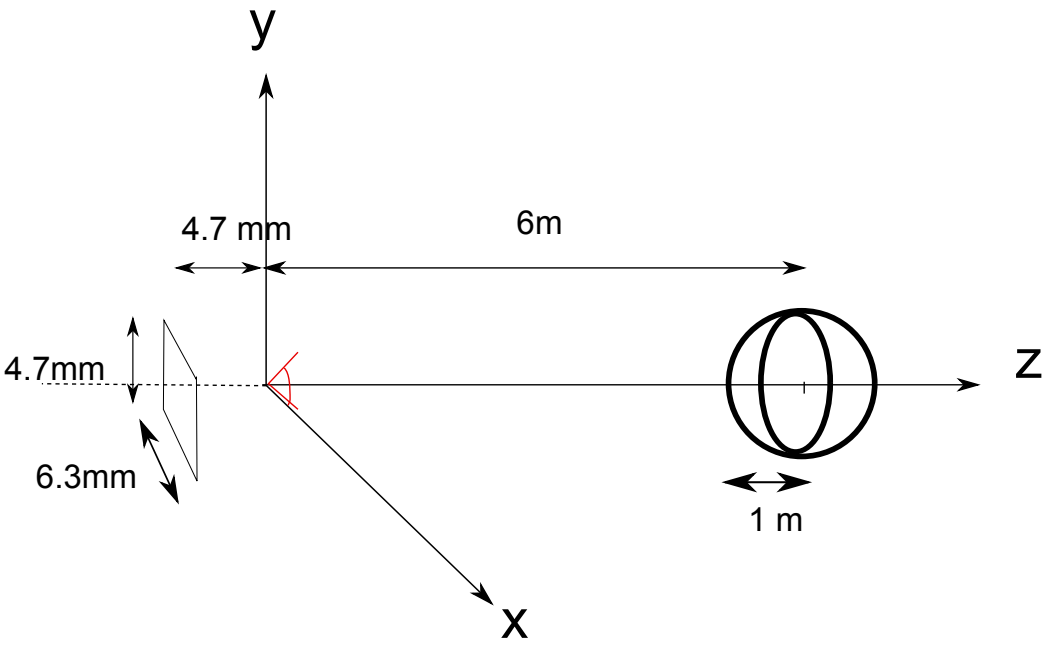


FIGURE 2.3 : Configuration de notre scène.

Pour la mise en correspondance 2D/2D, nous ne considérons pas Z comme une inconnue supplémentaire. Nous faisons l'hypothèse que la coordonnée Z de chaque point de la sphère peut être calculée à tout instant $n \cdot \Delta t$ grâce à son équation dans

le repère caméra :

$$(X - T_{x,n-1})^2 + (Y - T_{y,n-1})^2 + (Z - T_{z,n-1})^2 = R^2 \ , \tag{2.21}$$

où $(T_{x,n-1}, T_{y,n-1}, T_{z,n-1})$ est la position du centre de la sphère estimée à l'instant $(n-1) \cdot \Delta t$.

Pour la mise en correspondance 3D/2D, nous faisons l'hypothèse que le modèle de la sphère comprend le modèle géométrique et les modèles de diffusion et de spécularité de sa surface afin de tenir compte des deux composantes de la réflexion.

2.4.2 Résultats et discussion

Dans le cas d'une sphère de surface réfléchissante en rotation autour d'un de ses axes, le mouvement apparent de la composante diffuse témoigne de cette rotation Cependant, la partie de l'image correspondant à la composante spéculaire reste fixe au cours du temps. Ceci est illustré par la figure 2.4.

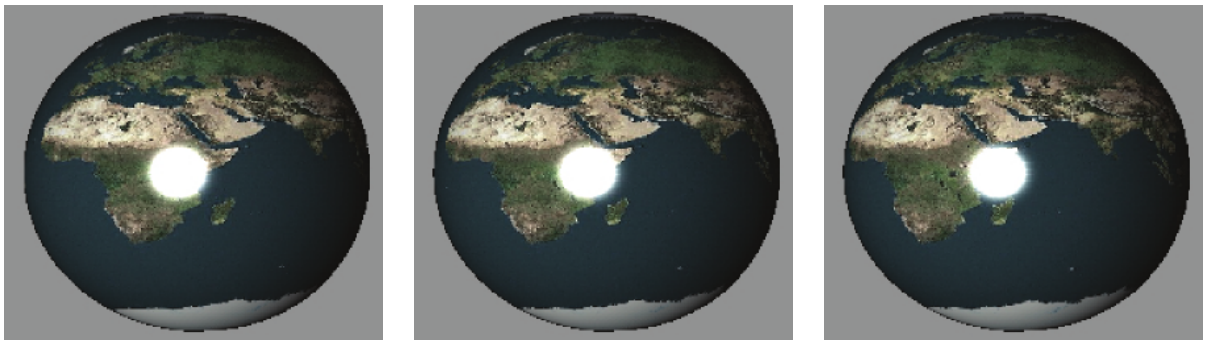


FIGURE 2.4 : La composante spéculaire (tâche blanche) reste fixe lorsque la sphère tourne autour de l'un de ses axes alors que la composante diffuse se déplace vers la gauche.

Estimation par mise en correspondance 2D/2D

La figure 2.5 illustre l'effet d'une transformation 2D, $\mathbf{W}(\mathbf{x}, \Delta\mathbf{p})$, non adaptée à la présence d'une composante spéculaire dans le voisinage. Cette transformation induit un déplacement de la droite vers la gauche de la tâche blanche alors qu'elle devrait rester fixe. Ceci a pour effet d'aboutir à une erreur d'estimation du vecteur Δp_n.

La précision de l'estimation est directement liée à la taille du patch utilisé. En

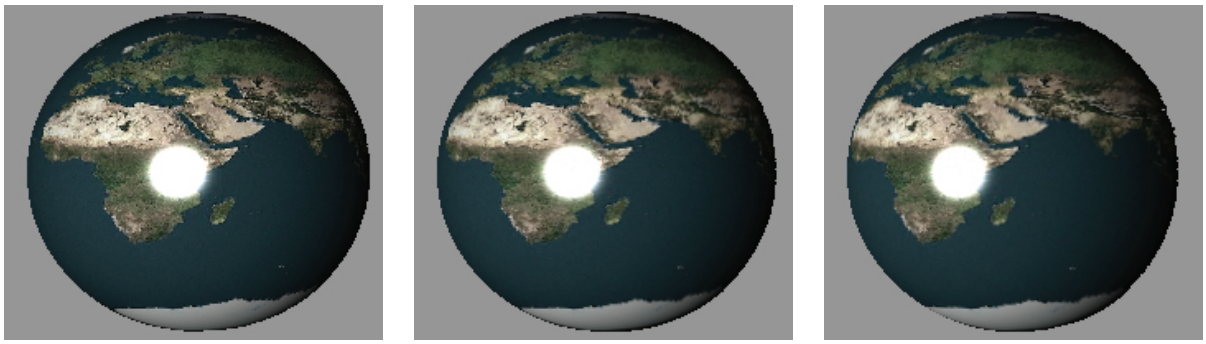


FIGURE 2.5 : Une transformation 2D non adaptée affecte du même mouvement apparent la composante diffuse et spéculaire : la tâche blanche se déplace vers la gauche comme le reste de la sphère.

effet, considérons le cas d'une rotation pure autour de l'axe vertical de la sphère $\Delta \mathbf{p}_1 = [0, 0, 0, 0, 0.01 \text{ rd}, 0]^T$. Nous avons étudié la précision de l'estimation pour différentes tailles d'un patch ($40 \times 40, 50 \times 50, 60 \times 60, 144 \times 144$) centrés sur la composante spéculaire de la réflexion (cf. figure 2.6). Dans les différents cas de figure, plus le patch est grand, plus la proportion de la surface du patch occupée par le reflet est petite et plus l'information provenant de la composante diffuse intervient de manière importante dans l'estimation.

FIGURE 2.6 : Différentes tailles de patchs utilisés afin d'obtenir différents taux de présence de la composante spéculaire par rapport à la composante diffuse : (a) Patch 40×40, (b) Patch 50×50, (c) Patch 60×60, (d) Patch 144×144

Comme nous le constatons sur la figure 2.7, pour le patch de taille 40×40, la fonction d'erreur n'atteint pas son minimum pour la valeur réelle de la rotation (ici égale à 0.01 rd) alors qu'aucune erreur n'apparaît pour les cinq autres paramètres recherchés. Ce résultat est dû au fait que la composante diffuse n'est qu'insuffisamment présente dans le patch alors qu'elle est la seule source d'information fiable. Nous remarquons que la précision augmente avec la taille du patch pour atteindre une convergence parfaite pour le patch de taille 144×144.

Ainsi pour garantir la convergence de l'approche 2D/2D la taille du patch doit

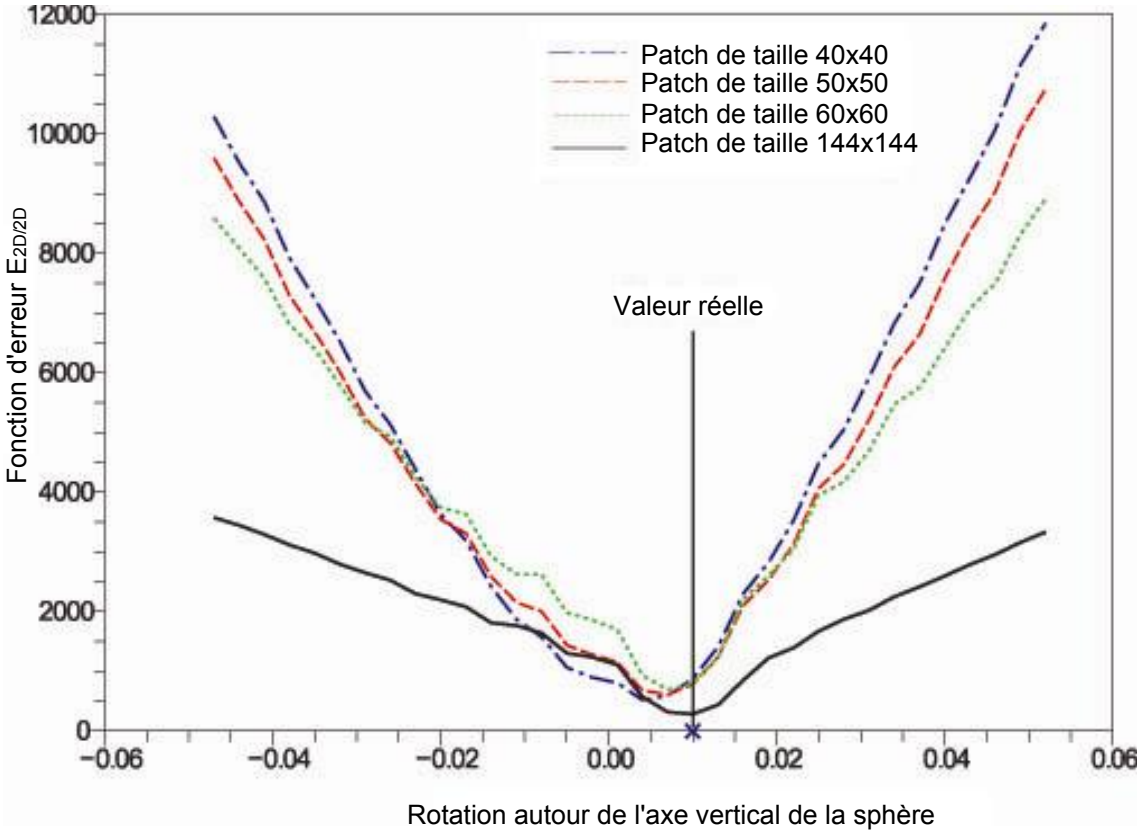


FIGURE 2.7 : Variation de la fonction d'erreur relative à la mise en correspondance 2D/2D par rapport à $\Delta\theta_y$. La valeur réelle de $\Delta\theta_y$ égale à 0.01 rd, ne correspond pas au minimum de la fonction d'erreur pour les patchs de petites tailles.

être suffisante afin qu'il contienne une information suffisante provenant de la composante diffuse. Cependant, lorsque la surface de l'objet est fortement réfléchissante ou lorsque le patch est positionné de manière non supervisée, cette condition s'avère difficile à respecter.

Estimation par mise en correspondance 3D/2D

La mise en correspondance 3D/2D intègre un modèle physique de la surface de la sphère qui tient compte à la fois les composantes diffuse et spéculaire de la réflexion. Le figure 2.8 présente la variation de la fonction d'erreur autour de la variable angulaire θ_y recherchée. Nous constatons que la convergence parfaite (valeur de la fonction d'erreur nulle) est assurée ce qui signifie que l'image de synthèse correspond exactement à l'image courante analysée.

Il faut relativiser ce résultat à deux niveaux. Tout d'abord, cette perfection «théorique» apparaît puisque les images analysées sont issues de la synthèse réalisée à partir du même modèle que celui utilisé lors de la mise en correspondance

3D/2D. Par ailleurs, deux opérations de rendu 3D sont nécessaires pour calculer le gradient du vecteur de paramètres dans la relation 2.20 entraînant un temps de calcul important. En effet, nous atteignons des temps de traitement 10 fois supérieurs à ceux obtenus lors de la mise en correspondance 2D/2D pour le patch de taille 40×40.

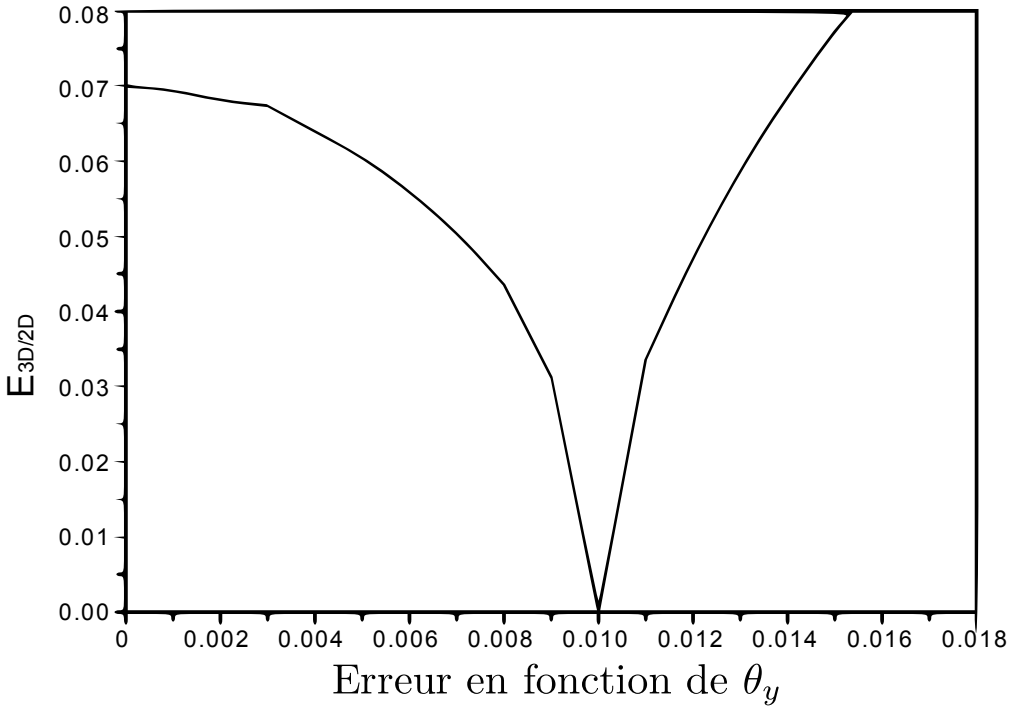


FIGURE 2.8 : Variation de l'erreur relative à la mise en correspondance 3D/2D.

Pour terminer, nous avons étudié le comportement de l'erreur d'estimation à l'issue de l'analyse d'une séquence complète comportant plusieurs images. Nous avons constaté que la méthode de mise en correspondance 2D/2D se caractérise par une accumulation des erreurs d'estimation obtenues lors du traitement de chaque image de la séquence. Ce qui n'est pas le cas avec la mise en correspondance 3D/2D (voir chapitre 4).

2.5 Conclusion

Dans ce chapitre, nous avons présenté deux méthodes d'estimation du mouvement 3D décrites dans la littérature : la méthode par mise en correspondance

3D/2D et la méthode par mise en correspondance 2D/2D. Ensuite, nous avons examiné les particularités de ces deux méthodes dans le cas du suivi d'une sphère de surface réfléchissante. Nous avons remarqué que la méthode 3D/2D fournit de bons résultats malgré la présence des reflets. Cependant, elle nécessite le calcul d'une image synthétique plusieurs fois à chaque itération de l'algorithme de minimisation de la fonction d'erreur, augmentant de ce fait le temps de calcul de façon très significative.

En comparaison, la mise en correspondance 2D/2D est une méthode plus rapide qui fournit des résultats précis à condition que le mouvement soit de faible amplitude. La limitation majeure de cette méthode est son incapacité à tenir compte des propriétés de réflexion de l'objet en mouvement tout particulièrement la composante spéculaire qui perturbe la convergence de l'algorithme.

Par ailleurs, les deux méthodes souffrent d'un phénomène d'accumulation de l'erreur d'estimation, voire de divergence de l'algorithme, lors de l'analyse d'une longue séquence d'images.

Notre approche hybride 2D/2D et 3D/2D, détaillée dans le chapitre suivant, tente de tirer partie des avantages de l'une et de l'autre des deux méthodes, tout en évitant leurs inconvénients. Dans cette nouvelle approche, la composante spéculaire est utilisée comme source d'information supplémentaire, qui permet de discriminer des mouvements ambigus détaillés dans le paragraphe 2.3.

Chapitre 3

Analyse du mouvement d'une sphère réfléchissante

3.1 Introduction

Dans le chapitre précédent, nous avons décrit deux méthodes d'estimation de mouvement, l'une reposant sur la recherche d'une correspondance entre un modèle 3D de l'objet et les images 2D de la séquence, l'autre sur la recherche d'une correspondance entre des fenêtres extraites de deux images successives. Nous avons précisé les limitations de ces deux méthodes lorsqu'elles sont utilisées pour estimer le mouvement d'une sphère de surface réfléchissante.

La première méthode consiste à analyser l'image $\mathbf{I}_{n+1}(\mathbf{x})$ de la séquence afin de déterminer les paramètres de position et d'orientation de la sphère à l'instant $(n+1)\cdot\Delta t$. Ces paramètres scalaires sont regroupés dans un vecteur $\mathbf{p}_{n+1}$. Dans cette approche, un modèle de la scène est requis, incluant notamment le type et la position des sources lumineuses, ainsi que les paramètres de texture et de réflectance de la sphère. Dans un premier temps, il s'agit de calculer une image synthétique de la scène lorsque la sphère est positionnée dans l'espace via la transformation paramétrique $M_{3D}(\mathbf{p}_{n+1})$. Cette image synthétique est ensuite comparée à l'image réelle $\mathbf{I}_{n+1}(\mathbf{x})$ par l'intermédiaire d'une fonction d'erreur $E_{3D/2D}(\mathbf{p}_{n+1})$. La minimisation de la fonction d'erreur par rapport à $\mathbf{p}_{n+1}$ fournit les paramètres de position et d'orientation recherchés.

Cette méthode présente l'avantage de permettre l'estimation de la position et de l'orientation de la sphère même lorsque sa surface est fortement réfléchissante, du fait que cette propriété est introduite explicitement dans le modèle. Le principal inconvénient de cette méthode est qu'elle nécessite de calculer un rendu réaliste de

la scène plusieurs fois à chaque itération de minimisation de la fonction d'erreur, d'où un temps de calcul prohibitif.

La deuxième méthode, introduite par Lucas et Kanade en 1981 consiste en une mise en correspondance 2D/2D [C21]. Cette technique exploite l'information de texture contenue dans les images. Son objectif est de retrouver la variation $\Delta\mathbf{p}_n$ des paramètres de position et d'orientation décrivant le mouvement de l'objet entre les instants $n\cdot\Delta t$ et $(n+1)\cdot\Delta t$. Ceci est accompli en comparant deux images successives $\mathbf{I}_n(\mathbf{x})$ et $\mathbf{I}_{n+1}(\mathbf{x})$. La méthode de Lucas-Kanade consiste à exploiter la texture de l'objet visible dans $\mathbf{I}_n(\mathbf{x})$ dans le but de calculer la nouvelle disposition de cette texture lorsque l'objet suit un mouvement décrit par $\Delta\mathbf{p}_n$. Ce calcul consiste en une simple transformation 2D, équivalente localement à la projection dans le plan image du mouvement 3D. La nouvelle disposition de texture est ensuite comparée à l'image analysée $\mathbf{I}_{n+1}(\mathbf{x})$ grâce à une fonction d'erreur $E_{2D/2D}(\Delta\mathbf{p}_n)$ dont la minimisation par rapport à $\Delta\mathbf{p}_n$ fournit le déplacement recherché.

Cette méthode converge en général vers des résultats assez précis avec des temps de calcul relativement faibles lorsque l'objet suivi possède une surface lambertienne et que son mouvement est de faible amplitude. Cependant, lorsque l'objet possède une surface réfléchissante, cette méthode rencontre de sérieux problèmes. En effet, dans la technique de Lucas-Kanade, la transformation 2D appliquée à une région Ω de $\mathbf{I}_n(\mathbf{x})$ est la même pour tout point image $\mathbf{x} \in \Omega$. Cependant, le mouvement apparent d'un élément de surface diffusant la lumière n'est pas forcément le même que celui d'un élément de surface réfléchissant la lumière de façon spéculaire, même si leurs mouvements réels dans l'espace sont identiques.

Dans le cas de la sphère réfléchissante, une rotation pure autour d'un axe n'entraîne pas de mouvement apparent de la composante spéculaire, alors que le mouvement apparent de la composante diffuse correspond directement à la projection du mouvement 3D dans le plan image. De ce fait, si la méthode de Lucas-Kanade est appliquée directement, aucune déformation 2D ne permet d'aligner parfaitement les textures entre les deux images, ce qui exclut la convergence vers la bonne solution.

Comme nous l'avons présenté dans le chapitre bibliographique, le problème

de la reconstruction du mouvement 3D d'objets de surface réfléchissante n'a pas été spécifiquement analysé jusqu'alors. Les méthodes déjà proposées tendent à contourner le problème des reflets, soit en considérant des patchs dans lesquels la valeur de la fonction image provient essentiellement de la composante diffuse, soit en prenant en compte des caractéristiques géométriques de l'objet, tels les contours, lesquels ne sont pas affectées par les reflets.

Lorsque la composante spéculaire est fortement présente sur les images, le choix automatique d'un patch ne contenant pas le reflet est un problème complexe. En outre, dans le cas spécifique qui nous intéresse dans ce travail, c'est à dire l'analyse du mouvement d'une sphère réfléchissante, le mouvement apparent du contour extérieur n'apporte aucune information relative aux mouvements de rotation.

L'intérêt majeur de l'approche présentée dans ce chapitre consiste à exploiter l'information provenant de la composante spéculaire, plutôt qu'à tenter de l'éliminer, et ce dans le but d'aboutir à des résultats plus précis. Cette méthode est une méthode hybride entre les méthodes 3D/2D et 2D/2D présentées auparavant. Elle est aussi précise que la méthode 3D/2D, mais moins gourmande en temps de calcul du fait qu'une partie des traitements consiste à déformer en 2D des voisinages extraits des images, plutôt qu'à calculer une image globale de rendu.

3.2 Approche proposée

3.2.1 Spécification du problème

Nous supposons que nous disposons d'une séquence d'images visualisant les positions successives d'une sphère de surface fortement réfléchissante. Nous supposons également connu le modèle de scène, qui inclut :

- le modèle de la sphère qui comprend le rayon R ainsi qu'une représentation de la texture intégrant les propriétés spéculaire et diffuse de la surface ;
- les paramètres p_0 de position et d'orientation de la sphère pour la première image de la séquence par rapport au repère caméra ;
- la position du (ou des) spot(s) lumineux par rapport à ce repère ainsi que les caractéristiques de l'éclairage. Le modèle d'éclairage de type Phong est utilisé

pour la synthèse des images ;

- les propriétés intrinsèques de la caméra telles sa distance focale f et la taille de son capteur ($Taille_x$ et $Taille_y$).

Notre méthode est récursive : supposant que le vecteur $\mathbf{p}_n$ représentant les paramètres de position et d'orientation de la sphère par rapport au repère caméra à l'instant $n \cdot \Delta t$ a été estimé, notre algorithme consiste à rechercher le mouvement entre les instants $n \cdot \Delta t$ et $(n+1) \cdot \Delta t$ décrit par le vecteur $\Delta\mathbf{p}_n$ en analysant $\mathbf{I}_{n+1}(\mathbf{x})$. Le mouvement estimé de façon incrémentale permet ensuite de mettre à jour les paramètres absolus $\mathbf{p}_{n+1}$ de position et d'orientation de la sphère à l'instant $(n+1) \cdot \Delta t$ en appliquant l'équation (2.8).

Dans la première partie de cette section nous présentons le principe de base de notre méthode, qui consiste à séparer les deux composantes spéculaire et diffuse lors du calcul des images synthétiques. L'étape de rendu fournit deux images, l'une pour la composante diffuse, l'autre pour la composante spéculaire. Chacune de ces deux images est traitée séparément par une transformation 2D qui lui est adaptée. Les deux images transformées sont ensuite combinées et comparées à l'image réelle $\mathbf{I}_{n+1}(\mathbf{x})$. Par la suite, nous détaillons les étapes de la méthode proposée.

3.2.2 Séparation spéculaire / diffus

Considérons en premier lieu le cas d'une sphère en rotation pure autour d'un de ses axes. Si on focalise l'analyse du mouvement sur une petite zone proche du centre du disque correspondant à la projection de la sphère, le mouvement apparent de la composante diffuse est très similaire à une translation dans une direction orthogonale à l'axe de rotation. En revanche, comme les reflets ne bougent pas dans le cas d'un mouvement de rotation, le mouvement apparent de la composante spéculaire est nul.

Considérons maintenant le cas d'une sphère en translation pure. Le mouvement apparent de la composante diffuse à proximité du centre du disque est également une translation. Par contre, cette fois-ci, le mouvement apparent de la composante spéculaire est non nul et correspond également à une translation.

Ces deux cas particuliers permettent de comprendre l'intérêt de tenir compte

de la différence entre les mouvements apparents des composantes spéculaire et diffuse. Cette différence est une source d'information primordiale dont il faut tenir compte lorsqu'on analyse le mouvement d'une sphère. Cette information est utilisée explicitement dans la méthode que nous proposons.

Notre approche s'inspire à la fois de la mise en correspondance 3D/2D, étant donné qu'un modèle 3D est utilisé pour synthétiser des images, et de la mise en correspondance 2D/2D, du fait que la recherche d'une correspondance est réalisée en transformant des patchs des images dans le domaine 2D. Afin de prendre en compte la différence qui peut exister entre les mouvements apparents des composantes diffuse et spéculaire, nous proposons d'utiliser une image synthétique pour chacune de ces deux composantes. Nous utilisons ensuite deux transformations 2D différentes pour déformer ces images. Enfin, les composantes déformées sont combinées et le résultat est comparé à l'image réelle. Cette méthode est illustrée par le synoptique de la figure 3.1.

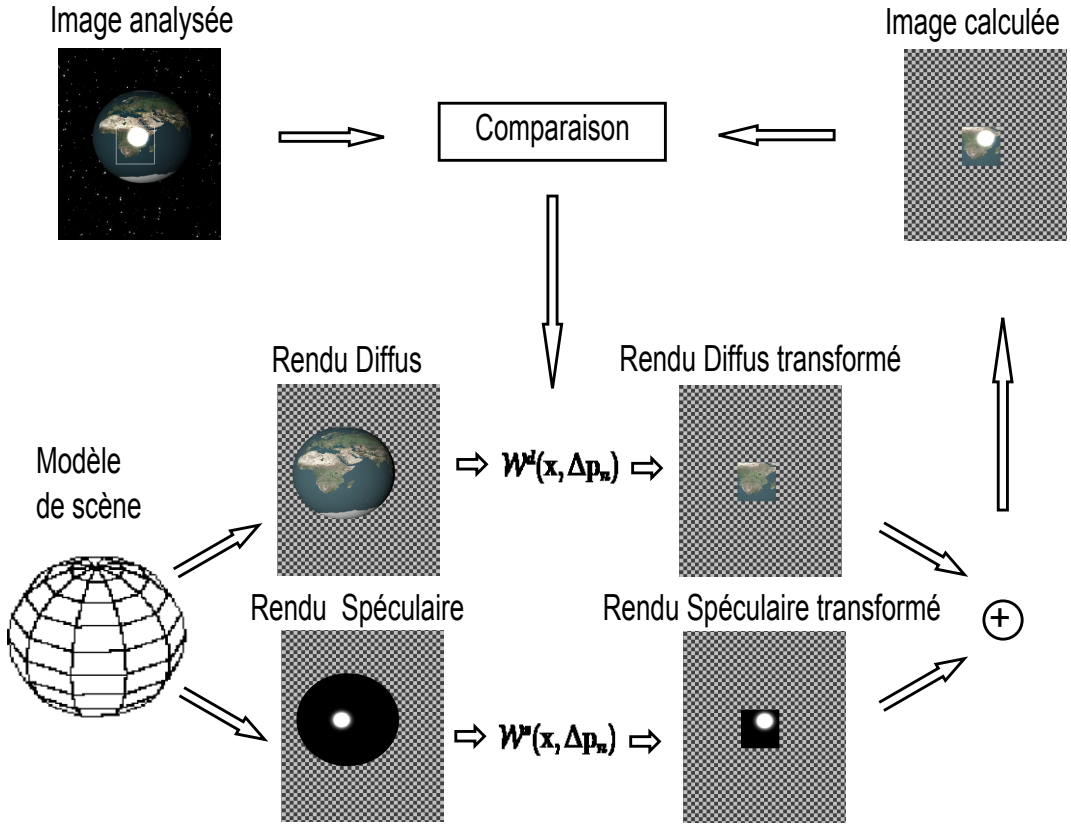
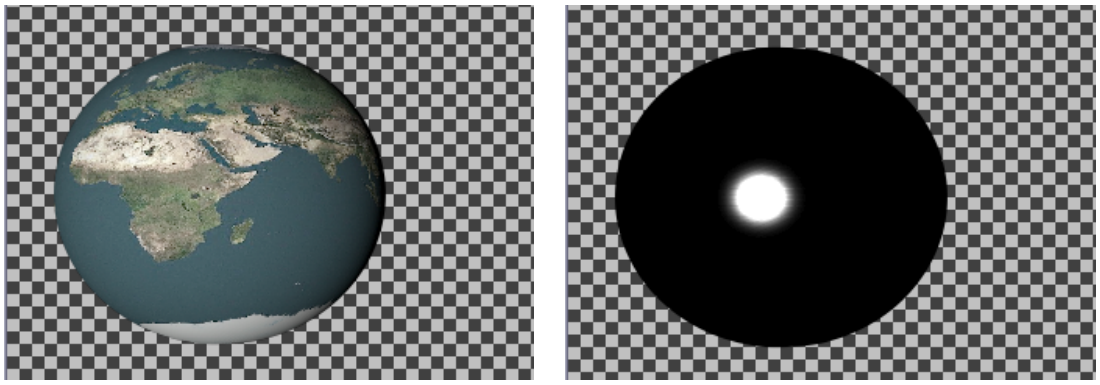


FIGURE 3.1 : Notre approche hybride

Ayant à disposition le modèle de la scène, ainsi que les paramètres p_n de posi-

tion et d'orientation retrouvés jusqu'à l'instant $n \cdot \Delta t$, nous pouvons calculer deux images de rendu. La première image, appelée image diffuse $\mathbf{R}_{n,d}(\mathbf{x}, \mathbf{p}_n)$, est calculée en considérant une surface purement lambertienne (figure 3.2(a)). La deuxième, appelée image spéculaire $\mathbf{R}_{n,s}(\mathbf{x}, \mathbf{p}_n)$, est calculée en considérant une surface noire parfaitement réfléchissante (figure 3.2(b)).



(a) Rendu diffus $\mathbf{R}_{n,d}(\mathbf{x}, \mathbf{p}_n)$ (b) Rendu spéculaire $\mathbf{R}_{n,s}(\mathbf{x}, \mathbf{p}_n)$

FIGURE 3.2 : Les rendus diffus et spéculaire calculés en fonction des paramètres $\mathbf{p}_n$ estimés pour l'image précédente.

3.2.3 Déformation et combinaison des images

Les deux images synthétiques sont ensuite transformées à l'aide de deux transformations 2D, $\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n)$ pour la composante diffuse et $\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n)$ pour la composante spéculaire. La transformation diffuse $\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n)$ est en tout point similaire à celle utilisée dans la méthode de mise en correspondance 2D/2D détaillée dans la section 2.3. La transformation spéculaire est détaillée dans ce qui suit.

En fonction du vecteur $\Delta\mathbf{p}_n$ recherché, représentant les variations des paramètres de position et d'orientation du mouvement par rapport au repère caméra, nous pouvons calculer $M_{2D}(\Delta\mathbf{p}_n)$ et donc $\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n)$ comme précédemment. Dans le cas d'une sphère, les mouvements de rotation pure n'ayant pas d'influence sur le mouvement apparent, il suffit d'annuler les paramètres de rotation lors du calcul de $\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n)$. Les paramètres de position pour la partie spéculaire sont ainsi donnés par : $\Delta\mathbf{p}'_n = [\Delta T_x, \Delta T_y, \Delta T_z, 0, 0, 0]^T$. La transformation finale, adaptée pour

traiter spécifiquement la composante spéculaire, est ainsi donnée par :

$$\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n) = \mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}'_n) \quad . \tag{3.1}$$

Dans la méthode de mise en correspondance 2D/2D initialement proposée par Lucas et Kanade, les fonctions image déformées par l'intermédiaire de la transformation 2D étaient vectorielles de type RVB. Dans notre approche, afin de mieux tenir compte des éventuels défauts de superposition qui peuvent apparaître quand les patchs sont situés à proximité du contour extérieur de la sphère, nous utilisons des images vectorielles de type RVBα, qui intègrent un coefficient d'opacité.

L'introduction d'un coefficient d'opacité permet de limiter les erreurs, mais pas de les supprimer complètement. En effet, les points qui se trouvent projetés à proximité du contour extérieur de la sphère dans l'image initiale peuvent être déplacés à l'extérieur du contour après transformation. Inversement, toujours à proximité du contour extérieur, des points du fond de la scène dans les images de rendu peuvent se retrouver superposés à des points de l'image de la sphère après transformation. Le coefficient d'opacité permet de compenser partiellement l'erreur liée à l'effet de bord, en limitant l'influence des points situés à la frontière de l'image de la sphère dans l'expression de l'erreur.

Le coefficient d'opacité est identique pour les deux images de rendu, diffuse et spéculaire. De ce fait, la composante correspondante des images sera notée indifféremment $\mathcal{R}^{\alpha}_{n,d}(\mathbf{x}, \mathbf{p}_n)$ ou $\mathcal{R}^{\alpha}_{n,s}(\mathbf{x}, \mathbf{p}_n)$ par la suite.

L'image synthétique finale $\tilde{\mathbf{R}}_n(\mathbf{x}, \Delta\mathbf{p}_n)$, utilisée pour l'estimation des paramètres du mouvement 3D par comparaison à $\mathbf{I}_{n+1}(\mathbf{x})$, est une combinaison des deux images déformées, correspondant aux composantes spéculaire et diffuse. Pour les composantes RVB, elle est déterminée par une simple addition point par point des deux images déformées $\mathbf{R}_{n,d}(\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n)$ et $\mathbf{R}_{n,s}(\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n)$. C'est ce modèle purement additif qui est utilisé dans les logiciels de synthèse d'images, quand on suppose qu'il n'y a pas de saturation du capteur.

$$\begin{aligned}
\tilde{\mathcal{R}}_n^r(\mathbf{x}, \Delta\mathbf{p}_n) &= \mathcal{R}_{n,d}^r(\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) + \mathcal{R}_{n,s}^r(\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) \ , \\
\tilde{\mathcal{R}}_n^v(\mathbf{x}, \Delta\mathbf{p}_n) &= \mathcal{R}_{n,d}^v(\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) + \mathcal{R}_{n,s}^v(\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) \ , \\
\tilde{\mathcal{R}}_n^b(\mathbf{x}, \Delta\mathbf{p}_n) &= \mathcal{R}_{n,d}^b(\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) + \mathcal{R}_{n,s}^b(\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) \ .
\end{aligned} \tag{3.2}$$

Pour la composante d'opacité α, il n'existe pas d'expression standard permettant de combiner deux images. Comme la combinaison n'a pas de sens physique, les divers opérateurs de combinaison qu'on peut imaginer sont aussi valables les uns que les autres. Nous avons choisi de combiner les deux images d'opacité en calculant la moyenne des images transformées diffuse et spéculaire :

$$\tilde{\mathcal{R}}_n^\alpha(\mathbf{x}, \Delta\mathbf{p}_n) = \frac{\mathcal{R}_{n,d}^\alpha(\mathbf{W}^d(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n) + \mathcal{R}_{n,s}^\alpha(\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n), \mathbf{p}_n)}{2} \ . \tag{3.3}$$

La fonction d'erreur permettant la comparaison de la combinaison des images déformées et de l'image réelle est obtenue en sommant, sur tous les points pour lesquels le coefficient d'opacité est non nul, les différences au carré entre les composantes couleur des deux images, chaque valeur étant pondérée par le coefficient d'opacité :

$$\begin{aligned}
E_{hybride}(\Delta\mathbf{p}_n) &= \frac{1}{\sum_{\mathbf{x};\tilde{\mathcal{R}}_n^\alpha(\mathbf{x},\Delta\mathbf{p}_n)\neq 0} \tilde{\mathcal{R}}_n^\alpha(\mathbf{x}, \Delta\mathbf{p}_n)} \cdot \sum_{\mathbf{x};\tilde{\mathcal{R}}_n^\alpha(\mathbf{x},\Delta\mathbf{p}_n)\neq 0} \tilde{\mathcal{R}}_n^\alpha(\mathbf{x}, \Delta\mathbf{p}_n) \cdot \\
&\quad \Big[(\tilde{\mathcal{R}}_n^r(\mathbf{x}, \Delta\mathbf{p}_n) - \mathcal{I}_{n+1}^r(\mathbf{x}))^2 \quad + \quad (\tilde{\mathcal{R}}_n^v(\mathbf{x}, \Delta\mathbf{p}_n) - \mathcal{I}_{n+1}^v(\mathbf{x}))^2 \\
&\quad + \quad (\tilde{\mathcal{R}}_n^b(\mathbf{x}, \Delta\mathbf{p}_n) - \mathcal{I}_{n+1}^b(\mathbf{x}))^2\Big] \ .
\end{aligned} \tag{3.4}$$

En pondérant chaque terme de différence au carré par le coefficient d'opacité, nous diminuons la confiance attribuée aux points qui se trouvent projetés dans l'image à proximité du contour extérieur de la sphère. Cette confiance devient nulle pour les points se projetant sur le fond de l'image, ce qui est un avantage supplémentaire de notre méthode par rapport à la mise en correspondance 2D/2D de Lucas-Kanade.

