HAL
open science

Etude du mouvement dans les séquences d'images par analyse d'ondelettes et modélisation markovienne hiérarchique. Application à la détection d'obstacles dans un milieu routier

Cédric Demonceaux

▶ **To cite this version:**

Cédric Demonceaux. Etude du mouvement dans les séquences d'images par analyse d'ondelettes et modélisation markovienne hiérarchique. Application à la détection d'obstacles dans un milieu routier. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université de Picardie Jules Verne, 2004. Français. NNT : . tel-00862980

HAL Id: tel-00862980

https://theses.hal.science/tel-00862980

Submitted on 18 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée à l'Université de Picardie Jules Verne

U.F.R. Mathématiques et d'Informatique

Pour obtenir le titre de :

Docteur de l'Université de Picardie Jules Verne

Spécialité : Mathématiques appliquées et Traitement des images

par :

Cédric Demonceaux

Titre :

Etude du mouvement dans les séquences d'images par analyse d'ondelettes et modélisation markovienne hiérarchique. Application à la détection d'obstacles dans un milieu routier

---

Soutenue le 17 décembre 2004 devant le jury composé de :

M. Jean-Pierre COCQUEREZ, Professeur, UTC Compiègne	Président
M. Patrick BOUTHEMY, Directeur de Recherche, Inria Rennes	Rapporteur
M. Florian DE VUYST, Professeur, Ecole Centrale Paris	Rapporteur
M. Sylvain DURAND, Maître de Conférences, Université de Picardie / Ens Cachan	Examinateur
M. El Mustapha MOUADDIB, Professeur, Université de Picardie Jules Verne	Examinateur
M. Olivier GOUBET, Professeur, Université de Picardie Jules Verne	Directeur de Thèse
Mme Djemâa KACHI, Maître de conférences, Université de Picardie Jules Verne	Directeur de Thèse

Remerciements

Ce travail a été réalisé conjointement entre deux laboratoires de recherche amiénois, le C.R.E.A. E.A. 3299 (Centre de Robotique, d'Electrotechnique et d'Automatique) et le L.A.M.F.A. U.M.R. 6140 (Laboratoire Amiénois de Mathématique Fondamentale et Appliquée). Je tiens donc tout d'abord à remercier les deux directeurs de ces laboratoires, El Mustapha Mouaddib et Olivier Goubet de m'avoir accueilli dans leur équipe respective : Perception en Robotique et Analyse Appliquée.

Je remercie mes deux directeurs de thèse : Djemâa Kachi pour m'avoir proposé ce sujet très intéressant et pour la confiance qu'elle m'a accordée pendant ces trois années, Olivier Goubet pour ses nombreux conseils, sa disponibilité et d'avoir fait tout son possible pour que cette thèse se passe dans les meilleures conditions.

Je remercie Jean Pierre Cocquerez de m'avoir fait l'honneur de présider le jury de cette thèse. Florian De Vuyst et Patrick Bouthemy pour avoir accepté d'être rapporteur de ce mémoire. Je tiens à leur exprimer toute ma gratitude pour ce fastidieux travail et pour la pertinence de leurs remarques. Je remercie également Sylvain Durand et El Mustapha Mouaddib qui ont accepté de participer à ce jury.

Je remercie une fois de plus Patrick Bouthemy de m'avoir accueilli une semaine dans son équipe de recherche ainsi que l'ensemble des membres de son équipe pour l'accueil chaleureux qu'ils m'ont réservé.

Bien sûr, je n'oublie pas l'ensemble des collègues de l'équipe Perception en Robotique, tout particulièrement les thésards : Cyril Charron, Khaled Kaaniche, Asli-Gül Oncel et les anciens membres de l'équipe : Arnaud Dupuis, David Fofi et Josselin Harp. Je remercie aussi Annick Crépy et Pierre Détaille pour leur soutien et leur bonne humeur.

Enfin je remercie Sophie, dont les encouragements et le soutien moral constant furent si précieux lors de la rédaction et sans qui tout ceci n'aurait été possible. Une dernière pensée à Mathis qui a toujours réussi à me rendre le sourire même dans les moments difficiles.

Table des matières

Introduction Générale **9**

I Outils mathématiques **13**

1 Méthodes markoviennes en traitement d'images **15**

1.1 Modèles markoviens et problèmes inverses 15

1.1.1 Champs markoviens 15

1.1.2 Estimation bayésienne 17

1.2 Algorithmes d'optimisation sur les champs de Markov 19

1.2.1 Algorithmes stochastiques : Recuit simulé 19

1.2.2 Optimisation déterministe 20

1.3 Méthodes hiérarchiques 22

1.3.1 Méthodes de hiérarchies induites 22

1.3.2 Méthodes de hiérarchies explicites 24

1.4 Méthode hiérarchique semi-itérative 27

1.5 Conclusion 30

II Estimation et analyse du mouvement **33**

2 Estimation du mouvement par analyse d'ondelettes **35**

2.1 Positionnement du problème 36

2.2 Les méthodes d'estimation du flot optique 38

2.2.1 Méthodes de mise en correspondance 38

2.2.2 Méthodes fréquentielles 39

2.2.3 Méthodes différentielles 40

2.3 Modélisations paramétriques du mouvement 42

2.3.1 Modélisations polynomiales 42

2.3.2 Modélisation sur une base de fonctions 43

2.3.3 Estimation des paramètres 43
2.4 Estimation du mouvement dans l'image par analyse d'ondelettes 44
2.4.1 Estimation du flot optique par projection sur une base de fonctions 45
2.4.2 Estimation du mouvement d'une région de l'image 51
2.5 Résultats expérimentaux 57
2.5.1 Comportement de l'algorithme sur des images animées de mouvements synthétiques 57
2.5.2 Expérimentations sur des séquences d'images réelles 61
2.6 Conclusion 66

3 Détection de mouvement par une approche hiérarchique 67
3.1 Éléments d'état de l'art des méthodes de détection de mouvement 67
3.2 Détection de mouvement entre deux images 69
3.2.1 Modélisation hiérarchique 70
3.2.2 Mesure d'erreur d'estimation du mouvement 71
3.2.3 Terme d'attache aux données 76
3.3 Détection de mouvement dans une séquence d'images 79
3.3.1 Principe de la méthode 79
3.3.2 Modélisation temporelle-hiérarchique 80
3.4 Résultats expérimentaux 82
3.4.1 Séquences synthétiques 83
3.4.2 Séquences réelles 92
3.5 Conclusion 95

4 Segmentation d'images au sens du mouvement par multirésolution 97
4.1 Éléments de l'état de l'art 97
4.1.1 Les méthodes de segmentation séquentielle 98
4.1.2 Les méthodes de regroupement de région 98
4.1.3 Les méthodes d'estimation du mouvement et de segmentation conjointes 99
4.2 Segmentation mono-résolution du mouvement entre deux images 100
4.2.1 Formulation markovienne du problème de segmentation 101
4.2.2 Détection de nouvelles régions 102
4.3 Segmentation du mouvement par une approche multirésolution 104
4.3.1 Estimation du mouvement des régions par compensation 105
4.3.2 Segmentation par une formulation markovienne multi-modèles . . . 105
4.4 Initialisation à l'échelle grossière 107
4.5 Résultats 109

4.5.1 Séquence Taxi 109
4.5.2 Séquence Yosemite 110
4.5.3 Séquence "Mobile and Calendar" 113
4.5.4 Séquence "Flower Garden" 114
4.6 Conclusion 114

III Application à la détection d'obstacles dans un milieu routier 117

5 Les méthodes de détection d'obstacles dans un milieu routier 119
5.1 Introduction 119
5.2 Détection de la chaussée 120
5.2.1 Les méthodes basées sur la segmentation en régions 120
5.2.2 Les méthodes de délimitation de la route 121
5.3 Détection d'obstacles 122
5.3.1 Détection d'obstacles prédéfinis 122
5.3.2 Détection de tous types d'obstacles 124
5.4 Conclusion 126

6 Détection d'obstacles par une approche hiérarchique 127
6.1 Introduction 127
6.2 Estimation du mouvement apparent de la route 129
6.3 Détection d'obstacles 130
6.4 Estimation du mouvement des obstacles et prédiction de la route 131
6.5 Expérimentations 132
6.5.1 Séquence Van 132
6.5.2 Séquence Autoroute 133
6.5.3 Séquence Rond-Point 134
6.6 Conclusion 134

Conclusion et perspectives 141

IV Annexes 143

A Les modèles paramétriques du mouvement 145
A.1 Relation entre le mouvement réel et le mouvement dans l'image 145
A.2 Quelques exemples de modélisations paramétriques 147

B Les estimateurs robustes **149**
B.1 Estimateur des moindres carrés tronqués 149
B.2 Estimateur des moindres carrés médians 150
B.3 Les M-estimateurs 150

C Publications de l'auteur **153**

Introduction Générale

Cadre général et objectifs

L'une des ambitions de la vision par ordinateur est de permettre à un robot de percevoir et d'assimiler l'environnement dans lequel il se déplace. La communauté des chercheurs s'inspire la plupart du temps de la vision humaine en construisant des systèmes de vision stéréoscopique. Ces systèmes permettent effectivement de déterminer à chaque instant la profondeur de la scène à partir de la mise en correspondance des deux images générées par ceux-ci. Mais ces systèmes de vision stéréoscopique présentent un inconvénient freinant leur exploitation industrielle : leur coût élevé. Un autre système permet de contourner ce problème, c'est la vision monoculaire. Dans ce cas, la scène ne peut être interprétée que si les objets ou la caméra sont en mouvement. C'est l'étude de ce mouvement qui va être l'objet de ce mémoire.

Le cadre de cette thèse est l'analyse du mouvement dans des séquences d'images à partir d'une caméra mobile. Cette analyse regroupe trois niveaux différents d'interprétation de séquences d'images : l'estimation du mouvement, la détection des éléments mobiles et la segmentation au sens du mouvement. Le premier objectif est l'estimation du mouvement dans les images. Ce problème consiste à affecter à chaque pixel un vecteur qui correspond à la projection sur le plan image du déplacement du point 3D de la scène. Cette mesure fournit une information très importante sur la scène filmée. Johansson [Joh73] a, par exemple, montré qu'à partir de données purement dynamiques, nous sommes capables de reconnaître une scène ou une activité. Ce problème dit de "bas niveau" donne un renseignement indispensable pour des problèmes de plus "haut niveau" tels que la détection de mouvement et la segmentation au sens du mouvement qui sont les deux autres objectifs de cette thèse. La détection du mouvement vise à rechercher dans les images les objets mobiles dans la scène. Ce problème est relativement simple à résoudre dans le cas d'une caméra fixe puisqu'elle ne nécessite pas une estimation préalable du mouvement dans les images. Une analyse de la différence entre deux images consécutives couplée avec des méthodes robustes comme les champs de Markov ou la morphologie mathématique peuvent suffire. Par contre, dans le cas d'une caméra mobile qui est le cadre de ce travail, ce problème se révèle beaucoup plus difficile. Enfin, le troisième objectif de ce mémoire

est la segmentation au sens du mouvement qui consiste à décomposer les images en zones homogènes au sens du mouvement.

Ces trois champs d'étude (estimation/détection/segmentation) trouvent de nombreuses applications en vision par ordinateur. Citons par exemple, la compression de séquences d'images vidéo (norme MPEG-4), la reconstruction 3D de scènes, la reconnaissance d'objets, la télésurveillance, le suivi d'objets dans une scène ou encore la détection d'obstacles. Cette dernière fera l'objet d'une étude plus approfondie dans ce mémoire dans le domaine routier.

Contexte

Cette thèse a été en partie financée par le conseil régional de Picardie dans le cadre d'un projet DI.V.A. (DIagnostic et Véhicules Avancés) du pôle régional de recherche et de développement technologique. Ce projet intitulé "Perception coopérative inter-véhicules pour la détection d'imprévus dans un milieu routier" a débuté il y a trois ans. Son objectif est double, il consiste dans un premier temps à détecter tous les obstacles susceptibles d'entraver la voie de circulation d'un véhicule sur lequel est fixée une caméra. Ce système de vision monoculaire doit être capable de détecter de façon autonome tous les obstacles autour du véhicule puis d'avertir les véhicules suiveurs par GPS de la présence d'obstacles éventuels. L'autre partie consiste à faire coopérer les véhicules munis du même système de vision et d'obtenir plusieurs images au même instant d'une même scène et ainsi de rendre plus sûre la détection des obstacles. Ce projet se fait en collaboration entre deux laboratoires, le C.R.E.A. (Centre de Robotique, d'Electrotechnique et d'Automatique) de l'Université de Picardie Jules Verne, chargé de la partie vision, et le laboratoire Heudiasyc (Heuristique et Diagnostic des Systèmes Complexes) de l'Université de Technologie de Compiègne chargé, de la partie communication inter-véhicules.

Ce travail se situe dans la première partie de ce projet, à savoir, la détection d'obstacles dans un milieu routier à partir d'un système monoculaire. Nous avons comme objectif de détecter tout type d'obstacles sur tout type de routes (marquées ou non). Notre application devant être implantée en temps réel, nous nous devons de développer des algorithmes d'estimation et de détection rapides et robustes. Pour cela, nous avons développé :

- une nouvelle méthode d'estimation paramétrique du mouvement à partir d'une analyse d'ondelettes,
- une méthode de détection fondée sur une modélisation markovienne hiérarchique,
- une méthode d'estimation conjointe du mouvement et de la segmentation par une approche multirésolution.

Plan du mémoire

Bien que l'objectif final de cette thèse soit la détection d'obstacles dans un milieu routier par vision monoculaire, nous avons volontairement choisi de ne traiter ce problème qu'en dernière partie du mémoire. En effet, les méthodes proposées ici peuvent s'appliquer sur tous types de séquences d'images acquises à partir d'une caméra fixe ou mobile. Les séquences d'images routières ne sont qu'un cas particulier d'application où nous pourrons faire une hypothèse supplémentaire sur le mouvement dans les images.

L'ensemble de ce travail est fondé sur l'utilisation de deux outils mathématiques : les ondelettes et les méthodes markoviennes. Dans une première partie composée de deux chapitres, nous avons donc choisi de faire un rappel sur ces deux techniques qui sont indispensable à la compréhension de cette thèse. Le premier chapitre expose le cadre théorique de l'analyse en ondelettes. Dans un deuxième chapitre, nous discuterons des différentes méthodes markoviennes disponibles pour résoudre les problèmes issus de l'analyse d'images, et nous expliquerons pourquoi une modélisation markovienne hiérarchique est une solution intéressante pour notre problématique.

Dans la deuxième partie, nous introduirons nos solutions pour résoudre les problèmes d'estimation, de détection du mouvement et de segmentation au sens du mouvement. Cette partie est divisée en trois chapitres traitant consécutivement de ces trois problèmes. Dans le premier chapitre, nous verrons les différentes méthodes d'estimation du mouvement dans les séquences d'images, puis nous introduirons notre méthode d'estimation du mouvement paramétrique à partir d'une analyse en ondelettes de l'équation de contrainte du flot optique. Ensuite, dans un deuxième chapitre, nous exploiterons le caractère multirésolution de l'estimation du mouvement pour résoudre le problème de détection du mouvement grâce à une modélisation markovienne hiérarchique. Enfin, un troisième chapitre traitera du problème de la segmentation au sens du mouvement par une approche multirésolution. Chaque chapitre fera l'objet d'un état de l'art (sans malheureusement prétendre à l'exhaustivité) des différents problèmes traités.

Pour conclure, dans la troisième partie, nous appliquerons notre méthode de détection du mouvement dans le cas précis d'une caméra montée sur un véhicule. Cette hypothèse supplémentaire permet de modéliser le mouvement de la caméra induit par le déplacement du véhicule par un modèle quadratique. La recherche d'obstacles sans connaissance *a priori* consistera à détecter tous les points qui ne subissent pas ce mouvement quadratique à partir d'une modélisation markovienne hiérarchique introduite au chapitre 4.

Dans la conclusion générale, une synthèse de cette étude sera présentée, suivie de quelques perspectives liées à ces travaux.

Première partie

Outils mathématiques

Chapitre 1

Méthodes markoviennes en traitement d'images

L'utilisation des champs de Markov en traitement d'images est attractif puisque ceux-ci fournissent un cadre probabiliste très général pour l'analyse d'images et permettent d'avoir une interprétation à la fois globale et locale des dépendances entre les variables aléatoires. Ils sont devenus très populaires depuis les travaux de Geman et Geman [GG84]. Ces travaux ont permis de développer des algorithmes de simulation des champs de Markov en les interprétant comme des champs de Gibbs.

Nous allons introduire dans un premier temps quelques définitions et propriétés générales sur les modèles markoviens puis, nous nous intéresserons aux algorithmes d'optimisation sur les champs de Markov. Enfin, nous nous attarderons plus particulièrement sur les méthodes hiérarchiques qui seront à la base de notre méthode de détection de mouvement et de segmentation au sens du mouvement.

1.1 Modèles markoviens et problèmes inverses

1.1.1 Champs markoviens

Les champs de Markov spatiaux ont été définis par Besag [Bes74] et popularisés en traitement d'images par Geman et Geman [GG84]. Ces modèles sont définis à partir d'un graphe. Considérons donc un graphe simple $\mathcal{G} = (\mathcal{S},\mathcal{U})$ où $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$ est l'ensemble des sommets (dit ensemble des sites dans le cas des champs de Markov) et $\mathcal{U}$ l'ensemble des arêtes.

Définition 1.1 Deux sites s_i et s_j sont dits voisins s'il existe une arête $u_{ij} \in \mathcal{U}$ entre eux. L'ensemble des voisins d'un site s est noté $\mathcal{V}_s$.

Définition 1.2 $\mathcal{V} = \{\mathcal{V}_s | s \in \mathcal{S}\}$ est un système de voisinage pour $\mathcal{G}$ si :

- $s \notin \mathcal{V}_s$
- $\forall (s,t) \in \mathcal{S} \qquad s \in \mathcal{V}_t \Leftrightarrow t \in \mathcal{V}_s$

A partir de $\mathcal{V}$, nous définissons l'ensemble $\mathcal{C}$ dont les éléments c (Fig 1.1), appelées cliques, sont les parties de $\mathcal{S}$ vérifiant

$$\begin{cases} \exists s \in \mathcal{S} \text{ tel que } c = \{s\} \\ \qquad\qquad ou \\ \forall \{s,t\} \subset c, \quad t \in \mathcal{V}_s \end{cases}$$

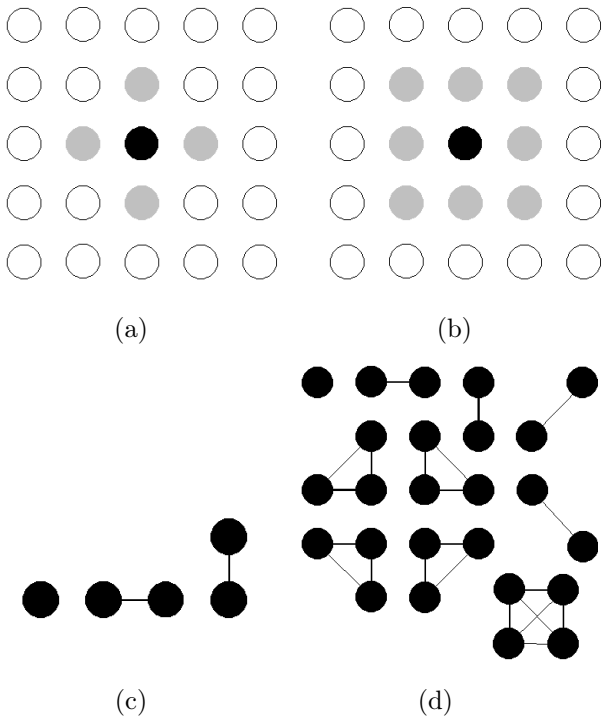


FIGURE 1.1 – Système de voisinage et cliques associées. (a) et (c) Voisinage d'ordre 1, (b) et (d) Voisinage d'ordre 2.

Considérons un champ aléatoire $\mathcal{X} = \{X_1, \cdots, X_N\}$ de réalisation $\omega = \{\omega_1, \cdots, \omega_N\}$

Définition 1.3 $\mathcal{X}$ est un champ de Markov par rapport à $\mathcal{V}$ si :

$$\begin{array}{lll} (i) & \forall \omega & P(\mathcal{X} = \omega) > 0 \\ (ii) & \forall (s,\omega) & P(X_s = \omega_s | X_r = \omega_r, r \neq s) = P(X_s = \omega_s | X_r = \omega_r, r \in \mathcal{V}_s) \end{array} \tag{1.1}$$

La définition des champs de Markov est très locale. Un champ de Markov est un champ aléatoire où chaque site n'est influencé que par ses sites voisins. Utiliser la caractérisation (1.1) ne permet pas d'avoir une expression simple de la distribution jointe $P(\mathcal{X} = \omega)$. Cependant, cette distribution peut être facilement obtenue à partir du théorème de Hammersley-Clifford établissant l'équivalence entre les champs de Markov et les champs de Gibbs. Ces derniers sont définis à partir de la forme particulière de leur distribution jointe suivante

Définition 1.4 $\mathcal{X}$ est un champ de Gibbs si sa distribution de probabilité est de la forme

$$P(\mathcal{X} = \omega) = \frac{1}{Z} \exp - \sum_{c \in \mathcal{C}} V_c(\omega),$$

où Z est la constante de normalisation

$$Z = \sum_{\omega} \exp(- \sum_{c \in \mathcal{C}} V_c(\omega)).$$

Théorème 1.1 (Hammersley-Clifford) [Bes74]

$$\begin{cases} \mathcal{X} \text{est un champ de Markov par rapport à } \mathcal{V} \\ P(\mathcal{X} = \omega) > 0 \quad \forall \omega \in \Omega \end{cases} \Leftrightarrow \mathcal{X} \text{ est un champ de Gibbs associé à } \mathcal{C}.$$

1.1.2 Estimation bayésienne

Les problèmes de restauration, classification, segmentation d'images peuvent se schématiser de la façon suivante : à partir d'observations (informations connues) sur l'image ou sur la séquence d'images (niveaux de gris, couleurs, contours, déplacements ...), on souhaite trouver un jeu d'étiquettes (informations cachées) qui pourrait être à l'origine de ces observations. La résolution de ces problèmes nécessite d'expliciter la relation entre les étiquettes et les observations que l'on définit à partir d'un modèle dit d'*attache aux données*. Cependant, ces modèles à eux seuls ne suffisent pas à résoudre ces problèmes inverses. En effet, à cause de la perte d'information intervenant lors de la formation des images et du bruit sur les observations, ces problèmes inverses sont en général mal posés au sens de Hadamard. Nous devons donc faire une hypothèse supplémentaire sur le jeu d'étiquettes recherché. C'est pourquoi nous ajoutons une contrainte au problème appelée connaissance *a priori* qui contraint le jeu d'étiquettes à être markovien.

Posons Y le champ aléatoire d'observations $y = \{y_r, r \in \mathcal{R}\}$ définies sur l'ensemble des sites $\mathcal{R} = \{r_1, r_2, \cdots, r_M\}$ et Φ l'ensemble des configurations d'observations. Notons Ω l'ensemble des configurations d'étiquettes $x = \{x_s, s \in \mathcal{S}\}$ définies sur $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$ du champ aléatoire d'étiquettes X. Les deux ensembles de sites $\mathcal{R}$ et $\mathcal{S}$ peuvent être confondus.

L'estimation bayésienne va nous permettre de modéliser conjointement les étiquettes et

les observations grâce au champ aléatoire couplé (X, Y) de distribution de probabilité P_{XY} .
Notons x^* une application appelée estimateur qui associe à tout champ d'observations sa configuration d'étiquette estimée

$$\begin{array}{rccc} x^* : & \Phi & \longrightarrow & \Omega \\ & y & \mapsto & x^*(y). \end{array} \tag{1.2}$$

On définit le risque de Bayes par :

$$R(x^*) = \mathbb{E}[C(X, x^*(Y))] = \int_{y\in\Phi}\int_{x\in\Omega} C(x, x^*(y))P_{XY}(x, y)dxdy, \tag{1.3}$$

où C est une distance sur Ω appelée fonction de coût.
D'après la règle de Bayes, on obtient

$$R(x^*) = \int_{y\in\Phi} P_Y(y)\int_{x\in\Omega} C(x, x^*(y))P_{X|Y}(x|y)dxdy. \tag{1.4}$$

On appelle estimateur bayésien optimal $\widehat{x}$ pour la fonction de coût C, l'estimateur qui minimise le risque de Bayes, c'est-à-dire :

$$\widehat{x}(y) \in \arg\min_{z\in\Omega} \int_{x\in\Omega} C(x, z)P_{X|Y}(x|y)dx. \tag{1.5}$$

Les trois estimateurs bayésiens les plus connus sont :

- L'estimateur du Maximum a Posteriori (MAP) dont la fonction de coût est définie par $C(x, z) = 1 - \delta_z(x)$ (où δ_z est la masse de dirac en z) ,
- L'estimateur du Mode des marginales *a posteriori* (MPM) de fonction de coût $C(x, z) = \sum_{s\in S}(1 - \delta_{z_s}(x_s))$,
- L'estimateur du champ moyen (MF) de fonction de coût $C(x, z) = \sum_{s\in S}(x_s - z_s)^2$.

Si l'on prend l'exemple de l'estimateur bayésien MAP, alors (1.5) devient

$$\widehat{x}(y) \in \arg\max_{x\in\Omega} P_{X|Y}(x|y). \tag{1.6}$$

Cela revient à rechercher le champ d'étiquettes le plus probable conditionnellement à la réalisation du champ d'observation y. C'est en fait l'estimateur le plus souvent employé car il permet de résoudre le problème de segmentation par un problème d'optimisation d'énergie de la forme :

$$\min_{x\in\Omega} U(x, y) = \min_{x\in\Omega}(U_1(x, y) + U_2(x)) \tag{1.7}$$

Ce problème est résolu par des algorithmes de minimisation stochastiques ou déterministes.

1.2 Algorithmes d'optimisation sur les champs de Markov

Dans ce paragraphe, nous nous restreignons à l'étude des algorithmes d'optimisation dans le cas de l'estimateur du MAP puisqu'il est le plus utilisé et le seul intervenant dans la suite de ce mémoire. Dans le cas des estimateurs du MPM et MF, nous invitons le lecteur à se référer au livre de Chalmond [Cha00a]. Dans le cas du MAP, il existe deux grandes classes d'algorithmes pour résoudre le problème (1.7) : les algorithmes stochastiques et les algorithmes déterministes ayant chacun leurs avantages et leurs inconvénients. Les premiers permettent en théorie de trouver le minimum global mais en un temps très long alors que les seconds sont beaucoup plus rapides mais convergent souvent vers un minimum local.

1.2.1 Algorithmes stochastiques : Recuit simulé

Ces algorithmes furent introduits par Kirkpatrick [KGV83] qui s'est inspiré des phénomènes physico-chimiques liés au refroidissement de systèmes cristallins. L'idée est de minimiser (1.7) de manière itérative à partir de la simulation des champs de Gibbs suivants

$$\pi_{y,T_n}(x) = \frac{1}{Z_{y,T_n}} \exp[-\frac{U(x,y)}{T_n}], \tag{1.8}$$

où la suite $(T_n)_{n\in\mathbb{N}}$ de paramètres strictement positifs appelés températures converge vers 0 en décroissant.
Geman et Geman [GG84] ont montré que si la suite $(T_n)_{n\in\mathbb{N}}$ vérifiait la relation suivante où c est une constante suffisamment grande :

$$\lim_{n\to+\infty} T_n \ln n \geq c \tag{1.9}$$

alors la chaîne de Markov engendrée par (1.8) converge vers la distribution uniforme $\pi_{y,0}(x) = \frac{1}{\sharp|E_{\min}|}\chi_{E_{\min}}$ (où $E_{\min} = \{\widetilde{x} \in \Omega | U(\widetilde{x}, y) = \min_x U(x,y)\}$) et ce, indépendamment de la configuration initiale. Cette propriété permet de trouver la solution de (1.7) par un algorithme dit de recuit-simulé présenté dans le Tableau 1.1. Dans cet algorithme, nous remplaçons souvent la décroissance logarithmique (1.9) numériquement beaucoup trop lente par une descente géométrique de la forme :

$$\forall n \in \mathbb{N}^* \qquad T_n = a \times T_{n-1}, \tag{1.10}$$

où a est un paramètre positif proche de 1. Bien que cette condition ne vérifie pas (1.9), ces algorithmes permettent de trouver une configuration proche du minimum global de

U tout en étant peu sensibles à la condition initiale. Cependant, ces algorithmes restent extrêmement lents même avec ce type de modification. C'est pourquoi on introduit des algorithmes déterministes.