Comme pour la méthode de mise en correspondance 2D/2D, Z est calculé pour tous les points image $\mathbf{x} \in \Omega$ grâce à l'équation paramétrique de la sphère, dont les paramètres $\mathbf{p}_n$ de position et d'orientation sont connus pour l'image précédente (à l'instant $n \cdot \Delta t$).

3.3 Performances et limitations de la méthode

3.3.1 Comportement sur un exemple simple

Dans cette section, nous reprenons simplement le problème considéré dans la section 2.4 afin de prouver l'efficacité de notre technique. Nous considérons ainsi un déplacement de la sphère entre deux images qui est défini par le vecteur paramètre $\Delta\mathbf{p} = [0, 0, 0, 0, 0.01 \text{ rad}, 0]^T$, soit un mouvement apparent d'environ 1 pixel (0.96 pixel exactement) au niveau de la projection du centre de la sphère sur l'image. Comme précédemment, nous calculons la fonction d'erreur aux alentours de la valeur réelle du paramètre de rotation, en considérant les différentes tailles de patchs du paragraphe 2.4. Les courbes obtenues sont représentées sur la figure 3.3.

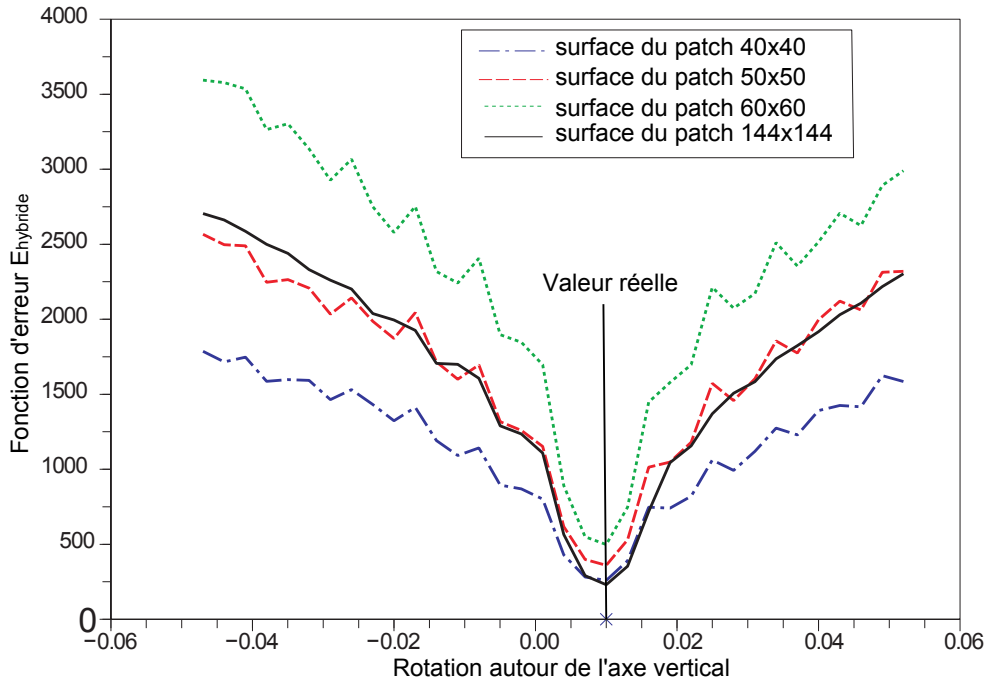


FIGURE 3.3 : Variation de la fonction d'erreur relative à notre méthode hybride par rapport à $\Delta\theta_y$. La valeur réelle de $\Delta\theta_y$ est égale à 0.01 radian et correspond au minimum de la fonction d'erreur même pour les patchs de petites tailles.

Nous remarquons que grâce à notre approche, le problème généré par la spécularité de la surface est résolu. En effet, le minimum de la fonction d'erreur pointe vers la valeur réelle égale à 0.01 radian même pour les patchs de très petite taille dans lesquels l'information provenant de la composante diffuse est très pauvre.

3.3.2 Intérêt de la séparation des composantes

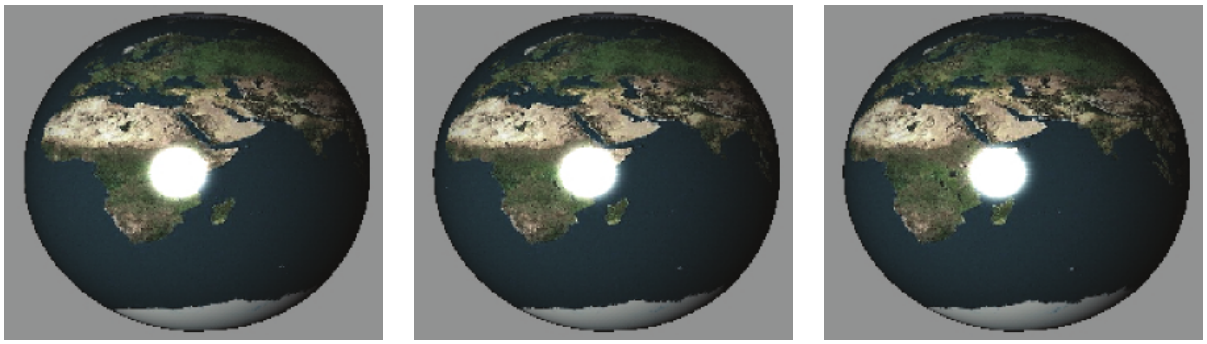
Dans notre méthode, l'image déformée qui est comparée à l'image réelle est obtenue par superposition de deux images déformées, l'une pour la composante diffuse, l'autre pour la composante spéculaire. Les déformations sont censées décrire au mieux le mouvement apparent en fonction des paramètres du mouvement 3D, sans qu'il soit nécessaire d'avoir recours à une synthèse à chaque itération.

L'intérêt d'avoir recours à deux transformations séparées est illustré par les images de la figure 3.4. La première série présente trois images d'une séquence obtenue par rendu complet de la scène, c'est à dire telles qu'elles seraient utilisées dans la méthode de mise en correspondance 3D/2D. On remarque aisément sur ces images que le mouvement apparent de la partie diffuse (texture) et de la partie spéculaire (reflet) sont différents dans le cas du mouvement de rotation.

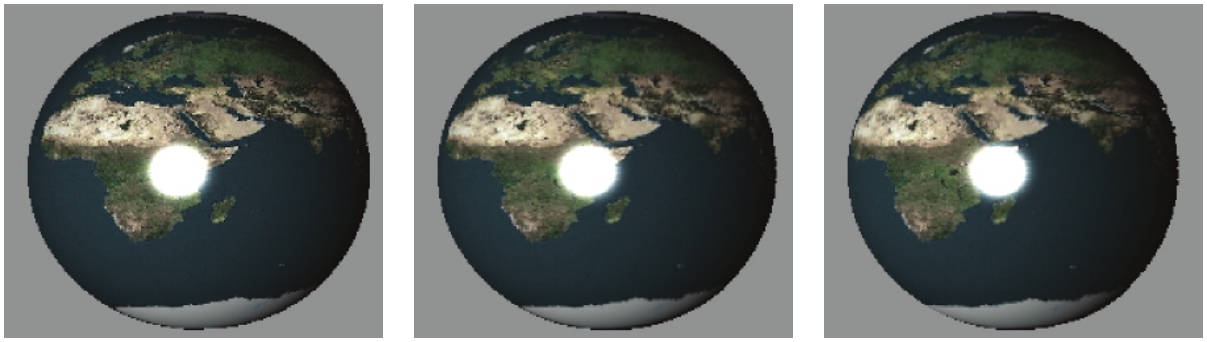
Les images de la deuxième série sont obtenues par la méthode de combinaison de deux images, diffuse et spéculaire, obtenues par déformation par deux transformations 2D/2D différentes. Les deux images initiales sont obtenues via un rendu séparé des composantes diffuse et spéculaire. Nous pouvons remarquer sur l'image de différence (figure 3.4(c)) que le warping séparé considéré dans notre approche est presque aussi précis que la synthèse complète, alors que le temps de calcul nécessaire à l'ajustement des paramètres est beaucoup plus faible.

3.3.3 Information apportée par la composante spéculaire

Dans cette partie, nous montrons que la composante spéculaire présente dans l'image est une source d'information et non un bruit qu'il faut minimiser. Pour ce faire, nous considérons les deux cas suivants : dans le premier la sphère a une surface réfléchissante, dans le second sa surface est purement lambertienne. Pour chacun de ces deux cas nous calculons deux images synthétiques entre les-



(a) Images obtenues par rendu complet



(b) Images obtenues par déformation des composantes séparées

(c) Images obtenues par différence des deux résultats précédents

FIGURE 3.4 : Mouvement apparent d'une sphère en rotation pure autour de son axe vertical. Images de rendu complet vs. images transformées par notre approche hybride.

quelles la sphère s'est déplacée en translation selon le vecteur paramètre $\Delta\mathbf{p} = [0, 0.01\ \mathbf{m}, 0, 0, 0, 0]^T$.

Dans ce cas, le déplacement réel de 0.01 m correspond à un mouvement ap-

parent d'environ 1 pixel dans l'image de l'ensemble de la projection de la sphère. Comme nous l'avons déjà souligné, il est difficile de faire la différence entre cette translation et une rotation selon un l'axe horizontal quand on cherche à mettre en correspondance des patchs par la méthode standard 2D/2D.

Nous appliquons ensuite la méthode hybride sur chacune de ces deux paires d'images, sphère réfléchissante ou de surface lambertienne, sur le patch 40×40 de la figure 2.6(a). Dans le cas où la sphère a une surface lambertienne, notre méthode correspond exactement à la mise en correspondance 2D/2D. Les résultats [1] sont présentés sur le tableau 3.3.3.

	2D/2D ($\times 10^{-3}$)	Hybride ($\times 10^{-3}$)	Valeurs réelles ($\times 10^{-3}$)
ΔT_x en m	0	0	0
ΔT_y en m	-1	-8	-10
ΔT_z en m	1	0	0
$\Delta\theta_x$ en rd	-7	0	0
$\Delta\theta_y$ en rd	0	1	0
$\Delta\theta_z$ en rd	-4	-2	0

Nous pouvons remarquer sur ces résultats que la présence de la composante spéculaire lève partiellement l'ambiguïté entre les mouvements de translation suivant l'axe vertical et de rotation pure autour de l'axe horizontal de la sphère. En effet, la translation suivant l'axe vertical est estimée avec une erreur d'environ 0.2 pixel en présence de la composante spéculaire lorsque la scène est modélisée avec précision. Dans le cas d'une surface lambertienne, l'estimation du mouvement d'environ 1 pixel est effectuée avec une erreur importante de 0.9 pixel.

Ceci est dû au fait que la translation suivant la verticale est confondue avec une rotation pure, estimée à 0.007 rd, autour de l'axe horizontal de la sphère. En revanche, ce mouvement est correctement estimé à une valeur nulle lorsqu'on tient compte de la composante spéculaire. Nous remarquons donc qu'avec notre méthode nous sommes capables d'estimer des mouvements de très faible amplitude, qui sur l'image apparaissent avec une résolution sub-pixellique.

De plus, les erreurs d'estimation des autres paramètres de mouvement sont généralement plus importantes dans le cas d'une surface lambertienne, comme

1. D'après un test préalable, nous avons fixé le nombre d'itérations à 200 pour toutes les simulations présentées dans ce chapitre. En fixant le nombre d'itérations, nous sommes capables de mener des comparaisons sur des bases équivalentes.

nous le remarquons sur l'estimation de ΔT_z et $\Delta\theta_z$. Ce qui n'est pas le cas pour $\Delta\theta_y$, estimé à 0.001 rd alors que la vraie valeur est nulle.

Nous pouvons donc conclure que la prise en compte de la composante spéculaire apporte une information additionnelle utile pour lever les ambiguïtés sur les mouvements apparemment similaires, lesquels ne peuvent pas être différenciés lorsqu'on ne tient compte que de la composante diffuse. Dès lors, l'estimation est obtenue avec une précision sub-pixellique.

Lorsque nous analysons deux images successives, notre méthode est équivalente à la méthode de mise en correspondance 2D/2D dans le cas d'une surface lambertienne et d'une séquence ne comportant que deux images. Lorsque la séquence est plus longue, même en absence de composante spéculaire, notre méthode présente un avantage par rapport à la méthode standard de mise en correspondance 2D/2D. En effet, une correction globale est apportée à chaque mise à jour du modèle 3D, lequel sert à calculer la valeur initiale de l'image synthétique lors de l'analyse d'une nouvelle image. Ceci permet d'éviter l'accumulation d'erreurs lors de la mise en correspondance 2D/2D.

3.3.4 Contrainte sur les sources lumineuses

Pour définir la transformation qui permet de déformer l'image de la composante spéculaire, nous avons supposé que les rotations de la sphère n'ont pas d'influence sur le mouvement apparent. Cette supposition reste vraie quelle que soit la position des sources lumineuses dans la scène. Par contre, nous avons implicitement supposé qu'une translation de la sphère entraîne les mêmes mouvements apparents de translation pour les composantes diffuses et spéculaires. Cette supposition n'est pas correcte quand les sources lumineuses ne sont pas situées à l'infini.

Le respect de cette contrainte est essentiel pour que notre approche aboutisse à de bons résultats. Les sources d'éclairage doivent être suffisamment éloignées afin qu'on puisse considérer que les mouvements apparents des composantes spéculaire et diffuse sont identiques pour la partie du mouvement 3D correspondant à une translation. Cette contrainte pourrait être levée en introduisant une technique plus complexe de détermination de la transformation $\mathbf{W}^s(\mathbf{x}, \Delta\mathbf{p}_n)$, qui ne consisterait

pas simplement à annuler les paramètres de rotation.

3.4 Déclinaisons de la méthode

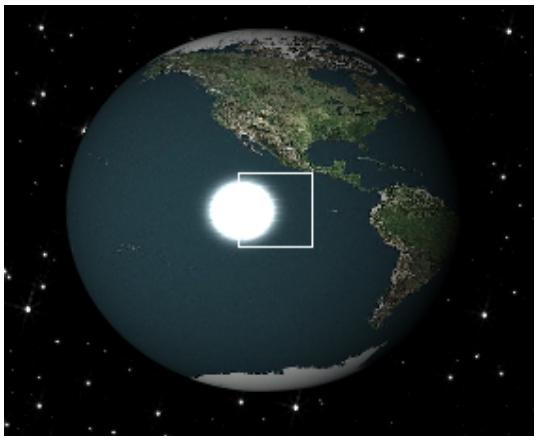
La méthode proposée repose sur une recherche de correspondance optimale entre une combinaison d'images de synthèse déformées et l'image réelle. A la base, cette comparaison est réalisée sur un patch rectangulaire dont la position et la taille doivent être correctement sélectionnées. Nous indiquons dans la suite les critères qui permettent de choisir correctement ce patch. Ensuite, nous montrons qu'il est possible de généraliser l'expression de la fonction d'erreur afin d'intégrer plusieurs patchs dans le calcul et indiquons plusieurs approches visant à déterminer la position et la taille de ces patchs.

3.4.1 Problème du choix d'un patch unique

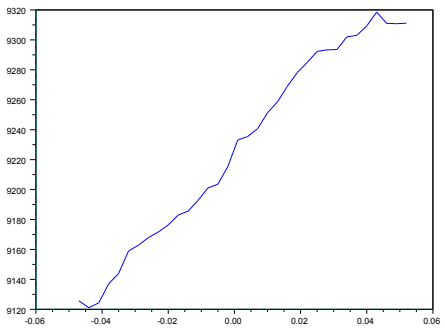
Le choix du patch sur lequel est réalisée la comparaison est un point crucial pour la réussite de la mise en correspondance. Dans le cas des méthodes standard de mise en correspondance 2D/2D, ce choix à lui seul a fait l'objet de plusieurs études [C32]. Quand on analyse uniquement le mouvement de façon incrémentale, c'est à dire sur la base de deux images successives, les critères de choix sont les suivants :

- En premier lieu, le patch doit contenir les deux sources d'information spéculaire et diffuse. Il s'agit donc de centrer le patch sur une zone de la sphère contenant une texture suffisante et intégrant le reflet. Vu que la profondeur de chaque élément de surface est calculée pour chaque point image, plus le patch est grand, plus l'information est riche, et donc plus le résultat est précis. En revanche, le temps de calcul augmente proportionnellement à la surface du patch.
- Ensuite, plus l'amplitude du mouvement apparent est importante, plus le patch doit être grand. Sur les séquences utilisées comme exemple de référence, un patch de 60×60, centré sur le disque correspondant à la projection de la sphère dans l'image, offre un bon compromis entre temps de calcul et précision des résultats.

Cependant, lorsque l'analyse du mouvement est réalisée sur une séquence complète, durant laquelle la sphère se déplace parfois de façon importante, il devient moins évident de rassembler les deux informations spéculaire et diffuse dans un patch de taille fixe (par exemple 60×60) et toujours centré sur le même point de la projection de la sphère. En effet, pour certaines positions et orientations de la sphère, le patch peut ne contenir qu'une information diffuse très pauvre. Par exemple, considérons le cas de la figure 3.5(a) lors de l'analyse du mouvement de translation $\Delta\mathbf{p} = [0.01 \text{ m}, 0, 0, 0, 0, 0]^T$ entre $\mathbf{I}_n$ et $\mathbf{I}_{n+1}$. Pour un patch positionné sur le centre de la projection de la sphère, la fonction d'erreur aux alentours de la valeur réelle égale à 0.01 m (fig.3.5(b)) n'est pas fiable puisque la zone visible, ici le bleu uniforme de l'océan, n'est pas suffisamment texturée.



(a) Image $\mathbf{I}_n$ avec le patch encadré en blanc



(b) Fonction d'erreur où la valeur recherchée correspond à -0.01 m.

FIGURE 3.5 : Le patch sélectionné dans (a) ne contient pas assez d'information diffuse puisque la zone qu'il englobe est peu texturée.

De plus, un patch positionné dans la même zone tout au long de la séquence peut contenir peu (ou pas) d'information d'origine spéculaire dans certaines situations. Dans ce cas, notre méthode redevient équivalente à la mise en correspondance 2D/2D sur une surface lambertienne et nous risquons alors d'avoir des résultats erronés tels que ceux présentés dans le tableau 3.3.3.

3.4.2 Extension multi-patch

Pour éviter les problèmes liés au positionnement et au dimensionnement d'un patch unique, nous proposons en premier lieu d'utiliser plusieurs patchs distribués

d'une façon aléatoire sur la surface projetée de la sphère. Afin que les patchs non significatifs, au sens des critères évoqués précédemment, ne perturbent pas le processus de recherche de la bonne correspondance, nous imposons a posteriori une condition pour valider chaque patch : la variance des valeurs de l'image pour au moins une des trois composantes RVB doit être supérieure à un certain seuil. Un patch sélectionné de façon aléatoire mais dont la variance maximale est inférieure au seuil est rejeté et remplacé par un autre.

Le fait d'avoir plusieurs patchs distribués sur la partie visible de la sphère, au lieu d'un seul, présente l'avantage de procurer une information plus riche. D'autre part, les transformations 2D utilisées pour déformer chaque patch peuvent approcher plus fidèlement la projection de la transformation 3D.

Par exemple, lorsque la sphère tourne autour de son axe vertical, le champ de vecteurs mouvement apparent correspond à celui présenté sur la figure 3.6(b). Nous remarquons que le module du vecteur mouvement apparent dépend de la position du point sur la sphère : plus le point est proche du contour extérieur de la sphère, plus l'amplitude du mouvement apparent est faible. Par contre, pour une sphère en translation horizontale, tous les vecteurs mouvement apparent ont le même module (fig. 3.6(a)).

On comprend aisément qu'on peut différencier plus simplement ces deux mouvements en utilisant plusieurs patchs distribués sur la surface visible de la sphère, du fait que les transformations 2D locales adaptées permettent une estimation plus précise du mouvement global.

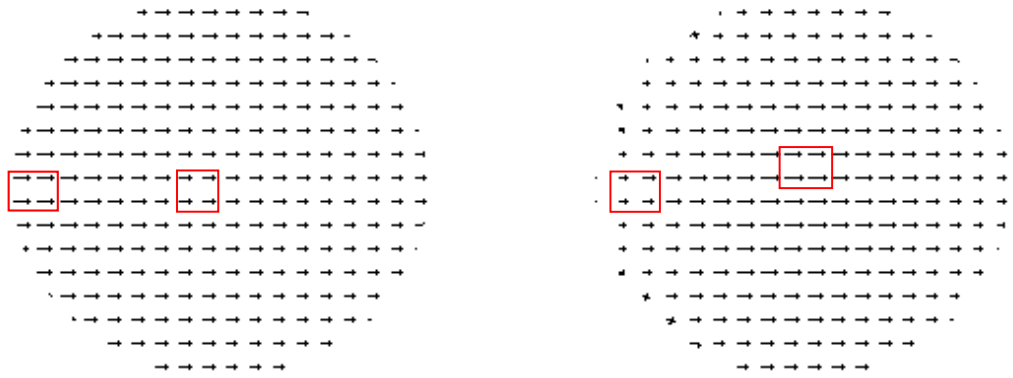
Dans le cas multi-patch, la fonction d'erreur $E^{total}_{hybride}(\Delta\mathbf{p}_n)$ totale correspond à une combinaison des erreurs $E^{patch}_{hybride}(\Delta\mathbf{p}_n)$ relatives à chacun des patchs :

$$E^{total}_{hybride}(\Delta\mathbf{p}_n) = \frac{1}{V_{tot}} \sum_{patch=0}^{N_{patch}-1} [V^{patch}_{max} * E^{patch}_{hybride}(\Delta\mathbf{p}_n)] \tag{3.5}$$

où :

- N_{patch} est le nombre de patchs ;
- $V_{tot} = \sum_{patch=0}^{N_{patch}-1} [V^{patch}_{max}]$ est la somme des variances maximales calculées sur chacun des patchs.

Avec cette expression, qui inclut des coefficients de pondération, nous attribuons



(a) Champ de mouvement 2D dans le cas d'une translation suivant l'horizontale

(b) Champ de mouvement 2D dans le cas d'une rotation autour de la verticale

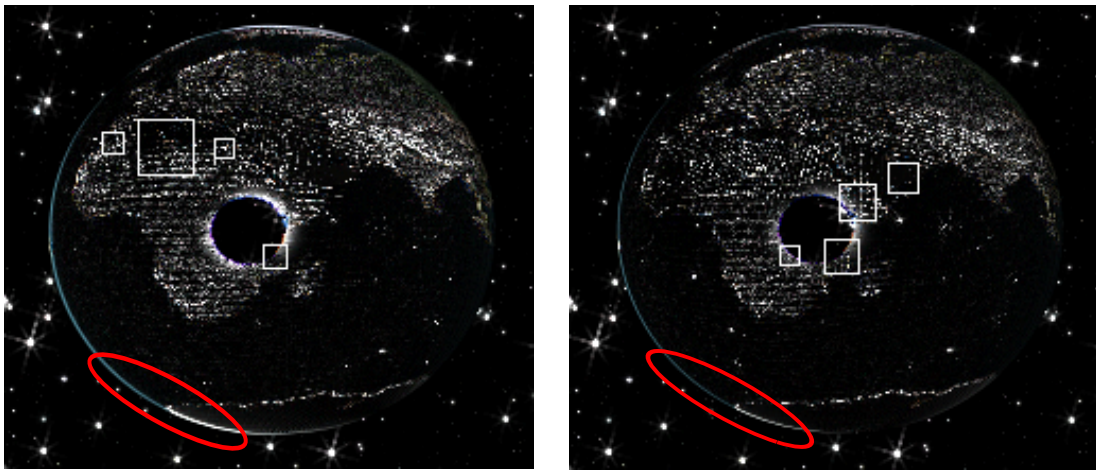
FIGURE 3.6 : Comparaison des champs de mouvement pour une rotation et une translation de la sphère. Nous remarquons que la translation est caractérisée par un champ uniforme alors que pour la rotation les vecteurs calculés sont plus courts à la périphérie qu'au centre.

plus de confiance à un patch ayant une variance plus élévée, ce critère indiquant qu'il contient une information suffisamment riche.

3.4.3 Stratégie de mise à jour des patchs

Lors de l'analyse de $\mathbf{I}_{n+1}$ visant à déterminer le vecteur $\Delta\mathbf{p}_n$ qui représente les paramètres du mouvement 3D, le choix aléatoire des patchs n'est pas forcément optimal. Cela peut entraîner des erreurs sur les valeurs estimées du vecteur paramètre. Du fait que l'estimation du vecteur paramètre global est réalisée de façon incrémentale, on peut supposer que ces erreurs successives s'accumulent, risquant ainsi de faire diverger le processus d'estimation du mouvement.

En fait, dans la plupart des simulations que nous avons réalisées (décrites dans le chapitre suivant) il s'avère que l'algorithme proposé est capable de compenser les erreurs successives. Par exemple, la figure 3.7 présente deux images d'erreur, obtenues par soustraction pixel à pixel de l'image déformée en fin d'itération et de l'image réelle, pour des images successives de la séquence. Nous pouvons remarquer, dans la partie encerclée de ces deux images, que l'erreur pour l'image de la figure 3.7(b) est inférieure à celle pour l'image de la figure 3.7(a).

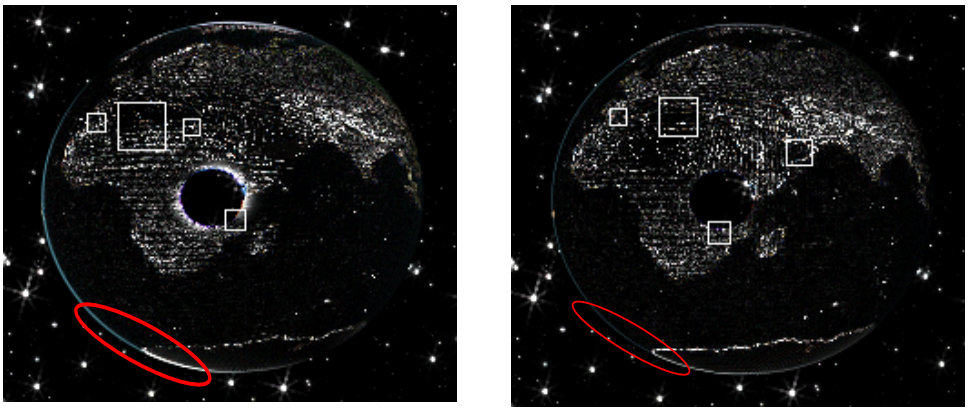


(a) Image de comparaison entre l'image calculée et l'image réelle à l'instant n de la séquence

(b) Image de comparaison entre l'image calculée et l'image réelle à l'image $n+1$ de la séquence.

FIGURE 3.7 : Images de comparaison entre l'image calculée et l'image réelle de la séquence mettant en relief les patchs considérés. Nous remarquons que l'algorithme corrige à l'image $n+1$ l'erreur causée par une disposition de patch non adéquate lors de l'analyse de l'image n.

Malgré tout, afin de diminuer le risque de divergence lié à une succession de choix non optimaux d'un ensemble de patchs, une amélioration de la méthode consiste à réinitialiser l'ensemble des patchs après un certain nombre d'itérations de recherche du minimum de la fonction d'erreur.



(a)Image de comparaison entre l'image calculée et l'image réelle à l'image 8 de la séquence sans mise à jour de patch

(b) Image de comparaison entre l'image calculée et l'image réelle à à l'image 8 de la séquence avec mise à jour du patch.

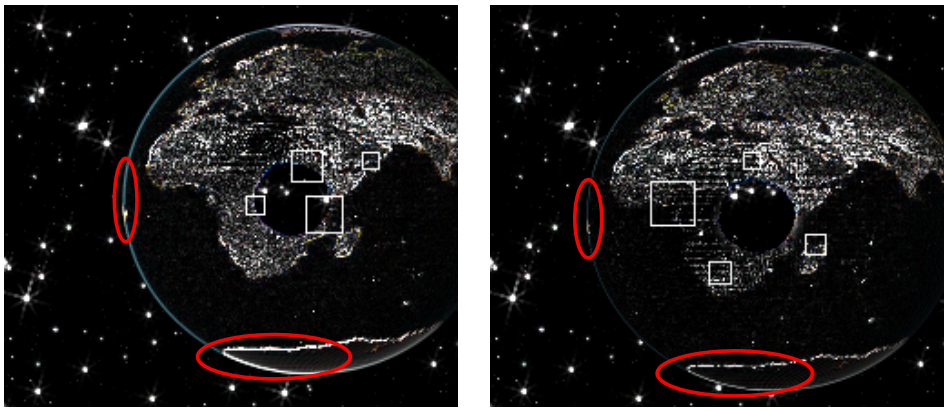
FIGURE 3.8 : Images de comparaison entre l'image calculée et l'image réelle lors de l'analyse d'une image d'une séquence complète suivant les deux méthodes de choix des patchs.

La figure 3.8 présente deux images d'erreur pour deux versions de l'algorithme, avec ou sans réinitialisation des patchs. On constate aisément, notamment dans la partie encerclée, que l'erreur est plus faible lorsque les patchs sont réinitialisés périodiquement en cours d'estimation.

3.4.4 Mise à jour des patchs en ciblant l'erreur

Une autre technique de choix de patch, visant à améliorer la rapidité et la qualité de convergence de notre algorithme, consiste à sélectionner des patchs dans lesquels l'erreur entre l'image réelle et l'image prédite est supérieure à un seuil fixé empiriquement. On suppose dans ce cas que ce sont dans les zones à l'intérieur desquelles l'erreur est initialement importante qu'il faut rechercher la bonne correspondance en ajustant les paramètres des transformations.

Avant de sélectionner un ensemble de patchs significatifs selon ce critère, nous mettons à jour le modèle 3D et recalculons les deux images synthétiques. Cette mise à jour périodique du modèle permet d'augmenter la précision des résultats, sans pour cela augmenter de façon significative le temps de calcul. D'autre part, elle permet de limiter l'erreur sur la fonction de mise en correspondance, du fait que les images de rendu intermédiaire décrivent mieux la scène que celles qui ont été calculées initialement lors du passage à la nouvelle image $\mathbf{I}_{n+1}$.



(a)L'erreur subsiste malgré les améliorations apportées.

(b) Une amélioration est aperçue.

FIGURE 3.9 : Images de comparaison entre l'image calculée et l'image réelle lors de l'analyse de l'image 50 de la séquence suivant deux méthodes de choix du multi-patch. Nous pouvons remarquer que sur le contour gauche de la sphère l'erreur a diminué.

Nous pouvons remarquer sur la figure 3.9(b) une amélioration sur la fonction d'erreur. Les patchs schématisés représentent les patchs utilisés entre les itérations 150 à 200. Dans cet exemple, les images de rendu sont calculées toutes les 50 itérations.

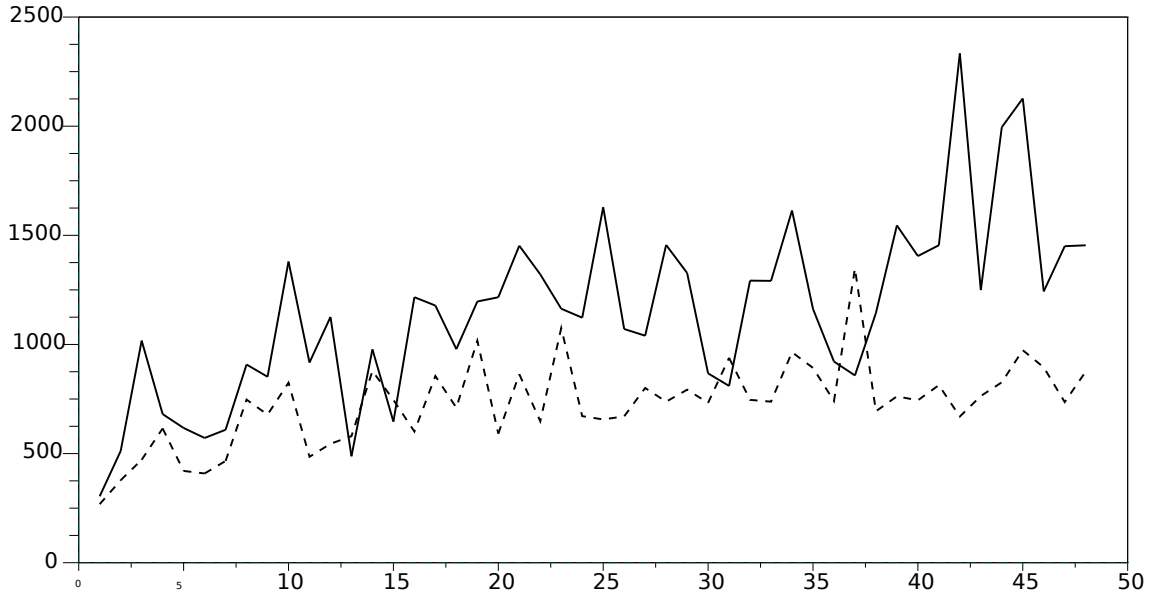


FIGURE 3.10 : fonctions d'erreur $E_{hybride}(\Delta \mathbf{p}_{n,200})$ calculées sur l'ensemble des points image pour $n = [2 \dots 50]$, pour la méthode standard de choix de patch (en trait plein) et pour la méthode ciblant l'erreur (en pointillés).

La figure 3.10 présente deux évolutions de la fonction $E_{hybride}(\Delta \mathbf{p}_{n,200})$, pour $n = [2 \dots 50]$, calculées sur l'ensemble des points image suivant les deux méthodes du choix du patch. Pour chaque image, l'algorithme de recherche de l'erreur minimale est appliqué pendant 200 itérations. On constate aisément qu'avec l'approche de choix du patch ciblant l'erreur, la fonction d'erreur converge globalement vers des valeurs plus faibles.

3.5 Conclusion

Dans ce chapitre, nous avons présenté une méthode d'estimation du mouvement 3D d'une sphère réfléchissante. A la différence de la mise en correspondance 2D/2D, notre méthode considère la propriété de spécularité comme source d'information et non comme source de bruit. De plus, elle permet d'éviter l'accumulation de l'erreur grâce à la mise à jour régulière du modèle 3D.

Notre méthode est aussi précise que la mise en correspondance 3D/2D, alors qu'elle ne nécessite pas de calculer systématiquement les images de rendu. Dans les deux implantations que nous avons réalisées, qui utilisent les mêmes fonctions de synthèse et de comparaison des images, la méthode proposée est 7 fois plus rapide que la mise en correspondance 3D/2D. Cela est dû au fait que nous effectuons beaucoup moins d'opérations de rendu complet de la scène, lesquelles sont très gourmandes en temps de calcul.

Plusieurs variantes de la méthode ont été décrites, fondées sur des techniques légèrement différentes de sélection ou de mise à jour du ou des patchs sur lesquels la fonction d'erreur est évaluée.

En résumé, lors de l'analyse d'une séquence constituée de N images, la méthode peut être implantée au moyen des étapes suivantes :

- $n = 0$; $\mathbf{p}_0$ est fixé selon l'application. Une connaissance a priori de la position de la sphère sur la première image peut-être disponible sinon il faut la définir manuellement.
- le modèle 3D est mis à jour grâce à $M_{3D}(\mathbf{p}_n)$.
- les deux images de rendu $\mathbf{R}_{n,d}(\mathbf{x}, \mathbf{p}_n)$ et $\mathbf{R}_{n,s}(\mathbf{x}, \mathbf{p}_n)$ sont calculées séparément.
- on calcule les positions des patchs ainsi que la surface de chacun sachant leur nombre et leur surface totale ainsi que la surface de l'objet projetée sur $\mathbf{I}_n(\mathbf{x})$.
- disposant des deux rendus et de l'image $\mathbf{I}_{n+1}(\mathbf{x})$ ainsi que des patchs on recherche le minimum de la fonction d'erreur.
- après les itérations, on passe à l'image $n + 1$ et on initialise $\mathbf{p}_{n+1} = \mathbf{p}_n + \Delta\mathbf{p}_n$.

Dans le chapitre suivant, nous présentons en détail les performances obtenues par cette méthode sur diverses séquences d'images. La méthode de mise en correspondance 3D/2D sert de référence en termes de qualité de l'estimation du mouvement. Les différentes options concernant la sélection et la mise à jour des patchs sont évalués.