Algorithme $\widehat{x} = RecuitSimule(T_0, a, x_0, U, y)$

T_0 : température initiale
a : taux de refroidissement
x_0 : configuration initiale
U : énergie à minimiser
y : champ d'observation

- $T = \frac{T_0}{a}$;
- $Compteur = 1$;
- $x(0) \leftarrow x_0$
- Tant que $Compteur \geq 1$
  - $Compteur \leftarrow 0$
  - $x' \leftarrow x(k)$
  - $T \leftarrow T \times a$
  - Pour chaque site s
    - $\lambda = x_s$
    - Choisir aléatoirement une étiquette $\mu \in \Omega$ et calculer $\Delta U = U(x', y) - U(x(k), y)$ avec $x' = (x'_1, \cdots, \mu, \cdots, x'_N)$
      - Si $\Delta \leq 0$ $x'_s \leftarrow \mu$
      - Sinon choisir un nombre aléatoire uniforme ξ dans $[0, 1]$
        - Si $\xi < exp(-\frac{\Delta U}{T})$, $x'_s \leftarrow \mu$
    - Si $x_s \neq \lambda$ alors $Compteur \leftarrow Compteur + 1$
  - $x(k+1) \leftarrow x'$
- $\widehat{x} \leftarrow x(k)$

TABLE 1.1 – Algorithme du recuit simulé

1.2.2 Optimisation déterministe

Des méthodes déterministes dérivées des méthodes de recuit simulé ont été développées pour éviter les problèmes de lenteur de convergence des algorithmes stochastiques. Ces méthodes de descente recherchent à partir d'une configuration courante, une configuration d'énergie plus faible. Elles peuvent être interprétées comme des méthodes de recuit

Algorithme $\widehat{x} = ICM(x_0, U, y)$

x_0 : configuration initiale
U : énergie à minimiser
y : champ d'observation

- $Compteur = 1$;
- $x(0) \leftarrow x_0$
- Tant que $Compteur \geq 1$
  - $Compteur \leftarrow 0$
  - $x' \leftarrow x(k)$
  - Pour chaque site s
    - $\lambda = x_s$
    - assigner au site s l'étiquette x'_s qui minimise $\Delta U = U(x', y) - U(x(k), y)$
    - Si $x_s \neq \lambda$ alors $Compteur \leftarrow Compteur + 1$
  - $x(k+1) \leftarrow x'$
- $\widehat{x} \leftarrow x(k)$

TABLE 1.2 – Algorithme I.C.M.

simulé à température nulle (aucune remontée de l'énergie n'est autorisée). Cependant comme l'énergie U est en général non convexe, sans bonne initialisation, ces techniques ne convergeront pas vers le minimum global de U mais vers un minimum local. Elles ont néanmoins l'avantage de converger rapidement. L'algorithme le plus utilisé est l'algorithme appelé I.C.M (Iterated Conditional Modes)[Bes74] Tab 1.2. Il existe plusieurs variantes de cet algorithme selon la façon de visiter les sites. En effet, contrairement aux méthodes stochastiques où la stratégie de visite des sites est en théorie peu importante, dans le cas des méthodes déterministes celles-ci influent fortement sur l'état final du système. Citons par exemple :

- la technique de balayage séquentiel des sites souvent utilisée pour sa simplicité.
- la technique qui utilise une pile d'instabilité binaire où initialement tous les sites sont considérés instables. Dans ce cas, seul les sites instables sont visités. Le site visité devenant stable et si son étiquette se voit modifiée, ses voisins sont alors considérés comme instables [BL93], et ce jusqu'à stabilité de l'ensemble des sites.
- la technique H.C.F. (Highest Confidence First) [CB90], utilise quant à elle une pile d'instabilité correspondant à la variation d'énergie la plus favorable (plus forte baisse) qui pourrait résulter de la substitution de l'étiquette du site par une autre. Les sites sont ordonnés dans cette pile en fonction de leur instabilité et visités selon

cet ordre jusqu'à stabilité de l'ensemble des sites.

L'algorithme I.C.M. converge beaucoup plus rapidement que l'algorithme du recuit-simulé. Cependant, contrairement à son homologue stochastique, il ne permet de trouver qu'un minimum local du problème (1.7). Il existe plusieurs solutions pour contourner ce problème :

- approcher l'énergie non convexe par une énergie convexe. Blake [Bla89] construit une suite d'approximations de l'énergie convergeant vers cette dernière. Cette méthode appelée Graduated Non-Convexity (GNC), minimise une première approximation convexe de la fonctionnelle par un algorithme déterministe. Puis, chaque approximation est ensuite minimisée en utilisant comme initialisation la solution approchée précédente. La principale limitation de cette méthode réside dans la construction des fonctions auxiliaires souvent très difficiles à déterminer.
- des méthodes hiérarchiques que nous allons expliciter.

1.3 Méthodes hiérarchiques

Nous distinguons deux types de méthodes : les méthodes de "hiérarchie induite" et les méthodes de "hiérarchie explicite" [GHP+97]. Les premières résolvent le problème markovien classique mono-échelle par une transformation de celui-ci sur plusieurs échelles. Les seconds intègrent directement la hiérarchie dans la définition du modèle.

1.3.1 Méthodes de hiérarchies induites

Approches par groupe de renormalisation

La technique par groupe de renormalisation est issue de la physique statistique. Gidas [Gid89] l'a adaptée aux modèles markoviens pour l'analyse d'image. A partir du champ d'étiquettes X original, la méthode consiste à définir une pyramide de champs d'étiquettes. Ces champs d'étiquettes $(X^0 = X, X^1, \cdots, X^N)$ sont définis respectivement sur les ensembles $(\mathcal{S}^0 = \mathcal{S}, \mathcal{S}^1, \cdots, \mathcal{S}^N)$ obtenus par décimations $(\downarrow 2)$ successives. Les champs X^i sont définis récursivement à partir de la probabilité de transition entre deux niveaux par la formule suivante

$$P(X^i = x^i | Y = y) = \sum_{x^{i-1}} P(X^i = x^i | X^{i-1} = x^{i-1}) P(X^{i-1} = x^{i-1} | Y = y). \tag{1.11}$$

Ainsi, en supposant que la distribution jointe (X, Y) soit markovienne de Hamiltonien U, on en déduit que l'estimateur MAP à chaque niveau de résolution i vaut

$$\begin{aligned} P(X^i = x^i | Y = y) &= \tfrac{1}{Z} \exp\{-U^i(x^i, y)\}, \\ \text{avec } U^i(x^i, y) &= -\log[\textstyle\sum_{x^{i-1}} P(X^i = x^i | X^{i-1} = x^{i-1}) \exp\{-U^{i-1}(x^{i-1}, y)\}]. \end{aligned} \tag{1.12}$$

Cette méthode bien que simple et rigoureuse est difficile à exploiter. En effet, même pour des modèles simples de transition entre chaque résolution, il est difficile d'exploiter les énergies définies par (1.12). De plus, en règle générale, les lois $P(X^i = x^i | Y = y)$, $i \neq 0$ ne sont pas des distributions de Markov locales (U^i ne se décomposant pas en somme de potentiels locaux). Cette perte de localité implique que nous ne pouvons pas utiliser les algorithmes classiques pour les champs de Markov.

Approche multiéchelle

Ces méthodes sont inspirées des méthodes multigrilles en analyse numérique [Hac85] fondées sur la résolution des équations sur une suite d'espaces emboîtés. Certains auteurs [BL91], [HPB94], [Pér93], ont adapté ces techniques pour résoudre le problème de minimisation d'énergie (1.7).

L'idée de l'approche multiéchelle est de résoudre le problème (1.7) dans des espaces de

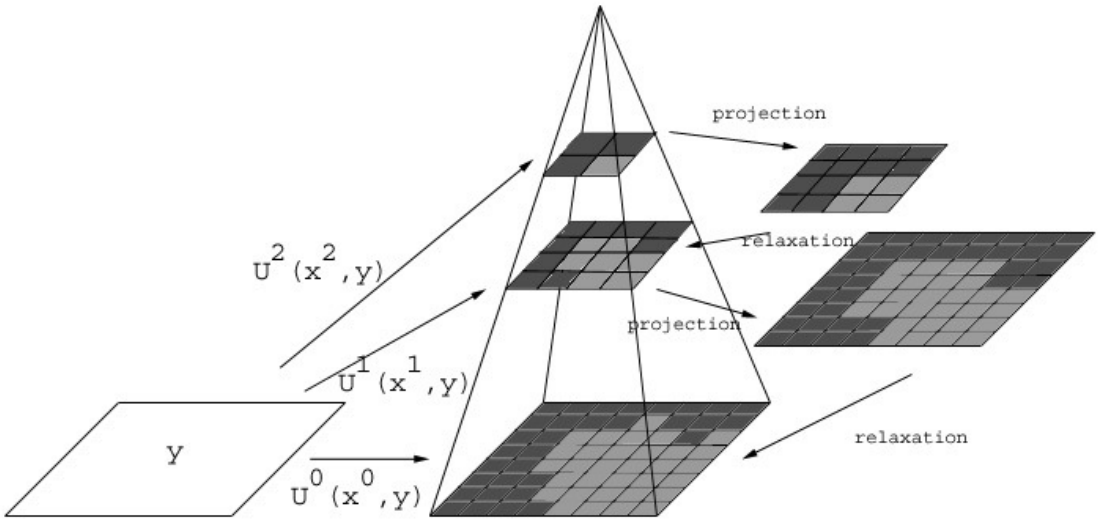


FIGURE 1.2 – Structure pyramidale des approches multiéchelles

configurations emboîtés $\Omega^N \subset \cdots \subset \Omega^i \cdots \subset \Omega^0 = \Omega$ où Ω^i correspond généralement

aux configurations de Ω constantes sur des blocs de taille $2^i \times 2^i$. L'approche consiste ensuite à reformuler l'énergie U sur chaque échelle en exploitant la structure particulière des éléments x^i de Ω^i pour obtenir une formulation pour $U^i(x^i, y)$. Une stratégie "coarse-to-fine" est ensuite appliquée à chaque échelle en utilisant un des algorithmes de relaxation précédent pour résoudre l'estimateur du MAP

$$\widehat{x}^i = \arg\min_{x^i} U^i(x^i, y). \tag{1.13}$$

A chaque étape, la solution de l'étape précédente permet d'initialiser l'algorithme (Fig 1.2).
Ces méthodes ont permis d'accélérer les processus de relaxation déterministes classiques, tout en obtenant des résultats de qualité similaire aux algorithmes stochastiques fort coûteux en temps de calcul.

1.3.2 Méthodes de hiérarchies explicites

Approche multirésolution

Les approches multirésolutions dans le cas des méthodes hiérarchiques explicites se rapprochent des méthodes multiéchelles vues précédemment. En effet, une stratégie "coarse-to-fine" est aussi adaptée pour initialiser le problème d'optimisation à chaque étape. Contrairement au cas précédent, une transformation est appliquée aux observations y pour définir les observations y^i à chaque échelle.

On distingue deux approches différentes : les approches multi-modèles [AG92], [CTPPB98] qui définissent un champ markovien avec des caractéristiques différentes à chaque échelle et les approches mono-modèles où les champs de Markov sont identiques [BBFC92], [HB93], [KD92]. Ces dernières construisent une pyramide de champs d'étiquettes identique aux méthodes vues précédemment. Elles décomposent les observations y de l'image originale par filtrage et sous-échantillonnage pour définir des observations y^i sur chaque échelle. Enfin, à partir de ces données, elles définissent les énergies $U^i(x^i, y^i)$ à chaque niveau à partir du problème original (1.7). Puis une stratégie de descente "coarse-to-fine" est appliquée (Fig 1.3). Ces méthodes ont été largement employées et ont l'avantage d'être simples d'utilisation. Les caractéristiques des modèles markoviens définis aux différents niveaux de résolution sont choisies de manière heuristique. Par conséquent, cette approche ne permet pas d'assurer un "bon" comportement de l'algorithme sur certains problèmes tant par la qualité des résultats que de la vitesse de convergence.

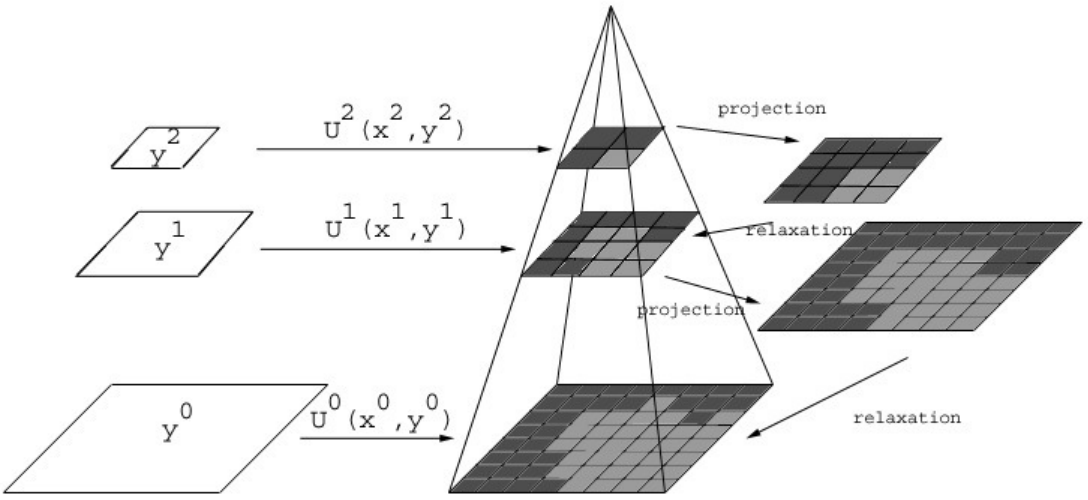


FIGURE 1.3 – Structure pyramidale des approches multirésolution

Approche sur graphes hiérarchiques

Les méthodes précédentes sont fondées sur des champs de Markov spatiaux. Les approches, présentées ici, sont définies de façon récursive sur des graphes hiérarchiques, ce qui leur donne par nature une représentation hiérarchique. On ne cherche pas ici à résoudre le problème (1.7) par une technique "coarse-to-fine" sur la base d'un modèle markovien spatial classique mais à mettre en place un champ d'étiquettes global sur la structure pyramidale $S = \cup_{i=0}^{N} S^i$ définie précédemment.

Luettgen [Lue93] s'est inspiré des travaux de Chou [CB88] et de Basseville [BBC+92]

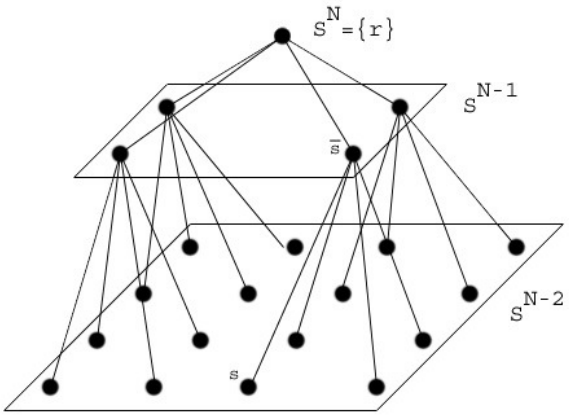


FIGURE 1.4 – Quadarbre

dans le cadre des signaux mono-dimensionnels qu'il a adaptés pour l'estimation du mouvement dans les séquences d'images [LKW94]. Pour cela, il définit le champ d'étiquettes X sur l'ensemble S constitué par les grilles $S^0, \cdots, S^N$ où S^N est réduit à un unique site r appelé racine (Fig 1.4). Le modèle *a priori* sur X est défini à partir de l'équation récursive et linéaire suivante

$$\begin{cases} x_s = A_s x_{\bar{s}} + B_s \omega_s, \quad \forall s \in S, s \neq r, \\ x_r \sim \mathcal{N}(0, \Gamma_r), \end{cases} \tag{1.14}$$

où $\bar{s}$ est le parent, A_s et B_s sont deux matrices qui ne dépendent que du niveau auquel le site s se trouve et ω_s est un bruit blanc gaussien centré et normalisé. Il est à remarquer que les connaissances *a priori* sur les étiquettes ne sont pas introduites à l'aide d'interactions spatiales (comme dans le cas markovien classique) mais par des interactions inter-échelles. Le champ d'observation est défini de la façon suivante

$$\begin{cases} y_s = C_s x_{\bar{s}} + v_s, \quad \forall s \in S, s \neq r, \\ v_s \sim \mathcal{N}(0, R_s). \end{cases} \tag{1.15}$$

La nature linéaire de (1.14) permet de calculer l'estimateur du MAP à l'aide d'algorithmes non itératifs en deux passes. La passe montante, où l'estimation de la configuration de x_s, est obtenue par filtrage de type Kalman sur ces quatre fils jusqu'à la racine du graphe. Une technique similaire est ensuite utilisée en redescendant le graphe, où l'estimée du site s est obtenue à partir de la prédiction issue de son père. Cette technique par son caractère non itératif, a permis d'obtenir dans le cas du problème de l'estimation du flot optique des résultats comparables à des méthodes d'optimisation classiques en un temps très inférieur. Cependant, de par la définition du modèle (1.14), cette méthode ne peut s'appliquer que pour des problèmes linéaires.

Parallèlement, Bouman [BS94] développe un modèle hiérarchique à partir d'un modèle markovien sur les transitions inter-échelles :

$$P(X^i = x^i | X^{i+1} = x^{i+1}, \cdots, X^N = x^n) = P(X^i = x^i | X^{i+1} = x^{i+1}), \quad \forall i < N. \tag{1.16}$$

L'estimateur du MAP ne permet pas de moduler les pénalisations d'une erreur en fonction de l'échelle à laquelle elle se produit. De ce fait, il n'est pas adapté aux modèles hiérarchiques. Partant de ce constat, Bouman [BS94] introduit un nouvel estimateur appelé SMAP (pour "Sequential MAP") défini de la façon suivante :

$$C^{SMAP}(s, z) = \frac{1}{2} + \sum_{i=0}^{N} 2^{i-1} [1 - \prod_{k=i}^{n} \delta(x^i - z^i)]. \tag{1.17}$$

Cet estimateur pénalise davantage les erreurs à une résolution grossière qu'à une résolution fine. L'inconvénient est que le risque de Bayes (1.3) associé à cette fonction de coût ne

peut être calculé qu'en faisant une approximation :

$$\begin{cases} \widehat{x}^N \approx \arg\max_{x^N}[\log P(Y=y|X^N=x^N)+\log P(X^N=x^N)] \\ \widehat{x}^i \approx \arg\max_{x^i}[\log P(Y=y|X^N=x^N)+\log P(X^i=x^i|X^{i+1}=x^{i+1})], \end{cases} \tag{1.18}$$

où Y n'est défini que sur la résolution la plus fine.
D'autres auteurs ont depuis repris cette modélisation hiérarchique [PCL00], [LHPF95], [LPH00],[PCP99] en les étendant au cas où Y est défini sur la pyramide entière... Ces techniques sont à l'heure actuelle très attractives. En effet, de par leur caractère non itératif, elles sont peu coûteuses en temps de calcul. Depuis, Chardin [Cha00b] a généralisé cette modélisation sur un graphe tronqué (Fig 1.5) où la résolution grossière n'est pas réduite à un singleton. Ceci permet d'utiliser ce type de méthode sur des images de taille $2^p \times 2^q$, $p \neq q$. Nous verrons dans la suite que cette modélisation se prête bien à notre problématique. C'est pourquoi, nous allons l'étudier plus en détail.

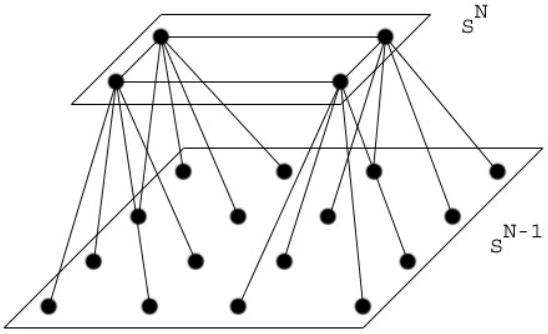


FIGURE 1.5 – Arbre tronqué

1.4 Méthode hiérarchique semi-itérative

Considérons $S = \cup_{n=0}^{N} S^n$ où S^i désigne le niveau de résolution i de l'image, $X = \{X_s, s \in S\}$ et $Y = \{Y_s, s \in S\}$ respectivement le champ aléatoire des étiquettes et le champ aléatoire des observations. Notons X^n l'ensemble des étiquettes des sites de niveau n, *i.e* $X^n = \{X_s, s \in S^n\}$ et de la même manière $Y^n = \{Y_s, s \in S^n\}$. Enfin, posons $\bar{i}$ le parent du site i, $\underline{i}$ l'ensemble des enfants de i et $\underline{\underline{i}}$ l'ensemble des sites formant l'arbre de racine i.
On suppose que :

– le champ aléatoire X est une chaîne de Markov d'ordre 1 en échelle :

$$P(X^n|X^k, k>n) = P(X^n|X^{n+1}), \quad \forall n < N. \tag{1.19}$$

– X^n indépendant conditionnellement à X^{n+1}

$$P(X^n|X^{n+1}) = \prod_{i \in S^n} P(X_i^n|X^{n+1}), \quad \forall n < N. \tag{1.20}$$

– X_i est conditionnellement à son père indépendant de $X_{j \neq \bar{i}}^{n+1}$,

$$P(X_i|X^{n+1}) = P(X_i|X_{\bar{i}}) \neq 0, \forall n < N, \forall i \in S^n. \tag{1.21}$$

– X^N est un champ de Markov spatial. Dès lors, d'après le théorème de Hammersley-Clifford, on peut définir la distribution de X^N par

$$P(X^N = x^N) = \frac{1}{Z^N} \exp\left[-\sum_{<i,j> \in S^N} v_{ij}(x_i, x_j)\right], \tag{1.22}$$

où Z^N est la constante de normalisation

$$Z^N = \sum_{x^N} \exp\left[-\sum_{<i,j> \in S^N} v_{ij}(x_i, x_j)\right]. \tag{1.23}$$

– Y indépendant conditionnellement à X

$$P(Y|X) = \prod_{i \in S} P(Y_i|X). \tag{1.24}$$

– Y_i est conditionnellement à X indépendant de $X_{j \neq i}$,

$$\forall i \in S, P(Y_i|X) = P(Y_i|X_i). \tag{1.25}$$

Sous ces hypothèses, on peut écrire la distribution *a priori* de la façon suivante

$$\begin{aligned} P(X) &= P(X^N) \prod_{0 \leq n < N} P(X^n|X^{n+1}) \\ &= P(X^N) \prod_{0 \leq n < N} \prod_{i \in S^n} P(X_i|X^{n+1}) \\ &= P(X^N) \prod_{i \in S \setminus S^N} P(X_i|X_{\bar{i}}), \end{aligned} \tag{1.26}$$

et la distribution jointe (X, Y)

$$P(X = x, Y = y) = P(X^N) \prod_{i \notin S^N} P(X_i|X_{\bar{i}}) \prod_{i \in S} P(Y_i = y_i|X_i = x_i). \tag{1.27}$$

En notant :

– les probabilités de transition parent-enfant : $P(X_i = x_i|X_{\bar{i}} = x_{\bar{i}}) \propto \exp[-w_i(x_i, x_{\bar{i}})]$
– les vraisemblances : $P(Y_i = y_i|X_i = x_i) \propto \exp[-l_i(x_i, y_i)]$,

on peut réécrire la distribution jointe (X, Y) sous la forme [PCL00], [CP98]

$$P(X = x, Y = y) \propto \exp\{-[\sum_{<i,j>\in S^N} v_{i,j}(x_i, x_j) + \sum_{i\notin S^N} w_i(x_i, x_{\bar{i}}) + \sum_{i\in S} l_i(x_i, y_i)]\}, \quad (1.28)$$

où v_{ij} et w_i sont respectivement des termes d'attaches en espace et en échelle du champ d'étiquettes et l_i est un terme d'attache aux données.
L'estimateur du MAP associé à cette distribution est

$$\begin{aligned}
\widehat{x} &= \arg\max_x P(x|y) = \arg\max_x P(x, y) \\
&= \arg\min_x [\sum_{<i,j>\in S^N} v_{i,j}(x_i, x_j) + \sum_{i\in S^N} l_i(x_i, y_i) + \sum_{i\notin S^N} [w_i(x_i, x_{\bar{i}}) + l_i(x_i, y_i)]] \\
&= \arg\min_{x^N} [\sum_{<i,j>\in S^N} v_{i,j}(x_i, x_j) + \sum_{i\in S^N} l_i(x_i, y_i) + \min_{x_{S\setminus S^N}} \sum_{i\notin S^N} [w_i(x_i, x_{\bar{i}}) + l_i(x_i, y_i)]] \\
&= \arg\min_{x^N} [\sum_{<i,j>\in S^N} v_{i,j}(x_i, x_j) + \sum_{i\in S^N} l_i(x_i, y_i) + \sum_{i\in S^{N-1}} \underbrace{\min_{x_{\underline{\underline{i}}}} \sum_{j\in \underline{\underline{i}}} [w_j(x_j, x_{\bar{j}}) + l_j(x_j, y_j)]}_{U_i(x_{\bar{i}})}].
\end{aligned} \quad (1.29)$$

Comme pour le modèle de Bouman, cette minimisation se fait en deux passes. Tout d'abord, une passe montante sur les sous-arbres dont la racine est un site de S^N qui consiste à calculer les probabilités $P(X^n = x^n | Y^n = y^n, X^{n-1} = x^{n-1})$. Puis une passe descendante qui estime $\widehat{x}^n$. Entre ces deux étapes, une procédure itérative (I.C.M.) au niveau de la grille grossière S^N est effectuée pour calculer $\widehat{x}^N$. C'est donc un algorithme semi-itératif qui peut se résumer de la façon suivante :

<u>Passe montante</u>

- Feuilles, $i \in S^0$

$$\begin{cases} U_i(x_{\bar{i}}) = \min_{x_i}[w_i(x_i, x_{\bar{i}}) + l_i(x_i, y_i)] \\ x_i^*(x_{\bar{i}}) = \arg\min_{x_i}[w_i(x_i, x_{\bar{i}}) + l_i(x_i, y_i)] \end{cases}$$

- Récurrence, pour $n = 1, 2, \cdots, N-1, i \in S^n$

$$\begin{cases} U_i(x_{\bar{i}}) = \min_{x_i}[w_i(x_i, x_{\bar{i}}) + l_i(x_i, y_i) + \sum_{j\in \underline{i}} U_j(x_i)] \\ x_i^*(x_{\bar{i}}) = \arg\min_{x_i}[w_i(x_i, x_{\bar{i}}) + l_i(x_i, y_i) + \sum_{j\in \underline{i}} U_j(x_i)] \end{cases}$$

<u>Echelle grossière N</u>

- Initialisation

$\forall i \in S^N, \quad x_i^N = \arg\min_\lambda[l_i(\lambda, y_i) + \sum_{j\in \underline{i}} U_j(\lambda)]$

$-ICM$

$\widehat{x}^N = \arg\min_{x^N}[\sum_{<i,j>\in S^N} v_{i,j}(x_i, x_j) + \sum_{i\in S^N}[l_i(x_i, y_i) + \sum_{j\in \underline{i}} U_j(x_i)]$

<u>Passe descendante</u>

$\forall n = N-1, J-2\cdots, 0, \quad \forall i \in S^n :$

$\widehat{x}_i = x_i^*(\widehat{x}_{\bar{i}})$

Remarquons que l'algorithme I.C.M. n'est appliqué qu'à une résolution grossière de l'image où le jeu de données est restreint. C'est pourquoi cet algorithme est moins sensible à l'initialisation qu'un simple algorithme de minimisation déterministe sur l'ensemble de l'image.
Cet algorithme a montré son efficacité dans la cas de la segmentation d'images [CP98]. En effet, les résultats sont comparables aux méthodes multigrilles [HPB94] tout en étant moins coûteux en temps de calcul.

1.5 Conclusion

Nous avons fait un bref rappel des méthodes markoviennes en traitement d'images. Nous avons volontairement restreint notre étude au cas des méthodes dites "supervisées", c'est-à-dire au cas où nous connaissons *a priori* les modèles probabilistes rentrant en jeu. Dans le cas contraire, les méthodes sont dites "non supervisées" et nécessitent l'utilisation d'algorithmes plus coûteux en temps de calcul (algorithme E.M., I.C.E.,...). Pour ce type de problèmes, nous invitons le lecteur à se référer au livre de Chalmond [Cha00a].

Pour résoudre les problèmes des méthodes "supervisées" à l'aide de l'estimateur du M.A.P., deux solutions s'offrent à nous : les méthodes stochastiques et les méthodes déterministes. Les premières, bien que convergeant vers le minimum global de l'énergie, ne sont pas appropriées pour des applications "temps réel". C'est pourquoi, nous optons pour des méthodes déterministes convergeant rapidement vers un minimum local. Nous avons vu qu'il existe différentes méthodes pour amoindrir la sensibilité de l'initialisation de ces algorithmes. Les méthodes hiérarchiques "induites" qui résolvent le problème sur une pyramide de champ d'étiquettes à partir du problème initial et les méthodes hiérarchiques "explicites" qui redéfinissent le problème par multirésolution ou sur un graphe hiérarchique. Ces méthodes ont montré que l'on peut obtenir des résultats similaires aux algorithmes stochastiques tout en étant moins coûteux en temps de calcul qu'un simple algorithme déterministe mono-résolution.

Dans notre cas, comme nous sommes en possession d'observations sur différentes échelles et au vu des différentes méthodes existantes, nous avons opté pour la méthode hiérarchique semi-itérative de Chardin [CP98] lorsque nous traiterons le problème de détection de mouvement (Chapitre 4). Cette modélisation nous permettra de surmonter le problème de l'aliasing temporel. Nous modifierons cependant cette méthode en lui rajoutant une dépendance temporelle (entre le champ d'étiquettes à l'instant t et à l'instant $t+1$) pour résoudre notre problème de détection d'objets au sens du mouvement. Par contre, dans le cas du problème de la segmentation au sens du mouvement, pour éviter de construire une méthode itérative du type détection/segmentation, nous opterons pour

une méthode markovienne multirésolution multi-modèles.