Chapitre 4

Comparaisons et analyse des performances

Dans ce chapitre, nous présentons les résultats obtenus par la méthode décrite précédemment sur des séquences d'images synthétiques. En premier lieu, nous comparons notre approche (pour les différentes méthodes de choix du patch) à celle de Lucas-Kanade et à la mise en correspondance 3D/2D. Ensuite, nous appliquons notre technique à des paires d'images successives, dans des situations où le mouvement est élémentaire mais d'amplitude croissante afin de déterminer les marges de bon fonctionnement de notre algorithme. Enfin, nous présentons quelques résultats d'analyse de séquences synthétiques contenant plusieurs dizaines d'images afin de vérifier la stabilité de notre méthode.

Afin d'étudier la convergence de notre algorithme, nous analysons l'évolution de l'erreur globale. Cette erreur est calculée entre l'image $\mathbf{I}_{n+1}(\mathbf{x})$ et le rendu $\mathbf{R}(\mathbf{x}, \mathbf{p}_{n,\infty} + \Delta\mathbf{p}_{n,\lambda\cdot 50})$ obtenu à l'itération $50 \cdot \lambda$, où λ est une valeur entière lorsque le choix des patchs correspond à la méthode détaillée à la section 3.4.4. L'erreur est calculée par l'expression :

$$\begin{aligned} E_{global}(\mathbf{p}_{n,\infty} + \Delta\mathbf{p}_{n,\lambda\cdot 50}) &= (\sum_{\mathbf{x}\in\Omega} \boldsymbol{\alpha}_{n+1}(\mathbf{x}, \mathbf{p}_{n+1,k}))^{-1} \sum_{\mathbf{x}\in\Omega} \boldsymbol{\alpha}_{n+1}(\mathbf{x}, \mathbf{p}_{n+1,k}) \\ &\quad \big[(\mathcal{R}^r(\mathbf{x}, \mathbf{p}_{n,\infty} + \Delta\mathbf{p}_{n,\lambda\cdot 50}) - \mathcal{I}^r_{n+1}(\mathbf{x}))^2 \\ &+ (\mathcal{R}^v(\mathbf{x}, \mathbf{p}_{n,\infty} + \Delta\mathbf{p}_{n,\lambda\cdot 50}) - \mathcal{I}^v_{n+1}(\mathbf{x}))^2 \\ &+ (\mathcal{R}^b(\mathbf{x}, \mathbf{p}_{n,\infty} + \Delta\mathbf{p}_{n,\lambda\cdot 50}) - \mathcal{I}^b_{n+1}(\mathbf{x}))^2\big] \quad , \end{aligned}$$

dans laquelle le voisinage Ω est l'ensemble des pixels pour lesquels le coefficient d'opacité est non nul dans l'image de rendu : $\Omega = \{\mathbf{x} \mid \boldsymbol{\alpha}_{n+1}(\mathbf{x}, \mathbf{p}_{n+1,k}) \neq 0\}$.

4.1 Comparaisons des performances

4.1.1 Comparaison avec les méthodes 2D/2D et 3D/2D

Dans cette section, nous présentons quelques exemples montrant les apports de notre méthode par rapport aux méthodes de Lucas / Kanade et de mise en correspondance 3D/2D. Pour cela, nous commençons par analyser le comportement de ces algorithmes en considérant uniquement deux images successives. Ces deux images visualisent dans un premier temps un mouvement élémentaire, à savoir une rotation pure de la sphère autour d'un de ses axes.

Ensuite, nous considérons un mouvement plus difficile à discriminer car combinant deux mouvements élémentaires, à savoir une rotation et une translation simultanées. Nous décrivons le comportement en terme de précision et de temps de calcul de chacun de ces algorithmes face à ces situations particulières.

Enfin, nous considérons une séquence de plusieurs images visualisant une translation uniforme (de faible amplitude) de la sphère, afin de mettre en évidence le fait que notre algorithme présente une erreur non-cumulative contrairement à celui de Lucas-Kanade.

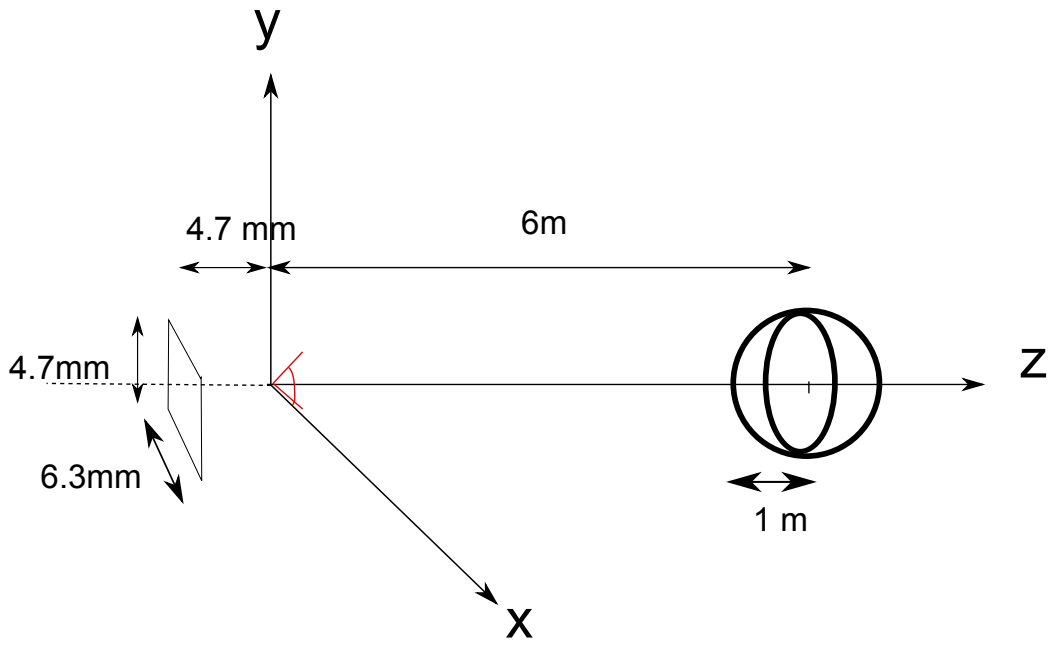


FIGURE 4.1 : Configuration de notre scène.

Avant de présenter les comparaisons, rappelons le contenu de la scène analysée

(fig. 4.1). Les images sont acquises par une caméra de distance focale $f = 4.7 \cdot 10^{-3}$ m. Les dimensions de son capteur sont $6.3 \cdot 10^{-3} \times 4.7 \cdot 10^{-3}$ m^2. Son point focal est supposé à l'origine du repère fixe considéré. La position de la sphère dans la première image est $(-2.5\text{m} \quad 0 \quad 6\text{m})^T$. Une seule source d'éclairage positionnée à l'infini est considérée. Nous pouvons voir sur la figure 4.2 l'image initiale que nous garderons pour toutes les séquences, sauf en cas d'indication contraire.

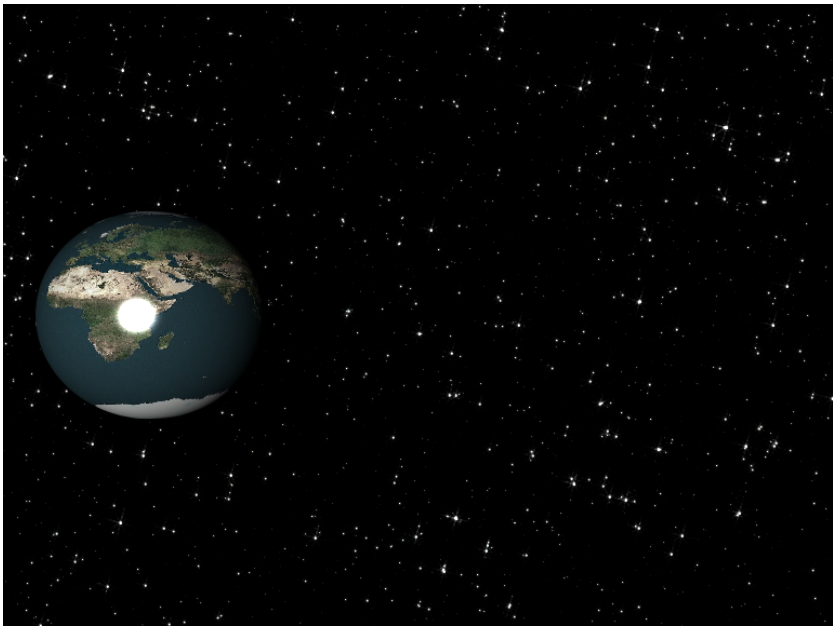


FIGURE 4.2 : Image initiale de toutes les séquences.

La position et l'orientation de la sphère dans la première image sont supposées connues. Pour l'estimation des paramètres de descente de gradient dans le cas de cette étude, le lecteur peut se référer à A.5 pour le réglage des paramètres d'estimation de dérivées et à A.6 pour le réglage des pas de descente de gradient.

4.1.1.1 Deux images, rotation élémentaire

Dans cette partie, nous considérons un mouvement simple de rotation pure autour de l'axe horizontal, soit $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0 \;\; 0.05\text{rd} \;\; 0 \;\; 0)^T$. Ce mouvement correspond à un mouvement apparent de translation au centre de la projection de la sphère d'environ 5 pixels entre deux images successives. Les deux images sont analysées par 4 déclinaisons des méthodes décrites auparavant :

- la méthode 2D/2D, avec des patchs sélectionnés manuellement ;

- la méthode 2D/2D, avec des patchs sélectionnés automatiquement ;
- notre méthode hybride, avec une sélection des patchs ciblant l'erreur ;
- la méthode 3D/2D.

Pour la première méthode, nous avons choisi les patchs manuellement en évitant les zones de reflet. Nous avons considéré 4 patchs de surface totale égale à 60×60 pixels, lesquels sont répartis sur l'intégralité de la projection de la sphère sur l'image. Sur la figure 4.3, nous représentons ces patchs entourés d'un rectangle blanc. En utilisant ces patchs, nous avons estimé le mouvement à l'aide de l'algorithme de Lucas-Kanade en considérant $E_{2D/2D} < seuil_1$ comme critère d'arrêt des itérations. $seuil_1$ a été réglé de façon empirique à la valeur 900 qui permet d'obtenir une convergence satisfaisante de l'algorithme de descente du gradient pour la méthode 2D/2D.

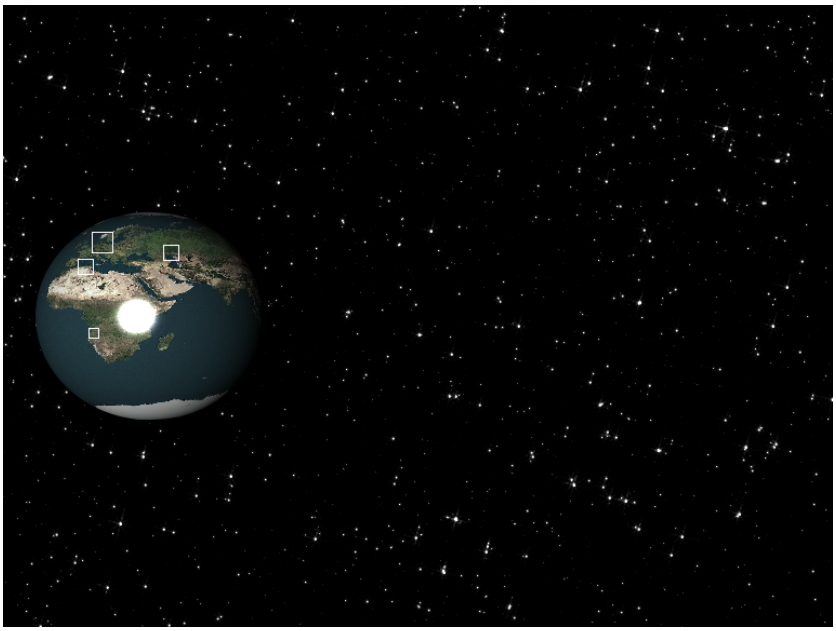


FIGURE 4.3 : Patchs choisis manuellement.

Pour la seconde déclinaison, nous avons appliqué l'algorithme de Lucas-Kanade en considérant une méthode aléatoire automatique pour le choix des patchs (décrite au paragraphe 3.4.2). Nous considérons ici 4 patchs de surface totale égale à 60×60, de surface minimale 10×10, de taille et de position aléatoires, et situés à l'intérieur du disque correspondant à la projection de la sphère sur l'image. Dans ce cas, les patchs peuvent contenir partiellement le reflet provenant de la composante spéculaire. Le critère d'arrêt considéré dans ce cas reste $E_{2D/2D} < seuil_1$.

Comme troisième méthode pour la comparaison, nous utilisons notre approche hybride avec la technique automatique de choix des patchs ciblant l'erreur (présentée à la section 3.4.4). Les patchs ont les mêmes caractéristiques que précédemment, à savoir 4 patchs de surface totale 60×60 et de surface minimale 10×10. Le critère de convergence considéré est ici $E_{global} < seuil_2$. $seuil_2$ a également été réglé de façon empirique à la valeur 400 qui permet une bonne convergence de l'algorithme de descente du gradient.

Enfin, comme quatrième méthode pour la comparaison, nous avons appliqué aux mêmes images la technique de mise en correspondance 3D/2D en supposant que la scène est modélisée avec précision. Les itérations de minimisation de l'erreur s'arrêtent lorsque $E_{3D/2D}$ devient inférieur à $seuil_3$, réglé à la valeur $3.5 \cdot 10^{-3}$ de façon empirique afin d'obtenir une bonne convergence. La valeur beaucoup plus faible de ce troisième seuil par rapport aux deux précédents s'explique par le fait que l'erreur $E_{3D/2D}$ est normalisée en la divisant par la surface apparente de la sphère dans l'image.

Les résultats de la comparaison sont présentés dans le tableau 4.1.

	2D/2D[1] $\times 10^{-2}$	2D/2D[2] $\times 10^{-2}$	Notre approche $\times 10^{-2}$	3D/2D $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	0	0.2	0	0	0
ΔT_y m	0.7	0.3	0.3	0	0
ΔT_z m	0.1	0.1	0	0	0
$\Delta\theta_x$ rd	4.2	-0.1	4.5	4.7	5
$\Delta\theta_y$ rd	0	0	0	-0.1	0
$\Delta\theta_z$ rd	-0.3	-3.7	-0.3	-0.1	0

TABLE 4.1 : Paramètres du mouvement obtenus avec une méthode de mise en correspondance 2D/2D où le choix des patchs est manuel[1] ou aléatoire[2], avec notre approche et un choix des patchs suivant la méthode présentée à la section 3.4.4 et enfin avec une mise en correspondance 3D/2D. La précision est de 10^{-3}.

Nous pouvons remarquer dans le tableau 4.1 que lorsque le choix des patchs est aléatoire dans la méthode 2D/2D (deuxième colonne), les erreurs sont relativement élevées. En effet, le mouvement de rotation de 0.05 rd autour de l'axe horizontal de la sphère est confondu avec un mouvement de rotation par rapport à l'axe de profondeur de -3.7×10^{-2} rd. Même en considérant un faible mouvement correspondant à un seul pixel (section 3.3), la mise en correspondance 2D/2D échoue à

estimer le mouvement avec précision lorsque le choix des patchs est aléatoire. Ceci est dû à la présence partielle du reflet sur les patchs analysés.

En effet, lorsque nous choisissons les patchs manuellement de sorte à éviter les reflets, la méthode de mise en correspondance 2D/2D donne de meilleurs résultats (*cf.* colonne 1 du tableau 4.1). Le mouvement est alors estimé avec une erreur de 0.008 rd sur $\Delta\theta_x$ et de 0.007 m sur ΔT_y.

Cependant, les résultats fournis par notre méthode sur les mêmes images restent plus précis (*cf.* colonne 3 du tableau 4.1 : erreurs de 0.05 rd sur l'angle de rotation autour de l'axe horizontal et de 0.003 m sur la translation suivant l'axe vertical) grâce à l'information procurée par la composante spéculaire. Avec notre approche hybride, la convergence est atteinte en moins de 50 itérations sans avoir recours à la mise à jour du modèle, étape gourmande en temps de calcul.

Nous pouvons remarquer que la mise en correspondance 3D/2D est la méthode qui donne les résultats les plus précis lorsque la scène est modélisée sans erreurs (*cf.* colonne 4 du tableau 4.1). Il faut toutefois noter que dans ce cas les temps de calcul requis sont environ 7 fois plus élevés que pour notre approche et 10 fois plus élevés que ceux requis par la mise en correspondance 2D/2D.

4.1.1.2 Deux images, combinaison translation / rotation

Dans cette partie, nous considérons un mouvement plus difficile à identifier, du fait qu'il combine deux mouvements élémentaires (rotation et translation) correspondant à un vecteur paramètre $\Delta\mathbf{p} = (0 \quad -0.02\text{m} \quad 0 \quad -0.02\text{rd} \quad 0 \quad 0)^T$. Le mouvement de translation de -0.02 m suivant l'axe vertical entraîne un mouvement apparent vers le haut de toute la sphère d'environ 2 pixels dans l'image. Le mouvement de rotation de 0.02 rd autour de l'axe horizontal de la sphère correspond à un mouvement apparent de son centre d'environ 2 pixels vers le bas. Il est donc difficile de distinguer ces deux mouvements sur la base d'informations 2D.

Nous procédons comme décrit précédemment afin de comparer les estimations de ce mouvement par trois méthodes. En effet, la méthode de sélection automatique des patchs pour l'approche 2D/2D ne converge plus sur cette paire d'images et ne peut donc pas être retenue dans la comparaison. Nous obtenons les résultats

suivants (tableau 4.2) avec une précision de 10^{-3} :

	2D/2D patchs fixés manuellement $\times 10^{-2}$	Notre approche $\times 10^{-2}$	3D/2D $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	0.1	0	0	0
ΔT_y m	-0.1	-1.6	-1.7	-2
ΔT_z m	0.1	-0.1	0	0
$\Delta\theta_x$ rd	-3.1	-2.6	-2.4	-2
$\Delta\theta_y$ rd	0	-0.1	0	0
$\Delta\theta_z$ rd	-0.3	0	0.1	0

TABLE 4.2 : Paramètres du mouvement obtenus avec une méthode de mise en correspondance 2D/2D où le choix des patchs est manuel[1] ou aléatoire[2], avec notre approche et un choix des patchs suivant la méthode présentée à la section 3.4.4 et enfin avec une mise en correspondance 3D/2D. La précision est de 10^{-3}.

Comme on pouvait s'y attendre, la méthode de mise en correspondance 2D/2D appliquée sur des patchs choisis manuellement ne parvient pas à estimer correctement les deux composantes du mouvement. En effet, les erreurs sur le mouvement estimé sont très importantes (*cf.* colonne 1 du tableau 4.2). La translation suivant l'axe vertical est estimée avec une erreur de 0.019 m et la rotation autour de l'axe horizontal avec une erreur de -0.011 rd.

Avec notre approche, quatre mises à jour du modèle ont été requises avant convergence. Cependant, les résultats fournis sont beaucoup plus précis que ceux obtenus avec l'approche 2D/2D (*cf.* colonne 2 du tableau 4.2 : erreurs de 0.006 rd sur l'angle de rotation autour de l'axe horizontal et de 0.004 m par rapport à la translation suivant l'axe vertical).

Encore une fois, la mise en correspondance 3D/2D est la méthode qui donne les résultats les plus précis lorsque la scène est modélisée sans erreur. Cette fois, les temps de calcul requis sont environ 5 fois plus élevés que pour notre approche et 10 fois plus élevés que ceux exigés par la mise en correspondance 2D/2D.

4.1.1.3 Analyse des erreurs sur une séquence d'images

Dans cette partie, nous comparons la méthode de mise en correspondance 2D/2D avec notre méthode lorsque le but est d'estimer le mouvement 3D d'une sphère sur plusieurs images successives d'une séquence. Pour cela, nous avons considéré le cas de petits déplacements entre deux images successives correspondant à des

translations selon l'axe vertical de vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0.01\mathbf{m} \;\; 0 \;\; 0 \;\; 0 \;\; 0)^T$. Le mouvement apparent de la sphère relatif à ce mouvement 3D est une translation d'environ 1 pixel vers le bas de toute l'image de la sphère.

Pour la mise en correspondance 2D/2D appliquée avec un choix de patch comme décrit dans la section 3.4.2, nous remarquons que la convergence (après $40\ 000$ itérations) n'est plus atteinte à partir de la sixième image analysée. Afin de visualiser ces résultats, nous avons reconstruit les 5 premières images avec les vecteurs mouvement 3D obtenus par la méthode 2D/2D, lesquelles sont présentées sur la figure 4.4.

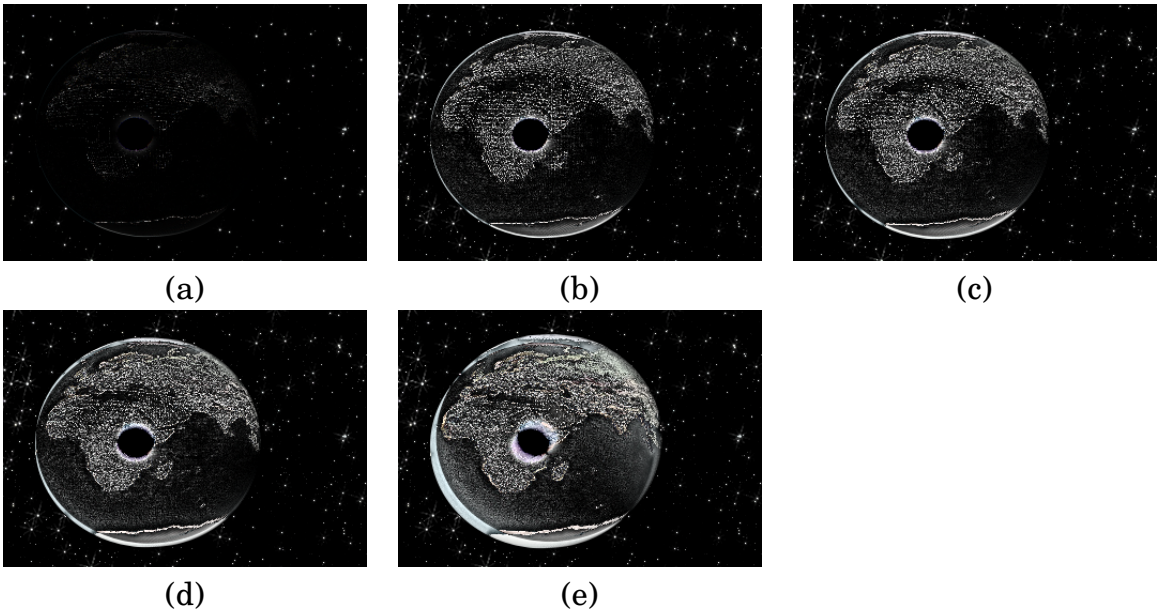


FIGURE 4.4 : Image différence entre les images réelles et les images reconstruites avec les vecteurs mouvement 3D déduits grâce à la mise en correspondance 2D/2D. Nous remarquons que l'erreur est cumulative.

Sur la figure 4.4, nous pouvons constater que l'erreur de modélisation augmente au cours du déroulement de la séquence d'images. Sur la figure 4.5, nous représentons l'évolution de cette erreur E_{global} pour ces 5 premières images de la séquence. On constate aisément que vu l'absence de correction globale apportée lors du calcul du rendu du modèle 3D pour chaque nouvelle image, l'erreur est cumulative.

Par contre, pour notre approche appliquée avec un choix de patchs ciblant l'erreur (décrite dans la section 3.4.4), la convergence est atteinte pour toutes les images de la séquence. Nous remarquons que lorsque l'algorithme converge lors

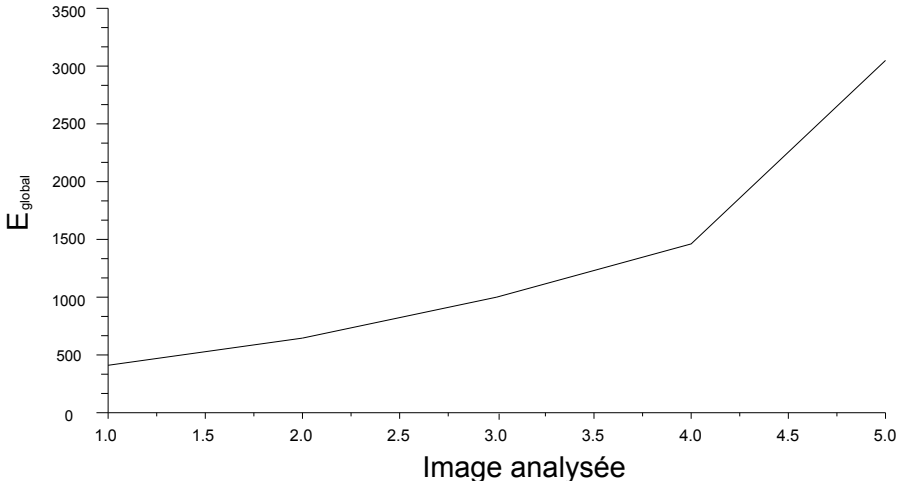


FIGURE 4.5 : Erreur globale vs. n ($0 \leq n \leq 5$)

de l'analyse de l'image à l'instant n, il converge également pour l'image à l'instant $n+1$, à condition que le mouvement soit de faible amplitude (cela sera précisé dans la section 4.2).

Comme notre approche hybride ne diverge pas sur cette séquence, nous pouvons présenter les résultats d'analyse obtenus sur un nombre plus important d'images de cette séquence. Sur la figure 4.6, nous avons représenté les images montrant la différence entre l'image réelle et l'image reconstruite (grâce aux vecteurs mouvement 3D estimés avec notre approche) pour $n = 2, 10, 20, 30, 40$ et 50. Les images différence restent sombres lorsque n augmente, ce qui indique une bonne adéquation du modèle et de l'image réelle. Dans la section 4.3, nous présenterons les résultats obtenus sur quelques exemples de séquences longues.

Il est important de noter que le critère d'arrêt pour la mise en correspondance 2D/2D, à savoir $E_{2D/2D}(\Delta p) < seuil_1$, ne prend pas en compte d'une façon directe la position et l'orientation calculées après qu'elles soient appliquées au modèle 3D. En revanche, dans notre approche, le critère d'arrêt $E_{global}(\Delta p) < seuil_2$ tient compte expressément de la position et de l'orientation estimées pour chaque nouvelle image.

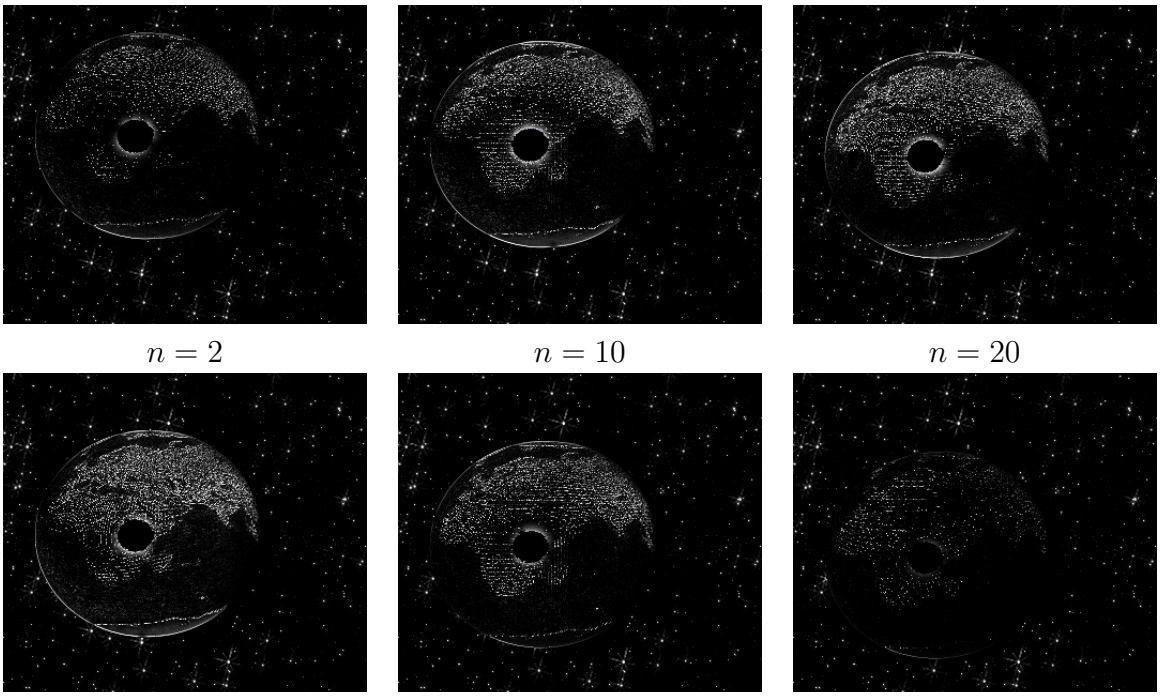


FIGURE 4.6 : Image différence entre les images réelles et les images reconstruites avec les vecteurs mouvement 3D déduits grâce à notre approche lorsque $n = 2$, 10, 20, 30, 40 et 50.

4.1.1.4 Conclusion sur les comparaisons réalisées

En premier lieu, nous avons mis en évidence les apports de notre approche hybride par rapport à la mise en correspondance 2D/2D. Dans le cas d'une surface réfléchissante, la méthode de mise en correspondance 2D/2D ne peut donner des résultats précis que si les patchs évitent les reflets. Cette condition ne peut pas être garantie lorsque l'analyse porte sur une longue séquence et que le choix des patchs est automatique. Il faudrait pour ce faire disposer d'une information *a priori* concernant la source lumineuse, ce qui conviendrait à modifier profondément la méthode 2D/2D.

En revanche, notre approche donne de meilleurs résultats lorsque les patchs choisis d'une façon aléatoire intègrent une partie du reflet. La méthode hybride parvient à résoudre correctement le problème de l'estimation des composantes d'un mouvement ambigu combinant une translation et une rotation 3D dont l'effet s'annule dans certaines parties de l'image. La correction apportée lors de la mise à jour du modèle 3D nous permet d'aboutir à des résultats qui restent précis tout au long

de la séquence d'images.

Mis à part l'estimation efficace des mouvements complexes, la prise en compte de la mise à jour du modèle 3D est une particularité de notre méthode qui permet d'éliminer l'effet d'accumulation des erreurs. En effet, même en considérant un mouvement simple de quelques pixels entre deux images successives, nous avons remarqué que lorsque l'analyse est menée sur une séquence longue, la méthode de mise en correspondance 2D/2D souffre d'une accumulation des erreurs, ce qui n'est pas le cas avec notre approche hybride.

Dans tous les cas étudiés, la mise en correspondance 3D/2D donne de meilleurs résultats que notre approche et que la mise en correspondance 2D/2D lorsque la scène est modélisée avec précision. Cependant, il faut souligner le fait que le temps de calcul est beaucoup plus important avec cette méthode. Sur les exemples traités, nous avons constaté que la mise en correspondance 2D/2D est 10 fois plus rapide. Avec notre approche, le temps de calcul est globalement 5 à 6 fois plus faible, étant entendu qu'il augmente parfois en fonction du nombre de mises à jour du modèle 3D requises avant convergence.

4.1.2 Comparaison par rapport à la taille et au choix des patchs

Afin d'analyser l'efficacité des différentes méthodes de choix des patchs énumérées dans le chapitre 3, nous entreprenons l'analyse suivante. Nous considérons une séquence de 50 images où le mouvement de la sphère d'une image à l'autre est constant et correspond à $\Delta\mathbf{p} = (0.01\mathrm{m} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0)^T$. Ensuite, nous essayons d'estimer ce mouvement en utilisant successivement chacune des méthodes de choix du patch. Dans ce cas, nous avons considéré le même critère d'arrêt pour chacune de ces déclinaisons de notre méthode hybride, lequel consiste à réaliser un nombre constant d'itérations égal à 200. Dans les cas où nous mettons à jour les patchs initialement sélectionnés, cette mise à jour est réalisée toutes les 50 itérations. Enfin, nous comparons les erreurs globales finales relatives à chacune des images analysées pour chacune de ces méthodes.

La figure 4.7 montre que lorsque nous considérons un seul patch de position relative fixe par rapport à la projection de la sphère, l'erreur globale augmente d'une

image à l'autre. Après un certain nombre d'images, cela entraîne une divergence de l'algorithme. D'autre part, cette erreur est souvent la plus élevée par rapport aux autres méthodes de choix de patchs.

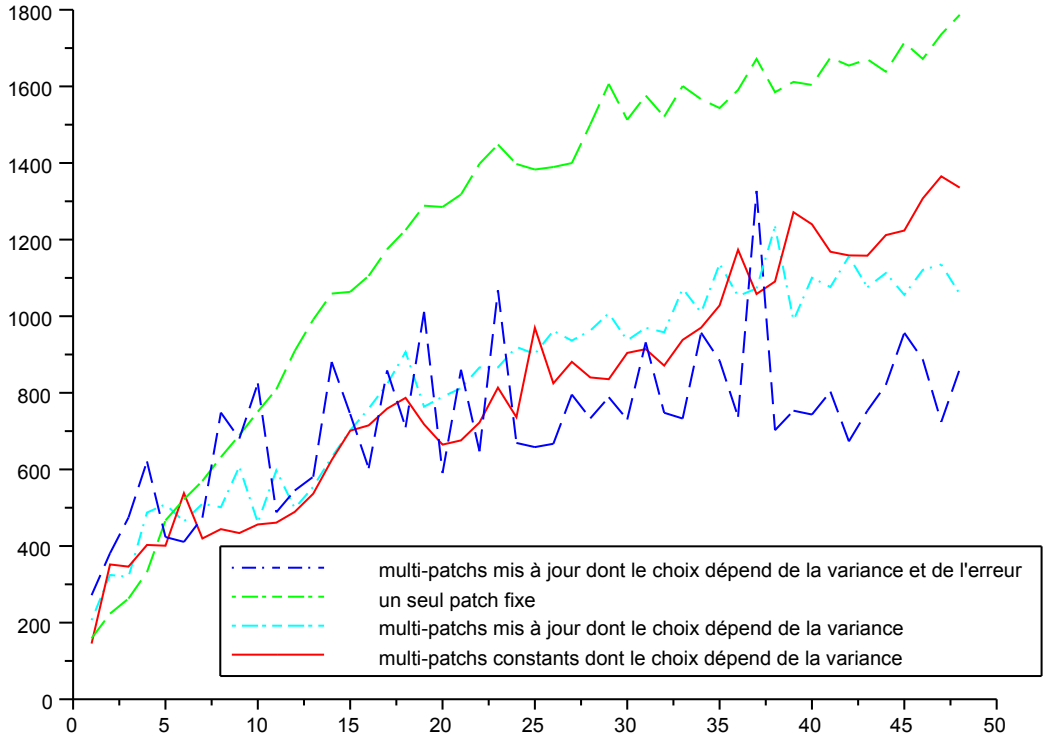


FIGURE 4.7 : Erreur globale finale par rapport au numéro de l'image analysée de la séquence.

Ensuite, vient la méthode de choix de quatre patchs aléatoires. Ce choix est soumis à la condition suivante : la variance de la texture du patch de l'image analysée doit être supérieure à un seuil fixé *a priori*. Pour cette méthode de choix des patchs, ceux-ci ne sont pas mis à jour lors de l'analyse d'une image (les mêmes patchs sont considérés pendant les 200 itérations). Nous pouvons voir que l'erreur globale lors de l'analyse de l'image 50 est inférieure que celle obtenue lorsque nous considérons un seul patch même si ce patch, centré, est censé contenir les deux informations spéculaire et diffuse. Nous expliquons cela par le fait que les patchs définis de façon aléatoire occupent des positions variées entre le centre de la projection de la sphère et ses bords, procurant de ce fait une information plus éparse et riche à l'algorithme de minimisation de l'erreur.

Ensuite, nous avons essayé de mettre à jour les 4 patchs toutes les 50 itérations. Nous remarquons à nouveau une amélioration sur l'erreur globale lors de

l'analyse de la dernière image de la séquence. Ceci prouve l'intérêt d'une mise à jour périodique des patchs, ce qui encore une fois tend à diversifier l'information mise à disposition de l'algorithme de minimisation de l'erreur, améliorant ainsi sa convergence.

Enfin, la dernière méthode consiste à mettre à jour le modèle 3D toutes les 50 itérations et à calculer les différentes images de rendu qui nous permettront d'ajouter une condition au choix des patchs. Cette condition consiste à choisir les patchs où l'erreur entre l'image calculée et l'image analysée est supérieure à un certain seuil. Nous remarquons dans ce cas que l'erreur globale n'augmente plus progressivement au cours de la séquence, mais qu'elle tend à se stabiliser vers la valeur ($E_{global} = 800$). Lorsque le nombre d'images analysées augmente, cette dernière méthode de sélection des patchs donne des erreurs globales généralement inférieures à celles obtenues avec les autres variantes.

Suite à cette analyse, nous pouvons conclure que notre algorithme fonctionne mieux avec plusieurs patchs dispersés qu'avec un seul patch, même si les surfaces sont équivalentes. Un autre apport de la technique de choix de patch ciblant l'erreur réside dans le fait qu'elle permet la mise à jour du modèle 3D entre chaque image. Cela apporte des améliorations notables par rapport aux autres techniques de sélection des patchs notamment en terme de précision de l'estimation du mouvement. Contrairement aux autres déclinaisons, avec la mise à jour du modèle 3D et la mise à jour de l'image de rendu, l'erreur n'est plus cumulative et l'algorithme continue à converger même sur des séquences d'image de longue durée.

4.1.3 Conclusion concernant les comparaisons

Sur cette série de comparaisons des performances respectives des différentes méthodes 2D/2D et 3D/2D et des déclinaisons possibles de notre approche hybride, nous avons constaté que :

- le calcul séparé des deux rendus spéculaire et diffus nous permet de calculer leurs transformations séparément sachant que leurs mouvements apparents ne sont pas identiques. Ceci nous permet de mieux discerner certains mouvements (voir section 3.3) ;

- la mise à jour du modèle 3D nous permet d'obtenir des résultats précis et d'éviter l'erreur cumulative, ce qui n'est pas possible avec la méthode de mise en correspondance 2D/2D ;
- le choix aléatoire du patch nous permet de récupérer une information riche et éparse permettant une meilleure analyse du mouvement ;
- grâce à l'intervention du modèle 3D dans notre approche, nous évitons une étape usuelle de l'analyse du mouvement, à savoir celle qui consiste à segmenter le fond de la scène dans l'image.

4.2 Plages de mouvements analysables

Dans cette section, nous analysons la plage de fonctionnement de notre approche vis à vis de la nature du mouvement et de son amplitude. Pour cela, nous analysons séparément chaque mouvement simple (c.a.d. lorsqu'un seul des paramètres du mouvement change) en considérant des amplitudes de plus en plus importantes pour chacun de ces mouvements. Le but est de déterminer pour quelle amplitude de chaque mouvement élémentaire notre technique ne converge plus.

Pour cette analyse, nous considérons 4 patchs de surface totale égale à 60×60, de surface minimale 10×10, de taille et de position aléatoires sur la projection de la sphère sur l'image. Le choix aléatoire de ces patchs est décrit dans la section 3.4.4. Une mise à jour du modèle 3D est calculée toutes les 50 itérations, donc à chaque changement de valeur de λ. Nous appliquons comme critère d'arrêt $(E_{global} < 400)$ sachant qu'au bout de $\lambda = 20$ notre approche requiert un temps de calcul aussi important que la mise en correspondance 3D/2D lorsque les pas de descente de gradient sont optimaux.

4.2.1 Mouvements de translation

4.2.1.1 Translation selon l'axe horizontal ou vertical

Les mouvements de translation selon les axes horizontal ou vertical dans la scène correspondent respectivement à des mouvements apparents de translation horizontale ou verticale de la projection de la sphère dans l'image. Le comporte-

ment de notre méthode étant similaire dans les deux cas, du fait que ces deux directions sont traitées de façon identique dans les équations de projection 3D/2D, nous ne décrivons ici que les simulations permettant d'évaluer les performances de notre méthode dans le cas d'une translation horizontale.

Une translation suivant l'axe horizontal correspond à un vecteur de paramètres de la forme $\Delta\mathbf{p} = (a \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^T$, dans lequel $|a|$ désigne l'amplitude du mouvement. Dans notre cas, lorsque $a = 0.01$ m le mouvement de translation apparent correspond à environ 1 pixel. Pour ce cas, la convergence ($E_{global} < 400$) est atteinte lorsque $\lambda = 1$ sans prise en compte de la mise à jour du modèle (*cf.* figure 4.8).

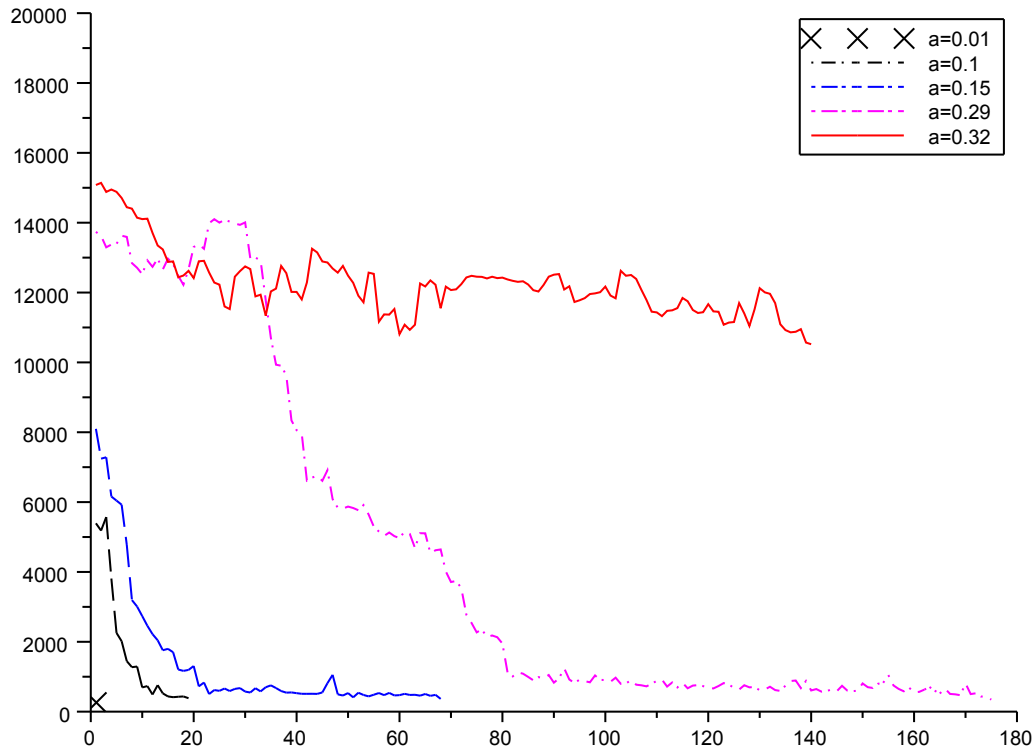


FIGURE 4.8 : Énergie globale par rapport à λ pour différentes valeurs de translation par rapport à l'axe horizontal (x).

Nous remarquons que lorsque l'amplitude du mouvement augmente, la mise à jour du modèle est opérée puisque $\lambda > 1$, et donc qu'on réalise plus de 50 itérations. Plus l'amplitude du mouvement augmente, plus le nombre d'itérations nécessaires à la convergence est élevé. Ainsi, pour $a = 0.1$ m (resp. $a = 0.15$ m, $a = 0.29$ m) correspondant à une translation apparente d'environ 10 pixels (resp. 15 pixels, 29 pixels), la convergence est atteinte au bout de $\lambda = 19$ (resp. $\lambda = 68$, $\lambda = 175$).

Nous pouvons en déduire qu'entre deux mises à jour du modèle, la mise en correspondance joue le rôle de prédicteur, la méthode hybride devenant similaire à une estimation du mouvement de type "coarse to fine".

Pour la courbe correspondant à un mouvement de translation d'amplitude $a = 0.32$ m, au bout de $\lambda = 140$ la convergence n'est toujours pas atteinte et l'erreur globale reste supérieure à $10\ 000$. On peut donc considérer que cette valeur constitue la limite de l'amplitude du mouvement de translation pouvant être correctement estimé. Jusqu'à un mouvement correspondant à $a = 0.29$ m, le mouvement peut toujours être estimé même si le temps de calcul devient très long. La convergence est atteinte dans ce cas au bout de $\lambda = 175$ et notre approche est alors plus coûteuse en temps de calcul que la mise en correspondance 3D/2D.

Nous pouvons voir sur la figure 4.9 l'image de comparaison (image différence) entre l'image calculée et l'image réelle pour différentes étapes de convergence lors de l'analyse du mouvement $\Delta\mathbf{p} = (0.29\text{m} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^T$. On remarque en premier lieu que cette amplitude du mouvement relativement élevée implique que les projections de la sphère sont situées à des positions éloignées dans les deux images, comme l'indiquent les premières images de l'erreur.

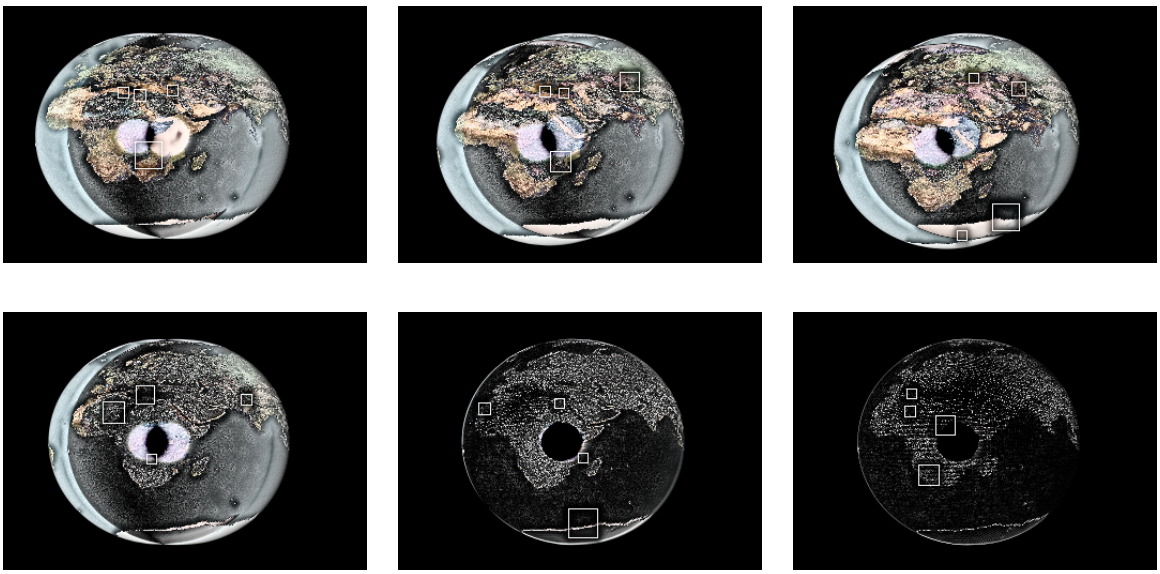


FIGURE 4.9 : Images de comparaison entre l'image calculée pour λ respectivement égal à 1, 18, 28, 51, 81, 175 et l'image réelle. Les patchs utilisés pour la comparaison sont encadrés en blanc.

Nous remarquons que l'image différence converge petit à petit vers une image

presque noire représentative d'une erreur globale minimale. Cependant, jusqu'à $\lambda = 30$ l'allure de l'erreur globale n'est pas décroissante (*cf.* figure 4.8). Ceci peut-être expliqué par le fait que pour les patchs tirés au hasard durant les premières itérations, l'erreur $E^{totale}_{hybride}$ n'est pas quadratique aux alentours des valeurs initiales des paramètres du mouvement. On constate que malgré cela, notre algorithme parvient à converger. Cela indique que l'aspect stochastique introduit dans l'algorithme par la mise à jour périodique des patchs contribue à rendre plus efficace la procédure d'optimisation.

Tant que la convergence est atteinte avec $\lambda <= 20$, notre approche reste plus rapide que la mise en correspondance 3D/2D. Ceci est le cas pour les mouvements de translation dont l'amplitude $|a|$ est inférieure à 0.1 m entre deux images. Pour $a = 0.1$ m le mouvement apparent correspondant est d'environ 10 pixels. Dans ces cas, nous obtenons les estimations présentées dans le tableau 4.3, avec une précision de 10^{-3}. L'amplitude du mouvement de translation par rapport à l'axe horizontal, égale à 0.1 m en réalité, est donc estimée à 0.086 m. L'erreur d'estimation est ici équivalente à environ 1 pixel.

	Notre approche $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	8.6	10
ΔT_y m	-0.3	0
ΔT_z m	1.2	0
$\Delta\theta_x$ rd	0.4	0
$\Delta\theta_y$ rd	-0.9	0
$\Delta\theta_z$ rd	0.2	0

TABLE 4.3 : Estimations des paramètres du mouvement et paramètres réels, mouvement de translation horizontal

4.2.1.2 Translation selon l'axe de profondeur

Quelques problèmes sont inhérents à la vision monoculaire, notamment celui de l'estimation de la distance séparant les objets de la caméra. Estimer le déplacement d'un objet le long de l'axe de profondeur est un problème aussi complexe que celui de l'estimation de sa distance.

Afin de tester le comportement de notre algorithme lorsque le mouvement ana-

lysé est une translation suivant l'axe de profondeur, nous avons considéré deux images successives sur lesquelles la sphère suit un mouvement de translation le long de cet axe, mouvement décrit par un vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; b \;\; 0 \;\; 0 \;\; 0)^T$. b désigne ici l'amplitude de la translation selon l'axe de profondeur. Nous commençons par considérer un mouvement de faible amplitude avec $b = 0.01$ m. Ceci correspond à une évolution de la surface apparente de la sphère d'un facteur 0.998 (environ -0.2 pixel sur le diamètre). Sur la figure 4.10, nous remarquons que la convergence a lieu pour $\lambda = 1$. Ensuite, nous augmentons progressivement l'amplitude de ce mouvement jusqu'à ce que notre algorithme ne puisse plus converger.

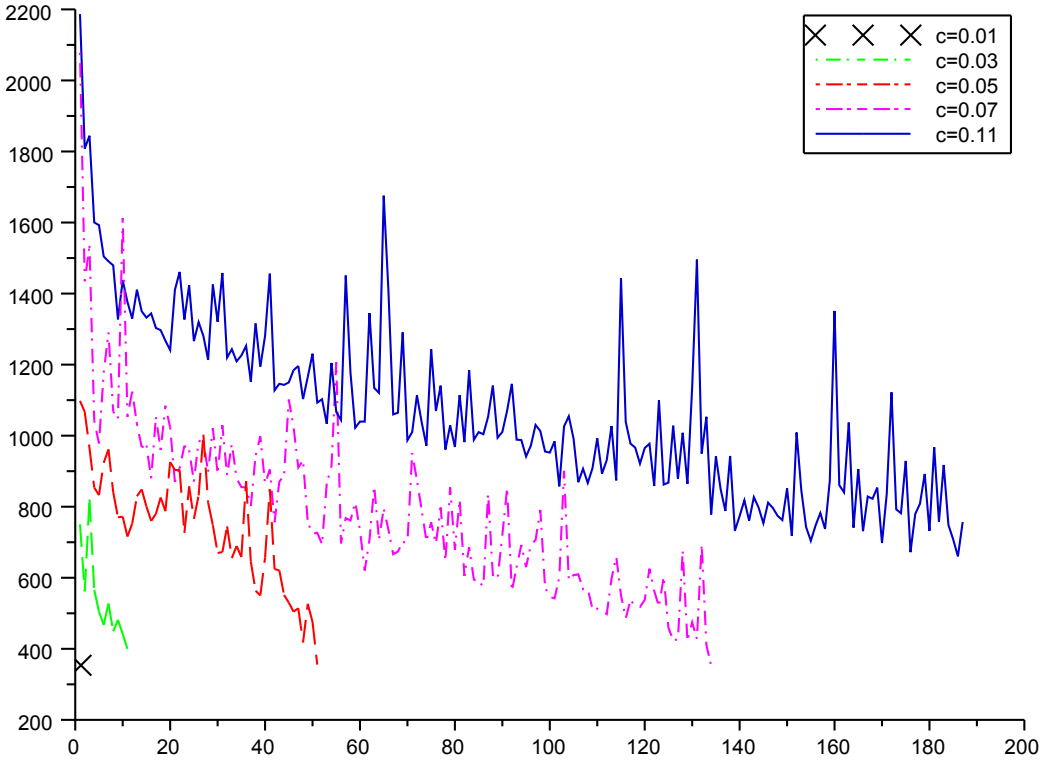


FIGURE 4.10 : Erreur globale par rapport à λ pour différentes valeurs de translation par rapport à l'axe de profondeur.

Nous remarquons que notre algorithme est beaucoup moins efficace quant à l'estimation d'une translation selon l'axe de profondeur que pour les autres translations. En effet, même pour une amplitude limitée de $b = 0.03$ m, $\lambda = 11$ lors de la convergence (pour $b = 0.05$ m $\lambda = 51$ et pour $b = 0.07$ m $\lambda = 134$).

La figure 4.10 montre également que la convergence de $E_{globale}$ est moins régulière que pour les autres translations. Le nombre d'itérations nécessaires à l'esti-

mation est également bien plus important. Lorsque l'amplitude b du mouvement augmente, son estimation est de moins en moins précise et nécessite de plus en plus d'itérations. Lorsque $b = 0.11$ m, ce qui correspond à une réduction d'environ 2 pixels du diamètre apparent de la sphère, notre algorithme ne parvient toujours pas à estimer le mouvement au bout de 190×50 itérations. Ceci prouve la difficulté qu'il y a à reconstituer le mouvement de profondeur à partir d'images monoculaires.

Notre méthode nécessite moins de calculs que la méthode 3D/2D jusqu'à une amplitude du mouvement de translation égale à $b = 0.03$ m. Cela correspond à une réduction de 1 pixel du diamètre apparent de la sphère. Dans ce cas, nous obtenons les résultats d'estimation présentés dans le tableau 4.4 avec une précision de 10^{-3}. On constate que dans ce cas, le mouvement de translation par rapport à l'axe de profondeur, d'amplitude réelle $b = 0.03$ m, est estimé à 0.018 m (erreur de 1.2 cm).

	Notre approche $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	0.5	0
ΔT_y m	0	0
ΔT_z m	1.8	3
$\Delta\theta_x$ rd	-0.1	0
$\Delta\theta_y$ rd	0.4	0
$\Delta\theta_z$ rd	0	0

TABLE 4.4 : Estimations des paramètres du mouvement et paramètres réels, mouvement de translation selon l'axe de profondeur

Comme précédemment, nous présentons sur la figure 4.11 les images de comparaison (images différence) entre l'image calculée et l'image réelle correspondant à $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0.07\text{m} \;\; 0 \;\; 0 \;\; 0)^T$ pour différentes étapes de convergence correspondant aux différentes valeurs de λ.

Nous remarquons que l'algorithme aboutit rapidement à une première estimation du mouvement. En revanche, la présence de minima locaux dans la fonction d'erreur entraîne des oscillations dans la méthode de descente du gradient, rendant la convergence plus lente et imprécise.

4.2.2 Mouvement de rotations

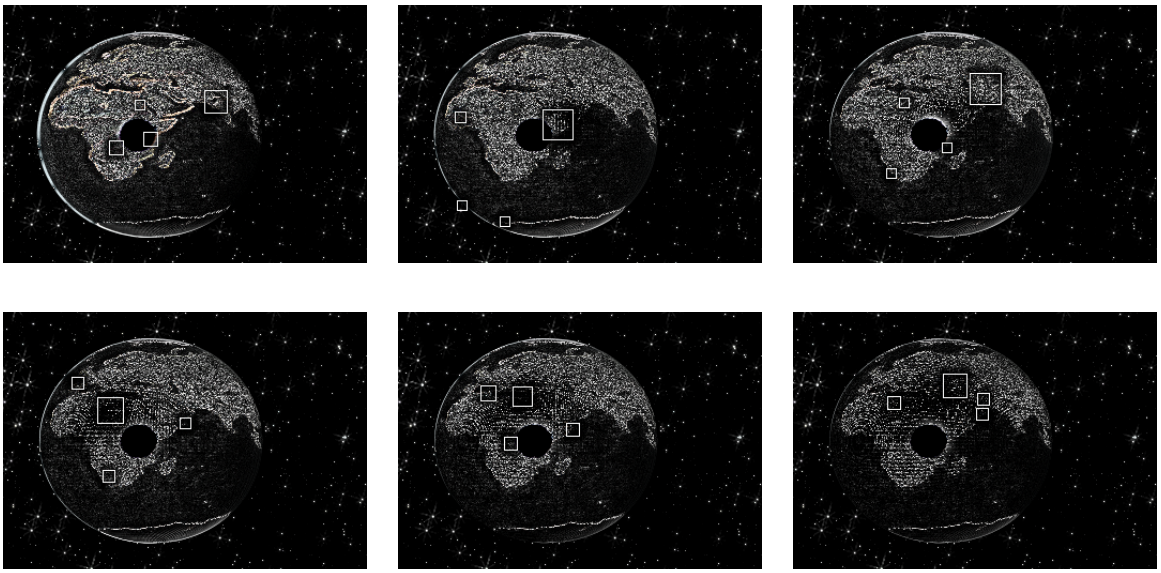


FIGURE 4.11 : Images de comparaison entre l'image calculée pour λ respectivement égal à 1, 19, 43, 66, 92, 134 et l'image réelle. Les patchs utilisés sur ces images pour estimer l'erreur sont encadrés en blanc.

4.2.2.1 Rotation autour de l'axe horizontal ou vertical

Comme pour les translations, les rotations pures autour des axes horizontal et vertical ont des propriétés similaires, du fait que ces derniers interviennent de façon équivalente dans les transformations 3D/2D. De ce fait, nous ne présentons ici que les évaluations des performances de notre méthode hybride pour l'estimation d'une rotation autour de l'axe horizontal.

Une rotation autour de l'axe horizontal correspond à un mouvement de vecteur paramètre $\Delta\mathbf{p} = (0 \quad 0 \quad 0 \quad c \quad 0 \quad 0)^T$ dans lequel c désigne l'angle de rotation. Les conditions d'expérimentation pour la rotation élémentaire sont exactement similaires à celles utilisées pour la translation.

Sur la figure 4.12, on constate que la convergence est atteinte en moins de 50 itérations ($\lambda = 1$) lorsque l'angle de rotation est de $c = 0.01$ rd. De ce fait, le modèle n'est pas mis à jour durant les itérations de minimisation de l'erreur. Au centre de la projection de la sphère, cette rotation 3D apparaît comme une translation d'environ 1 pixel dans la direction verticale.

Une rotation correspondant à $c = 0.04$ rd (resp. $c = 0.19$ rd, $c = 0.27$ rd) nécessite $\lambda = 2$ mises à jour du modèle (resp. $\lambda = 14$, $\lambda = 23$) avant convergence. λ est de plus en plus grand lorsque c augmente. Nous remarquons que pour une rotation d'angle

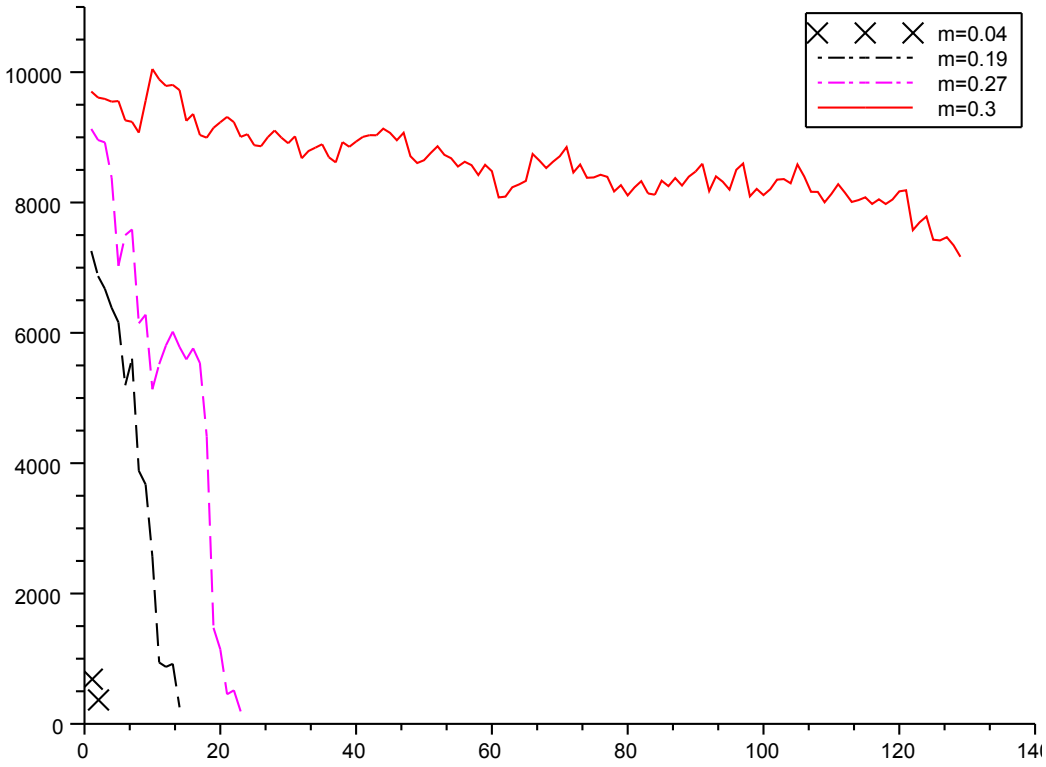


FIGURE 4.12 : Erreur globale par rapport à λ pour différentes valeurs de rotation autour de l'axe horizontal.

c supérieur à 0.27 rd, notre algorithme n'a toujours pas convergé pour $\lambda = 120$ (l'erreur E_{global} reste supérieure à $8\ 000$).

Sur la figure 4.13 nous présentons les images de comparaison (images différence) entre l'image calculée et l'image réelle correspondant à une rotation, de vecteur paramètre $\Delta\mathbf{p} = (0 \quad 0 \quad 0 \quad 0.27\text{rd} \quad 0 \quad 0)^T$, pour des étapes de convergence correspondant à différentes valeurs de λ. Bien que la rotation soit importante, nous remarquons que la convergence est atteinte relativement rapidement. L'estimation obtenue lorsque $\lambda = 19$ atteint brusquement les valeurs recherchées. Ceci est probablement dû en partie à des dispositions particulièrement informatives des patchs à cette étape.

Notre algorithme est capable d'estimer une rotation de grande amplitude, correspondant à un angle $c = 0.21$ rd, tout en restant plus rapide que la mise en correspondance 3D/2D et avec la même précision. Ce mouvement correspond à un mouvement apparent de translation d'environ 21 pixels au niveau du centre de la projection de la sphère. Dans ce cas, nous obtenons les résultats présentés dans le

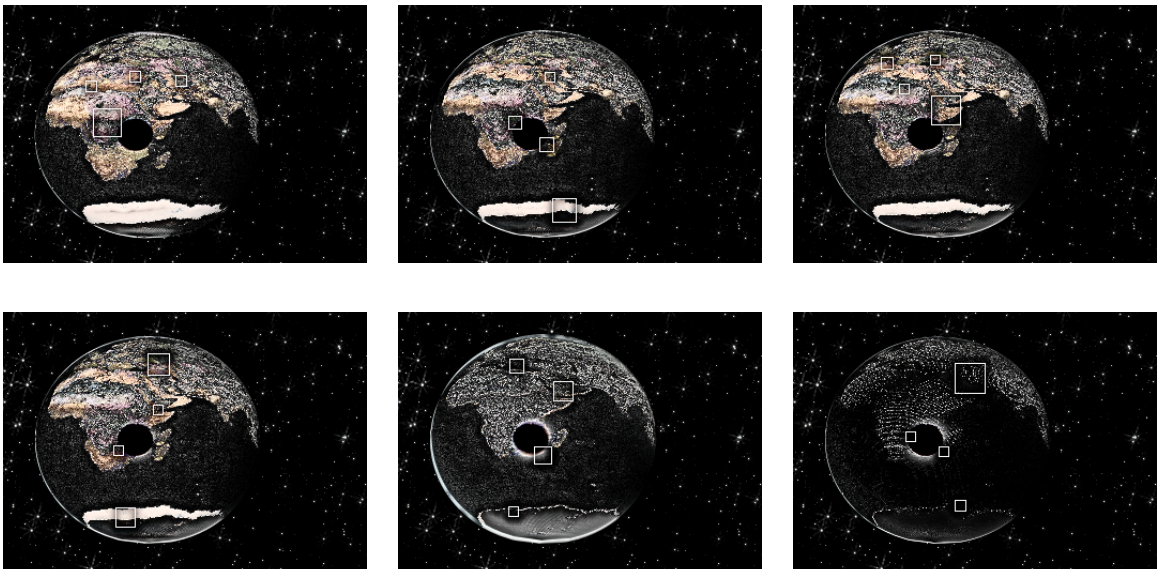


FIGURE 4.13 : Images de comparaison entre l'image calculée pour λ respectivement égale à 1, 9, 13, 17, 19, 23 et l'image réelle. Les patchs utilisés pour la comparaison sont encadrés en blanc.

tableau 4.5 (avec une précision de 10^{-3}). On constate que ce mouvement de rotation est estimé avec précision. En effet, le mouvement de rotation autour de l'axe horizontal de la sphère, d'angle réel 0.21 rd, est estimé à 0.206 rd, soit une erreur d'estimation inférieure à 1 pixel pour le centre de la projection de la sphère. Les erreurs par rapport à tous les autres paramètres sont également très faibles (tous inférieurs à 1 pixel en terme de mouvement apparent).

	Notre approche $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	0.4	0
ΔT_y m	0.3	0
ΔT_z m	-0.3	0
$\Delta\theta_x$ rd	20.6	21
$\Delta\theta_y$ rd	0.2	0
$\Delta\theta_z$ rd	-0.1	0

TABLE 4.5 : Estimations des paramètres du mouvement et paramètres réels, mouvement de rotation autour de l'axe horizontal

4.2.2.2 Rotation autour de l'axe de profondeur

Une rotation autour de l'axe de profondeur correspond à un vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; d)^T$, dans lequel d désigne l'angle de rotation. Afin d'évaluer les

performances et les limites de la méthode, nous utilisons à nouveau les conditions expérimentales décrites auparavant. La figure 4.14 présente les résultats de nos simulations pour ce cas particulier de mouvement élémentaire.

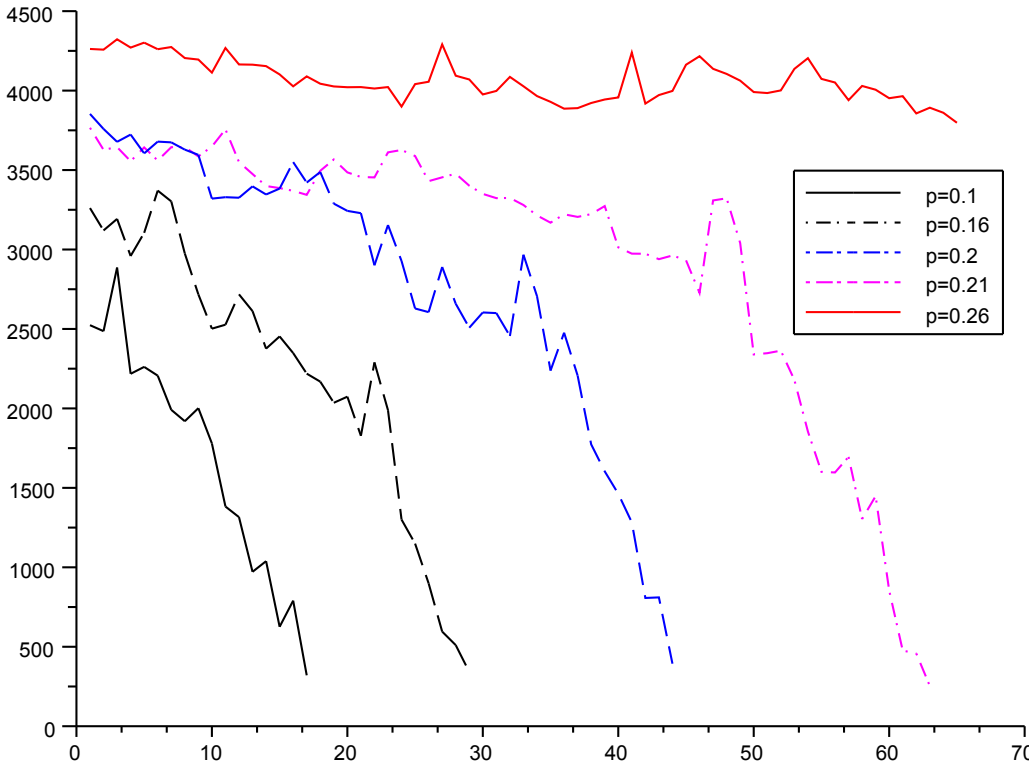


FIGURE 4.14 : Erreur globale par rapport à λ pour différentes rotations autour de l'axe de profondeur.

On constate que lorsque $d = 0.01$ rd, la convergence ($E_{global} < 400$) est atteinte avant 50 itérations, donc que la mise à jour du modèle n'est pas opérée durant la minimisation de l'erreur. Ce mouvement correspond environ à un déplacement de 1 pixel sur le pourtour de la projection de la sphère dans l'image (translation horizontale de 1 pixel sur les bords haut et bas de la sphère, et translation verticale de 1 pixel sur les bords gauche et droit).

Le nombre d'itérations requises pour aboutir à un minimum de l'erreur augmente avec l'angle de rotation. Sur la figure 4.14, on vérifie que pour $d = 0.21$ rd (resp. $d = 0.1$ rd, $d = 0.16$ rd, $d = 0.2$ rd), l'algorithme converge pour $\lambda = 63$ (resp. $\lambda = 17$, $\lambda = 29$, $\lambda = 44$). Nous remarquons que pour une rotation d'angle $d = 0.26$ rd, l'algorithme ne converge toujours pas pour $\lambda = 95$ (l'erreur E_{global} reste supérieure à $10\,000$, *cf.* figure 4.14).

La figure 4.15 présente les images correspondant à la différence entre l'image réelle et celles calculées pour différentes valeurs de λ. Pour ces images le mouvement est défini par le vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0.21\text{rd})^T$. Sur chaque image, les patchs utilisés pour évaluer l'erreur sont délimités par un rectangle blanc.

FIGURE 4.15 : Images de comparaison entre l'image calculée pour λ respectivement égal à 1, 16, 26, 48, 51, 63 et l'image réelle. Les patchs utilisés pour la comparaison sont encadrés en blanc.

On constate qu'au début du processus la convergence est lente. Les images de différence correspondant à $\lambda = 16$ et $\lambda = 26$ se ressemblent fortement. A nouveau, on peut vérifier l'intérêt de faire intervenir un tirage au sort des patchs dans la procédure de minimisation, l'exploration de l'espace des solutions étant amélioré par l'introduction de ce comportement stochastique.

Notre approche reste plus rapide que la mise en correspondance 3D/2D jusqu'à $d = 0.1$ rd. Avec cette valeur de l'angle de rotation, les résultats d'estimation sont présentés dans le tableau 4.6, avec une précision de 10^{-3}. On vérifie que le mouvement de rotation a été estimé avec précision, la valeur de 0.96 rd étant très proche de la valeur réelle 0.1 rd. Les erreurs d'estimation de tous les paramètres sont faibles, leur effet dans l'image restant inférieur à 1 pixel.

	Notre approche $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	0.1	0
ΔT_y m	0	0
ΔT_z m	0	0
$\Delta\theta_x$ rd	0	0
$\Delta\theta_y$ rd	0.2	0
$\Delta\theta_z$ rd	9.6	10

TABLE 4.6 : Estimations des paramètres du mouvement et paramètres réels, mouvement de rotation autour de l'axe de profondeur

4.2.3 Rotation et translation simultanées

Afin de tester le comportement limite de notre algorithme en présence d'un mouvement simultané de rotation autour de l'axe horizontal (resp. vertical) et d'une translation suivant l'axe vertical (resp. horizontal), nous avons considéré deux images successives sur lesquelles la sphère se déplace selon un vecteur paramètre $\Delta\mathbf{p} = (a \;\; 0 \;\; 0 \;\; 0 \;\; -a \;\; 0)^T$ (resp. $\Delta\mathbf{p} = (0 \;\; a \;\; 0 \;\; -a \;\; 0 \;\; 0)^T$). Ce mouvement particulier a été sélectionné car le mouvement apparent qui lui correspond dans l'image est nul au centre de la projection de la sphère.

Nous commençons par considérer un mouvement combiné de petite amplitude avec $a = 0.01$ (en mètres pour la translation et en radians pour la rotation). Ensuite, nous augmentons l'amplitude de mouvement combiné jusqu'à ce que notre algorithme ne converge plus. Notre algorithme réussit à estimer ce mouvement même dans les cas de mouvements de grande amplitude. Comme nous le voyons sur la figure 4.16, notre méthode parvient sans problème à estimer un mouvement combiné pour $a = 0.25$. Cela prouve que l'approche hybride parvient à lever l'ambiguïté relative à certains mouvements, pourtant inhérente à la vision monoculaire.

Notre approche reste plus rapide que la mise en correspondance 3D/2D jusqu'à ce que le mouvement atteigne une amplitude $a = 0.1$. Dans ce cas, nous obtenons les résultats présentés dans le tableau 4.7 avec une précision de 10^{-3}. On constate que le mouvement est estimé avec précision, l'erreur d'estimation étant très faible pour tous les paramètres (l'effet des ces erreurs dans l'image reste toujours inférieur à 1 pixel). Des résultats équivalents sont obtenus pour une translation selon l'axe vertical combinée à une rotation autour de l'axe horizontal.

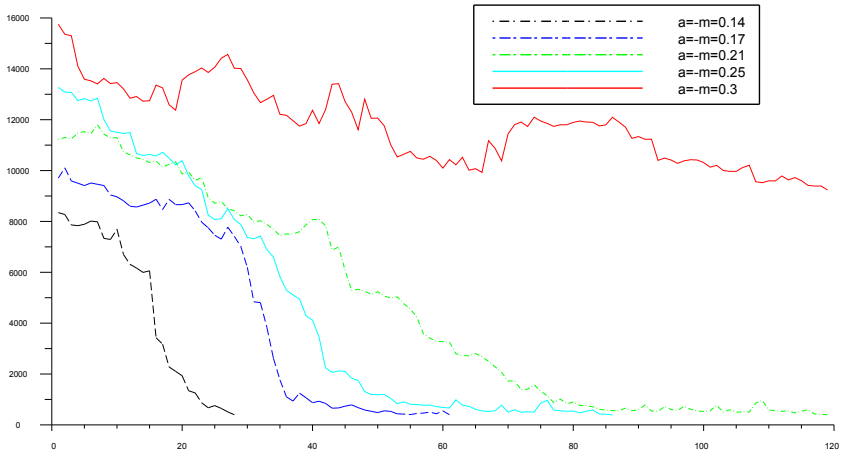


FIGURE 4.16 : Erreur globale par rapport à λ pour des mouvements combinant une translation et une rotation.