Deuxième partie

Estimation et analyse du mouvement

Chapitre 2

Estimation du mouvement par analyse d'ondelettes

Dans ce chapitre, nous nous intéressons au problème de l'estimation du mouvement dans les séquences d'images. Ce problème est un sujet important en analyse d'images puisqu'il est la première étape indispensable pour une interprétation automatique des séquences d'images. Les problèmes de compression de vidéo, de détection de mouvement, de résumé vidéo sont autant d'applications qui nécessitent une étude préalable d'estimation du mouvement. C'est pour cette raison que ce problème a fait l'objet de nombreux travaux depuis les années 80. On distingue deux approches pour caractériser le mouvement dans des séquences d'images en fonction des connaissances préalables que nous possédons sur celles-ci. Dans le cas où nous n'avons aucun *a priori* sur le type de mouvement étudié ni sur les zones de l'image qui suivent des déplacements distincts, le problème consiste à déterminer le vecteur de déplacement (flot optique) en chaque point de l'image. Par contre, si nous disposons d'un partitionnement de l'image en zones homogènes au sens du mouvement, il s'agit d'attribuer à chacune de ces régions un modèle paramétrique de mouvement. La méthode que nous développerons dans ce chapitre peut être utilisée dans les deux circonstances.

Après une présentation des difficultés rencontrées sur ce sujet, nous exposerons les principales techniques d'estimation du flot optique. Dans un troisième paragraphe, nous décrirons les méthodes d'estimation de mouvement d'une région à partir d'une modélisation paramétrique de celui-ci. La méthode que nous proposons sera ensuite expliquée, elle est fondée sur une projection de l'équation du flot optique sur une base de fonctions et est en fait une extension de la méthode d'estimation du flot optique de Bernard [Ber98]. Cette méthode supposant le flot localement constant permet d'aboutir à un bon compromis entre rapidité et exactitude de l'estimation grâce à la projection de l'équation du flot

optique sur une base d'ondelettes.

2.1 Positionnement du problème

D'après les travaux de Johanson [Joh73], la connaissance du mouvement est capitale dans la vision humaine pour l'analyse d'une scène. Cependant, une différence importante entre la vision monoculaire et la vision stéréoscopique humaine est la perte de la troisième dimension. En effet, une caméra ne fournit que la projection de la scène 3D sur un plan 2D. Par conséquent, l'information seule du mouvement dans une séquence d'images ne nous permettra pas d'obtenir les informations sur le mouvement 3D réel de la scène. Seul le mouvement dit "mouvement apparent" résultant de la projection du mouvement réel sur le plan image et du mouvement de la caméra peut être analysé. Néanmoins, bien que celui-ci ne nous permette pas de connaître le mouvement réel dans la scène, il peut cependant nous donner des informations primordiales pour différents problèmes. La détection du mouvement [IRP92][OB94], la segmentation d'images [CTS97] [MP02], le suivi de régions [GB00][MB94], la classification de scènes [FBP02] sont autant d'applications qui nécessitent au préalable une connaissance du mouvement apparent dans l'image.

Considérons une séquence d'images $I((x(t), y(t)), t)$, le mouvement apparent dans l'image, plus communément appelé **flot optique**, est la vitesse de déplacement $\overrightarrow{v}((x(t), y(t), t)$ des points physiques dans la séquence d'images, c'est-à-dire

$$\overrightarrow{v} = (v_1, v_2) = (\frac{dx}{dt}(t), \frac{dy}{dt}(t)). \tag{2.1}$$

Le flot optique est une variable cachée et n'est accessible que par l'analyse des variations de l'intensité $I((x(t), y(t)), t)$ au cours du temps. C'est pourquoi il est nécessaire de faire une hypothèse sur la variation de cette intensité quand t varie. Cette hypothèse dite "**contrainte du flot optique**" sera la base du calcul du flot optique.

Hypothèse 2.1 Contrainte du flot optique

L'intensité d'un point physique de l'image ne varie pas au cours du temps.

Ce qui d'un point de vue mathématique s'écrit :

$$I((x(t), y(t)), t) = I((x(t_0), y(t_0)), t_0) \quad \forall t. \tag{2.2}$$

Remarquons que la contrainte du flot optique est une hypothèse forte sur la séquence. En effet, un changement de luminosité dans la scène se traduit par un mouvement dans le calcul. Ceci n'est pas sans dangers puisqu'elle détecte un déplacement dans le cas d'un changement de luminosité. C'est ainsi que cette hypothèse peut être remise en cause dans les cas suivants :

- présence d'ombres ou de reflets dans la scène,
- déplacements ou changements de source lumineuse,
- occlusions des objets,
- bruit des capteurs $\cdots$

Notons que certains auteurs tentent d'éviter cette hypothèse afin de tenir compte des éventuels changement d'intensité. [NY93] [MN94] [GJ95] supposent que l'intensité varie de façon linéaire au cours du temps. Dans le cas de séquences d'images météorologiques ou de mouvements fluides, [Bér99] [CMP02] remplacent l'hypothèse de constance de luminosité par une hypothèse de conservation de densité. Néanmoins, bien que cette hypothèse de contrainte de luminosité ait ses limites, la plupart des méthodes d'estimation du flot optique la considère suffisante.
De l'hypothèse (2.2), il résulte que

$$\frac{dI}{dt}((x(t), y(t)), t) = 0. \tag{2.3}$$

Soit

$$\frac{\partial I}{\partial x}\frac{dx}{dt}((x,y),t) + \frac{\partial I}{\partial y}\frac{dy}{dt}((x,y),t) + \frac{\partial I}{\partial t}((x,y),t) = 0 \tag{2.4}$$

Ainsi, nous obtenons la relation suivante

$$\overrightarrow{\nabla} I((x,y),t).\overrightarrow{v}((x,y),t) + \frac{\partial I}{\partial t}((x,y),t) = 0, \tag{2.5}$$

où $\overrightarrow{\nabla} I$ est le gradient spatial de l'image. Cette équation est connue sous le nom d'**équation de contrainte du mouvement apparent** (notée E.C.M.A.). Cette équation à elle seule ne peut suffire à estimer le flot optique dans la séquence d'images. En effet, (2.5) est une équation scalaire à deux inconnues v_1 et v_2. Seule la composante normale aux contours de l'image $v_n((x,y),t)$ peut être déduite de (2.5) (figure 2.1)

$$v_n((x,y),t) = -\frac{\partial I}{\partial t}((x,y),t).\frac{\overrightarrow{\nabla} I((x,y),t)}{||\overrightarrow{\nabla} I((x,y),t)||^2}. \tag{2.6}$$

Ce problème connu sous le nom de "aperture problem" ou "**problème de l'ouverture**". Pour résoudre ce problème mal posé, il faut donc ajouter d'autres contraintes à celle de constance de luminosité. Une première méthode pour résoudre le problème du flot optique consiste à supposer une contrainte plus forte que la constance de luminosité. Il s'agit de supposer la conservation du gradient de l'intensité [Nag83] ce qui exclut les déformations premières de l'intensité telles que les rotations, les dilatations... Cette contrainte permet d'obtenir les équations suivantes

$$\begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \begin{pmatrix} v_1(t) \\ v_2(t) \end{pmatrix} + \begin{pmatrix} \frac{\partial^2 I}{\partial t \partial x} \\ \frac{\partial^2 I}{\partial t \partial y} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{2.7}$$

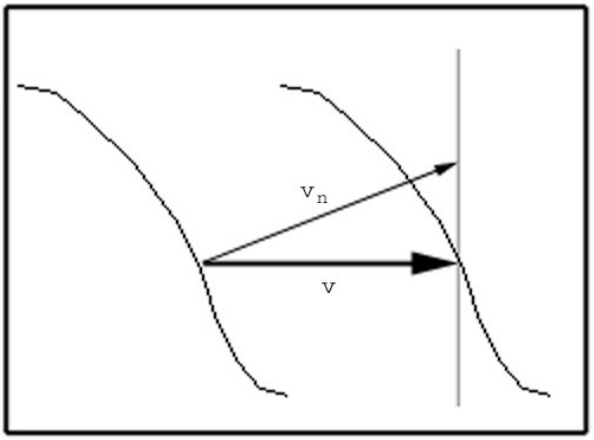


FIGURE 2.1 – Illustration du problème de l'ouverture : le mouvement apparent v est en gras, seule la composante normale au contour v_n peut être calculée à partir de l'E.C.M.A.

Les équations (2.7) peuvent se suffire à elles-mêmes ou bien être utilisées en parallèle avec la contrainte du flot optique. Remarquons que l'utilisation des dérivées secondes a pour conséquence une plus forte sensibilité au bruit. De nombreuses études ont été menées pour contourner le problème de l'ouverture. Ces méthodes sont généralement regroupées en trois grandes familles :

- Les méthodes de mise en correspondance.
- Les méthodes fréquentielles.
- Les méthodes différentielles.

Nous allons développer brièvement les principes de ces différentes approches dans les sections suivantes. Pour une étude plus exhaustive, nous invitons le lecteur à consulter l'article de Barron, Fleet et Jepson [BFB94].

2.2 Les méthodes d'estimation du flot optique

2.2.1 Méthodes de mise en correspondance

Les méthodes de mise en correspondance consistent à trouver le champ de déplacement $\overrightarrow{d} = (d_x, d_y)$ qui apparie au mieux des régions de la scène entre deux instants consécutifs [Ana89][Sin90]. La mise en correspondance entre les deux images se fait généralement à partir de la minimisation de la somme des carrés de la différence entre les deux images

aux instants t et $t+1$, SSD (*sum of squared differences*), définie de la façon suivante

$$SSD_t((x,y);\overrightarrow{d}) = \sum_{(i,j)\in V_{x,y}} \left[I\left((i,j),t\right) - I\left((i,j)+\overrightarrow{d},t+1\right)\right]^2, \tag{2.8}$$

où $V_{x,y}$ est un voisinage centré autour du point (x,y).
Ces méthodes sont souvent coûteuses en temps de calcul. Pour diminuer la complexité algorithmique, certains auteurs proposent de réduire l'espace de recherche des vitesses admissibles sur une gamme de valeurs discrètes [AP95] [HJ83]. Ces méthodes sont robustes et simples à mettre en oeuvre. Elles se retrouvent dans la majorité des standards de compression vidéo. Cependant, du fait de la discrétisation du déplacement estimé, elles deviennent très imprécises.

2.2.2 Méthodes fréquentielles

Ces méthodes inspirées des travaux de Adelson et Bergen [AB85] étudient l'E.C.M.A. dans le domaine fréquentiel. Considérons la séquence d'images $I(x,y,t)$ ayant pour transformée de Fourier $\widehat{I}(f_x,f_y,f_t)$. Dans le cas où les objets de la séquence d'images subissent un mouvement de translation uniforme $(x(t),y(t)) = (x_0+v_1t, y_0+v_2t)$, nous avons, d'après l'hypothèse de contrainte du flot optique (2.2), la relation suivante

$$I(x,y,t) = I(x-v_1t, y-v_2t, 0). \tag{2.9}$$

Ce qui est équivalent à

$$\widehat{I}(f_x,f_y,f_t) = \widehat{I}_0(f_x,f_y)\delta(f_t+v_1f_x+v_2f_y), \tag{2.10}$$

où $\delta(.)$ est la mesure de Dirac en 0.
Posons $k=(f_x,f_y)$ et $w=f_t$. Déterminer le flot optique dans la séquence d'images revient donc, d'après l'équation (2.10), à identifier dans l'espace des fréquences un plan de vitesse d'équation

$$v^Tk + w = 0. \tag{2.11}$$

L'estimation du mouvement est extrait de l'équation (2.11) par filtrage spatio-temporel. Le problème de l'ouverture est contourné en supposant le flot constant sur les supports des ces filtres. Cette hypothèse a pour conséquence de lisser le flot à la fois en espace et en temps et peut parfois poser des problèmes pour des séquences d'images rapides et saccadées. Ces méthodes sont divisées en deux branches suivant que l'estimation du flot est faite sur l'énergie [Hee88] ou la phase [FJ90] du signal filtré. Ces dernières permettent d'obtenir de très bons résultats mais sont très coûteuses en temps de calcul.

2.2.3 Méthodes différentielles

Les méthodes différentielles calculent le flot optique de l'image à partir de l'E.C.M.A. (2.5). Il existe deux approches différentes pour résoudre cette équation : les approches denses qui calculent le flot optique sur l'intégralité de l'image et les méthodes dites locales qui estiment le mouvement en chaque point de façon indépendante. Le problème de l'ouverture est contourné en ajoutant une hypothèse de régularité spatiale dans le cas des méthodes denses et une contrainte de constance sur un voisinage dans le cas des méthodes locales.

Méthodes variationnelles, régularisation sur l'ensemble de l'image

Les principes fondamentaux de l'estimation du flot optique par des méthodes variationnelles ont été proposés par Horn et Schunck en 1981 [HS81]. L'idée est de définir une fonctionnelle composée de deux termes. Le premier terme s'appuie sur l'hypothèse de conservation de la luminosité et est définie comme suit

$$H_1(\overrightarrow{v}) = \int\int \left(\overrightarrow{\nabla} I.\overrightarrow{v} + \frac{\partial I}{\partial t}\right)^2 dxdy. \tag{2.12}$$

Le deuxième terme dit terme de régularisation pénalise les variations du premier ordre du champ de vecteur $\overrightarrow{v}$

$$H_2(\overrightarrow{v}) = \int\int \|\overrightarrow{\nabla} v\|^2 dxdy. \tag{2.13}$$

Horn et Schunck cherchent alors le champ de vecteurs $\overrightarrow{v}$ minimisant la fonctionnelle :

$$E(\overrightarrow{v}) = H_1(\overrightarrow{v}) + \alpha^2 H_2(\overrightarrow{v}) \tag{2.14}$$

où α est un paramètre fixé empiriquement permettant de pondérer les effets du terme (H_2). Ainsi, Horn et Schunck cherchent le champ de vecteurs $\overrightarrow{v}$ vérifiant au mieux la contrainte de flot optique (terme H_1) et tel que ses dérivées soient les plus faibles possible (terme H_2). L'inconvénient majeur dans cette méthode est l'introduction du terme de Tikhonov (H_2) qui induit une forte pénalisation des gradients. Cette forte pénalisation conduit à un lissage excessif du flot optique. C'est pourquoi, il n'est pas possible de retrouver les discontinuités spatiales du flot optique.