	Notre approche $\times 10^{-2}$	Valeurs réelles $\times 10^{-2}$
ΔT_x m	9.3	10
ΔT_y m	0	0
ΔT_z m	1.5	0
$\Delta\theta_x$ rd	0	0
$\Delta\theta_y$ rd	-10	-10
$\Delta\theta_z$ rd	0	0

TABLE 4.7 : Estimations des paramètres du mouvement et paramètres réels, mouvements de rotation et de translation combinés

4.2.4 Conclusion

Sur les exemples présentés dans cette section, nous avons constaté que notre algorithme hybride parvient à estimer des mouvements d'amplitude importante avec une bonne précision. En effet, le choix aléatoire des patchs confère à la méthode d'optimisation une nature stochastique qui lui apporte la capacité d'estimer des mouvements importants en explorant efficacement l'espace des solutions.

Cependant, plus le mouvement est de grande amplitude, plus le nombre d'itérations nécessaire à son estimation est élevé, entraînant de ce fait une augmentation du temps de calcul. Globalement, lorsque λ devient supérieur à 20 ($1\ 000$ itérations de minimisation de l'erreur), notre approche devient aussi gourmande en temps de calcul que la mise en correspondance 3D/2D, laquelle fournit pourtant des résultats plus précis.

Nous avons vérifié que pour la rotation par rapport à l'axe horizontal (resp.

vertical) et pour la translation suivant l'axe vertical (resp. horizontal), deux mouvements 3D caractérisés par des mouvements apparents qui se ressemblent au niveau du centre de la projection de la sphère, nous avons obtenu des convergences assez rapides même lorsque l'amplitude des mouvements augmente. Les mêmes performances sont obtenues lors de l'estimation d'un mouvement de rotation autour de l'axe de profondeur de la scène.

En revanche, nous avons constaté qu'il est difficile d'estimer un mouvement de translation selon cet axe de profondeur à partir des images de la scène acquises dans cette configuration monoculaire. Même avec un modèle précis de la sphère, la perte d'information causée par la projection de la scène 3D sur l'image 2D est telle que notre méthode (comme toutes les autres) ne permet pas de recouvrer précisément les paramètres du mouvement.

4.3 Séquences complètes

Dans cette section, nous présentons quelques résultats obtenus par notre approche hybride sur des séquences longues d'environ 100 images chacune. Nous invitons le lecteur à consulter la page web de l'auteur, dans l'onglet *recherche*, pour une meilleure visualisation de ces résultats sous la forme de fichiers vidéo : http ://lagis-vi.univ-lille1.fr/∼yb/.

Tout d'abord, nous avons essayé d'estimer le mouvement 3D de la sphère dans une séquence visualisant des mouvements élémentaires de faible amplitude entre deux instants successifs d'acquisition. Le mouvement élémentaire considéré est modifié toutes les 15 images :

- images 0 à 14 : translation uniforme selon l'axe vertical de vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0.01\mathbf{m} \;\; 0 \;\; 0 \;\; 0 \;\; 0)^T$;
- images 15 à 29 : rotation pure autour de l'axe horizontal de vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0 \;\; 0.01\mathbf{rd} \;\; 0 \;\; 0)^T$;
- images 30 à 44 : translation uniforme selon l'axe horizontal de vecteur paramètre $\Delta\mathbf{p} = (0.01\mathbf{m} \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0)^T$;
- images 45 à 59 : rotation pure autour de l'axe vertical de vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0 \;\; 0 \;\; 0.01\mathbf{rd} \;\; 0)^T$;

- suite de la séquence : rotation pure autour de l'axe de profondeur de vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0.01\mathrm{rd})^T$.

Nous présentons sur la figure 4.17 des images calculées par une différence pixel à pixel entre le modèle estimé et l'image réelle à différents instants ($n = 2$, 20, 39, 56, 70 et 90).

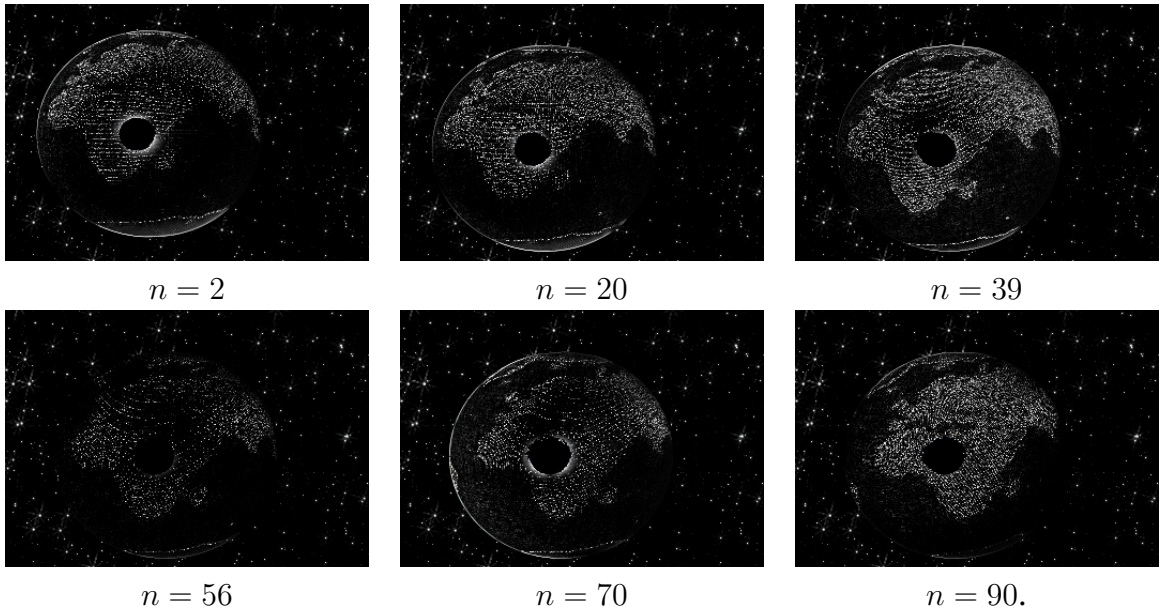
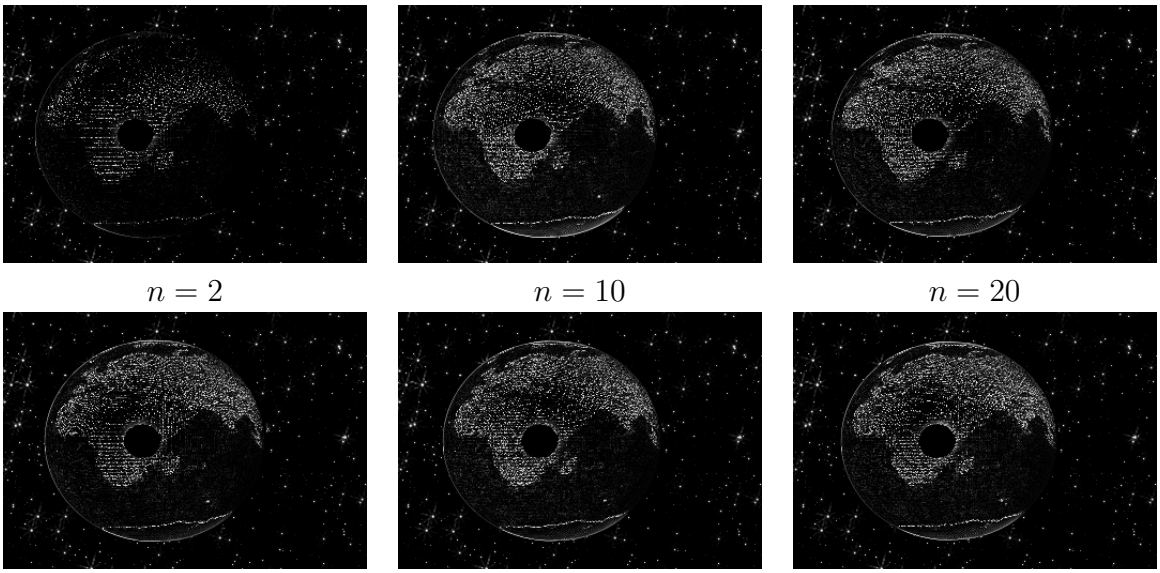


FIGURE 4.17 : Images différence entre les images réelles et les images reconstruites avec les vecteurs mouvement 3D déduits grâce à notre approche lorsque $n = 2$, 20, 39, 56, 70 et 90.

Sur cette figure, on constate que toutes les images différence sont presque noires, ce qui indique que la modélisation est excellente. De ce fait, le mouvement est correctement estimé, même lors des transitions entre des mouvements de translation et de rotation pour lesquels le mouvement apparent au centre de la sphère est similaire.

Dans la séquence suivante, nous avons considéré le cas où la sphère se déplace suivant l'axe de profondeur, avec un mouvement de translation uniforme de vecteur paramètre $\Delta\mathbf{p} = (0 \;\; 0 \;\; 0.005\mathrm{m} \;\; 0 \;\; 0 \;\; 0)^T$. Les résultats, sous la forme d'images différence entre le modèle estimé et l'image réelle est présenté sur la figure 4.18.

On constate que toutes les images différence sont presque noires, ce qui indique une modélisation correcte. On vérifie ainsi que notre algorithme est capable d'estimer ce mouvement de translation uniforme, d'amplitude toutefois limitée, sur



$n = 2$ $n = 10$ $n = 20$

$n = 30$ $n = 40$ $n = 50$.

FIGURE 4.18 : Image différence entre les images réelles et les images reconstruites avec les vecteurs mouvement 3D déduits grâce à notre approche lorsque $n = 2$, 20, 39, 56, 70 et 90.

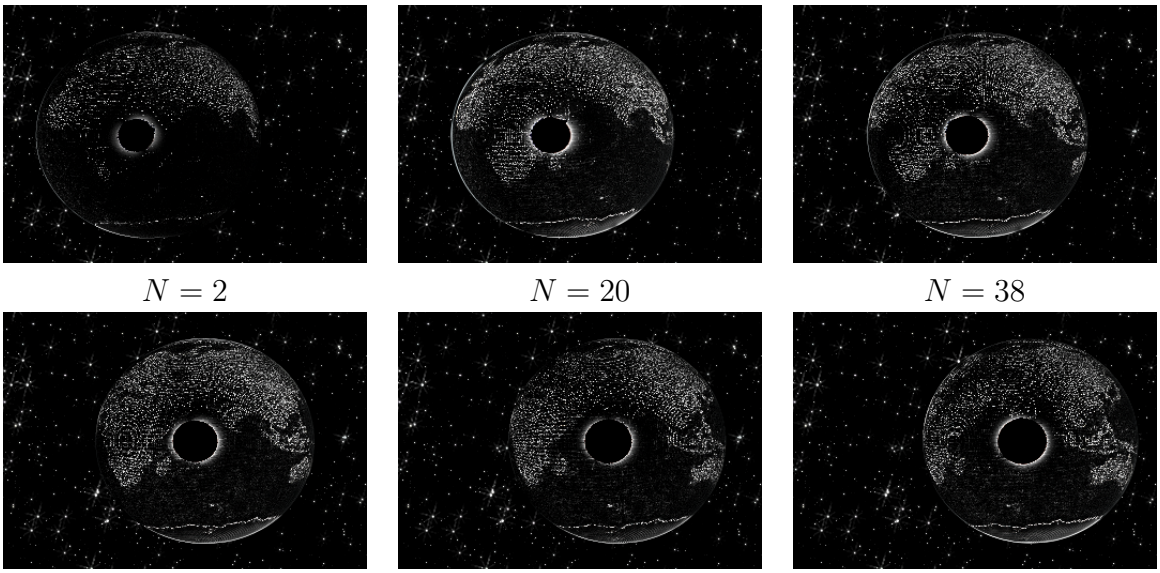
toutes les images de la séquence.

Enfin, nous présentons les résultats obtenus sur une séquence visualisant un mouvement complexe qui combine plusieurs mouvements élémentaires. Pour cela, nous avons considéré une séquence dans laquelle le mouvement de la sphère correspond à celui du globe terrestre, à savoir une rotation pure autour de l'axe vertical, combiné à une translation suivant l'axe horizontal et l'axe de profondeur. Cette dernière combinaison permet d'approcher le mouvement réel de rotation autour d'un point situé à grande distance de la sphère.

Nous pouvons voir sur la figure 4.19 les images montrant la différence entre les images réelles et les images reconstruites à partir des vecteurs mouvement 3D estimés pour $N = 2, 20, 38, 54, 70$ et 87.

Comme nous avons considéré un mouvement faible, la méthode parvient à estimer correctement les paramètres de ce mouvement, pourtant très complexe à étudier sur la base de séquence monoculaires.

Comme nous l'avons précisé auparavant, nous remarquons sur toutes les séquences présentées dans cette section qu'il n'y a pas de divergence de l'algorithme tant que l'amplitude du mouvement entre deux instants successifs reste limitée.



$N = 2$ $N = 20$ $N = 38$

$N = 54$ $N = 70$ $N = 87$.

FIGURE 4.19 : Image différence entre les images réelles et les images reconstruites avec les vecteurs mouvement 3D déduits grâce à notre approche lorsque $n = 2$, 20, 38, 54, 70 et 87.

Plus précisément, tant que les composantes du mouvement 3D réel restent inférieures aux valeurs limites évaluées dans la section 4.2, notre algorithme converge.

4.4 Conclusion

Dans ce chapitre, nous avons validé notre approche d'estimation du mouvement 3D d'une sphère de surface réfléchissante par expérimentation sur différentes séquence d'images. L'apport de cette approche consiste à appliquer deux traitements différents à la texture selon qu'elle provient du phénomène de réflexion ou de diffusion de la lumière. Un autre apport de cette approche réside dans la méthode du choix des patchs sur lesquels est évaluée l'erreur entre l'image réelle et le modèle.

Dans la première partie, nous avons comparé notre approche avec la méthode de mise en correspondance 2D/2D et la méthode de mise en correspondance 3D/2D. Nous avons montré qu'elle est beaucoup plus précise que la méthode de mise en correspondance 2D/2D, surtout dans le cas d'un mouvement complexe. Cette précision résulte du fait que le reflet provenant de la composante spéculaire n'est plus considéré comme du bruit mais au contraire comme source supplémentaire d'information.

D'autre part, nous avons vérifié que la méthode de choix des patchs, consistant à les sélectionner de façon aléatoire tout en ciblant les zones dans lesquelles l'erreur est maximale, est particulièrement efficace. Le tirage aléatoire des positions confère un aspect stochastique à l'algorithme de mise en correspondance, lui permettant de localiser plus efficacement le minimum significatif.

Comme on pouvait s'y attendre, nous avons constaté que la méthode de mise en correspondance 3D/2D aboutit quant à elle à des résultats plus précis que notre approche. Cependant, le temps de calcul nécessaire pour l'estimation avec cette méthode, surtout dans sa phase de calcul du rendu, est plus important que pour notre approche hybride.

Dans la deuxième section, nous avons analysé le fonctionnement de notre approche lorsque l'amplitude des mouvements considérés augmente, ou quand ces derniers deviennent plus complexes. Nous avons vérifié que la convergence a lieu même lorsque le mouvement est d'amplitude importante, correspondant parfois à des déplacements apparents de plus de 20 pixels. Ceci s'explique à nouveau par la nature stochastique de la technique de choix des patchs.

Conclusions et perspectives

Dans cette thèse, nous avons présenté une méthode d'estimation des paramètres 3D du mouvement d'un objet de surface réfléchissante. Cette méthode est l'hybridation d'une technique de mise en correspondance 3D/2D et d'une technique de mise en correspondance 2D/2D. A l'opposée des méthodes classiques, notre méthode traite de façon différente les zones texturées qui proviennent d'un phénomène de réflexion ou de diffusion de la lumière.

Le mémoire est organisé en quatre parties.

Dans le premier chapitre, nous présentons un état de l'art des méthodes d'estimation d'un mouvement 3D dans une séquence d'images. Parmi elles, aucune ne tient compte spécifiquement du cas des objets dont la surface est réfléchissante.

Trois conclusions importantes ont été portées au chapitre :

- l'option couramment retenue est celle de réduire l'influence des reflets en se basant sur la forme plutôt que sur l'aspect ou en effectuant une mise en correspondance 2D/2D à l'aide de patchs (zones des images de surface limitée).
- l'utilisation d'un unique modèle géométrique a montré une certaine efficacité en présence de réflexions spéculaires mais s'est avérée insuffisante pour l'estimation du mouvement 3D tout particulièrement dans le cas d'une sphère réfléchissante (mouvement apparent du contour extérieur de l'objet non informatif).
- la texture permet d'analyser avec succès le mouvement de la composante diffuse présente dans les images mais produit des résultats erronés en présence de régions spéculaires.

Le chapitre deux est consacré à la comparaison du comportement des méthodes d'estimation par mise en correspondance 2D/2D et 3D/2D face au mouvement d'un objet sphérique réfléchissant. La méthode 3D/2D fournit de bons résultats malgré

la présence des reflets mais nécessite le calcul d'une image synthétique plusieurs fois à chaque itération lors de la minimisation de la fonction d'erreur, augmentant de ce fait le temps de calcul de façon très significative. Une mise en correspondance 2D/2D est plus rapide et fournit des résultats précis à condition que le mouvement soit de faible amplitude. Elle est incapable de tenir compte de la composante spéculaire qui perturbe la convergence de l'algorithme même en limitant les régions d'analyse à quelques patchs. Nous avons finalement montré qu'elle souffre d'un phénomène d'accumulation de l'erreur d'estimation et de divergence de l'algorithme, lors de l'analyse d'une longue séquence d'images.

Sur la base des conclusions des deux précédents chapitres nous détaillons, dans le chapitre 3, notre approche hybride 2D/2D et 3D/2D. Dans cette nouvelle approche, la composante spéculaire est utilisée comme source d'information supplémentaire. Notre méthode permet d'éviter l'accumulation de l'erreur grâce à la mise à jour régulière du modèle 3D et à la synthétisation d'une image de référence. Cette mise à jour n'étant pas réalisée à chaque itération, la méthode proposée est plus rapide que la mise en correspondance 3D/2D classique, tout en garantissant une précision élevée. Plusieurs variantes de la méthode ont été décrites en implantant des techniques légèrement différentes de sélection ou de mise à jour du ou des patchs sur lesquels la fonction d'erreur est évaluée.

Dans le dernier chapitre, nous présentons en détail les performances obtenues par les différentes versions de la méthode sur diverses séquences d'images. D'une part, nous calculons une estimation des paramètres 3D plus pécise que celle obtenue avec une méthode de mise en correspondance 2D/2D surtout dans le cas d'un mouvement complexe. Ce résultat prend sa source dans le fait que la composante spéculaire est devenue une source d'informations importante tout particulièrement en présence de mouvements de rotation et de translation ambigüs. D'autre part, un choix aléatoire des patchs en ciblant les zones dans lesquelles l'erreur est maximale, est particulièrement efficace. Le tirage aléatoire des positions confère un aspect stochastique à l'algorithme de mise en correspondance, lui permettant de localiser plus efficacement le minimum significatif. La précision des estimations est plus faible que celle atteinte avec la méthode 3D/2D mais le temps de calcul est

bien plus faible tout particulièrement grâce à un calcul de l'image de synthèse non systématique. En présence de mouvements complexes et de fortes amplitudes, la convergence de la méthode hybride est garantie grâce à la nature stochastique de la technique de choix des patchs.

Les perspectives sont nombreuses.

Notre étude se plaçait dans le contexte du suivi de la position de la tête d'une libellule, nous avons donc limité notre évaluation au déplacement d'un objet sphérique. Toutefois, notre méthode reste applicable à des objets plus complexes qu'ils soient déformables ou non. Dans ce cas, le modèle géométrique 3D est évidemment plus complexe et un modèle dynamique de déformation est nécessaire afin de synthétiser l'image de l'objet la plus ressemblante possible à l'image réelle.

Nous envisageons d'intégrer à notre méthode des prédicteurs basés sur un développement de Taylor. Ceux-ci ont pour avantage de présenter une première approximation des paramètres de mouvement recherché et ainsi limiter les temps de calcul. Par ailleurs, ils offrent une robustesse élevée en présence de brusques variations des paramètres 3D de mouvement. Cette particularité apparaît très souvent lors du vol d'une libellule.

A ce jour, notre méthode n'a pas été évaluée sur des séquences réelles. Dans un premier temps, nous programmons de l'appliquer sur la séquence d'images d'un pendule sphérique réfléchissant en mouvement. Nous poursuivrons par le problème beaucoup plus complexe du suivi de la tête d'une libellule en mouvement puis de tous ses membres. Dans ce dernier cas, l'objet à suivre devient déformable et il est nécessaire de définir un modèle géométrique, un modèle de texture et un modèle cinématique et dynamique de l'insecte suffisamment précis et réalistes.

Annexes

A.1 Les limites de l'optimisation par descente de gradient

La méthode de descente du gradient présente une convergence initiale rapide. En revanche, la présence de minima locaux la fait osciller et rend sa convergence très lente au fur et à mesure que l'on s'approche du minimum. Si la fonction ne présente pas de minima locaux, la méthode du gradient permet d'atteindre à coup sûr la valeur optimale des paramètres.

La méthode de descente du gradient possède plusieurs inconvénients :

1. Rien ne garantit que le minimum trouvé soit un minimum global. Ce point constitue la limite la plus importante de cette méthode d'optimisation. En effet, si l'on considère un cas simple tel celui de la figure A.20, il est possible que l'on atteigne un minimum local au lieu du minimum global. Une solution est que le vecteur initial soit relativement proche du minimum global où la fonction a une forme quadratique d'où l'utilité des prédicteurs.

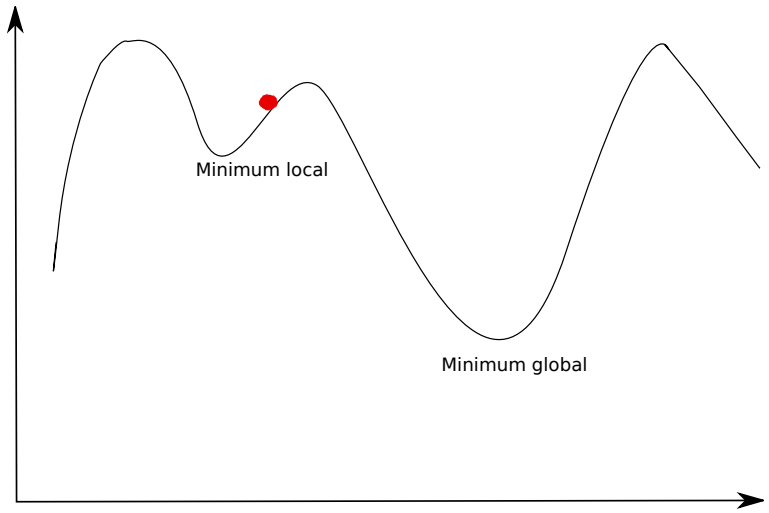


FIGURE A.20 : Convergence vers un minimum local

2. Le choix du pas μ est empirique. Si μ est trop grand, les valeurs de la suite risquent d'osciller autour du minimum sans converger. En revanche, s'il est trop petit, la convergence est trop lente. La figure A.21 montre trois valeurs différentes du pas et leur influence sur la convergence : pour (b) le pas est trop grand, pour (a) il est trop petit, pour (c) il semble adéquat. Enfin, parfois on opte pour un pas variable, c'est à dire que l'on réduit sa valeur au fur et à mesure qu'on s'approche de la solution.

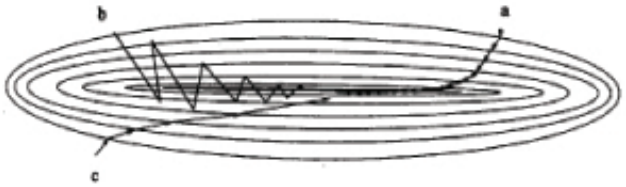


FIGURE A.21 : Influence du pas du gradient

3. Une autre limite de cette méthode consiste dans l'approximation du vecteur gradient. Une bonne estimation des dérivées partielles de E par rapport au 6 paramètres est requise. La méthode conventionnelle d'estimation de la dérivée partielle d'une fonction E par rapport à x est la suivante :

$$\frac{\partial E}{\partial x} = \frac{E(x + h_x) - E(x - h_x)}{2h} \quad , \tag{A.1}$$

où h_x est un paramètre délicat à régler suivant le bruit superposé à E. D'autres méthodes plus robustes au bruit ont été analysées.

A.2 Estimation numérique des dérivées partielles

Lors de nos travaux nous avons eu recours aux estimations numériques de dérivées partielles à deux reprises : une première fois lors de la construction d'un prédicteur (volume 2 de la thèse), une deuxième fois lors de l'estimation des dérivées partielles pour la méthode de descente du gradient. Hors, l'équipe ALIEN (ALgèbre pour l'Identification et l'Estimation Numérique) de l'INRIA Lille-Nord Europe travaille sur de nouvelles techniques de différentiation et d'estimation algébriques.

Ces techniques nouvelles datent de 2004, faisant suite aux travaux de M. Fliess. Elles sont très prometteuses tant en automatique qu'en traitement du signal. En traitement du signal, on peut assurer une estimation rapide en temps réel des dérivées successives d'un signal tout en restant robuste au bruit :

- les perturbations dites structurées, comme les biais constants, sont éliminées en convoluant la fonction par un opérateur différentiel approprié, car elles sont solutions d'équations différentielles ;
- Les bruits à fluctuations rapides, traités d'habitude par des méthodes probabilistes et statistiques, sont atténués par des filtres passe-bas, dont l'intégration est l'exemple le plus simple.

Notre contribution dans ce domaine se manifeste par une extension multidimensionnelle de la dérivée algébrique pour estimer les dérivées partielles successives d'un champ (scalaire ou vectoriel) en présence de bruit. Partons d'un développement en série de Taylor vectoriel tronqué à l'ordre N de la fonction image, donné par l'expression :

$$I(x) = \sum_{|\alpha|<N} \frac{\partial I^{\alpha}(\mathbf{x})}{\partial \mathbf{x}^{\alpha}} \mathbf{x}^{\alpha} \quad , \tag{A.2}$$

dans laquelle x désigne un point de l'image. L'application d'une transformation de Laplace vectorielle donne :

$$\hat{I}(\mathbf{s}) = \int_{\mathbb{R}^n_+} I(\mathbf{x}) \exp(-\mathbf{s}^T \mathbf{x}) \cdot d\mathbf{x} \quad , \tag{A.3}$$

dans laquelle s et x sont des vecteurs. Cette transformation nous permet de passer d'un domaine spatial (ou spatiotemporel, notons que le domaine temporel de la transformation de Laplace classique n'est plus utile) à un domaine opérationnel où des manipulations algébriques adéquates nous permettent de calculer une estimation opérationnelle de la dérivée partielle désirée.

Finalement, par application de la transformation de Laplace inverse vectorielle, nous obtenons une estimation spatiale de la dérivée partielle. Ce travail a été publié dans [B4]. Ces techniques algébriques multidimensionnelles sont prometteuses dans les problèmes réels où l'estimation de dérivées partielles est nécessaire en présence de bruit, en particulier en traitement d'images.

A.3 Warping et interpolation bi-linéaire

La méthode directe de warping (fig. A.22) qui consiste à passer du patch de l'image analysée à son transformé rencontre plusieurs problèmes :

- Les coordonnées résultant de la transformation ne sont pas des entiers.
- Cette transformation n'est pas bijective. Nous pouvons donc nous retrouver avec des pixels dont la luminosité n'a pas été calculée ou dans une situation où un même pixel peut avoir plusieurs luminosités.

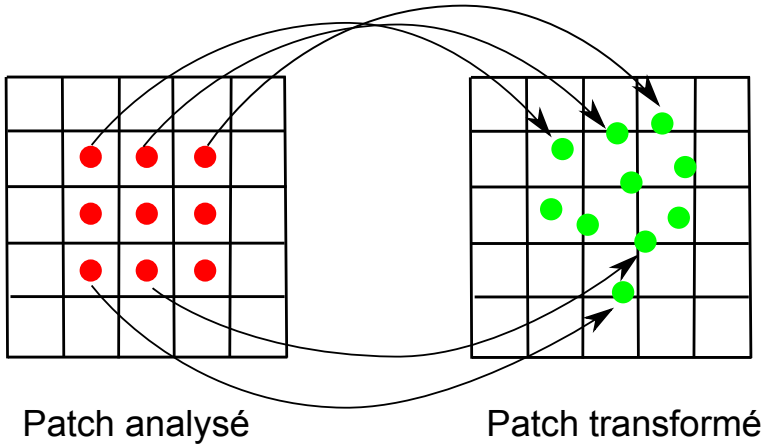


FIGURE A.22 : Transformation directe

Pour résoudre ce problème, nous considérons la transformation inverse (*cf.* figure A.23) :

- Nous considérons dans l'image transformée la région devant recevoir la transformation.
- Pour une transformation $\mathbf{W}(\mathbf{x}, \Delta\mathbf{p}_{n,k})$, nous calculons la transformation inverse $\mathbf{W}^{-1}(\mathbf{x}, \Delta\mathbf{p}_{n,k})$.
- Pour chacun des pixels $(i' \cdot \Delta x, j' \cdot \Delta y)$ de la région qui représente le patch transformé, nous retrouvons grâce à $\mathbf{W}^{-1}(i' \cdot \Delta x, j' \cdot \Delta y, \Delta\mathbf{p}_{n,k})$ et l'équation

$$(i' \cdot \Delta x, j' \cdot \Delta y)^T = \mathbf{W}(i' \cdot \Delta x, j' \cdot \Delta y, \Delta\mathbf{p}_{n,k}) \cdot (x, y)^T \quad , \tag{A.4}$$

  la position $(x\ y)^T$ du point dans l'image initiale analysée.
- Puisque cette position ne correspond pas forcément à un nombre entier, pour retrouver la luminosité au pixel $(i' \cdot \Delta x, j' \cdot \Delta y)$, nous effectuons une interpo-

lation bilinéaire sur les pixels voisins du point $(x\ y)^T$.

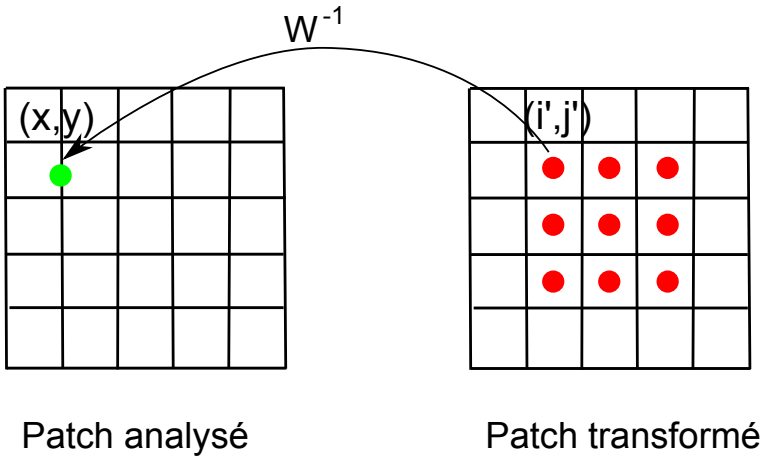


FIGURE A.23 : Transformation inverse

A.4 Initialisation de la position et de l'orientation de la sphère

Après l'étape de modélisation de la scène vient l'étape d'initialisation de la position et de l'orientation de la sphère, lesquelles doivent correspondre au mieux à la projection de la sphère dans la première image de la séquence. Il est possible d'estimer les paramètres de position en exploitant les informations liées au contour de la projection de la sphère dans l'image.

Pour ce faire, il s'agit en premier lieu d'estimer sur l'image le nombre de pixels r correspondant au rayon du disque projeté. A partir de r on peut ensuite estimer la profondeur de la sphère dans la scène, soit le paramètre initial de translation T_z, par l'expression :

$$T_z = f \times \frac{r}{R} \times \frac{N_x}{Taille_x} \tag{A.5}$$

Ensuite, il s'agit d'estimer la position $(x_g \quad y_g)^T$ du centre du disque correspondant à la projection de la sphère. A partir de ces coordonnées, on peut estimer les deux autres paramètres (T_x et T_y) de la translation définissant la position initiale de la sphère dans l'image :

$$\begin{aligned} T_x &= \frac{T_z}{f} \times (x_g - \frac{N_x}{2}) \times \frac{Taille_x}{N_x} \quad , \\ T_y &= \frac{T_z}{f} \times (y_g - \frac{N_y}{2}) \times \frac{Taille_y}{N_y} \quad . \end{aligned} \tag{A.6}$$

En revanche, il n'y a pas de méthode simple permettant d'estimer l'orientation initiale de la sphère. Si uniquement un seul degré de liberté (angle) est inconnu, on peut envisager de calculer l'erreur globale entre l'image initiale et des images de synthèse obtenues pour différentes valeurs de cet angle. Le minimum de l'erreur indique la valeur la plus adéquate de l'angle initial. En revanche, cette méthode n'est pas applicable dans le cas où plusieurs angles sont inconnus, le temps de calcul nécessaire à la recherche exhaustive des meilleurs paramètres devenant prohibitif.

Si l'orientation initiale est connue, mais avec imprécision, on peut considérer une procédure de recherche des valeurs des angles les plus adéquates en calculant l'erreur globale sur un voisinage de l'espace des solutions situé à proximité des valeurs approchées. On peut également tenter d'introduire les valeurs approximatives comme valeurs initiales et d'effectuer des itérations de minimisation de l'énergie comme pour l'estimation d'un mouvement. Si les valeurs approchées sont proches des valeurs réelles, l'algorithme converge et fournit les valeurs recherchées des angles initiaux.

A.5 Réglage des pas de calcul des dérivées

Le problème lié à l'estimation des dérivées numériques est un problème largement étudié dans la littérature. Le lecteur peut trouver plusieurs références et explications dans [B4].

L'une des méthodes les plus simples d'estimation de la dérivée d'une fonction f continue, consiste à partir des développements en série de Taylor à l'ordre 1 de cette dernière calculés pour deux voisins du point considéré :

$$\begin{aligned} f(x+h) &= f(x) + h \cdot f'(x) + o(h^2) \ , \\ f(x-h) &= f(x) - h \cdot f'(x) + o(h^2) \ , \end{aligned} \tag{A.7}$$

puis de soustraire ces deux expressions membre à membre. On obtient ainsi une équation aux différences finies fournissant une estimation de la dérivée de la fonc-

tion en fonction d'un pas de calcul h :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + o(h^2) \quad . \tag{A.8}$$

Graphiquement (*cf.* figure A.24), cela revient à remplacer la tangente à la courbe représentative de f par une sécante passant par deux points proches. La dérivée est estimée par la pente de cette sécante.

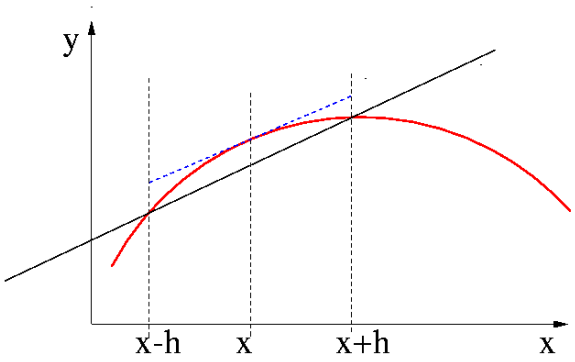


FIGURE A.24 : Dérivée d'une fonction f.

Fixer une valeur de h la plus petite possible pourrait paraître suffisant. Toutefois, cela revient à calculer le rapport de deux expressions dont les valeurs sont faibles, ce qui entraîne des erreurs numériques non négligeables. D'autre part, la présence de bruit, tout particulièrement celui provenant des approximations bilinéaires de la fonction (*cf.* Annexe A.3), conduit à des erreurs d'estimation de la dérivée.

Dans notre cas, étant donné que nous estimons le vecteur gradient de la fonction d'erreur qui dépend de 6 paramètres, nous devons régler le vecteur *pas de calcul* des dérivées directionnelles que nous noterons $\mathbf{h} = (h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6)^T$. Dans cette section, nous analysons le problème du réglage de ce vecteur dans le cas des séquences considérées dans ce manuscrit.

Considérons que les paramètres de position et d'orientation de la sphère perçue sur la première image de la séquence sont connus. A l'aide de cette information, nous allons voir qu'il est possible d'utiliser cette première image ainsi que le modèle de la scène afin de régler les paramètres de descente de gradient.

Ayant le modèle à notre disposition, nous calculons les deux rendus spéculaire

et diffus séparément. Pour chacun des éléments du vecteur h, nous procédons comme suit.

Ces deux images sont warpées avec un vecteur mouvement $\Delta \mathbf{p} = [-0.1 + j \times h_1, 0, 0, 0, 0, 0]^T$ tels que $j = 0 \ldots 2000$ pour différentes valeurs de h_1. Ensuite, elles sont combinées puis comparées à l'image initiale. Pour $h_1 = 0.0001$ (resp. $h_1 = 0.003$), nous obtenons la courbe (fig.A.25) en traits non continus (resp. en ligne continue). Nous pouvons remarquer que vue l'approximation imposée par l'interpolation bilinéaire lors de l'étape de warping, plus h_1 est petit plus la phase de mise en correspondance risque de converger sur des minimums locaux. La dérivée doit être estimée à l'aide d'un pas $h_1 >= 0.003$ afin que le signal soit lisse et la convergence vers un minimum global soit assurée.