Depuis, différents auteurs ont proposé d'autres termes de régularisation permettant de préserver les discontinuités du flot optique en introduisant des normes dites "robustes" [BA93]. Le principe est de modifier le terme de pénalisation en

$$H_2(\overrightarrow{v}) = \int\int [\phi(\|\nabla v_1\|) + \phi(\|\nabla v_2\|)] dxdy, \tag{2.15}$$

Auteurs	$\phi(s)$
Rudin	s
Geman et Reynolds	$\frac{\lvert s\rvert}{r+\lvert s\rvert}$
Geman et McClure	$\frac{s^2}{r^2+s^2}$
Malik et Perona	$\log(1+s^2)$
Green	$2\log(\cosh(s))$
Aubert	$\sqrt{r^2+s^2}$

TABLE 2.1 – Quelques fonctions de régularisation

où ϕ est une fonction permettant de tenir compte des discontinuités du flot (Tab 2.1). Par exemple, Cohen [Coh93] effectue la régularisation au moyen de la norme L_1, c'est-à-dire qu'il pose $\phi(s) = s$ au lieu de la norme L_2 dans le cas de la méthode de Horn et Schunck. Citons aussi Kornprobst et al. [DKA95], qui plus récemment ont introduit une fonctionnelle du type

$$\begin{aligned} E(\overrightarrow{v}) &= \int\int \left|\overrightarrow{\nabla} I.\overrightarrow{v} + \tfrac{\partial I}{\partial t}\right| dxdy + \alpha^r \textstyle\sum_{j=1}^{2} \int\int \phi\left(\|\nabla v_j\|\right) dxdy \\ &\quad +\alpha^h \int\int c(x,y)\|\overrightarrow{v}\|^2 dxdy, \end{aligned} \tag{2.16}$$

où c est une fonction qui pénalise les zones où le gradient de l'image est faible, c'est-à-dire où les contrastes sont peu marqués et où α^r et α^h sont deux constantes fixées.

Régularisation sur un voisinage

A la différence des méthodes variationnelles qui calculent un champ de vecteurs dense sur l'ensemble de l'image, il existe des méthodes de calcul local du flot optique.

Lucas et Kanade [LK81] furent les premiers à proposer une telle méthode. Pour cela, ils minimisent une énergie composée uniquement du terme exprimant l'équation de contrainte du flot optique. Pour contourner le problème de l'indétermination du flot optique, ils supposent le flot $\overrightarrow{v}$ constant sur un petit voisinage du point (x, y). Cette fonctionnelle est définie de la façon suivante :

$$\sum_{(p,q)\in\Omega} W^2(p,q)\left[\overrightarrow{\nabla} I(p,q).\overrightarrow{v}(p,q) + \frac{\partial I}{\partial t}(p,q)\right]^2 \tag{2.17}$$

où Ω est un voisinage du point (x, y) et $W(x, y)$ est une gaussienne.

2.3 Modélisations paramétriques du mouvement

Une approche différente pour caractériser le mouvement dans une séquence d'images consiste à déterminer un modèle spatial pour le flot. Le nombre de paramètres à estimer est alors moins important et permet d'obtenir une estimation plus robuste du mouvement global dans la scène. Cette approche est très utile dans les problèmes d'indexation de vidéos ou de reconnaissance d'activité [BP02b] [Gel98]. Cette information permet d'identifier une vidéo dans une base de données, de localiser un passage précis dans celle-ci ou encore d'en faire un résumé.

Le problème posé dans cette section est le suivant : il s'agit d'estimer le mouvement d'une partie Ω de l'image qui par hypothèse peut être décrit par une modélisation paramétrique. Le type de modélisation choisi dépend de l'application et de la connaissance *a priori* sur la région Ω. Une modélisation polynomiale peut suffire dans le cas où la région Ω correspond à un objet rigide ou que l'on recherche le déplacement de la caméra dans une scène dans laquelle peu d'objets sont en mouvement. Au contraire, dans le cas où Ω correspond à l'image entière et que la scène est composée de nombreux mouvements différents qui ne permettent pas d'extraire un mouvement dominant, ou bien que Ω subit un mouvement complexe, une modélisation plus riche est nécessaire.

2.3.1 Modélisations polynomiales

Les modèles paramétriques polynomiaux considèrent que le champ de vitesse peut se décomposer sous la forme (Cf. Annexe A)

$$\overrightarrow{v}(x, y, \Theta) = \sum_{(i,j)\in\Lambda} \Theta x^i y^j \quad \text{où} \quad \Theta = \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix}_{ij} \text{ et } (x, y) \in \Omega. \tag{2.18}$$

Ceci correspond à des transformations géométriques plus ou moins complexes et est suffisant si l'image n'est composée que d'objets rigides ou peu déformables en déplacement. Néanmoins, étant donné que les déplacements d'objets indépendants ne peuvent pas être décrits par une même quantité Θ, il est nécessaire d'avoir un découpage de l'image en régions homogènes au sens du mouvement. Ces différentes régions peuvent être obtenues par une segmentation spatiale de l'image [Gel98] ou déterminées simultanément avec le calcul de Θ [OB98].
Dans le cas de mouvements complexes : mouvement de feuilles, de flammes, de fluides, ces méthodes ne sont pas appropriées. On préfère alors utiliser des approches reposant sur la décomposition du champ de vitesses sur une base de fonctions.

2.3.2 Modélisation sur une base de fonctions

La modélisation du flot optique à partir de fonctions permet de modéliser des mouvements plus complexes qu'avec des modèles polynomiaux. Le principe est de modéliser le flot optique sous la forme

$$\overrightarrow{v}(x,y) = \sum_{k,l} c_{k,l}\phi_{kl}(x,y),$$

où $(\phi_{kl})_{kl\in\Lambda}$ est une famille de fonctions.
De nombreux modèles paramétriques de mouvement ont été développés dans la littérature. Le choix des fonctions ϕ_{kl} est assez vaste. Rakshit et Anderson [RA97] ont utilisé des fonctions interpolatrices, Bruno et Pellerin [BP02b] utilisent des B-splines...
Le nombre de fonctions de base utilisées pour calculer le flot optique définit la précision du modèle : en augmentant le nombre de degrés de liberté, la solution obtenue peut suivre au mieux les variations spatiales du flot optique ce qui permet d'obtenir une modélisation globale du champ de vitesses (Ω correspondant à l'image entière). En contrepartie, plus il y a de paramètres à estimer, plus le processus d'estimation est instable.

2.3.3 Estimation des paramètres

Nous montrons dans ce paragraphe comment calculer le mouvement apparent dans l'image à partir d'une modélisation polynomiale ou par interpolation. Nous venons de voir que de façon générale les modélisations paramétriques consistent à écrire le flot optique comme une somme pondérée de fonctions de base $\phi(p)$ avec $p = (x,y) \in \Omega$,

$$\overrightarrow{v}(p,\Theta) = \sum_{k=0}^{N} c_k\phi_k(p) \quad avec \quad c_k = \left(c_x^k, c_y^k\right)^T. \tag{2.19}$$

Le domaine Ω peut être l'image entière ou une partie de celle-ci et correspond à la région où l'on cherche à estimer le flot optique. Le mouvement dans cette région est caractérisé par le vecteur de paramètres du mouvement $\Theta = [c_0, \cdots, c_N]^T$. En considérant toujours l'hypothèse de conservation de la luminosité (2.2), l'estimation de Θ est obtenue en minimisant l'énergie E suivante :

$$E(\Theta) = \sum_{p=(x,y)\in\,\Omega} [r(p,\Theta)]^2, \tag{2.20}$$

$$\text{où } r(p,\Theta) = I(p + \overrightarrow{v}(p,\Theta), t+1) - I(p,t).$$

Cependant, l'expérience montre que déterminer les paramètres de Θ à partir de la minimisation de cette énergie (2.20) engendre beaucoup d'erreurs dues au non respect de

l'hypothèse de conservation de luminosité et de la non validité de la modélisation en certains points de Ω. C'est pourquoi plusieurs auteurs utilisent un M-estimateur $\rho(x, C)$ et minimisent alors l'énergie E

$$E(\Theta) = \sum_{(x,y)\in\,\Omega} \rho(r(p, \Theta), C). \tag{2.21}$$

Citons par exemple [OB95, BP02a] qui utilise le M-estimateur de Tukey

$$\rho(x, C) = \begin{cases} \frac{x^6}{6} - \frac{C^2 x^4}{2} \text{ si } |x| < C \\ \frac{C^6}{6} \text{ sinon} \end{cases} . \tag{2.22}$$

et Black [BA93] qui utilise le M-estimateur

$$\rho(x, C) = \frac{x^2}{C + x^2}. \tag{2.23}$$

L'utilisation d'un M-estimateur permet une estimation plus robuste du flot optique vis-à-vis des données aberrantes relativement aux hypothèses faites. La minimisation de (2.21) est faite de façon itérative par une méthode de Gauss-Newton. On décompose l'estimation de Θ comme la somme d'une estimation initiale et d'un incrément $\delta\Theta$:

$$E(\Theta + \delta\Theta) = \sum_{p\,\in\,\Omega} \rho(r(p, \Theta + \delta\Theta), C). \tag{2.24}$$

Etant donné une estimation $\widehat{\Theta}_k$ (initialement égale à 0) le problème est d'estimer le l'incrément $\delta\Theta_k$. Cette valeur est obtenue à partir de la linéarisation de (2.24) et donc par la minimisation de

$$\widetilde{E}(\widehat{\Theta}_k + \delta\Theta_k) = \sum_{p\,\in\,\Omega} \rho(r(p, \widehat{\Theta}_k) + \overrightarrow{v}(p, \delta\Theta_k) \cdot \overrightarrow{\nabla} I(p + v(p, \widehat{\Theta}_k), t + 1), C) \tag{2.25}$$

obtenue par une technique des moindres-carrés pondérés itérés. L'estimé de Θ devient alors $\widehat{\Theta}_{k+1} = \widehat{\Theta}_k + \widehat{\delta\Theta}_k$.

2.4 Estimation du mouvement dans l'image par analyse d'ondelettes

Dans un premier temps, nous allons détailler la méthode de Bernard [Ber99] sur l'estimation du flot optique dans une séquence d'images à partir d'une base d'ondelettes. Puis nous adapterons celle-ci à l'estimation du mouvement d'une région de l'image.

2.4.1 Estimation du flot optique par projection sur une base de fonctions

L'approche que nous défendons dans cette thèse est fondée sur les travaux de Weber et Malik [WM95], Bernard [Ber98], Bruno [BP02a] et se rapproche de la méthode de Lucas et Kanade [LK81] vue précédemment. Nous étudions l'équation de contrainte du flot optique à partir de filtrages spatiaux. Considérons une base de fonctions $(f^n)_{n=1\cdots N}$ de $L^2(\mathbb{R}^2)$ et construisons une famille de fonctions translatées (f_u^n) :

$$f_u^n(x,y) = f^n(x-u_1, y-u_2), \quad \forall n = 1\cdots N, \quad u = (u_1, u_2). \tag{2.26}$$

Nous estimons le flot optique au point u en projetant l'équation de contrainte du flot optique (2.5) sur ces différentes fonctions, ce qui permet d'obtenir un système de N équations

$$\int\int \left(\overrightarrow{\nabla} I.\overrightarrow{v}((x,y),t) + \frac{\partial I}{\partial t}\right) f_u^n(x,y)dxdy = 0, \quad \forall n = 1\cdots N. \tag{2.27}$$

En posant $\langle f, g\rangle = \int\int f(x,y)\overline{g(x,y)}dxdy$, le système d'équations devient

$$\langle \frac{\partial I}{\partial x} v_1, f_u^n\rangle + \langle \frac{\partial I}{\partial y} v_2, f_u^n\rangle + \langle \frac{\partial I}{\partial t}, f_u^n\rangle = 0, \quad \forall n = 1\cdots N. \tag{2.28}$$

A ce stade, pour contourner le problème de l'ouverture une nouvelle hypothèse sur le flot est nécessaire. Supposons le flot constant sur les supports des ondelettes.
Cela se traduit mathématiquement par

$$\overrightarrow{v}(x,y) = \begin{pmatrix} v_1(u) \\ v_2(u) \end{pmatrix}, \quad \forall (x,y) \in \text{support de } f_u^n. \tag{2.29}$$

Grâce à cette hypothèse, l'équation (2.28) devient

$$\begin{array}{c} \forall n = 1\cdots N, \\ \langle \frac{\partial I}{\partial x} v_1(u), f_u^n\rangle + \langle \frac{\partial I}{\partial y} v_2(u), f_u^n\rangle = -\langle \frac{\partial I}{\partial t}, f_u^n\rangle, \end{array} \tag{2.30}$$

soit,

$$\begin{array}{c} \forall n = 1\cdots N, \\ v_1(u)\langle \frac{\partial I}{\partial x}, f_u^n\rangle + v_2(u)\langle \frac{\partial I}{\partial y}, f_u^n\rangle \quad = \quad -\langle \frac{\partial I}{\partial t}, f_u^n\rangle, \end{array} \tag{2.31}$$

et par intégration par parties,

$$\begin{array}{c} \forall n = 1\cdots N, \\ v_1(u)\langle I, \frac{\partial f_u^n}{\partial x}\rangle + v_2(u)\langle I, \frac{\partial f_u^n}{\partial y}\rangle \quad = \quad \langle \frac{\partial I}{\partial t}, f_u^n\rangle. \end{array} \tag{2.32}$$

Par conséquent, déterminer le flot optique au point $u = (u_1, u_2)$ revient à résoudre le système linéaire d'inconnue $V = (v_1(u), v_2(u))^T$ suivant

$$M_u V = P_u, \tag{2.33}$$

où M_u est la matrice de taille $2 \times N$ définie par :

$$M_u = \begin{pmatrix} \langle I, \frac{\partial f_u^1}{\partial x} \rangle & \langle I, \frac{\partial f_u^1}{\partial y} \rangle \\ \vdots & \vdots \\ \langle I, \frac{\partial f_u^N}{\partial x} \rangle & \langle I, \frac{\partial f_u^N}{\partial y} \rangle \end{pmatrix},$$

et

$$P_u = \begin{pmatrix} \langle \frac{\partial I}{\partial t}, f_u^1 \rangle \\ \vdots \\ \langle \frac{\partial I}{\partial t}, f_u^N \rangle \end{pmatrix}.$$

Si $N \geq 2$, le système (2.43) est résolu par moindres carrés,

$$V = \left(M_u^T M_u\right)^{-1} M_u^T P_u. \tag{2.34}$$

Cette méthode a deux avantages considérables : elle calcule localement le flot par simple résolution d'un système linéaire et grâce à l'intégration par parties, le calcul, souvent bruité du gradient de l'image, est évité. Cependant un problème inhérent à ce type d'approche est ajouté. En effet, pour affaiblir l'hypothèse (2.29), nous serions tentés d'utiliser des fonctions ayant un support le plus petit possible. Or, l'échantillonnage temporel de la séquence nous oblige à faire le contraire.

Aliasage temporel

Étant donné que la séquence est échantillonnée en temps, nous sommes contraints d'approcher la dérivée temporelle de l'image par une différence finie du type

$$\frac{\partial I}{\partial t} \simeq I(t+1) - I(t). \tag{2.35}$$

Regardons l'influence de l'approximation (3.8) en fonction de la taille du support des fonctions. Pour cela, considérons les fonctions contractées f_{su}^n définies de la façon suivante,

$$f_{su}^n(x, y) = \frac{1}{s} f_u^n(\frac{x}{s}, \frac{y}{s}) = \frac{1}{s} f^n(\frac{x - u_1}{s}, \frac{y - u_2}{s})$$

et supposons que la séquence d'images soit régie par un mouvement de translation uniforme $\overrightarrow{v} = (v_1, v_2)$

$$I((x, y), t) = I(x - tv_1, y - tv_2);$$

les coefficients de P_u sont alors approchés par

$$\langle \frac{\partial I}{\partial t}, f_{su}^n \rangle \simeq \langle I(. - (t+1)v_1, . - (t+1)v_2) - I(. - tv_1, . - tv_2), f_{su}^n \rangle, \tag{2.36}$$

soit par changement de variable,

$$\langle \frac{\partial I}{\partial t}, f_{su}^n \rangle \simeq \langle I(t), f_{su}^n(. + v_1, . + v_2) - f_{su}^n \rangle. \tag{2.37}$$

Donc d'après l'équation de contrainte du flot optique et par intégration par parties, on déduit l'approximation

$$\langle I(t), \overrightarrow{v}.\overrightarrow{\nabla} f_{su}^n \rangle \simeq \langle I(t), f_{su}^n(. + v_1, . + v_2) - f_{su}^n \rangle. \tag{2.38}$$

Ainsi, faire l'approximation (2.35) conduit à l'approximation suivante

$$\overrightarrow{v}.\overrightarrow{\nabla} f_{su}^n \simeq f_{su}^n(. + v_1, . + v_2) - f_{su}^n. \tag{2.39}$$

Or, d'après un développement de Taylor, l'erreur relative commise en faisant cette approximation n'est valable que si on impose la relation

$$||\overrightarrow{v}|| \leq C \times s, \tag{2.40}$$

où C est une constante dépendant des fonctions f^n.
Nous venons donc de montrer que l'échantillonnage temporel de la séquence d'images impose une contrainte sur le flot estimé. Pour calculer des grands déplacements dans la scène, nous sommes obligés d'utiliser des fonctions de base avec de grands supports, mais dans ce cas, l'hypothèse (2.29) est trop forte. Il faut donc faire un juste compromis entre ces deux contraintes. Comme la taille des supports ne peut pas être choisie *a priori*, un schéma de raffinement multi-résolution est nécessaire. C'est pour cette raison que Bernard propose d'utiliser comme base de fonctions une base d'ondelettes, un outil naturel pour une approche multi-résolution.

Utilisation d'une base d'ondelettes

Considérons donc une base d'ondelettes $(\psi^n)_{n=1\cdots N}$ et la famille d'ondelettes discrètes associées
$(\psi_{jk}^n)_{n=1\cdots N, j\in\mathbb{Z}, k\in\mathbb{Z}^2}$ définie par

$$\psi_{jk}^n(x,y) = 2^{-j}\psi^n(2^{-j}x - k_1, 2^{-j}y - k_2), \tag{2.41}$$

où j est l'indice de résolution et $k = (k_1, k_2)$ l'indice de translation.
Le problème de l'aliasage temporel nous incite à utiliser des ondelettes ayant un grand

support (j grand) pour estimer les grands déplacements dans la scène et des ondelettes avec un petit support (j petit) pour les petites vitesses afin d'affaiblir l'hypothèse (2.29).

A une échelle grossière J, nous obtenons en chaque point $(2^J k_1, 2^J k_2)$ le système

$$M_{Jk} V_{Jk} = P_{Jk}, \tag{2.42}$$

où

$$M_{Jk} = \begin{pmatrix} \langle I, \frac{\partial \psi^1_{Jk}}{\partial x} \rangle & \langle I, \frac{\partial \psi^1_{Jk}}{\partial y} \rangle \\ \vdots & \vdots \\ \langle I, \frac{\partial \psi^N_{Jk}}{\partial x} \rangle & \langle I, \frac{\partial \psi^N_{Jk}}{\partial y} \rangle \end{pmatrix},$$

$$P_{Jk} = \begin{pmatrix} \langle I(t+1) - I(t), \psi^1_{Jk} \rangle \\ \vdots \\ \langle I(t+1) - I(t), \psi^N_{Jk} \rangle \end{pmatrix} \quad \text{et} \quad V_{Jk} = (v_1(2^J k_1, 2^J k_2), v_2(2^J k_1, 2^J k_2))^T.$$

La résolution de ce système permet d'obtenir une approximation grossière du mouvement sur une grille de résolution 2^J. Le raffinement de l'estimation se fait par compensation descendante.

Supposons qu'une estimation du mouvement à une échelle $j+1$ nous donne au point $(x, y) = 2^j(k_1, k_2) = 2^{j+1}(\frac{k_1}{2}, \frac{k_2}{2})$ une vitesse $\overrightarrow{v}^{j+1}(x, y)$. Le raffinement à l'échelle j consiste à estimer l'erreur commise entre le flot estimé précédemment et le flot réel. Pour éviter de calculer des coefficients d'ondelettes non entiers, Bernard propose de décomposer $\overrightarrow{v}^{j+1}$ de la façon suivante

$$\overrightarrow{v}^{j+1} = \overrightarrow{v}^N + \overrightarrow{v}^r,$$

où

$$\begin{cases} \overrightarrow{v}^N \in 2^j \mathbb{Z}^2 \\ \overrightarrow{v}^r \in 2^j[-0.5, 0.5)^2 \end{cases}.$$

Le raffinement de l'estimation du flot à l'échelle j au point $(x, y) = 2^j(k_1, k_2)$ consiste à estimer le résidu $\overrightarrow{v}^r = (v_1^r, v_2^r)$. Ce résiduel est obtenu par la résolution du système de N équations

$$M_{jk} \overrightarrow{v}^r = \widetilde{P}_{jk}, \tag{2.43}$$

où

$$\widetilde{P}_{jk} = \begin{pmatrix} \langle I(t+1), \psi^1_{j(k+2^{-j}v^N)} \rangle - \langle I(t), \psi^1_{jk} \rangle \\ \vdots \\ \langle I(t+1), \psi^N_{j(k+2^{-j}v^N)} \rangle - \langle I(t), \psi^N_{jk} \rangle \end{pmatrix}.$$

Si ce système (2.43) est bien conditionné et que sa solution au sens des moindres carrés $\overrightarrow{\widetilde{v}}^r = (M_{jk}^T M_{jk})^{-1} M_{jk}^T \widetilde{P}_{jk}$ ne transgresse pas la contrainte d'aliasage temporel (2.40), la nouvelle estimation du flot au point (x, y) devient

$$\overrightarrow{v}^j = \overrightarrow{v}^N + \overrightarrow{\widetilde{v}}^r.$$

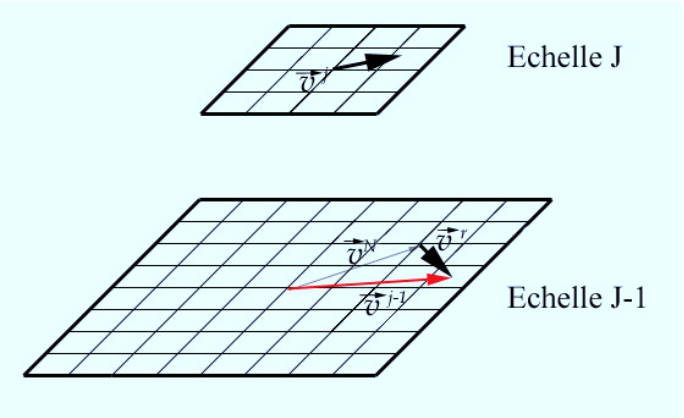


FIGURE 2.2 – Estimation du mouvement par compensation entre deux échelles.

Dans le cas contraire, l'estimation du flot optique en ce point n'est pas réévaluée. On procède ainsi de suite de l'échelle $j = J$ à l'échelle la plus fine. Cet algorithme est résumé dans le Tableau 2.2.

Cette approche multi-résolution permet d'obtenir une estimation du mouvement en contournant le problème de l'aliasage temporel. L'utilisation d'une base d'ondelettes comme base de projection permet à la fois de résoudre le problème de l'ouverture et de créer la pyramide d'images pour la multi-résolution.

Il est à noter que l'utilisation d'ondelettes réelles comme base de projection engendrent des systèmes d'équations (2.43) souvent mal conditionnés. Bernard propose donc de considérer des ondelettes analytiques (Cf. Chapitre 1 paragraphe 1.7). Celles-ci ont aussi l'avantage de réduire l'effet de l'aliasage temporel à partir d'un "décentrement de gamme" [Ber99].

Résultats comparatifs

Nous donnons ici à titre d'exemple les mesures comparatives effectuées sur différentes approches rencontrées dans la littérature. L'erreur entre le flot réel $\overrightarrow{v} = (v_1, v_2)$ et le flot estimé $\overrightarrow{\widetilde{v}} = (\widetilde{v}_1, \widetilde{v}_2)$ est mesurée par l'erreur angulaire définie par Fleet et Jepson [FJ90],

$$\alpha = E_{ang}(\overrightarrow{v}, \overrightarrow{\widetilde{v}}) = \arccos\left(\frac{v_1\widetilde{v}_1 + v_2\widetilde{v}_2 + 1}{\sqrt{v_1^2 + v_2^2 + 1}\sqrt{\widetilde{v}_1^2 + \widetilde{v}_2^2 + 1}}\right). \tag{2.44}$$

Cette mesure prend en compte l'erreur commise sur l'orientation et l'amplitude des vecteurs de vitesse. Une autre mesure importante entre en jeu pour la comparaison des techniques qui est la densité de pixels de l'image sur lesquels le flot optique a été estimé. Le tableau (2.4.1) donne les résultats obtenus sur les séquences synthétiques "Yosemite" et "Arbre en translation" mises à disposition par Barron et *al.* sur le site *ftp.csd.uwo.ca*.

Algorithme Estimation multirésolution du flot optique$(I(t), I(t+1), J)$

$I(t)$ image de la séquence à l'instant t
$I(t+1)$ image de la séquence à l'instant $t+1$
J échelle grossière

Initialisation :
Calcul des coefficients d'ondelettes de l'échelle 0 à J.

- Pour j allant de J à 0 faire
  - ∘ pour $k = (k_1, k_2)$ tel que $(2^j k_1, 2^j k_2)$ est un point de l'image
    - · Si $\det(M_{jk}^T M_{jk}) < \mathrm{Cst}_1 \mathrm{Tr}(M_{jk}^T M_{jk})^2$ % $M_{jk}^T M_{jk}$ est mal conditionnée
      $v^j(k) = v^{j+1}(k/2)$ % aucune estimation de la vitesse
      % n'est calculée à cette échelle
    - · Sinon
      % Décomposition de la vitesse précédemment
      % calculée sur la grille de résolution j
      $v^{j+1}(k/2) = v^N + v^r$
      $\Theta = \left(M_{jk}^T M_{jk}\right)^{-1} M_{jk}^T \widetilde{P_{jk}}$
      si $||\Theta|| > 2^j \mathrm{Cst}_2$ % transgression de la contrainte d'aliasage
        $v^j(k) = v^{j+1}(k/2)$
      sinon
        $v^j(k) = v^N + \Theta$
      finsi
    - · finSi
  - ∘ finpour
- finPour

TABLE 2.2 – Algorithme d'estimation du flot optique par multi-résolution.

De façon générale, on remarque que la méthode de Bernard est un bon compromis entre exactitude et densité de l'estimation du flot optique. De plus, grâce à l'utilisation d'ondelettes discrètes comme base de fonctions et de l'algorithme rapide de décomposition en ondelettes de Mallat [Mal89], cette méthode est une méthode rapide d'estimation du flot optique. En effet, la complexité de l'algorithme est de l'ordre $\mathcal{O}(N)$ pour des images composées de N pixels. Cette méthode a en outre l'avantage de pouvoir facilement s'adapter

Séquence	auteur	erreur angulaire	densité
"Yosemite"	Fleet et Jepson	4,25 °	34,1%
	Lucas et Kanade	4,10 °	35,1%
	Weber et Malik	4,31 °	64,2%
	Bruno	4,33 °	89,1%
	Bernard	6,50 °	96,5%
"Arbre en translation"	Fleet et Jepson	0,32 °	74,5%
	Lucas et Kanade	0,66 °	39,8%
	Weber et Malik	0,49 °	96,8%
	Bruno	0,83 °	94,3%
	Bernard	0,78 °	99,3%

TABLE 2.3 – Erreur angulaire et densité de l'estimation de différentes méthodes d'estimation du flot optique sur la séquence "Yosemite" et "Arbre en translation".

au problème de l'estimation du mouvement d'une région quelconque dans l'image. C'est pourquoi, nous avons choisi de généraliser cette méthode pour estimer le mouvement d'une région de l'image comme nous allons le voir dans le prochain paragraphe.

2.4.2 Estimation du mouvement d'une région de l'image

Dans ce paragraphe, nous nous intéressons à l'estimation du mouvement d'une région de l'image. Cette région peut soit correspondre à l'image entière, à une partie précise de l'image ou tout simplement réduite à un pixel. Dans la suite de ce paragraphe, nous ferons l'hypothèse que le mouvement de cette région peut être décrit par un modèle polynomial. Bien que cette hypothèse peut paraître très restrictive, puisqu'elle ne permet pas de calculer des mouvements complexes, elle est cependant suffisante pour l'application visée dans cette thèse. Le problème posé ici est donc de trouver les coefficients d'un modèle paramétrique qui décrivent au mieux le mouvement d'une région notée $\mathcal{R}$.

Pour ne pas alourdir les notations, nous supposerons le mouvement de la région régi par un mouvement affine [1]. Cette hypothèse permet de prendre en compte un grand nombre de mouvements sur $\mathcal{R}$ avec peu de coefficients (mouvement de translation, mouvement de

1. La méthode proposée ici se généralise à toute modélisation polynomiale du mouvement de la région, nous utiliserons un modèle quadratique dans le chapitre 7.

rotation, divergence, cisaillement ...). Nous supposons donc que

$$\begin{gathered}\forall u = (x,y) \in \mathcal{R},\\ \overrightarrow{v}(x,y) = \begin{pmatrix} v_1(x,y) \\ v_2(x,y) \end{pmatrix} = \begin{pmatrix} a_1x + a_2y + a_3 \\ a_4x + a_5y + a_6 \end{pmatrix} = B(x,y).\Theta,\\ \Theta = (a_1, a_2, a_3, a_4, a_5, a_6)^T.\end{gathered} \tag{2.45}$$

Le problème posé est donc de déterminer le vecteur Θ qui vérifie "au mieux" l'hypothèse de constance de luminosité sur $\mathcal{R}$, c'est-à-dire

$$I(u,t) = I(u + B(u).\Theta, t+1) \quad u \in \mathcal{R}. \tag{2.46}$$

Pour ce faire, comme nous avons vu précédemment, il existe les méthodes basées sur la différence d'images déplacées qui recherchent Θ solution de

$$\min \sum_{\mathcal{R}} \rho(I(u,t) - I(u + B(u).\Theta, t+1)) \tag{2.47}$$

où ρ est une fonction robuste qui permet de ne pas considérer les points où l'hypothèse de mouvement affine est fausse [Odo94], [Gel98]. On trouve aussi des méthodes différentielles [BA96] où la fonctionnelle à minimiser est définie de la façon suivante

$$\min \sum_{\mathcal{R}} \rho(\overrightarrow{\nabla} I.B(u)\Theta + \frac{\partial I}{\partial t}). \tag{2.48}$$

A cause de la non linéarité en Θ des méthodes basées sur la minimisation de l'énergie (2.47), elles requièrent des algorithmes de descentes qui alourdissent le temps de calcul. A contrario, les méthodes différentielles sont plus simples à résoudre mais souffrent du problème d'aliasage temporel. L'inconvénient majeur de ces deux méthodes est que la région $\mathcal{R}$ doit être suffisamment grande pour que l'estimation ne soit pas trop sensible au bruit et pour que suffisamment de contraintes puissent être considérées.