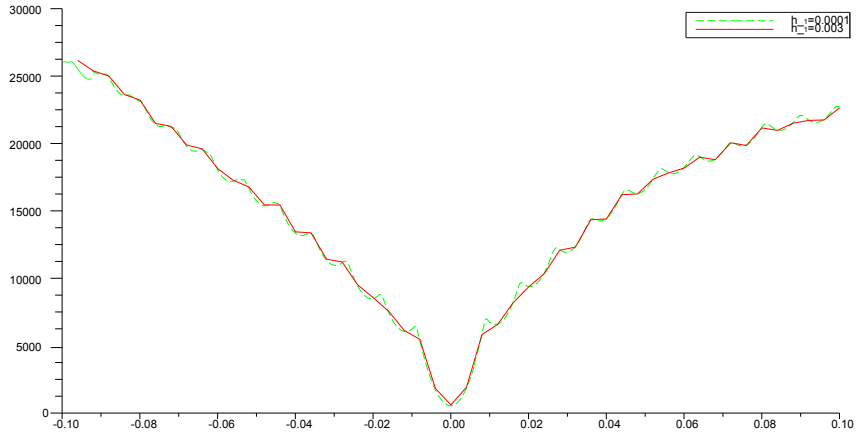


FIGURE A.25 : Allure de $E_{hybride}$ aux alentours du minimum pour un pas de 0.003 et de 0.0001. Les patchs sont les mêmes. Nous pouvons remarquer que, afin d'éviter les minimums locaux et avoir une estimation de la dérivée sans prendre en considération les erreurs produites par l'estimation bilinéaire , il faut que $h_1 >= 0.003$

A.6 Réglage des pas de descente de gradient

Dans cette section, nous examinons le problème de réglage des pas de descente de gradient pour les séquences présentées dans ce manuscrit. Pour les pas de descente de gradient, comme nous l'avons déjà mentionné, nous avons considéré le même pas pour la translation suivant l'axe vertical et la translation suivant l'axe

horizontal. Pareillement, nous avons considéré le même pas de descente de gradient pour la rotation autour de l'axe vertical de la sphère et la rotation autour de son axe horizontal. Nous avons fait ce choix parce que nous supposons que l'allure de la fonction d'erreur à minimiser est identique pour ces paramètres. Les paramètres à régler dans ce cas sont donc au nombre de 4, $\mu = [\mu_1, \mu_1, \mu_2, \mu_3, \mu_3, \mu_4]^T$.

Nous allons analyser le réglage d'un de ces paramètres : le paramètre μ_1. Le réglage des autres paramètres s'effectuera de façon identique.

Nous savons que :

$$\Delta T_x|_{k+1} = \Delta T_x|_k - \mu_1 \times \frac{\partial E^{total}_{hybride}}{\partial \Delta T_x} \tag{A.9}$$

Par soucis de précision, le pas de mise à jour du paramètre ΔT_x ne peut pas être supérieure à 0.01 m (1 pixel) aux alentours du minimum, $|\mu_1 \times \frac{\partial E^{total}_{hybride}}{\partial \Delta T_x}| < 0.01$. Hors, en examinant la figure A.25, la valeur de la dérivée par rapport à cette variable aux alentours du minimum en considérant un pas de calcul de dérivée égal à 0.003 est de $1.1 \cdot 10^6$. Ceci implique :

$$\begin{aligned} \mu_1 &< \frac{0.01}{1.1 \cdot 10^6} \\ \mu_1 &< 9 \cdot 10^{-9} \end{aligned} \tag{A.10}$$

D'autre part, vu le choix des patchs constamment variable le problème des minima locaux est atténué. Par la suite, plus le pas de descente est petit plus les résultats sont précis. Cependant, plus le pas de descente de gradient est petit plus le temps de calcul est long. La limite inférieure de ce paramètre dépend donc de la grandeur du mouvement. Plus le mouvement est grand plus cette limite doit être grande. Pour un mouvement de $0.1\ m$, la convergence est considérée rapide si elle est atteinte avec environ 50 itérations. Le pas correspondant de mise à jour des paramètres du mouvement est donc de $\frac{50}{0.1} = 0.002$. Le pas de descente de gradient relatif au paramètre ΔT_x doit donc être supérieur à :

$$\begin{aligned} \mu_1 &> \frac{0.002}{260000} \\ \mu_1 &> 7.6 \cdot 10^{-9} \end{aligned} \tag{A.11}$$

où 260000 est l'estimation de la dérivée moyenne. Au minimum, pour un mouvement de 0.01 m, le pas de descente correspond à environ :

$$\begin{aligned} \mu_1 &> \frac{0.0002}{260000} \\ \mu_1 &> 7.6^{-10} \end{aligned} \tag{A.12}$$

Jusque là tous les résultats présentés sont obtenus avec $\mu_1 = 2{\cdot}10^{-9}$. Pour le réglage des 3 autres paramètres, il suffit de suivre le même raisonnement. Cependant, nous avons remarqué que le réglage de μ_3 dépend de μ_1. Si μ_3 est relativement important par rapport à μ_1, le mouvement de rotation autour de l'axe horizontal de la sphère (resp. l'axe vertical) est privilégié au mouvement de translation suivant l'axe vertical (resp. horizontal). Nous gardons donc :

$\frac{\mu_1}{\mu_3} = constante.$

Bibliographie

Livres et chapitres de livres

[L1] A. Blake. *Introduction to Active Contours and Visual Dynamics*. Online Book, Juin 1999.

[L2] L. S. Davis. *Foundations of Image Understanding*, volume 628 of *The International Series in Engineering and Computer Science*, chapter 16, pages 469–489. Kluwer Academic Publishers, Boston, 2001.

[L3] J. J. Gibson. Boston : Houghton Mifflin, 1950.

[L4] A. D. Jepson et D. J. Fleet. *Measurement of Image Velocity*. Kluwer, Mars 1992.

Articles dans des revues

[A1] J. K. Aggarwal et Q. Cai. Human motion analysis : A review. *Computer Vision and Image Understanding*, 73(3) :428–440, Mars 1999.

[A2] G. Aubert, R. Deriche et P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1) :156–182, 1999.

[A3] L. Alvarez, J. Weickert et S. Javier. Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39(1) :41–56, 2000.

[A4] B. F. Buxton et H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2(2) :59–75, Mai 1984.

[A5] J. L. Barron, D. J. Fleet et S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1) :43–77, 1994.

[A6] J. Bigun, G. H. Granlund et J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(8) :775–790, Août 1991.

[A7] R. Basri et D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(2) :218–233, Février 2003.

[A8] A. Bruhn, J. Weickert et C. Schnorr. Lucas/kanade meets horn/schunck : combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3) :211–231, 2005.

[A9] V. Caselles, R. Kimmel et G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22 :61–79, 1995.

[A10] K. Chun et J. Ra. An improved block matching algorithm based on successive refinement of motion vector candidates. *Signal Processing : Image Communications*, 6(2) :115–122, May 1994.

[A11] T. W. Drummond et R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(7) :932–946, Juillet 2002.

[A12] R. Deriche et G. Giraudon. A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10(2) :101–124, 1993.

[A13] Y. Fu, A. Erdem et A. Tekalp. Tracking visible boundary of objects using occlusion adaptive motion snake. *IEEE Trans. on Image Processing*, 9(12) :2051–2060, Décembre 2000.

[A14] D. J. Fleet et A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, pages 77–104, 1990.

[A15] D. M. Gavrila. The visual analysis of human movement : A survey. *Computer Vision and Image Understanding*, 73(1) :82–98, Janvier 1999.

[A16] G. D. Hager et P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(10) :1025–1039, Octobre 1998.

[A17] D. J. Heeger. Model for the extraction of image flow. *Journal of the Optical Society of America*, 2(2) :1455–1471, 1987.

[A18] D. Hogg. Model-based vision : a program to see a walking person. *Image Vision Computing*, 1(1) :5–20, 1983.

[A19] P. Hammond et J. Reck. Influence of velocity on directional tuning of complex cells in cat striate cortex for texture motion. *Neuroscience Letters*, 19 :309–314, 1981.

[A20] B. Horn et B. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[A21] A. D. Jepson et D. J. Fleet. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1) :77–104, Août 1990.

[A22] M. R. M. Jenkin et A. D. Jepson. Recovering local surface-structure through local phase difference measurements. *Computer Vision Graphics and Image Processing*, 59(1) :72–93, Janvier 1994.

[A23] D. Koller, K. Daniilidis et H. H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3) :257–281, Juin 1993.

[A24] S. Kumar et D. Goldgof. Recovery of global nonrigid motion : A model-based approach without point correspondences. *Journal of the Optical Society of America, JOSA-A*, 17(9) :1617–1626, Septembre 2000.

[A25] J. K. Kearney, W. B. Thompson et D. L. Boley. Optical flow estimation : an error analysis of gradient-based methods with local optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2) :229–244, 1987.

[A26] P. M. Kuhn. Fast MPEG-4 motion estimation : Processor based and flexible VLSI implementations. *The Journal of VLSI Signal Processing*, 23(1) :67–92, Octobre 1999.

[A27] M. Kass, A. P. Witkin et D. Terzopoulos. Snakes : Active contour models. *International Journal of Computer Vision*, 1(4) :321–331, Janvier 1988.

[A28] V. Lepetit et P. Fua. Monocular Model-Based 3D Tracking of Rigid Objects : A Survey. *Foundations and Trends in Computer Graphics and Vision*, (1) :1–89, 2005.

[A29] F. Leymarie et M. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(6) :617–634, Juin 1993.

[A30] D. G. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2) :113–122, Août 1992.

[A31] E. Marchand, P. Bouthemy et F. Chaumette. A 2D-3D model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13) :941–955, Novembre 2001.

[A32] F. Mokhtarian et F. Mohanna. Performance evaluation of corner detectors using consistency and accuracy measures. *Computer Vision and Image Understanding*, 102(1) :81–94, April 2006.

[A33] K. Mikolajczyk et C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, 2005.

[A34] H. H. Nagel. Displacement vectors derived from second-order. *Computer Vision, Graphics, and Image Processing*, 21(1) :85–117, 1983.

[A35] H. H. Nagel et W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5) :565–593, 1986.

[A36] J. Nascimento et J. Marques. Adaptive snakes using the em algorithm. *IEEE Trans. on Image Processing*, 14(11) :1678–1686, Novembre 2005.

[A37] E. Polat, M. Yeasin et R. Sharma. A 2D/3D model-based object tracking framework. *Pattern recognition*, 36(9) :2127–2141, Septembre 2003.

[A38] K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision Graphics and Image Processing*, 59(1) :94–115, Janvier 1994.

[A39] S. M. Smith et J. M. Brady. Susan : A new approach to low-level image-processing. *International Journal of Computer Vision*, 23(1) :45–78, Mai 1997.

[A40] A. Singh. Optic flow computation : A unified perspective, ieee computer society press, los alamitos. *CA*, 1992.

[A41] C. Stiller, J. Konrad et R. Bosch. On models, criteria and search strategies for motion estimation in image sequences. *IEEE Signal Processing Magazine*, pages 1–41, 1998.

[A42] C. Stiller, J. Konrad et R. Bosch. Estimating motion in image sequences : A tutorial on modeling and computation of 2D motion. *IEEE Signal Processing Magazine*, 16 :70–91, 1999.

[A43] M. Trajkovic et M. Hedley. Fast corner detection. *Image Vision Computing*, 6(2) :75–87, 1998.

[A44] G. TSECHPENAKIS, K. RAPANTZIKOS, N. TSAPATSOULIS et S. KOLLIAS. A snake model for object tracking in natural sequences. *Signal processing, Image communication*, 19(3) :219–238, 2004.

[A45] C. Vieren, F. Cabestaing et J. Postaire. Catching moving-objects with snakes for motion tracking. *Pattern Recognition Letters*, 16(7) :679–685, Juillet 1995.

[A46] A. M. Waxman. Contour evolution, neighborhood deformation, and global image flow : Planar surfaces in motion. *The International Journal of Robotics Research*, 4(3) :95–108, 1985.

[A47] L. Wang, W. M. Hu et T. N. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3) :585–601, Mars 2003.

[A48] S. Wachter et H. H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3) :174–192, Juin 1999.

[A49] T. Wiegand, G. Sullivan, G. Bjntegaard et A. Luthra. Overview of the H.264/AVC video codingstandard. *IEEE Transactions on Circuits and Systems for VideoTechnology*, 13(7) :560–576, 2003.

[A50] A. M. Waxman, J. Wu et F. Bergholm. Convected activation profiles and the measurement of visual motion. *Computer Vion and Pattern Recognition*, pages 717–723, 1988.

[A51] Z. Y. Zhang, R. Deriche, O. D. Faugeras et Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2) :87–119, Octobre 1995.

Communications

[C1] E. H. Adelson et J. R. Bergen. The extraction of spatio-temporal energy in human and machine vision. Dans *Workshop on visual motion*, pages 151–155, 1986.

[C2] M. J. Black et P. Anandan. Robust dynamic motion estimation over time. Dans *Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR'91*, pages 296–302, Lahaina, Maui, HI, USA, Juin 1991.

[C3] N. Badler. Graphical behavior and animated agents. Dans *Advanced Techniques in Human Modeling, Animation, and Rendering, ACM SIGGRAPH*, 92.

[C4] J. Bigun et G. H. Granlund. Optical flow based on the inertia matrix of the frequency domain. Dans *Proceedings from SSAB Symposium on Picture Processing*, pages 132–135, Lund University, Sweden, Mars 1988.

[C5] H. Barman, L. Haglund, H. Knutsson et G. H. Granlund. Estimation of velocity, acceleration and disparity in time sequences. Dans *Workshop on visual motion*, pages 44–51, 1991.

[C6] C. Bregler et J. Malik. Tracking people with twists and exponential maps. Dans *Computer Vion and Pattern Recognition*, pages 8–15, 1998.

[C7] R. Deriche et B. Bascle. Region tracking through image sequences. Dans *INRIA*, 1994.

[C8] J. Deutscher, A. Blake et I. D. Reid. Articulated body motion capture by annealed particle filtering. Dans *Computer Vion and Pattern Recognition*, volume 2, pages 126–133, 2000.

[C9] W. Enkelmann. Investigation of multigrid algorithms for the estimation of optical flow fields in image sequences. Dans *Workshop on visual motion*, pages 81–87, 1986.

[C10] Y. Fu, A. T. Erdem et A. M. Tekalp. Occlusion adaptive motion snake. Dans *International Conference on Image Processing*, volume III, pages 633–637, 1998.

[C11] W. Forstner. A feature based correspondence algorithm for image matching. Dans *International Archives of Photogrammetry and Remote Sensing*, volume 26, pages 150–166, 1986.

[C12] D. Freedman et M. W. Turek. Illumination-invariant tracking via graph cuts. Dans *Computer Vion and Pattern Recognition*, volume 2, pages 10–17, Juin 2005.

[C13] D. Grosu et H. Galmeanu. Motion estimation in MPEG-2 video encoding using A parallel block matching algorithm. Dans *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 16–26, 1998.

[C14] B. Galvin, B. Mccane, K. Novins, D. Mason et S. Mills. On benchmarking optical flow. Dans *Computer Vision and Image Understanding*, volume 84, pages 126–143, Octobre 2001.

[C15] C. Harris, C. et Stennett. RAPID : A video rate object tracker. Dans *British Machine Vision Conference*, 1990.

[C16] D. J. Heeger. Optical flow using spatiotemporal filters. Dans *International Conference on Computer Vision*, pages 181–190, 1987.

[C17] H. Yang, G. Welch et M. Pollefeys. Illumination insensitive model-based 3D object tracking and texture refinement. Dans *Third International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 869–876, University of North Carolina, Chapel Hill, USA, Juin 2006.

[C18] C. Harris et M. J. Stephens. A combined corner and edge detector. Dans *Fourth Alvey Vision Conference*, pages 147–152, Manchester, 1988.

[C19] A. D. Jepson et D. J. Fleet. Stability of phase information. Dans *Workshop on visual motion*, pages 52–60, 1991.

[C20] H. Liu, T. Hong et M. Herman. Accuracy vs. efficiency trade-offs in optical flow algorithms. Dans *European Conference on Computer Vision*, volume 72, pages 271–286, Cambridge, UK, Avril 1996.

[C21] B. D. Lucas et T. Kanade. An iterative image registration technique with an application to stereo vision. Dans *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, Avril 1981.

[C22] D. Lowe. Object recognition from local scale-invariant features. Dans *International Conference on Computer Vision*, pages 1150–1157, 1999.

[C23] H. Moravec. Towards automatic visual obstacle avoidance. Dans *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, page 584, August 1977.

[C24] T. McInerney et D. Terzopoulos. Topologically adaptable snakes. Dans *International Conference on Computer Vision*, pages 840–845, 1995.

[C25] J. Malik et J. Weber. Robust computation of optical-flow in a multiscale differential framework. Dans *International Conference on Computer Vision*, pages 12–20, 1993.

[C26] T. Papadopoulo et O. D. Faugeras. Estimation of the second order spatio-temporal derivatives of deforming image curves. Dans *International Conference on Pattern Recognition*, volume A, pages 179–184, 1994.

[C27] E. P. Simoncelli, E. H. Adelson et D. J. Heeger. Probability distributions of optical flow. Dans *Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR'91*, pages 310–315, Lahaina, Maui, HI, USA, Juin 1991.

[C28] J. Skoglund et M. Felsberg. Covariance estimation for sad block matching. Dans *15th Scandinavian Conference on Image Analysis, SCIA 2007*, pages 374–382, 10-14 Juin 2007.

[C29] A. Singh. An estimation-theoretic framework for image-flow computation. Dans *Proceedings of the 3rd International Conference on Computer Vision, ICCV'90*, pages 168–178, Osaka, Japan, Décembre 1990.

[C30] L. Taycher, J. W. Fisher, III et T. J. Darrell. Combining simple models to approximate complex dynamics. Dans *Statistical Methods in Video Processing*, page 94, 2004.

[C31] C. Tomasi et J. Shi. Good features to track. Dans *Computer Vion and Pattern Recognition*, pages 593–600, 1994.

[C32] C. Tomasi et J. Shi. Good features to track. Dans *Proceedings of the Conference on Computer Vision and Pattern Recognition, CVPR'94*, pages 593–600, Seattle, WA, USA, 21-23 Juin 1994.

[C33] D. Terzopoulos, A. P. Witkin et M. Kass. Snakes : Active contour models. Dans *International Conference on Computer Vision*, pages 259–268, 1987.

[C34] L. Vacchetti, V. Lepetit et P. Fua. Combining edge and texture information for real-time accurate 3D camera tracking. Dans *ISMAR '04 : Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 48–57, Washington, DC, USA, 2004. IEEE Computer Society.

[C35] P. Wunsch et G. Hirzinger. Real-time visual tracking of 3D objects with dynamic handling of occlusion. Dans *Conference on Robotics and Automation*, pages 2868–2873, Albuquerque, NM, USA, 1997.

Thèses de Doctorat

[T1] P. Anandan. *Measuring visual motion from image sequences*. phdthesis, Measuring visual motion from image sequences, 1987.

[T2] C. Bernard. *Ondelettes et problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière*. PhD thesis, Centre de Mathématiques Appliquées, 1999.

[T3] L. Haglund. *Adaptive Multidimensional Filtering*. PhD thesis, Linkoping University, Sweden, SE-581 83 Linkoping, Sweden, Octobre 1992. Dissertation No 284, ISBN 91-7870-988-1.

[T4] B. D. Lucas. *Generalized Image Matching by the Method of Differences*. phdthesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Juillet 1984.

[T5] E. P. Simoncelli. *Distributed Representation and Analysis of Visual Motion*. phdthesis, Massachusetts Institute of Technology, 1993.

Publications relatives à cette thèse

[B1] Y. Bachalany, F. Cabestaing et S. Ambellouis, *Suivi de libellules par analyse de séquence d'images*, 29-31 Octobre 2008.
MajeSTIC'2008.

[B2] Y. Bachalany, F. Cabestaing, S. Ambellouis et C. Vieren, *An altered image alignment technique for 3d motion estimation of a reflective sphere*, Novembre 2008.
IPTA'2008.

[B3] Y. Bachalany, F. Cabestaing, S. Ambellouis et C. Vieren, *Tracking Dragonflies in Image Sequences*, Décembre 2008.
ICPR'2008.

[B4] S. Riachy, Y. Bachalany, M. M'boup et J.-P. Richard, *An algebraic method for multi-dimensional derivative estimation*, 2008.
MED'2008.

Rapports techniques

[R1] P. Anandan. A computational framework and an algorithm for the measurement of visual. Rapport technique, University of Massachusetts, Amherst, MA, USA, 1987.

[R2] S. Baker, R. Gross et I. Matthews. Lucas-kanade 20 years on : A unifying framework : Part 4. Rapport technique CMU-RI-TR-04-14, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Février 2004.

[R3] P. Barnum, B. Hu et C. Brown. Exploring the practical limits of optical flow. Rapport technique, University of Rochester, 2003.

Résumé

Dans le cadre de cette thèse, nous proposons une solution au problème de l'estimation du mouvement 3D d'objets dont la surface est réfléchissante, lequel est très complexe quand leurs caractéristiques géométriques à elles seules ne permettent pas de définir des indices caractéristiques d'un mouvement. Nous avons focalisé notre étude sur le cas particulier de la sphère, dont la parfaite symétrie complique au maximum le problème d'estimation du mouvement.

En effet, lorsqu'une sphère est affectée d'un mouvement de rotation pure autour d'un de ses axes, ses contours extérieurs apparaissent statiques. Ces derniers n'apportent donc pas d'information utilisable par le système pour estimer le mouvement. Nous proposons donc une approche se basant sur la texture dans un schéma de mise en correspondance 3D/2D modifié. Ici, cette information est exploitée différemment si elle provient d'une composante spéculaire ou diffuse.

Enfin, nous démontrons que la propriété réfléchissante de la surface n'est plus considérée comme un obstacle mais que au contraire, cette propriété procure une information supplémentaire sur le mouvement recherché.

Mots clés : vision artificielle ; estimation mouvement 3D ; patchs ; mise en correspondance modèle/images ; réflexion spéculaire ; réflexion diffuse ; Lucas/Kanade

Abstract

Recovering 3D motion of reflective objects in image sequences is still a cumbersome problem for computer vision. One common approach is to track geometric features of the object such as contours and edges since they are rather insensitive to light reflections. However, such basic features fail to recover the actual 3D motion in some cases. For example, the external contour of a sphere rotating about one of its axes remains static in the image. In this thesis, we propose a new approach to 3D motion recovery of a reflective sphere visible in an image sequence.

Instead of tracking only geometric features, our technique makes use of texture information in a slightly modified image alignment method. Unlike in classical image alignment methods, texture information is processed differently whether it comes from a diffuse or a specular light component.

Using this technique, we show that motion estimation is not only possible when dealing with reflective objects, but also that specular components can offer information about the 3D motion. Finally, we present some results obtained from the analysis of image sequences.

Keywords : artificial vision ; 3D motion estimation ; patches ; 3D/2D alignment ; specular reflection ; diffuse ; Lucas/Kanade

Numéro d'ordre : 40180

Volume 2 : Supplément

Thèse présentée et soutenue publiquement par

Yara BACHALANY

le 18/12/2009

pour obtenir le grade de

Docteur de l'Université des Sciences et Technologies de Lille

en Automatique, Génie Informatique, Traitement du Signal et des Images

Estimation du mouvement 3D de libellules par analyse de séquences d'images : Analyses préliminaires

P. Bonton	Rapporteur	Professeur à l'Université Blaise Pascal, Clermont-Ferrand
M. Rudko	Rapporteur	Professeur à Union College, Schenectady, NY, USA
O. Colot	Examinateur	Professeur à l'Université de Lille1, Villeneuve d'Ascq Cedex
P. Sayd	Examinateur	Chargé de recherche au CEA, SACLAY
S. Ambellouis	Co-directeur	Chargé de recherche à l'INRETS, Villeneuve d'Ascq Cedex
F. Cabestaing	Co-directeur	Professeur à l'Université de Lille1, Villeneuve d'Ascq Cedex

Thèse préparée au **L**aboratoire d'**A**utomatique, **G**énie **I**nformatique & **S**ignal

LAGIS - UMR CNRS 8146

Introduction

Dans le manuscrit de cette thèse, nous avons présenté une méthode hybride permettant d'analyser le mouvement 3D d'une sphère dont la surface est réfléchissante. Cette étude théorique a été motivée à l'origine par un sujet beaucoup plus appliqué, à savoir l'analyse du vol d'une libellule lors de la capture d'une proie à partir d'une séquence d'images.

Dans ce document complémentaire à notre manuscrit de thèse, nous décrivons ce problème initial et montrons comment il nous a amenés à nous focaliser sur l'analyse du mouvement d'une sphère dont la surface est réflechissante. La sphère est le modèle géométrique simple qui semble le plus adapté à décrire la tête de la libellule, dont nous décrirons ici les propriétés très particulières.

Nous présentons dans un premier temps les expérimentations qui ont permis aux biologistes d'acquérir les séquences d'images du vol d'une libellule lorsqu'elle capture une proie. Ensuite, nous décrirons les analyses qu'ils ont menées manuellement afin de modéliser le comportement de la libellule lorsqu'elle décolle et vole lors de la prédation. Enfin, nous décrirons les études préliminaires de vision artificielle que nous avons menées avant de nous focaliser sur le problème traité dans le manuscrit de thèse.

Table des matières

Table des figures

Chapitre 1

Capture d'une proie par une libellule

1.1 Introduction

Les libellules sont d'excellents prédateurs. La stratégie de chasse de la famille des *Libellulidae* consiste à attendre leur proie perchées sur une végétation et à choisir le moment propice pour décoller lorsque de petits insectes les survolent. Le plus étonnant, c'est que la capture se produit pendant le vol : elles approchent leur proie par le dessous et basculent leur corps au dernier instant afin de l'attraper avec leurs pattes tendues. L'interception est extrêmement précise avec des taux de réussite de 97% [A16]. C'est le système visuel extrêmement complexe des libellules qui guide ce processus de capture.

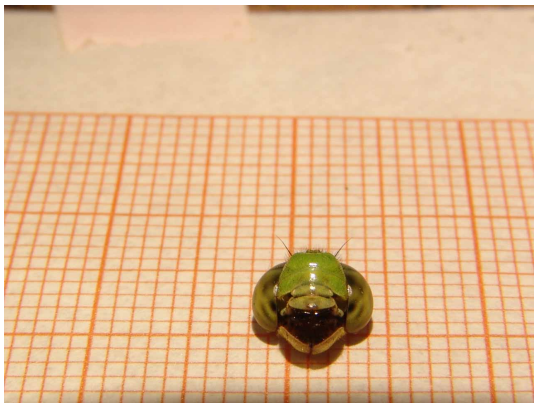
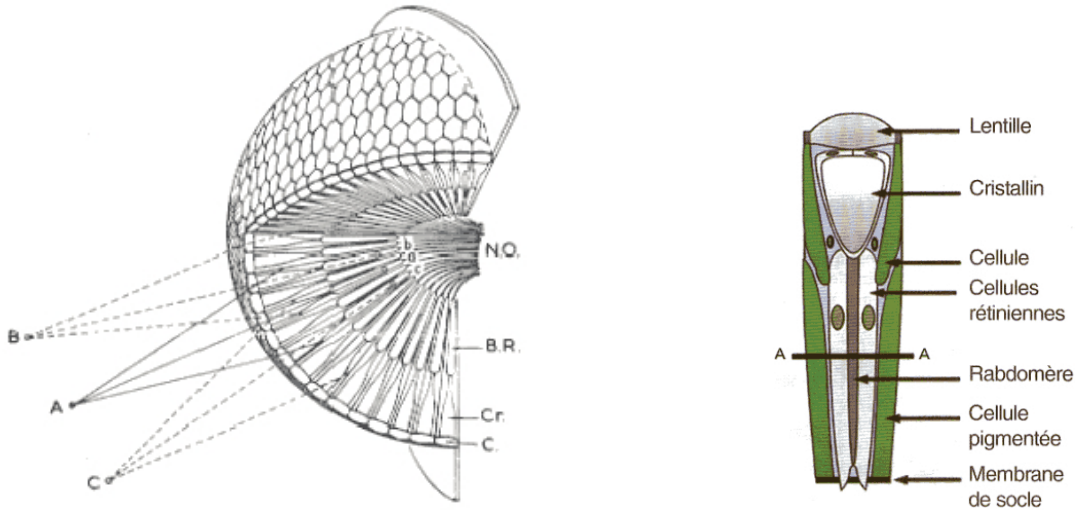


FIGURE 1.1 : Les yeux de libellule

Les libellules ont deux yeux énormes (voir figure 1.2(a)), composés par l'assemblage de milliers d'unités simples appelées ommatidies, qui correspondent chacune à une facette distinguable sur l'oeil de l'insecte. Cette structure d'oeil composé est commune à presque tous les invertébrés. Chaque ommatidie fonctionne comme un

capteur indépendant. En effet, un objet qui diffuse la lumière dans toutes les directions, est *observé* grâce aux rayons dirigés exactement dans l'axe du rabdomère ou bâtonnet rétinien (figure 1.2(b)). Ainsi, chaque ommatidie ne capte qu'un point lumineux de l'image entière. Cette dernière est ensuite recomposée par le rassemblement de toutes ces informations captées par l'ensemble des ommatidies.



(a) Vue générale de l'oeil (b) Coupe d'une ommatidie

FIGURE 1.2 : Oeil de l'abeille : C cornée, Cr cristallin, BR bâtonnet rétinien, NO nerf optique. Aux points A, B et C, situés dans le champ visuel, correspondent les points-images rétiniens a,b et c ; l'image qui se forme est donc droite (images extraites de [A1])

Plus l'objet est lointain, moins nombreuses sont les ommatidies excitées lors de son passage. Ceci le rend difficilement discernable. Il a été prouvé [A1] qu'un obstacle que l'humain peut distinguer à 18 m de distance, n'est visible pour une abeille qu'à 30 cm de distance. Par contre, l'oeil composé présente un énorme avantage : il permet à l'insecte d'estimer précisément le mouvement des objets qui se déplacent dans son champ de vision grâce à l'effet de compilation que procure l'activation/désactivation successive des ommatidies.

Les libellules sont les insectes dont les yeux composés possèdent le plus d'ommatidies (jusqu'à 30 000), ce qui leur confère une vision des mouvements très précise. Ces yeux occupent à peu près la moitié de la surface de la tête de l'insecte et fournissent à celui-ci un très grand champ de vision. Une zone de l'oeil, située sur la partie dorsale, joue un rôle primordial durant la capture d'une proie. Dans cette zone, les ommatidies ont une plus grande surface que sur le reste des yeux.

Le ommatidies de cette zone sont principalement sensibles à la lumière bleue et ultraviolette, permettant à l'insecte de distinguer clairement leur proie par rapport au "fond" que constitue le ciel bleu.

Les yeux des libellules leur permettent donc d'accomplir des performances inégalées dans le monde animal durant la capture des proies. Le comportement prédateur des libellules est un exemple remarquable d'interception de trajectoire par asservissement visuel, dont la compréhension pourrait guider à la construction de mécanismes biomimétiques. Robert Olberg, biologiste de Union College (Schenectady, NY, USA) avec qui nous collaborons, a entrepris ce projet afin de comprendre le fonctionnement des neurones permettant ce comportement visuellement guidé, précis et rapide.

Un comportement aussi complexe que la capture d'une proie pendant le vol consiste en au moins trois processus séparés mais interdépendants. Le premier est la décision de décollage ; le second est la navigation en suivant une trajectoire d'interception ; le troisième est la coordination des mouvements des pattes dans l'espace et le temps afin d'agripper la proie. Les biologistes envisagent d'analyser ces trois processus en émettant des hypothèses, puis en essayant de les vérifier grâce à des séquences vidéos qu'ils ont enregistrées. Une description détaillée des expérimentations et de la plate-forme qui a permis l'acquisition de ces séquences est présentée dans le paragraphe suivant. Nous présentons ensuite les analyses manuelles que les biologistes ont entreprises sur ces séquences. Nous concluons sur l'intérêt qu'il y aurait à mener des analyses automatiques sur ces séquences.

1.2 Description de la plate-forme

1.2.1 Cage à libellule

Les libellules ne chassent pas lorsqu'elles sont en captivité et ont tendance à s'affamer même quand leur cage contient des insectes volants. Pour éviter cela, une cage a été installée en plein air, dont les murs et le plafond laissent pénétrer tous les rayons de soleil y compris les rayons UV. La cage est de dimensions $2.9 \times 2.9 \times 2.5$ m^3, construite avec des tubes de cuivre de 1.9 cm de diamètre. Cette cage

est recouverte d'un filet polyéthylène/polypropylène (US Netting, ERIE, PA). La dimension des mailles (0.5 cm) permet aux petits insectes tels les moustiques et les moucherons de rentrer dans la cage, assurant ainsi une capture occasionnellement "naturelle" pour la libellule. Cette cage a été installée dans une cour procurant un abri contre le vent et permettant une bonne réception des rayons solaires pendant les périodes d'enregistrement, généralement comprises entre $9h$ et $12h : 30$. Des perches de bois ou de polystyrène expansé ont été dispersées dans la cage. Un bassin de 1 m de diamètre et quelques plantes ont été ajoutés pour procurer de l'eau et de l'ombre aux libellules.

6 à 8 libellules femelles *Erythmesis simplicicollis* (famille des *Libellulidae*) ont été capturées et introduites dans la cage. Il leur fallait une journée pour récupérer leur comportement naturel, se percher et éventuellement décoller pour capturer une proie. Les libellules mâles n'ont pas été utilisées parce qu'elles ne s'adaptent pas à la captivité et ne récupèrent pas leur comportement naturel.

Pour capturer la séquence de chasse sur vidéo, les chercheurs ont attaché une boule blanche de 2 mm de diamètre à un fil de tungstène très fin (75 μm de diamètre), donc invisible pour l'insecte. Cette boule blanche a été choisie pour sa ressemblance avec de petits insectes dont se nourrit la libellule. Ensuite, ils ont déplacé cette boule blanche dans la cage au dessus de la libellule pour l'attirer et l'inciter à chasser. Pour restreindre le mouvement de la libellule autant que possible dans un plan, ils ont déplacé la boule dans un plan parallèle au plan image de la caméra.

D'autres expériences consistaient à attacher la proie artificielle sur un fil en nylon suspendu entre les deux sommets d'un instrument en forme de U. L'avantage de ce montage un peu plus rigide est qu'il procure plus de contrôle sur le mouvement de la boule. Néanmoins, aucune différence n'a été constatée vis à vis du comportement de la libellule. Après quelques essais de capture, la libellule s'habituait aux conditions expérimentales et arrêtait de chasser la proie artificielle. Dans quelques cas, un véritable insecte était attaché au fil afin de récompenser la libellule et l'inciter à maintenir son comportement prédateur. Cela a permis de conserver les mêmes libellules pendant 2 à 3 jours d'expérimentation avant de devoir les remplacer.

1.2.2 Vidéos

Les séquences d'images ont été enregistrées à l'aide d'une caméra à haute cadence Redlake Motion pro 2000 (Redlake MASD LLC, Tucson, AZ, USA). Une lentille Elicar 90 mm est montée sur cette caméra. La caméra est installée à une distance variant entre 0.5 et 1.5 m de la libellule. Pour aider à la localisation de la tête de la libellule avant la capture de proie, les perches ont été montées de façon à encourager la libellule à se positionner dans l'axe de la caméra ou dans un plan perpendiculaire à l'axe de la caméra. De plus, la caméra a été fixée à une position et orientation permettant de garder la libellule et sa proie dans le champ de vision durant la capture.

La caméra a été connectée à un ordinateur Acme KB-108-1 tournant sous Microsoft Windows 2000. Les séquences ont été acquises grâce au logiciel Redlake MIDAS permettant une résolution temporelle de 500 images par seconde et une résolution spatiale de 1280×1024. L'expérimentation a permis d'enregistrer 128 séquences représentant une *Erythmesis Simplicicollis* capturant sa proie.

La libellule reste dans l'axe de la caméra ou dans un plan parallèle à celui de la caméra dans 48 de ces vidéos (respectivement 33 et 15). Environ 25 libellules différentes apparaissent sur ces séquences. La proie est habituellement déplacée à 15 cm de la libellule, mais parfois à seulement 5 cm. Les séquences sont donc de très courtes durées, soit environ 200 ms (168 ms de moyenne et variant entre 62 et 480 ms). Ces durées de vols sont proches de celles remarquées pour les libellules chassant en milieu naturel (184 ms). Ces séquences ont été analysées afin de localiser la tête de la libellule avant et durant la capture de sa proie ainsi que la trajectoire de la libellule et celle de sa proie.

1.3 Analyses manuelles

Dans ce paragraphe nous présentons toutes les analyses qui ont été effectuées manuellement par les biologistes. Nous présentons ensuite les résultats qu'ils ont obtenus.

Les séquences ont été parcourues image par image, en utilisant le logiciel d'ana-

lyse Redlake MIDAS. Les coordonnées des points d'intérêt ont été transférées dans un tableur Microsot Excel pour l'analyse. Pour chaque image, les points d'intérêts recueillis sont : la position de la proie, le centre de la tête ainsi que deux autres points de la tête. Dans les séquences où la libellule se déplace dans un plan parallèle à celui de la caméra, les coordonnées de la pointe de l'abdomen ont également été extraites. L'extraction des coordonnées de points d'intérêts dans douze des séquences disponibles ont été menées par des personnes différentes afin de vérifier la validité des sélections. Les études menées par les biologistes sont décrites dans ce qui suit.

– *Trajectoire de poursuite*. Une première hypothèse que l'analyse manuelle a permis de vérifier facilement est celle d'une interception de trajectoire : la libellule vise un point situé devant sa proie (figure 1.3(b)) et non la position actuelle de sa proie (figure 1.3(a))(figures présentées dans [A16]). Dans la plupart des cas où la libellule maintient une vitesse et une direction assez constantes, sa trajectoire ressemble à une ligne droite conduisant au point d'interception. Les biologistes ont aussi pu constater que la trajectoire de la libellule n'est pas pré-planifiée puisqu'elle réagit efficacement à un changement de direction de sa proie. Cela a incité à mener l'étude suivante.

(a) Poursuite de la position actuelle (b) Trajectoire d'interception

FIGURE 1.3 : Trajectoire de la libellule

– *Temps de réponse*. Des trajectoires comme celles présentées dans la figure (figure 1.4) ont permis de calculer le temps de réponse de la libellule à un changement de direction de sa proie dans une étude menée en 2007 [A14]. Durant certains vols, les biologistes ont remarqué un battement d'ailes de plus forte amplitude précédant la correction de trajectoire. Le temps de latence entre la déviation de la trajectoire de la proie et la réaction remarquée sur les ailes

varie entre 26 et 42 ms (moyenne = 29 ms, s = 6.4 ms, n=6). Une latence de 48 ms en moyenne a été observée entre le changement de direction de la proie et un changement observable de la direction de la libellule (s = 5.1 ms, n = 16). La figure 1.4 illustre une réponse au niveau des ailes après 28 ms induisant une variation de la direction après 36 ms. Hors, une étude menée en 2000 (voir [A16]), avec des résolutions temporelle et spatiale inférieures, a abouti à des résultats différents : 33 à 50 ms. Ceci dit, ces résultats pourraient gagner en précision si une localisation des ailes, de la tête et de l'abdomen de la libellule était rendue possible. Cela permettrait aux biologistes de calculer la vitesse de transmission du signal (ou la vitesse de la conduction nerveuse) vers les neurones des aires motrice et prémotrice de la région thoracique, lesquelles excitent les muscles afin de permettre le changement de direction.

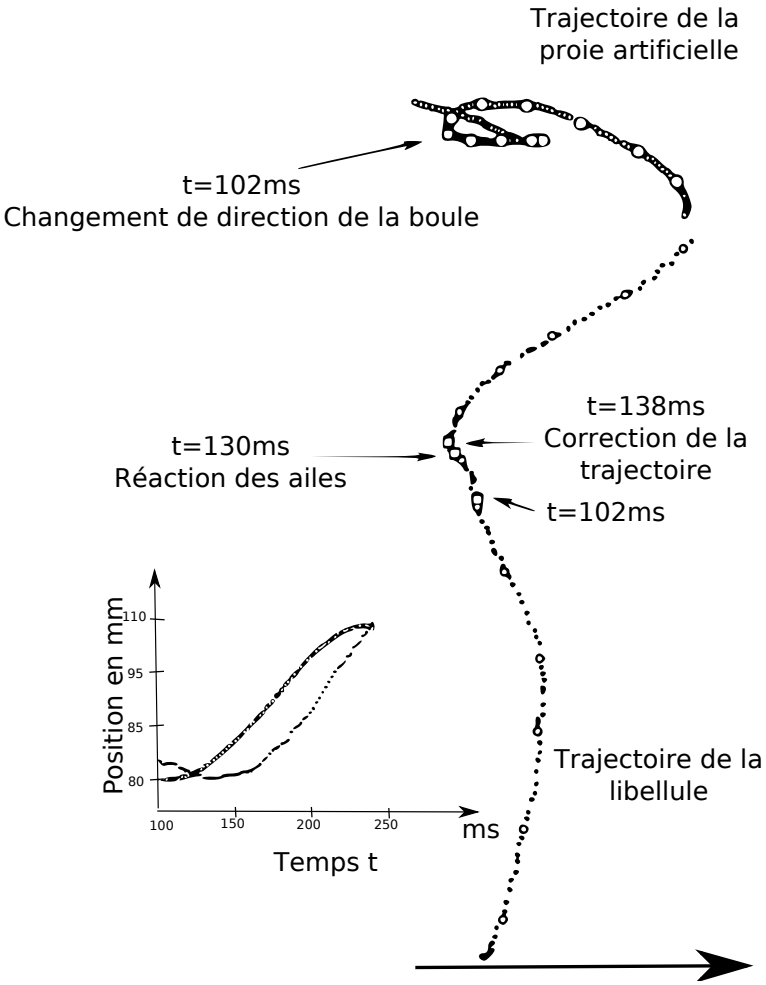


FIGURE 1.4 : Réponse de la libellule au changement de direction de sa proie.

– *Orientation de la tête*. Une étude a été menée sur l'orientation de la tête de la libellule durant la capture de proie. Le but de cette étude est de déterminer la direction de vue de la libellule afin de vérifier l'hypothèse suivante : l'orientation de la tête de la libellule est ajustée par rapport à la position de sa proie. Un petit déplacement de l'image de sa proie sur sa rétine induit des mouvements au niveau de la tête de la libellule afin de rétablir l'image de sa proie à sa position stable. Par la suite, une commande est envoyée aux muscles des ailes afin de corriger la trajectoire. Pendant leurs analyses manuelles, les biologistes ont remarqué que seules 18 séquences sont exploitables manuellement : 7 montrant la libellule de face et 11 la montrant de profil. A une exception près, la libellule semble maintenir la position de sa proie à une position presque fixe sur son oeil grâce à des rotations effectuées au niveau du cou. C'est cette nouvelle orientation de la tête qui déterminera la commande envoyée aux ailes. Par contre, cette étude manque en précision pour plusieurs raisons :
  – Le nombre de séquences exploitables n'est pas suffisant pour valider l'hypothèse.
  – Afin d'estimer les distances, les biologistes ont supposé les dimensions des différentes *Erythemis* fixes à 6 mm pour la largeur de la tête et 41 mm pour la longueur du corps (thorax et abdomen). Ces mesures sont entachées d'erreurs de calibration d'une variabilité estimée à 3%.
  – L'orientation de la libellule n'est pas précisément dans l'axe de la caméra ou perpendiculairement à celle-ci, ce qui entraîne des erreurs sur les mesures de position.
  – Il n'est pas évident de définir des points d'intérêts sur la tête de la libellule (ce qui entraîne une erreur sur l'angle estimée à 5 degrés) ou de les localiser sur les différentes images comportant du flou de bougé et des occultations de ces points et une variation de leur apparence due à la réflexion spéculaire et au phénomène de pseudopupille (voir chapitre 2).
  – Sur la séquence d'exception une nouvelle stratégie est rendue possible : la libellule garde sa tête fixe au lieu de l'orienter afin de garder l'image de sa

proie au milieu de sa rétine.

– *Estimation de la distance*. La décision de décollage à la poursuite d'un objet en mouvement peut-être le résultat d'une estimation de la distance de la proie et par suite de sa taille. Une étude menée en 2004 (voir [A15]) visait à déterminer le mécanisme d'estimation de distance que les libellules emploient. Cette estimation peut être générée par deux mécanismes équiprobables :
  – Le mouvement de la tête avec une contribution des pattes : la flexion du cou et les mouvements des segments du thorax génèrent des données stéréoscopiques permettant à l'insecte d'estimer la distance de la proie.
  – Les objets volants distants comme les avions et les oiseaux gardent une vitesse angulaire relativement constante sur la rétine alors que les objets plus proches provoquent une variation importante de la vitesse angulaire pendant leur passage au-dessus de l'insecte. Grâce à sa vision exceptionnelle des mouvements, la libellule est donc capable de repérer la position de sa proie en estimant la variation de sa vitesse angulaire apparente.

1.4 Conclusion

Les trois processus dont la capture d'une proie est composée sont loin d'être complètement élucidés. Un élément essentiel à cette étude est l'estimation de la position et l'orientation de tous les membres de la libellule avant et durant la capture de proie. A partir de ces données, il serait possible de trancher sur la technique utilisée durant cette activité étonnante.

Cette étude est trop complexe pour être résolue par une simple extraction manuelle de points d'intérêt, d'où l'intérêt d'une analyse automatique. En effet, l'analyse automatique présente l'avantage d'éviter les erreurs humaines, de faciliter l'extraction de données, d'obtenir une précision sub-pixel, de suivre des textures et des formes difficilement repérables par l'oeil humain. Sous certaines hypothèses, elle permettrait même de résoudre les problèmes d'occultation et de variation d'apparence dues aux réflexions spéculaires, au flou de bougé et au phénomène de pseudopupille au niveau de la tête de la libellule. Ces phénomènes sont décris dans le chapitre 2 dans lequel nous détaillons les défis auxquels nous serons confrontés

lors de l'estimation du mouvement 3D de la libellule.

Chapitre 2

Analyses préliminaires

2.1 Description des séquences

Dans ce chapitre, nous présentons une analyse préliminaire des séquences du point de vue de la vision artificielle. Cette analyse nous permet de souligner les défis qu'il s'agit de relever pour estimer le mouvement 3D de la tête et des autres parties du corps de la libellule afin d'en tirer des conclusions concernant les approches possibles d'estimation automatique du mouvement.

Dans les séquences dont nous disposons, la caméra est fixée à une position et une orientation permettant de conserver la libellule et sa proie dans le champ de vision durant la capture. Une seule vue, obtenue à partir d'une seule caméra, est alors disponible. La figure 2.1 présente 12 images successives de l'une des séquences dont nous disposons, correspondant à la phase de décollage.

Tenir compte au maximum de l'information portée par les variations temporelles de l'intensité lumineuse des points de l'image de l'insecte représente le premier défi. Cette variation est due à la nature réfléchissante de la surface du corps et de la tête, au flou de bougé et au problème d'occultation causé par les ailes de la libellule. Au niveau de la tête, s'ajoute un processus non standard de formation de l'image causé par le phénomène de pseudopupille. Ce dernier est décrit dans la première section de ce chapitre.

Ensuite, nous présentons une analyse préliminaire sur laquelle nous pouvons nous baser pour construire un modèle cinématique et dynamique du corps de la libellule.

Enfin, puisque le déplacement apparent de certaines parties du corps de l'in-

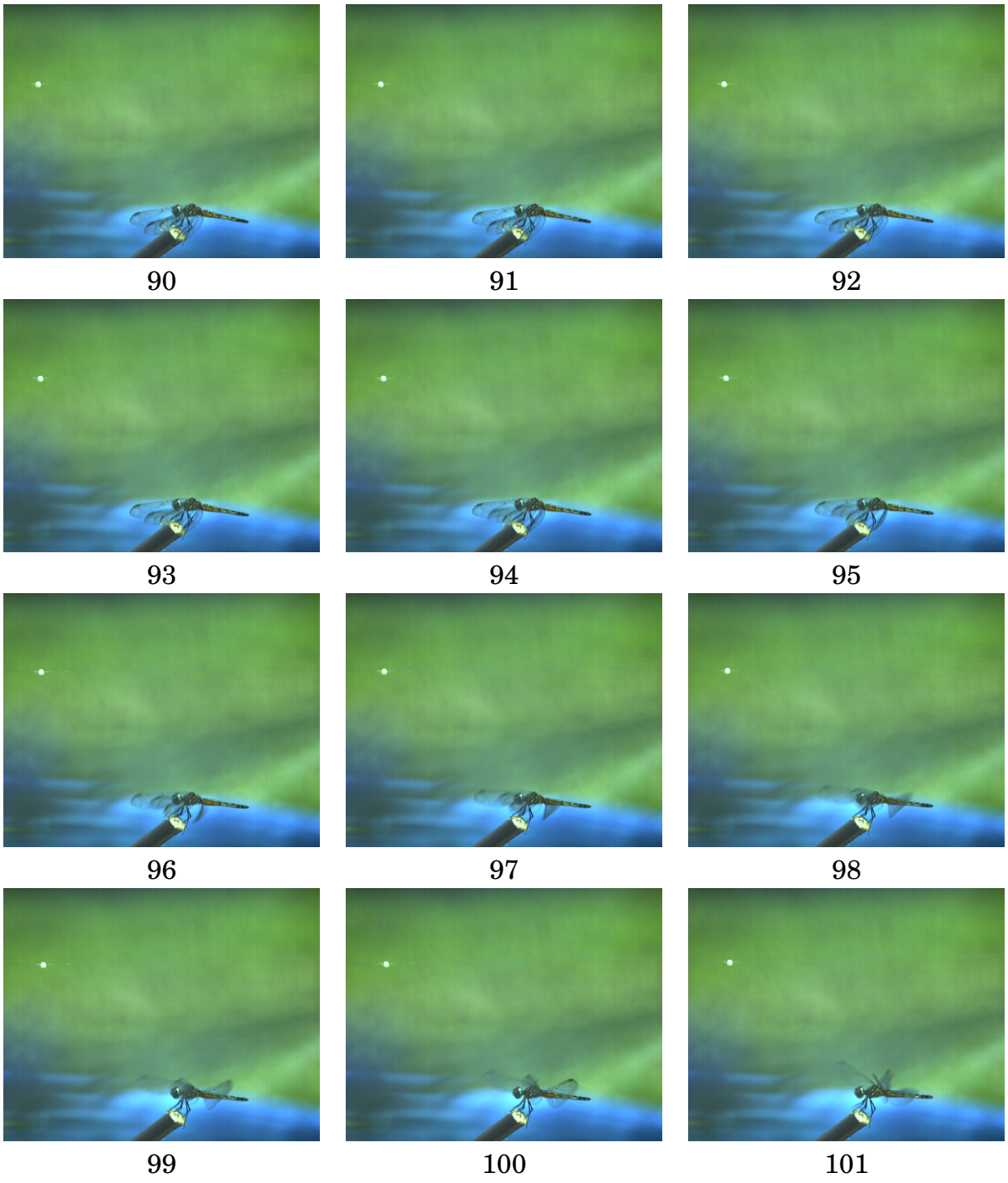


FIGURE 2.1 : 12 images successives montrant la libellule au moment du décollage.

secte est assez important entre deux images successives (parfois plus de dix pixels), nous montrons l'intérêt qu'il peut y avoir à prédire les paramètres du mouvement 3D grâce à différentes techniques.

2.2 Validité de l'hypothèse de conservation temporelle de l'intensité lumineuse

La majorité des méthodes d'estimation du mouvement se basent sur une hypothèse de conservation temporelle de l'intensité lumineuse des points images. La plupart du temps, cette hypothèse permet de prévoir la disposition de la texture après un mouvement 3D, et de comparer cette prédiction à la mesure que constitue la nouvelle image analysée. Cette comparaison autorise la correction du mouvement 3D estimé *a priori*, puis son utilisation pour la prédiction de la disposition de la texture dans l'image à venir. Dans les séquences dont nous disposons, plusieurs phénomènes physiques remettent en question la validité de cette hypothèse dans le cas des images d'une libellule.

La nature réfléchissante de la peau de la libellule constitue le premier obstacle. En effet, sous l'influence d'un mouvement 3D, les méthodes de la littérature prévoient la même disposition de texture indépendemment du fait que cette texture provienne d'un phénomène de réflexion spéculaire ou de diffusion de la lumière. En réalité, ces deux phénomènes impliquent des mouvements apparents différents de la texture. De ce fait, dans le cas où le mouvement 3D réel est exploité dans la prédiction, une différence notable sera présente entre la texture prédite et la texture observée.

Un autre problème compliquant l'étape de prédiction est celui du phénomène de pseudopupille (figure 2.2). Ce phénomène est causé par la structure de l'oeil composé de la libellule. Ce dernier intègre des milliers de petites lentilles situées aux extrémités des ommatidies, lesquelles absorbent complètement les rayons lumineux. De ce fait, aucun rayon n'est réfléchi et la surface correspondante dans l'image est très sombre. Ce phénomène se manifeste ainsi par l'apparition de tâches noires dans les régions de l'image pour lesquelles la direction des ommatidies est la même que celle de l'axe optique de la caméra. L'intensité lumineuse des points image au niveau des yeux de la libellule est donc loin d'être constante dans le temps et doit être modélisée par une fonction de réflectance bidirectionnelle non standard.

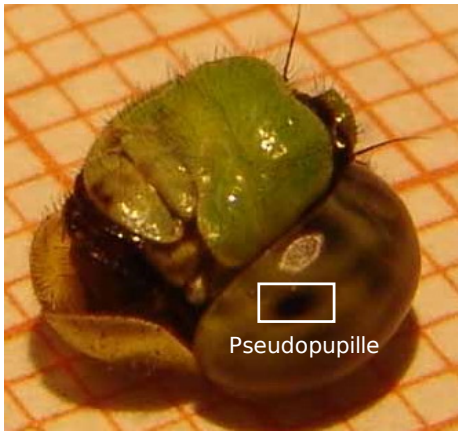


FIGURE 2.2 : Phénomène de pseudopupille

Outre la modélisation du phénomène de pseudopupille, la prise en compte du flou de bougé est également indispensable. Le déplacement de la libellule étant extrêmement rapide, il faut utiliser une cadence d'acquisition des images très élevée. Dans ces conditions, pour que l'image soit suffisamment claire, il n'est pas possible de diminuer le temps d'intégration, technique habituellement retenue pour diminuer le flou de bougé.

Enfin, un défi supplémentaire se pose du fait de l'occultation causée par les ailes de la libellule et la difficulté d'estimer le mouvement très rapide de cette partie transparente du corps de l'insecte.

2.3 Modélisation cinématique et dynamique : Analyses préliminaires

2.3.1 Étude cinématique

Afin de parvenir à une modélisation cinématique efficace, nous commençons par étudier l'anatomie des libellules. Cette étude vise à identifier les parties qui composent le corps de la libellule et les degrés de liberté des articulations qui les relient.

Les libellules font partie de la famille des odonates. Les odonates comprennent deux groupes : les libellules ou anisoptères et les demoiselles ou zygoptères. On les distingue grâce à leurs ailes : les anisoptères (aniso = inégale, ptère = aile) ont

leurs ailes postérieures plus larges et plus courtes que les antérieures, alors que les zygoptères ont leurs quatre ailes presque identiques en longueur et en largeur. Généralement, on confond ces deux groupes sous le nom unique de libellule.

Comme pour tous les insectes, le corps des odonates est formé de trois parties principales : la tête, le thorax et l'abdomen. 4 900 espèces ont été rescencées au niveau mondial, lesquelles sont parfaitement adaptées à la prédation car toutes carnivores au stade adulte.

Chacune de ces espèces diffère dans son anatomie et dans une même espèce les tailles et poids des individus changent en fonction de l'environnement. Nous détaillons dans les paragraphes suivants les similarités et les différences au niveau des membres dont il faut tenir compte lors de la modélisation cinématique, en nous appuyant sur le livre de R.J. Tillyard, "The Biology of Dragonflies" [L2]. Puisque que c'est l'espèce étudiée par les biologistes dans les séquences dont nous disposons, nous nous focalisons sur le cas des libellules (anisoptères) afin de limiter les paramètres du modèle.

2.3.1.1 La tête

La tête des libellules est formée de deux grands yeux composés qui occupent une très grande surface (*cf.* figure 2.3). Les yeux sont orientés vers le haut et vers l'arrière. Le côté facial inférieur de la tête est occupé par des pièces buccales permettant aux libellules de broyer leurs proies. Sur la frontière supérieure de ces pièces buccales nous pouvons remarquer une sorte de plateau central appelé *clypeus*. Ce plateau est séparé du front par une ligne de suture horizontale. Dans certaines espèces de libellules, le front a une couleur distinctive et ressemble à un "T" majuscule d'où l'appellation *T-spot*. Entre le front et l'oeil composé, nous pouvons observer le vertex auprès duquel sont situées les antennes.

La tête de la libellule est reliée au prothorax. La liaison peut être modélisée par une rotule (3 degrés de liberté correspondant aux rotations suivant les trois axes).

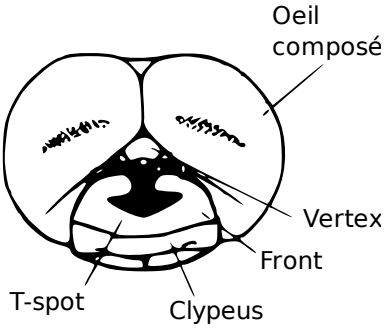


FIGURE 2.3 : La tête d'une libellule

2.3.1.2 Le thorax

Le thorax des libellules est principalement formé de deux segments : le prothorax et le ptérothorax (figure 2.4). Le prothorax est un petit segment relié à la tête et porte une première paire de pattes. Le ptérothorax quant à lui, est large et représente une base pour les deux paires d'ailes ainsi que les deux paires de pattes restantes. Il est couramment nommé "thorax". Le mouvement relatif de ces deux segments semble nul sur les séquences analysées.

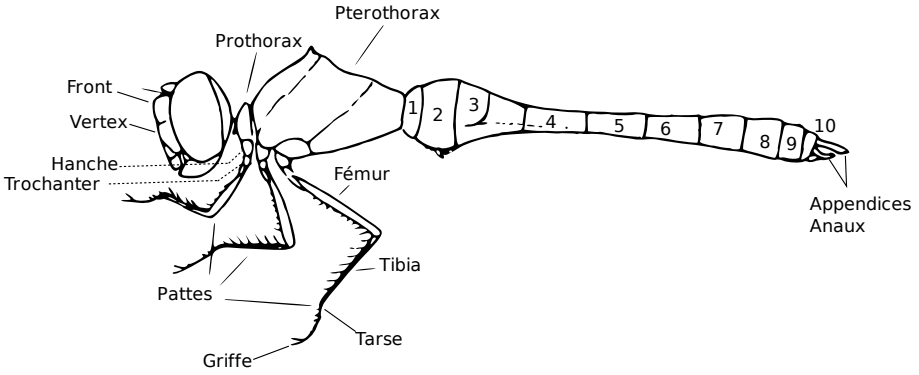


FIGURE 2.4 : L'anatomie d'une libellule

2.3.1.3 Les pattes

Nous pouvons remarquer le rôle primordial que jouent les pattes de la libellule durant la capture de la proie. En premier lieu, la libellule change d'orientation avant de décoller à l'aide de ses pattes. Pendant le vol les six pattes sont maintenues pliées sous le thorax. Ce sont les pattes qui attrapent la proie pendant le vol. Les deux paires de pattes antérieures maintiennent la proie pendant que l'insecte

la mâche. On constate que les deux pattes antérieures sont très proches de la tête. De ce fait, les libellules ont des difficultés pour marcher.

Chaque patte est formée de cinq segments liés par des joints : le trochanter, la coxa (la hanche), le fémur (la cuisse), le tibia (la jambe), le tarse (figure 2.4). Le tarse comprend lui-même plusieurs segments, appelés pré-tarses. Ils se terminent par deux griffes. Sur les différentes séquences, ces éléments semblent liés entre eux et avec le thorax grâce à des liaisons pivot, avec un seul degré de liberté de rotation par rapport à l'axe de l'articulation.

2.3.1.4 L'abdomen

L'abdomen se compose de 10 segments complets et distincts. Un onzième et même douzième segments peuvent être distingués. Ces segments sont semblables à un cylindre étroit (figure 2.4). L'abdomen semble lié au thorax grâce à une liaison pivot semblable à celle liant les différentes parties des pattes.

2.3.1.5 Les ailes

Les libellules ont deux paires d'ailes. Leurs ailes sont attachées au ptérothorax. Elles sont presque transparentes et donc difficiles à suivre dans les séquences d'images. Elles sont munies de nervures qui constituent le support de la membrane transparente. Selon la disposition et les formes de ces nervures, nous pouvons distinguer quatre familles d'anisoptères. Les nervures importantes à connaître sont : la sous-costale, la radiale, le secteur de la radiale et l'anale ; sur le bord antérieur est situé un point appelé Nodus, un ptérostigma coloré à l'apex ; une petite nervure transverse à la base ou arculus et encore un triangle ailaire important (figure 2.5).

Pour l'erythemis simplicicollis, espèce observée dans les vidéos dont nous disposons, la taille totale varie entre 36 et 48 mm. La longueur de l'abdomen varie entre 24 et 30 mm et celle des ailes entre 30 et 34 mm. La majorité des libellules vivant dans les zones tempérées ont une envergure de 5 à 8 cm, mais certaines espèces tropicales peuvent atteindre 20 cm.

2.3.2 Etude Dynamique

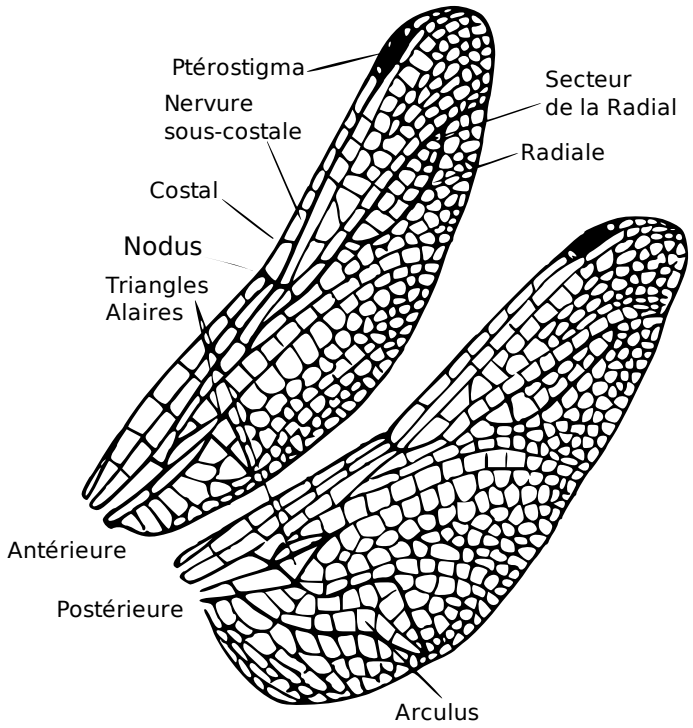


FIGURE 2.5 : Les ailes d'une libellule

2.3.2.1 Introduction

Une fois le modèle articulé est choisi, il suffit d'appliquer la deuxième loi de Newton sur chacune des i liaisons :

$$M_i\vec{\ddot{X}}_i = \vec{F_{ex}} + \vec{F_m} + \vec{F_c} \ , \tag{2.1}$$

et

$$I_i\vec{\ddot{\theta}} = M_i/g(\vec{F_{ex}}) + M_i/g(\vec{F_m}) + M_i/g(\vec{F_c}) \ , \tag{2.2}$$

où M_i est la masse de l'élément i, $\ddot{X}_i$ l'accélération du centre de gravité g_i de cet élément, $\vec{F_{ex}}$ la résultante des forces extérieures aérodynamiques, $\vec{F_m}$ la résultante des forces motrices générées par les muscles, $\vec{F_c}$ la résultante des contraintes exercées par les liaisons entre les articulations, I_i le moment d'inertie de l'élément i, et enfin $\vec{\ddot{\theta}}$ l'accélération angulaire par rapport au centre de gravité g_i.

Le deuxième membre de l'équation 2.2 correspond à la somme des moments résultants des forces extérieures, motrices et de contraintes.

Dans la littérature, nous avons trouvé une étude similaire qui visait à établir

un modèle biomécanique de la salamandre [C2]. Cependant, les forces extérieures terrestres et aquatiques sont très différentes de celles qui s'appliquent sur le corps des libellules (forces aérodynamiques). D'autre part, plusieurs hypothèses simplificatrices ont été formulées dans [C2] concernant les forces aquatiques. On peut remarquer que pour déterminer les forces exercées par les muscles, ceux-ci ont été modélisés par des ressorts et des amortisseurs. Chaque paire de muscles de flexion et d'extension exerce un moment sur le centre de gravité de l'élément correspondant. C'est l'activité des motoneurones (M_f et M_e) qui détermine le moment moteur exercé sur les objets :

$$M = \alpha(M_f - M_e) + \beta(M_f + M_e + \gamma)\Delta\varphi + \delta\Delta\dot{\varphi} \tag{2.3}$$

où α, β, γ et δ sont respectivement le gain, la raideur, la raideur tonique et le coefficient d'amortissement du muscle. Pour le moment, ces informations ne sont pas à notre disposition pour les libellules.

2.3.2.2 Forces aérodynamiques

Pour la modélisation des forces aérodynamiques, deux possibilités se présentent : considérer le cas d'un état stable (steady aerodynamics) ou celui d'un état instable (unsteady aerodynamics). En considérant le cas de stabilité aérodynamique, les calculs sont relativement simples, comme cela a été indiqué dans l'article [C1] où un simulateur du vol des drosophiles est proposé. Par contre, les résultats ne sont pas précis, du fait que beaucoup d'hypothèses simplificatrices ont été formulées :

- sur la masse, le moment d'inertie et le centre de gravité. Une estimation de la masse, du moment d'inertie et du centre de gravité de chacun des éléments du modèle articulé est nécessaire. Pour la drosophile, le modèle articulé considéré est formé de trois parties composées de la tête, du thorax et de l'abdomen. Ensuite, à partir d'une collection d'images calibrées de la drosophile, un modèle est suggéré. Puis, une représentation polygonale du modèle est appliquée. Enfin, à partir de la représentation polygonale, en considérant une densité uniforme égale à celle de l'eau, la masse, le centre de gravité et le moment d'inertie sont calculés grâce à l'algorithme de Mirtich [A9].

Concernant ces paramètres, nous avons pu trouver dans la littérature les masses moyennes suivantes relatives à différentes familles et espèces de libellules :

Famille	Espèces	Masse du corps(g)	N
Aeshnidae	Anax junius	1.06 ± 0.12	10
	Aeshna umbrosa umbrosa	0.62 ± 0.06	3
Libellulidae	Plathemis lydia	0.45 ± 0.04	9
	Tramea lacerata	0.44 ± 0.05	6
	Libellula luctuosa	0.35 ± 0.04	11
	Erythemis simplicicollis	0.23 ± 0.03	8
	Sympetrum janae	0.12 ± 0.02	6
	Sympetrum vicinum	0.11 ± 0.007	2

Les valeurs représentées dans ce tableau sont des valeurs moyennes obtenues sur plusieurs mesures.

– sur les forces aérodynamiques : les forces aérodynamiques qu'ils entreprennent de calculer sont celles appliquées aux ailes et au corps de la drosophile. Ils considèrent le cas d'un état aérodynamique stable et négligent les interactions ailes-ailes et ailes-corps. Ce travail aboutit à des résultats qualitatifs reflétant bien le comportement de la drosophile pendant le vol. Quantitativement, les résultats ne sont pas du tout fiables. D'autre part, les libellules ont des ailes plus grandes que celles des drosophiles et opèrent moins de battements par minute. Les libellules peuvent faire plus d'acrobaties aériennes. De ce fait, les forces aérodynamiques résultant de ces battements sont plus complexes à modéliser. Plusieurs travaux ont été entrepris dont le but est de retrouver les caractéristiques aérodynamiques du corps et des ailes de la libellule [A17] [C3]. Ils ont abouti à la conclusion qu'une hypothèse d'état aérodynamique stable ne suffit pas à résoudre le problème. Étant donné que les théories relatives à l'aérodynamique instable ne sont pas assez développées pour parvenir à la résolution du problème, tous les travaux effectués jusqu'à aujourd'hui sont restés expérimentaux.

2.4 Prédicteurs de Taylor

Comme nous l'avons mentionné précédemment, le mouvement rapide des libellules implique un mouvement apparent parfois supérieur à 10 pixels. Ces grands déplacements représentent un défi pour les méthodes classiques d'estimation du mouvement, telles la méthode d'Horn et Schunck ou la méthode de Lucas Kanade. Nous proposons ici de résoudre ce problème grâce à une première approximation du vecteur mouvement 3D. Cette approximation est basée sur les vecteurs mouvements 3D relatifs aux instants antérieurs précédemment calculés. Pour ce faire, nous nous basons sur le développement de Taylor. Dans un premier temps, nous essayons de valider ce prédicteur sur le signal représentant les coordonnées dans l'image de l'appendice anal de la libellule. Ce signal est extrait manuellement sur les différentes images d'une séquence et noté $\mathcal{D}(t)$:

$$\mathcal{D}(t)^T = (x(t), y(t))^T. \tag{2.4}$$

Il est important de noter que cette extraction manuelle aboutit à un signal bruité (figure 2.9). Supposons que les dérivées jusqu'à certain ordre du signal représentant les coordonnées successives de l'appendice anal sont continues. Alors, à un instant t donné, nous pouvons utiliser les valeurs du signal à des instants antérieurs à t sur un intervalle $[t-T, t]$ afin d'estimer les dérivées successives $\mathcal{D}^n(t)$, où n représente l'ordre de différenciation. Grâce au développement de Taylor, nous pouvons prédire les coordonnées de l'appendice anal sur un intervalle de temps fini $[t, t+\delta]$ comme suit :

$$\tilde{\mathcal{D}}(t+\delta) = \mathcal{D}(t) + \dot{\mathcal{D}}(t)\delta + \ddot{\mathcal{D}}(t)\frac{\delta^2}{2} + \cdots + \mathcal{D}^{(n)}(t)\frac{\delta^n}{n!}. \tag{2.5}$$

Notons que lorsque δ augmente, la prédiction devient moins précise. L'apport de cette approche réside dans la méthode d'estimation des dérivées. En effet, l'estimation numérique des dérivées d'un signal temporel bruité est un problème ancien, mal posé, qui a longtemps attiré l'attention vu son importance dans les domaines de l'ingénierie et des mathématiques appliquées. Des estimations rapides et ro-

bustes sont désormais possibles grâce à une approche algébrique initiée dans [A5], et adaptée à la dérivation de signaux dans [A4, A10]. Notre estimation est basée sur les travaux décrits dans [A10].

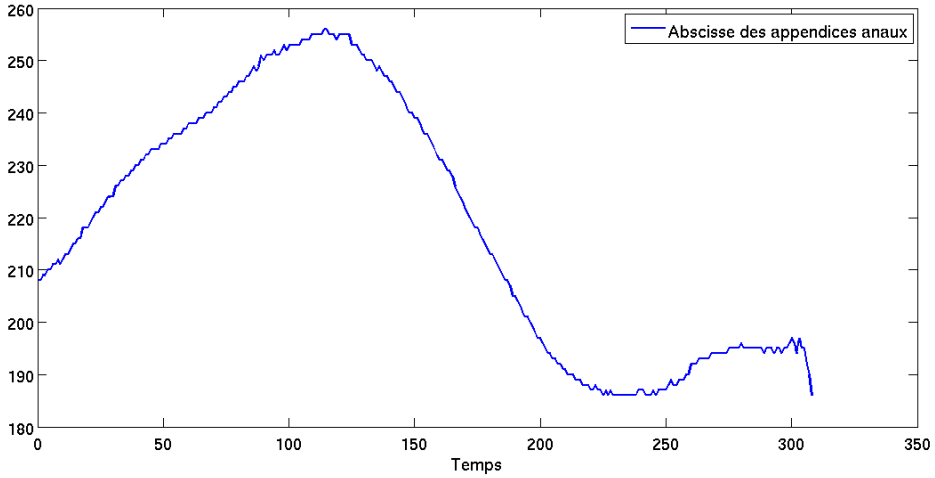


FIGURE 2.6 : Abscisses de l'appendice anal en fonction du temps.

Dans le cas où nous cherchons tout d'abord à prédire l'abscisse de l'appendice anal présenté sur la figure 2.6, nous nous basons sur les estimateurs suivants pour l'estimation des dérivées première et seconde de ce signal :

$$\dot{x}(0) = -\frac{12}{T}\int_0^1 (3 - 16\tau + 15\tau^2)\mathcal{D}(T\tau)d\tau \tag{2.6}$$

$$\ddot{x}(0) = \frac{60}{T^2}\int_0^1 (8 - 90\tau + 216\tau^2 - 140\tau^3)\mathcal{D}(T\tau)d\tau \tag{2.7}$$

Nous pouvons voir sur les figures 2.7 et 2.8 respectivement les dérivées première et seconde estimées à partir des mesures initiales.

Les prédictions de l'abscisse $x(t)$ pour des instants postérieurs à t tels que $\delta = 2,\ 4,\ 5$ sont présentées sur la figure 2.9. Nous pouvons remarquer que la précision de l'estimation se dégrade avec l'augmentation de δ. Les prédictions commencent à l'image 40 de la séquence puisque les estimations de dérivées sont réalisées en utilisant 40 échantillons. Cependant, ces résultats peuvent être améliorés grâce à des travaux plus récents [A11].

Enfin, notons que notre prédicteur est robuste au bruit de mesure, il est facile

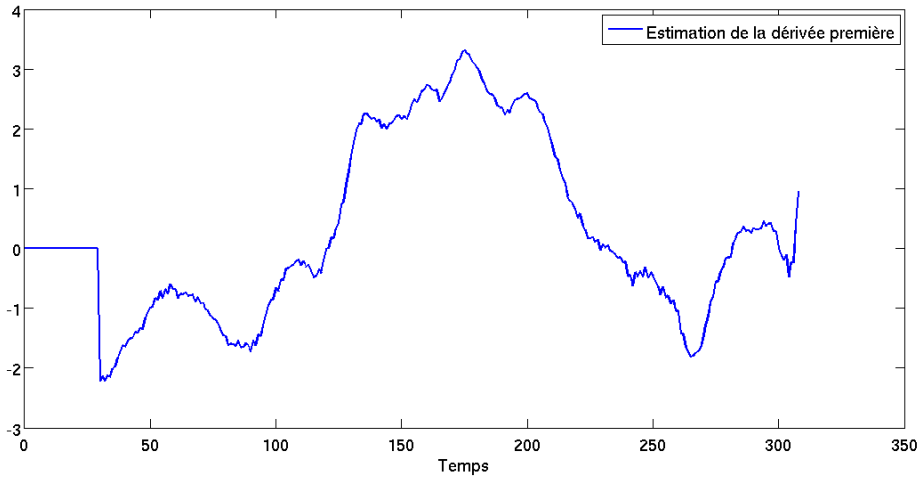


FIGURE 2.7 : Estimation de la dérivée première.

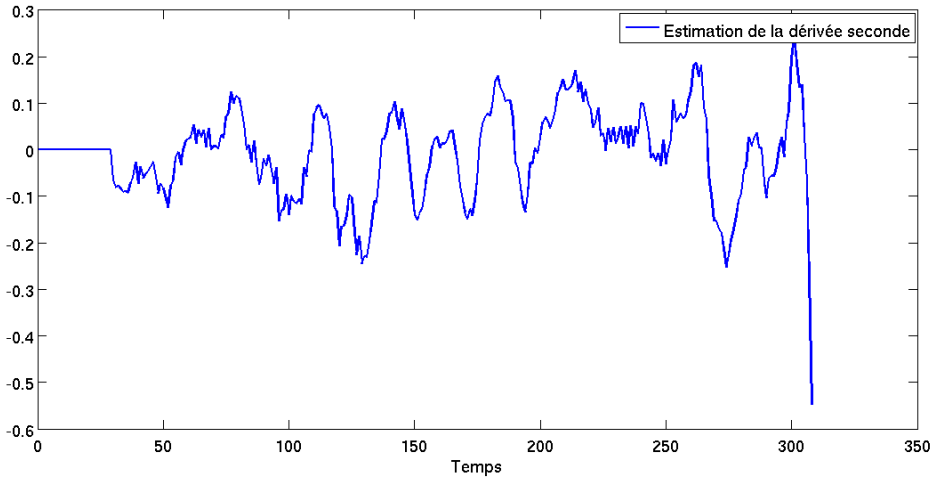


FIGURE 2.8 : Estimation de la dérivée seconde.

à implémenter et de complexité algorithmique faible. La méthode que nous avons présentée ici peut-être appliquée aux 6 paramètres du mouvement 3D d'un élément du modèle de l'insecte. Cependant, il est important de noter que le signal est considéré analytique ce qui constitue une hypothèse forte.

2.5 Conclusion

Outre les prédictions de Taylor et la possibilité d'utiliser les modèles dynamique et cinématique comme prédicteurs, nous avons également envisagé d'utiliser des

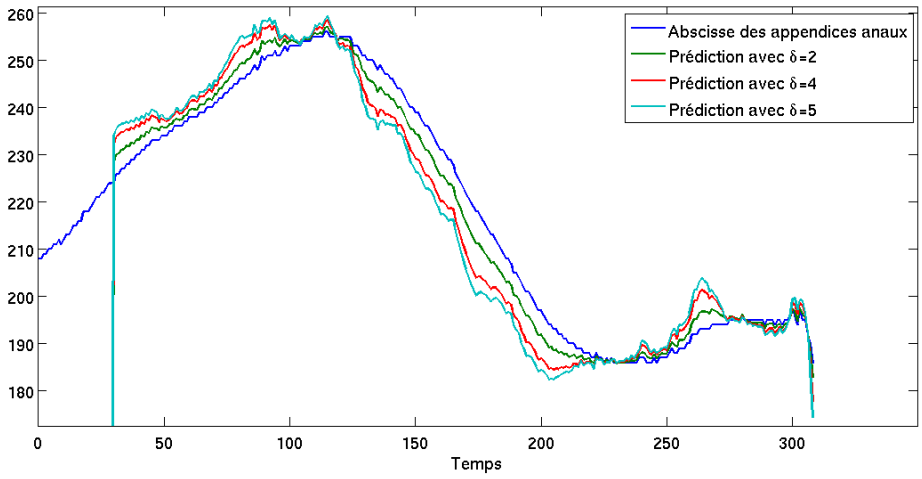


FIGURE 2.9 : Prédiction de l'abscisse des appendices anaux pour $\delta = 2,\ 4,\ 5$.

prédicteurs basés sur une étude statistique du comportement prédateur de la libellule. Cette approche n'a pas encore été examinée. Une autre possibilité consisterait à introduire comme prédicteur le modèle issu des hypothèses des biologistes. S'il s'avère efficace pour l'analyse automatique du mouvement, on aboutirait simultanément à une validation du modèle et à une méthode originale d'analyse du mouvement d'un insecte.

Nous avons présenté dans ce chapitre des analyses préliminaires au suivi de tous les membres de la libellule dans une séquence d'images. Vue la complexité du problème et l'importance qu'accordent les biologistes à l'estimation du mouvement 3D de la tête de la libellule avant le décollage, nous avons décidé au cours de la thèse de nous focaliser sur cet unique problème. Ainsi, vue la forme géométrique quasi sphérique de la tête de libellule, nous avons sélectionné ce modèle géométrique simple dans nos travaux.

Reste à modéliser la texture intégrant les propriétés spéculaire et diffuse de la surface et la modélisation du phénomène de pseudopupille. La modélisation des propriétés spéculaire et diffuse de la surface est un sujet abondamment relaté dans la littérature. Cependant, la modélisation du phénomène de pseudopupille est quant à elle plus problématique. Nous abordons ces problèmes dans le chapitre suivant.

Chapitre 3

Modélisation de la tête de libellule

3.1 Introduction

Proposer un modèle le plus précis possible de la tête suivie dans la séquence représente une étape cruciale pour la réussite de la méthode décrite dans notre thèse. Une fois cette étape réussie, nous pourrons espérer aboutir à une reconstruction assez précise de son mouvement 3D.

Modéliser la tête de libellule revient donc à extraire le plus d'informations 3D possibles de plusieurs images 2D de cet objet. Ces informations portent sur sa géométrie et sur sa texture.

Dans ce chapitre, nous commençons par présenter quelques méthodes utilisées dans la littérature pour la reconstruction géométrique, puis concluons sur les avantages de la méthode que nous préconisons. Elle consiste en une méthode d'intersection de cônes. Les silhouettes sont extraites de chacune des images de la libellule. Chacune de ces silhouettes représente la base d'un cône généralisé dont le sommet n'est autre que la position de la caméra dans un repère fixe de position connue.

L'intersection de tous les cônes relatifs aux différentes images montrant l'objet à modéliser sous différents angles de vue représente le modèle géométrique recherché. La précision du modèle augmente avec le nombre d'images et d'angles de vue de l'objet à modéliser. Outre le nombre d'images et les angles de vue de l'objet, deux autres paramètres sont importants pour le succès de cette méthode de modélisation, à savoir le calibrage de la caméra ainsi que l'extraction des points de contour. Une partie de ce chapitre est donc consacrée à proposer une méthode pour l'estimation de ces paramètres.

3.2 Méthodes passives de reconstruction

Dans la littérature, on peut retrouver plusieurs méthodes actives pour la reconstruction 3D. Certaines se basent sur l'utilisation d'outils coûteux comme des scanners. Ces méthodes aboutissent en général à des résultats de très bonne qualité et sont plus performantes que les méthodes passives utilisant par exemple des caméras.

Par contre, les méthodes actives requièrent un temps d'acquisition élevé et parfois, une modification de la scène s'avère indispensable (par exemple, peindre la surface des objets).

Vu les inconvénients liés à ces méthodes, nous avons eu recours aux méthodes passives. Les méthodes passives de reconstruction de la scène sont très nombreuses. Elles ont en commun la construction d'un modèle à partir de plusieurs images. Mais, selon les conditions d'acquisition de ces images une méthode peut plus ou moins convenir et aboutir au modèle le plus précis. Nous détaillons ces méthodes dans ce qui suit.

3.2.1 Méthodes de reconstruction stéréo

Ces méthodes s'inspirent de la vision humaine. L'homme, grâce à ses deux yeux peut retrouver l'information de profondeur des objets qu'il observe. La stéréovision permet donc la détermination de la position tridimensionnelle de points matériels d'une scène à partir de deux images ou plus prises au même moment mais depuis des points de vue légèrement différents.

Les techniques de stéréovision consistent à retrouver la profondeur à partir des disparités (figure 3.1). Étant données deux images ou plus, il faut résoudre deux problèmes :

1. Le problème de correspondance : Ce problème est connu sous le nom d'appariement stéréoscopique. Pour un point m dans la première image, il s'agit de retrouver le point m' qui lui correspond dans l'autre image, sachant que ces deux points représentent les projections du même point M de l'espace 3D. Plusieurs problèmes se posent durant l'étape de la mise en correspon-

dance tels les occultations, la spécularité de la surface analysée ainsi que son homogénéité. Il n'existe donc pas une méthode générale pour la mise en correspondance stéréoscopique. Cependant, plusieurs contraintes (ex : géométrie épipolaire) et suppositions (ex : intensité lumineuse supposée constante) sont exploitées afin de résoudre le problème.

2. Le problème de reconstruction : Ce problème consiste à partir de deux points m et m' à retrouver le point M de la scène dont ils sont la projection.

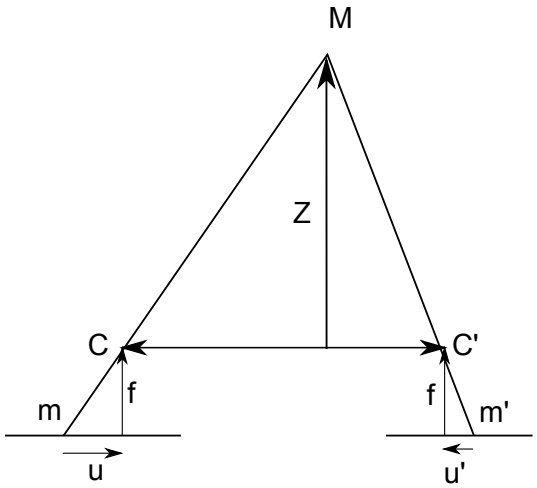
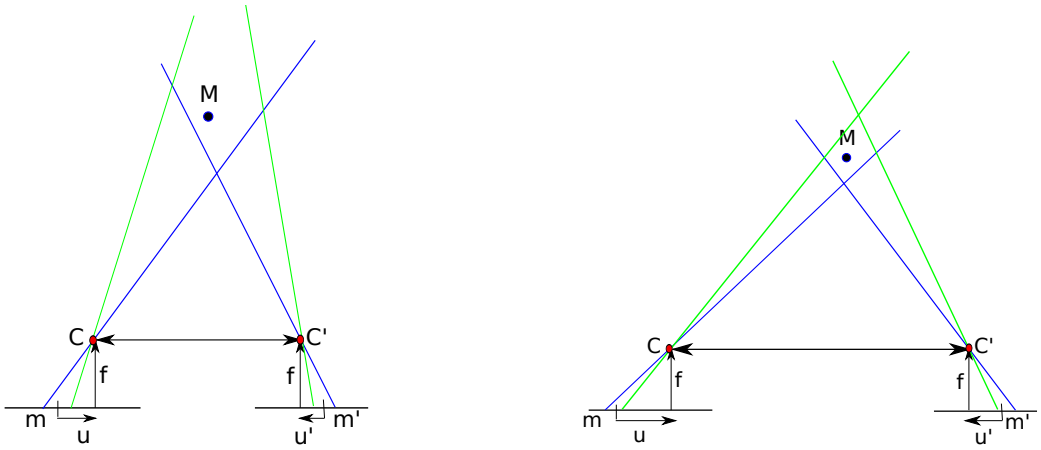


FIGURE 3.1 : Disparité$= u - u' = baseline * f/z$

Le choix de la ligne de base est d'une grande importance. Une petite ligne de base (figure 3.2(a)) aboutit à des erreurs plus importantes sur la profondeur, laquelle est estimée durant la phase de reconstruction. D'autre part, une grande ligne de base (figure 3.2(b)) aboutit à des difficultés pour la mise en correspondance et peut conduire à des estimations incorrectes.

Ce domaine a évolué de façon significative depuis quelques années, surtout par rapport aux méthodes de mise en correspondance. De plus, d'autres méthodes globales sont apparues telles la programmation dynamique [A12], les graph-cuts [A2] et les propagations de croyance [A19].

Une solution envisagée consiste à utiliser plusieurs lignes de base générées par des déplacements latéraux de la caméra. Citons par exemple les travaux d'Okutomi et Kanade [A13] qui ont travaillé sur une fonction SSD appliquée à l'inverse de la distance (au lieu de l'image de disparité). L'avantage de leur méthode est qu'elle



(a) Petite ligne de base (b) Grande ligne de base

FIGURE 3.2 : Choix de la ligne de base

permet d'éliminer les erreurs de correspondance dûes aux grandes ligne de base, tout en augmentant la précision de la phase de reconstruction.

3.2.2 Méthodes de reconstruction volumétrique à partir de plusieurs vues

Grâce à ces méthodes la reconstruction de la scène se fait directement dans un espace 3D. Le but est de partir de plusieurs images de l'objet prises selon différents angles de vue et d'en déduire un volume V exprimé dans un repère fixe. Ce volume doit être consistant avec les images d'entrée et contenir l'objet réel observé sur ces images.

Pour cela, en partant du volume initial contenant l'objet observé, deux possibilités se présentent : 1) créer l'ensemble des cônes et calculer leur intersection afin de retrouver l'enveloppe convexe représentant le modèle géométrique ; 2) commencer par discrétiser le volume V en un ensemble de petits cubes appelés $voxels$ pour ensuite projeter ces voxels sur les images et tester leur appartenance aux silhouettes.

Il est important de noter que le problème d'occultation qui se pose dans les méthodes de reconstruction stéréoscopique est évité grâce à ces méthodes, du fait qu'elles exploitent un nombre important d'images acquises sous des orientations variées. Quelques méthodes de reconstruction géométrique sont détaillées par la suite. Pour plus de détails, le lecteur peut se référer au chapitre 16, *"Volumetric Scene Reconstruction from Multiple View"*, du livre *"Foundations of Image Understanding"* écrit par Charles Dyer [L1].

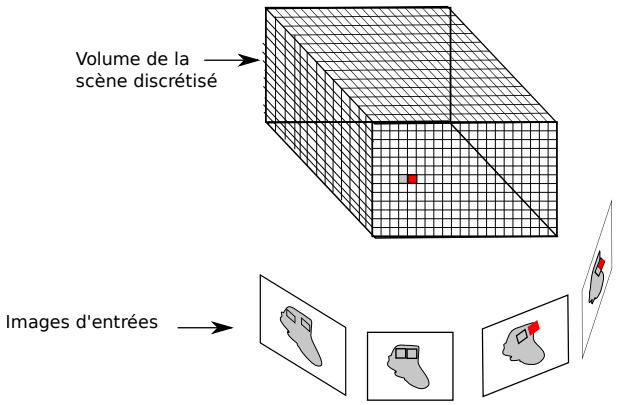


FIGURE 3.3 : Volume V discrétisé en voxels.

3.2.2.1 Reconstruction à l'aide de voxels

Le volume V recherché peut être représenté de différentes façons. L'une d'entre elles consiste à le discrétiser sous la forme d'un ensemble de voxels. Le problème revient ensuite à déterminer si chacun de ces voxels est $opaque$ ou $transparent$ en se référant à la projection de ce volume dans les différentes images disponibles. Dans certains cas, le problème revient à déterminer un degré d'opacité et le résultat n'est plus binaire pour chaque voxel.

Une méthode consiste à commencer par des voxels de grande taille. Les voxels qui ont leurs projections relatives aux différents centres optiques des différentes images d'entrée à l'intérieur de toutes les silhouettes sont considérés opaques. Ceux dont les projections tombent toutes à l'extérieur des différentes silhouettes sont considérés transparents. Enfin, ceux dont les projections intersectent simultanément les silhouettes et le fond sont subdivisés en sous-voxels, lesquels sont ensuite étudiés de la même façon.

3.2.2.1.1 Coloriage de voxels Cette méthode vient compléter la méthode décrite auparavant en ajoutant une texture à la forme obtenue. Elle consiste à attribuer des couleurs aux voxels de l'espace 3D afin de maximiser la ressemblance entre le volume 3D formé par les voxels et sa projection sur l'ensemble des images d'entrée. La projection de cet espace 3D par rapport aux différents centres de projection relatifs aux différentes prises de vue doit correspondre à chacune des images. L'un des problèmes de cette méthode réside dans le fait que plusieurs colo-

riages des voxels peuvent être cohérents avec l'ensemble des images. Ainsi, le défi est de retrouver l'ensemble de tous les modèles photo-cohérents.

L'autre problème est de pouvoir proposer un test de visibilité, par exemple : *si la ligne joignant le centre du voxel x et le centre optique d'une caméra intersecte le voxel y, le voxel x est situé dans une couche plus éloignée de la caméra que le voxel y* (*cf.* figure 3.4).

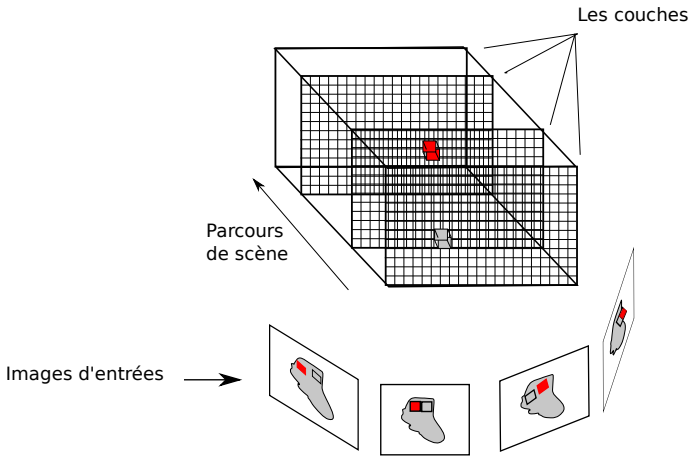


FIGURE 3.4 : Ordre de profondeur

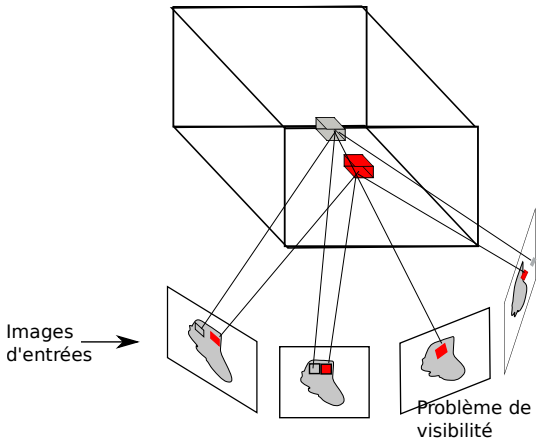


FIGURE 3.5 : Problème de visibilité

Après discrétisation de la scène par ordre de profondeur, indépendamment de l'ordre des prises de vue, l'affectation des couleurs est opérée selon les étapes suivantes (*cf.* figure 3.5) :

1. choisir un voxel ;

2. le projeter ;
3. le colorier s'il est photo-cohérent, c.à.d. si l'écart type des couleurs des pixels projetés et réels est inférieur à un certain seuil.

L'algorithme identifie donc un certain nombre de voxels invariants après plusieurs itérations, lesquels forment un modèle géométrique et photométrique consistant par rapport aux images.

Steven M. Seitz et Charles R. Dyer [A18] ont mis en place un algorithme balayant une seule fois les voxels. Chaque voxel passe un test de photo-consistance qui détermine son opacité (appartenance à l'objet observé) ou sa transparence (appartenance à l'environnement). L'image de la figure 3.6 présente un résultat de cet algorithme.

FIGURE 3.6 : Reconstruction par coloriage de voxels [A18]

3.2.2.1.2 Space carving Parfois, un algorithme balayant les voxels en une seule passe et aboutissant à un modèle photo-consistant par rapport à toutes les prises de vue est difficile à mettre en place. Une autre approche consiste à balayer l'espace suivant des plans d'orientations variables. Chaque voxel est donc visité plusieurs fois et à chaque fois le test de photo-consistance est appliqué. A chaque passage, le résultat du test par rapport à un voxel peut varier, entraînant de ce fait une variation du résultat pour d'autres voxels. Plusieurs balayages sont donc nécessaires jusqu'à ce qu'un équilibre soit atteint. Le volume initial convergera ainsi vers le volume réel. Enfin, l'union de toutes les scènes photo-consistantes représente l'en-

veloppe recherchée. L'inconvénient réside dans la complexité de l'algorithme de base et plus précisément dans la procédure de mise à jour de l'enveloppe.

Kutulakos et Seitz [A7] ont prouvé que grâce à cette méthode les voxels du bord seront éliminés successivement jusqu'à ce qu'il n y ait plus de voxels non photo-consistants et que la forme restante représente l'enveloppe convexe de l'objet observé. Afin de limiter le nombre de plans balayés, un ensemble de plans est fixé en choisissant l'orientation. Un exemple d'orientation des plans consiste à ce qu'ils soient considérés parallèles aux bords du cube définissant le volume initial. Pour garantir l'aboutissement à l'enveloppe recherchée à partir de l'ensemble des plans fixés au départ, il suffit d'appliquer le test de photo-consistance par rapport à tous les points de vue à la fin de chaque itération de balayage. Ainsi, d'autres pixels du bord seront retirés et l'enveloppe intermédiaire convergera vers l'enveloppe recherchée.

La figure 3.6 montre un modèle reconstruit à partir de 100 images à l'issue de six balayages pour chaque itération.

FIGURE 3.7 : Reconstruction par Space Carving [A7]

3.2.2.2 Reconstruction à partir de silhouettes

Une silhouette est généralement représentée par une image binaire. Chaque pixel de cette image représente donc un pixel de l'objet ou un pixel du fond. Cette image binaire est obtenue par segmentation ou par d'autres méthodes d'extraction du fond de l'image.

Une étape importante de cette méthode est le calibrage de la caméra et l'estimation de pose, étape détaillée à la fin de ce chapitre. Après rassemblement de ces informations, il suffit de suivre la demi-droite de projection du centre optique vers tous les points de la silhouette. Ces demi-droites forment un cône dans lequel

l'objet observé est inclus (figure 3.8). L'intersection de tous les cônes formés par les différentes images représente une approximation de l'objet observé. Dans le cas idéal où le nombre de prises de vue est infini, et où la segmentation de l'image ainsi que le calibrage de la caméra (intrinsèque et extrinsèque) sont précis, l'intersection aboutit à une représentation exacte de l'objet observé (si celui-ci ne contient pas de surfaces concaves).

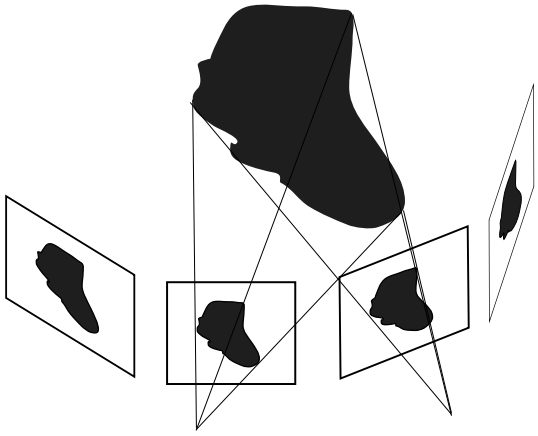


FIGURE 3.8 : Projection des silhouettes

En réalité, seulement un nombre fini d'images est à notre disposition. L'intersection des cônes aboutit donc à une enveloppe convexe contenant l'objet réel.

La reconstruction à partir de silhouettes n'est pas forcément une reconstruction volumétrique, mais peut également être une reconstruction surfacique se basant sur les contours et l'enveloppe représentant l'objet observé. La représentation volumétrique peut-être basée sur des voxels ou consister en une intersection volumétrique directe en 3D (figure 3.9).

Les méthodes de reconstruction 3D à partir de silhouettes sont les plus populaires, vu la facilité relative de l'extraction des points de contour sous éclairage et poses contrôlés. De plus, l'implémentation est facile en comparaison à d'autres méthodes de reconstruction. Ce concept a été introduit par Baumgart en 1974 dans sa thèse de Doctorat. Depuis, cette méthode a été largement améliorée [A3, C4, R1].

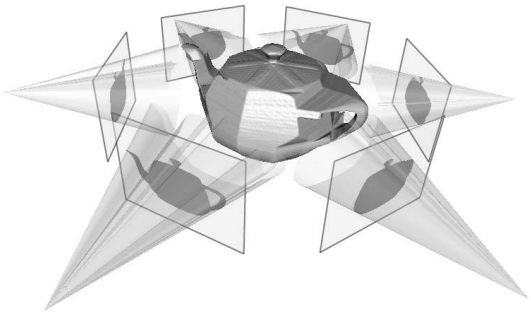


FIGURE 3.9 : Intersection de cônes (référence [R1]

3.3 Conclusion

Dans cette section, nous avons présenté quelques méthodes de construction du modèle géométrique d'un objet 3D, en distinguant les méthodes passives des méthodes actives. Parmi les méthodes passives, nous distinguons les méthodes de reconstruction stéréoscopique et les méthodes de reconstruction volumétrique à partir de voxels ou d'intersection de cônes et de surfaces.

Parmi toutes ces méthodes, nous avons testé celle qui consiste à déterminer l'intersection de cônes. La validation a été menée sur des images synthétiques. Nous présentons sur la figure 3.10(a) une des images de l'objet synthétique considéré. Sur la figure 3.10(b), nous présentons le modèle géométrique reconstruit à partir de 4 vues de cet objet et sur la figure 3.10(c), le modèle reconstruit à partir de 25 prises de vue réparties uniformément autour de son axe de symétrie. La texture n'est pas analysée pour l'instant.

Une étape de prétraitement des images est indispensable afin d'extraire les contours, autorisant une reconstruction précise des cônes. L'extraction des contours nous fournit un polygone dont l'orientation est connue par rapport à un repère fixe. L'intersection est calculée à partir des cônes dont ces polygones constituent les bases.

Pour l'exemple présenté dans la figure 3.10, les positions de la caméra pour toutes les vues sont connues avec précision, du fait que la scène est synthétique. En pratique, sur les images de la tête de libellule dont nous disposions, dont quelques unes sont présentées dans la figure 3.11, il faut en premier lieu estimer la position

FIGURE 3.10 : Modélisation géométrique par intersection de cônes : (a) Image représentant une vue de l'objet à modéliser, (b) Modèle obtenu à partir de 4 angles de vue différents, (c) Modèle obtenu à partir de 25 angles de vue différents.

et l'orientation de la caméra par rapport à un repère fixe, et ce pour chaque image avant de déterminer les cônes et enfin le modèle. La technique d'estimation de pose que nous avons utilisée est décrite dans la section suivante.

3.4 Estimation de pose

Pour l'estimation de pose, nous avons implémenté la méthode de Helder *et coll.* [R2], que nous détaillons dans ce qui suit. Nous supposons que les paramètres intrinsèques de la caméra sont connus avec précision. Partant de cette hypothèse, et à partir d'une image d'une grille régulière, cette méthode permet de déduire la pose de la caméra. Helder *et coll.* se basent sur l'algorithme de Lowe [A8] et Ishii *et coll.* [A6], y apportant toutefois quelques améliorations. L'algorithme proposé fournit un résultat précis avec un nombre limité d'itérations.

Étant donné un objet 3D et son image par rapport à une certaine position de la caméra, Lowe considère que l'image est obtenue par une simple transformation projective. Cette méthode peut donc être utilisée afin de retrouver la pose (position et orientation) de la caméra par rapport au repère réel fixe considéré. Outre la pose,

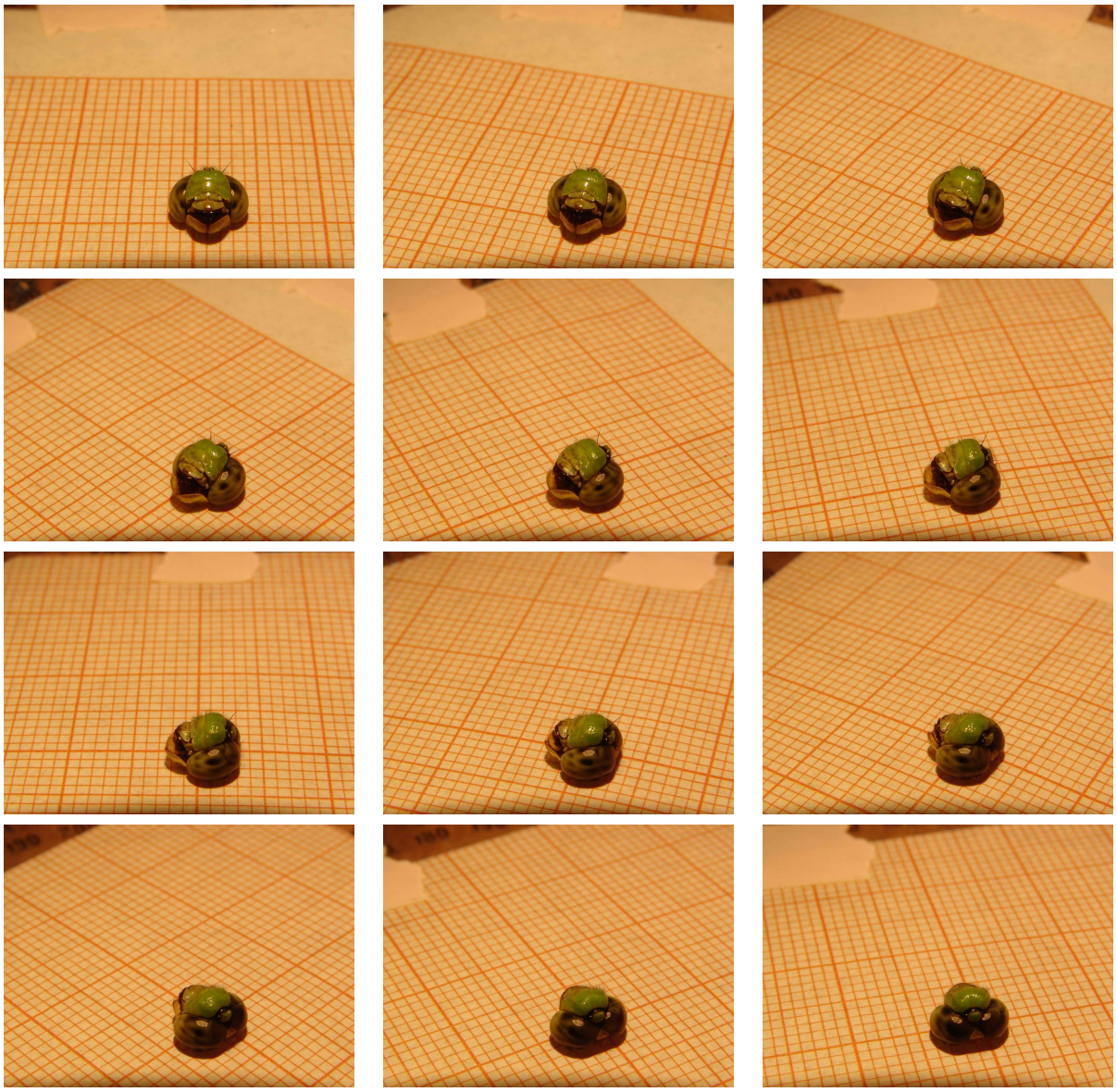


FIGURE 3.11 : Quelques images dont nous disposons pour la modélisation de la tête de la libellule.

cet algorithme peut servir à estimer la distance focale de la caméra.

L'algorithme se base sur une linéarisation du problème via l'utilisation de la méthode de Newton. Au lieu de chercher directement la solution s du système non-linéaire d'équations, la méthode de Newton consiste à calculer les valeurs d'un vecteur de correction γ à retrancher du vecteur s^i, estimation de s à l'itération i, afin de calculer l'estimation s^{i+1} à l'itération suivante :

$$s^{i+1} = s^i - \gamma. \tag{3.1}$$

Soit e le vecteur représentant les erreurs de mesure entre les composantes du modèle et celles de l'image. Le but est de retrouver le vecteur de correction γ qui élimine ces erreurs : $J\gamma = e$ où : $J_{ij} = \frac{\partial e_i}{\partial x_j}$.

Dans le cas de l'estimation de pose, les paramètres à retrouver sont la position et l'orientation de la caméra par rapport à un repère fixe. Il suffit d'initialiser la position et l'orientation, et d'itérer jusqu'à convergence vers une solution représentant la position et l'orientation de la caméra permettant de minimiser l'erreur e.

Pour ce faire, nous fixons un repère réel 3D. Nous représentons un point p dans ce repère. La projection de ce point p, obtenue par transformation de ses coordonnées du repère réel au repère de la caméra est comparée aux coordonnées de son image dans le même repère caméra. Cette comparaison nous indique les erreurs suivant l'axe horizontal et vertical de la caméra. Chaque point nous procure donc deux équations relatives à l'erreur. Pour estimer les six paramètres de rotation et de translation, nous avons besoin de connaître au moins trois points non alignés de l'objet.

3.5 Conclusion

Dans ce chapitre, nous avons introduit les méthodes qu'il s'agit de mettre en oeuvre pour construire le modèle géométrique 3D de la tête de libellule. A partir de plusieurs vues de cet objet, une technique d'intersection de cônes généralisés (dont les bases sont des polygones) fournit l'enveloppe convexe du modèle 3D. Auparavant, une méthode d'estimation de pose permet d'estimer la position et l'orientation de la caméra lors de l'acquisition de ces vues.

Bibliographie

Livres et chapitres de livres

[L1] L. S. Davis. *Foundations of Image Understanding*, volume 628 of *The International Series in Engineering and Computer Science*, chapter 16. Volumetric Scene Reconstruction from Multiple Views, C. Dyer, pages 469–489. Kluwer Academic Publishers, Boston, 2001.

[L2] R. J. Tillyard. Cambridge University Press, Cambridge, 1917.

Articles dans des revues

[A1] F. Anchling. La vision chez l'abeille. *Abeille de France*, 900, 2004.

[A2] Y. Boykov, O. Veksler et R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1222–1239, 2001.

[A3] K. Cheung, S. Baker et T. Kanade. Shape-from-silhouette across time part i : Theory and algorithms. *International Journal of Computer Vision*, 62(3) :221 – 247, Mai 2005.

[A4] M. Fliess, C. Join, M. Mboup et H. Sira-Ramírez. Compression differentielle de transitoires bruitï£¡s. *C.R. Acad. Sci. Paris*, Ser. I, vol.339 :821–826, 2004.

[A5] M. Fliess et H. Sira-Ramírez. An algebraic framework for linear identification. *ESAIM : COCV*, 9 :151–168, 2003. Available at http ://hal.inria.fr/inria-00188435.

[A6] M. Ishii, S. Sakane, M. Kakikura et Y. Mikami. A 3-d sensor system for teaching robot paths and environments. 6(2) :45–59, 1987.

[A7] K. N. Kutulakos et S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3) :307–314, 2000.

[A8] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3) :355–395, 1987.

[A9] B. Mirtich. Fast and accurate computation of polyhedral mass properties. *J. Graph. Tools*, 1(2) :31–50, 1996.

[A10] M. Mboup, C. Join et M. Fliess. A revised look at numerical differentiation with an application to nonlinear feedback control. 15^{th} *Mediterranean conference on control and automation*, june 27-29 2007. Available at http ://hal.inria.fr/inria-00142588.

[A11] M. Mboup, C. Join et M. Fliess. Numerical differentiation with annihilators in noisy environment. *Numerical Algorithm*, pages 1017–1398, 2008. Available at http ://hal.inria.fr/inria-00319240_v1.

[A12] Y. Ohta et T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on pattern analysis and machine intelligence*, 7(2) :139–154, 1985.

[A13] M. Okutomi et T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4) :353–363, Avril 1993.

[A14] R. M. Olberg, R. C. Seaman, M. I. Coats et A. F. Henry. Eye movements and target fixation during dragonfly prey-interception flights. *Journal of Comparative Physiology A*, Volume 193(7) :685–693, Juillet 2007.

[A15] R. M. Olberg, A. H. Worthington, J. L. Fox, C. E. Bessette et M. P. Loosemore. Prey size selection and distance estimation in foraging adult dragonflies. *Journal of Comparative Physiology A*, 191(9) :791–797, Septembre 2005.

[A16] R. M. Olberg, A. H. Worthington et K. R. Venator. Prey pursuit and interception in dragonflies. *Journal of Comparative Physiology A*, 186(2) :155–162, Février 2000.

[A17] M. Okamoto, K. Yasuda et A. Azuma. Aerodynamic characteristics of the wings and body of a dragonfly. *Journal of Experimental Biology*, 199(Issue 2) :281–294, Février 1996.

[A18] S. M. Seitz et C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 25(3), November 1999.

[A19] J. Sun, N. Zheng et H. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7) :787–800, 2003.

Communications

[C1] W. Dickson, A. Straw, C. Poelma et M. Dickinson. An integrative model of insect flight control. Dans AIAA, editor, *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, USA, Jan 2006.

[C2] A. Ijspeert. A 3-d biomechanical model of the salamander. Dans I. conference on virtual worlds No2, editor, *Lecture notes in computer science (Lect. notes comput. sci.) ISSN 0302-9743*, volume 1834, pages 225–234, Paris, Jul 2000. Springer, Berlin, ALLEMAGNE (1973) (Revue) Springer, Berlin, ALLEMAGNE (2000) (Monographie).

[C3] W. Lai, J. Yan, M. Motamed et S. Green. Force measurements on a scaled mechanical model of dragonfly in forward flight. Dans *12th*

International Conference on Advanced Robotics ICAR '05, pages 595–600, Jul 2005.

[C4] W. Matusik, C. Buehler, R. Raskar, S. Gortler et L. McMillan. Image-based visual hulls. Dans *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

Rapports techniques

[R1] C. H. Esteban et F. Schmitt. Using silhouette coherence for 3D image-based. Département Traitement du Signal et des Images 2003D011, Ecole Nationale Supérieure des Télécommunications, Paris, 2003.

[R2] A. Helder, R. L. Carceroni et C. M. Brown. A fully projective formulation for lowe's tracking algorithm. Rapport technique 641, Comp Sci Dept University of Rochester, Rochester, November 1996.