La méthode que nous proposons ici a l'avantage de pouvoir calculer le mouvement d'une région réduite à un pixel et donc de contourner le problème de l'ouverture. De plus, le fait de ne pas calculer le gradient de l'image réduit sa sensibilité au bruit.
Reprenons l'équation (2.28)

$$\langle \frac{\partial I}{\partial x} v_1, f_u^n \rangle + \langle \frac{\partial I}{\partial y} v_2, f_u^n \rangle + \langle \frac{\partial I}{\partial t}, f_u^n \rangle = 0 \quad \forall n = 1 \cdots N; \tag{2.49}$$

de l'hypothèse du mouvement affine de $\mathcal{R}$, nous déduisons

$$\begin{gathered}\forall u = (x,y) \in \mathcal{R}_f = \{u \text{ dans l'image} \mid \text{supp}(f_u) \subset \mathcal{R}\} \quad \forall n = 1 \cdots N,\\ \langle \tfrac{\partial I}{\partial x}(a_1x + a_2y + a_3), f_u^n \rangle + \langle \tfrac{\partial I}{\partial y}(a_4x + a_5y + a_6), f_u^n \rangle = -\langle \tfrac{\partial I}{\partial t}, f_u^n \rangle\end{gathered} \tag{2.50}$$

Soit

$$\begin{array}{l} \forall u = (x,y) \in \mathcal{R}_f \quad \forall n = 1 \cdots N, \\ a_1 \langle \frac{\partial I}{\partial x} x, f_u^n \rangle + a_2 \langle \frac{\partial I}{\partial x} y, f_u^n \rangle + a_3 \langle \frac{\partial I}{\partial x}, f_u^n \rangle + \\ a_4 \langle \frac{\partial I}{\partial y} x, f_u^n \rangle + a_5 \langle \frac{\partial I}{\partial y} y, f_u^n \rangle + a_6 \langle \frac{\partial I}{\partial y}, f_u^n \rangle \quad = \quad -\langle \frac{\partial I}{\partial t}, f_u^n \rangle, \end{array} \tag{2.51}$$

et par intégration par parties,

$$\begin{array}{l} \forall u = (x,y) \in \mathcal{R}_f \quad \forall n = 1 \cdots N, \\ a_1 \left[\langle xI, \frac{\partial f_u^n}{\partial x} \rangle + \langle I, f_u^n \rangle \right] + a_2 \langle yI, \frac{\partial f_u^n}{\partial x} \rangle + a_3 \langle I, \frac{\partial f_u^n}{\partial x} \rangle + \\ a_4 \langle xI, \frac{\partial f_u^n}{\partial y} \rangle + a_5 \left[\langle yI, \frac{\partial f_u^n}{\partial y} \rangle + \langle I, f_u^n \rangle \right] + a_6 \langle I, \frac{\partial f_u^n}{\partial y} \rangle \quad = \quad \langle \frac{\partial I}{\partial t}, f_u^n \rangle. \end{array} \tag{2.52}$$

En théorie, si $N \geq 6$, un seul point de $\mathcal{R}_f$ peut suffire pour calculer Θ et donc estimer le mouvement de la région $\mathcal{R}$. Néanmoins, les bruits d'acquisition des images nous obligent, pour avoir une estimation plus correcte du mouvement, à utiliser le plus de points possible. En pratique, l'estimation de Θ au sens des moindres carrés à partir des $N \times \text{card}\mathcal{R}_f$ équations de (2.52) n'est pas adaptée au problème. En effet, certaines équations du système (2.52) ne sont parfois pas vérifiées. Par exemple, certains points de $\mathcal{R}$ peuvent ne pas vérifier la contrainte de constance de luminosité ou comme la région $\mathcal{R}$ n'est souvent pas connue de façon précise, certains points de celle-ci ne vérifient pas (2.45). C'est pourquoi, certaines équations du système (2.52) dites "*outliers*" ne sont pas valables. Une méthode pour ne pas considérer ces mesures aberrantes consiste à utiliser un estimateur robuste pour résoudre le système sur-déterminé.

Estimation robuste

Nous devons donc résoudre le système

$$\begin{array}{c} \forall u \in \mathcal{R}_f \\ M_u \Theta = P_u, \end{array} \tag{2.53}$$

où M_u est la matrice de taille $6 \times N$ définie par

$$M_u = \begin{pmatrix} \left[\langle xI, \frac{\partial f_u^1}{\partial x} \rangle + \langle I, f_u^1 \rangle \right] & \langle yI, \frac{\partial f_u^1}{\partial x} \rangle & \langle I, \frac{\partial f_u^1}{\partial x} \rangle & \langle xI, \frac{\partial f_u^1}{\partial y} \rangle & \left[\langle yI, \frac{\partial f_u^1}{\partial y} \rangle + \langle I, f_u^1 \rangle \right] & \langle I, \frac{\partial f_u^1}{\partial y} \rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left[\langle xI, \frac{\partial f_u^N}{\partial x} \rangle + \langle I, f_u^N \rangle \right] & \langle yI, \frac{\partial f_u^N}{\partial x} \rangle & \langle I, \frac{\partial f_u^N}{\partial x} \rangle & \langle xI, \frac{\partial f_u^N}{\partial y} \rangle & \left[\langle yI, \frac{\partial f_u^N}{\partial y} \rangle + \langle I, f_u^N \rangle \right] & \langle I, \frac{\partial f_u^N}{\partial y} \rangle \end{pmatrix},$$

$$P_u = \begin{pmatrix} \langle \frac{\partial I}{\partial t}, f_u^1 \rangle \\ \vdots \\ \langle \frac{\partial I}{\partial t}, f_u^N \rangle \end{pmatrix},$$

et où certains points u de $\mathcal{R}_f$ sont des points aberrants.

Il existe dans la littérature de nombreuses méthodes d'estimation robuste (Cf. Annexe B). Nous optons ici pour l'utilisation du M-estimateur robuste "*biweight*" de Tukey. La solution du système sur-déterminé (2.53) est alors le minimum de la fonctionnelle

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_{u \in \mathcal{R}_f} \sum_{i=1}^{N} \rho(M_u^i \Theta - P_u^i, C), \tag{2.54}$$

avec

$$\rho(r, C) = \begin{cases} \frac{r^6}{6} - \frac{C^2 r^4}{2} + \frac{C^2 r^2}{2} & \text{si } |r| < C \\ \frac{C^6}{6} & \text{sinon,} \end{cases}$$

où M_u^i (respectivement P_u^i) est la i^{ème} ligne de la matrice M_u (respectivement P_u) et où C est la valeur à partir de laquelle le résidu $r_i = M_u^i \Theta - P_u^i$ est rejeté pour l'estimation de Θ. Cette valeur dépend du bruit d'acquisition et de la validité des hypothèses de constance de luminosité et de mouvement affine. Si le bruit est faible et que les hypothèses sont vérifiées, la valeur de C peut être choisie faible. Dans le cas contraire, il est préférable de choisir une valeur élevée. Pour obtenir une méthode s'adaptant automatiquement aux données, on peut choisir une valeur de C proportionnelle à la médiane des écarts absolus

$$\sigma = 1.483 \times \text{med}_i(|r_i - \text{med}_j(r_j)|).$$

[HW77] montrent que la valeur $C = 4,7\sigma$ permet d'obtenir une bonne efficacité du M-estimateur en cas de bruit gaussien.

Comme le résidu est linéaire, le minimum (2.54) est obtenu facilement par la méthode des moindres carrés pondérés itérés (M.C.P.I.) (Cf. Annexe B).

De cette façon, nous obtenons une estimation robuste du mouvement de la région $\mathcal{R}$ ne tenant pas compte des équations du système (2.52) non respectées.

Schéma multi-résolution

Comme pour l'estimation du flot optique, les problèmes d'aliasage temporel nous incitent à calculer le mouvement supposé affine de $\mathcal{R}$ à partir d'un schéma multi-résolution. Cette méthode de calcul multi-résolution du mouvement nous permettra une nouvelle fois d'estimer les grands déplacements à échelle grossière puis de raffiner cette estimation aux échelles plus fines.

Considérons donc de nouveau une base d'ondelettes discrètes (ψ_{jk}^n) et posons $\mathcal{R}_j = \{k = (k_1, k_2) | \text{ supp}(\psi_{jk}) \subset \mathcal{R} \, \forall n = 1 \cdots N\}$. A une échelle grossière J, l'hypothèse du mouvement affine nous fournit le système

$$\begin{gathered} \forall k \in \mathcal{R}_J, \\ M_{Jk} \Theta^J = P_{Jk}. \end{gathered} \tag{2.55}$$

La solution robuste

$$\Theta^J = \arg\min_{\Theta} \sum_{k\in\mathcal{R}_J} \sum_{i=1}^{N} \rho(M^i_{Jk}\Theta - P^i_{JK}, C), \tag{2.56}$$

permet d'obtenir une première estimation du mouvement affine de $\mathcal{R}$

$$\overrightarrow{v}^J(x,y) = B(x,y).\mathcal{P}^J(\Theta^J), \quad \forall (x,y) \in \mathcal{R}, \tag{2.57}$$

où $\mathcal{P}^J$ est l'opérateur de projection de l'échelle J sur l'échelle la plus fine,

$$\mathcal{P}^J(\Theta) = (a_1, a_2, 2^J a_3, a_4, a_5, 2^J a_6)^T.$$

Le passage de l'échelle J à une échelle $J-1$ se fait en décomposant le mouvement réel $\overrightarrow{v}$ de $\mathcal{R}$ sous la forme,

$$\overrightarrow{v}(x,y) = \overrightarrow{v}^J(x,y) + \overrightarrow{\widetilde{v}}^{J-1}(x,y)$$
$$\text{avec } ||\overrightarrow{\widetilde{v}}^{J-1}|| < ||\overrightarrow{v}^J||.$$

Posons alors l'image compensée

$$\widetilde{I}((x,y),t+1) = I((x + v_1^J(x,y), y + v_2^J(x,y)), t+1). \tag{2.58}$$

Le déplacement de la région $\mathcal{R}$ entre l'image $I(t)$ et $\widetilde{I}(t+1)$ est exactement de $\overrightarrow{\widetilde{v}}^{J-1}$. Cette vitesse est alors estimée à partir de la résolution robuste du système sur-déterminé suivant

$$\begin{gathered} \forall k \in \mathcal{R}_{J-1} \\ M_{(J-1)k}\Theta^{J-1} = \widetilde{P}_{(J-1)k}, \end{gathered} \tag{2.59}$$

où

$$\widetilde{P}_{jk} = \begin{pmatrix} \langle \widetilde{I}(t+1) - I(t), \psi^1_{jk} \rangle \\ \vdots \\ \langle \widetilde{I}(t+1) - I(t), \psi^N_{jk} \rangle \end{pmatrix} = \begin{pmatrix} \langle I(t+1), \psi^1_{j(k+2^{-j}v^{j+1})} \rangle - \langle I(t), \psi^1_{jk} \rangle \\ \vdots \\ \langle I(t+1), \psi^N_{j(k+2^{-j}v^{j+1})} \rangle - \langle I(t), \psi^N_{jk} \rangle \end{pmatrix},$$

les coefficients d'ondelettes non entiers de $\widetilde{P}_{jk}$ étant calculés par interpolation linéaire. Ainsi,

$$\overrightarrow{v}^{J-1}(x,y) = \overrightarrow{v}^J(x,y) + \overrightarrow{\widetilde{v}}^{J-1}(x,y) = B(x,y).(\mathcal{P}^J(\Theta^J) + \mathcal{P}^{J-1}(\Theta^{J-1}))$$

devient une estimation plus précise du mouvement de $\mathcal{R}$. On procède ainsi de suite de l'échelle $j = J$ à l'échelle la plus fine. Cet algorithme est résumé dans le Tableau 2.4.

Nous venons donc d'introduire une nouvelle méthode d'estimation de mouvement paramétrique d'une région. Cette méthode présente plusieurs intérêts :

Estimation robuste multirésolution du mouvement affine de $\mathcal{R}(I(t), I(t+1), J, \mathcal{R})$

$I(t)$ image de la séquence à l'instant t
$I(t+1)$ image de la séquence à l'instant $t+1$
J échelle grossière
$\mathcal{R}$ région de l'image où le mouvement est calculé

Initialisation :
Calcul des coefficients d'ondelettes de l'échelle 0 à J.
$\Theta = (0, 0, 0, 0, 0, 0)^T$

- Pour j allant de J à 1 faire
  - $\overrightarrow{v}^j = B.\mathcal{P}^j(\Theta)$
  - Evaluation de M_{jk} et $\widetilde{P}_{jk} \quad \forall k \in \mathcal{R}_j$
  - $\Delta\Theta =$ solution de (2.54)
  - Si $||B.\mathcal{P}^j(\Delta\Theta)|| > 2^j \mathrm{Cst}_2$ % transgression de la contrainte d'aliasage
    - $\overrightarrow{v}(x, y) = B(x, y).\mathcal{P}^j(\Theta) \quad \forall (x, y) \in \mathcal{R}$
    - arrêt de l'algorithme
  - Sinon
    - $\Theta = \mathcal{P}^1(\Theta) + \Delta\Theta$
  - finSi
- finPour

$\overrightarrow{v}(x, y) = B(x, y).\mathcal{P}^1(\Theta) \quad \forall (x, y) \in \mathcal{R}$

TABLE 2.4 – Algorithme multi-résolution d'estimation robuste du mouvement d'une région.

- Nous ne calculons pas le gradient spatial de l'image, ce qui permet de nous épargner des erreurs d'estimation dues à l'aliasage spatial.
- Le problème de l'aliasage temporel est contrôlé par les fonctions de base,
- Nous pouvons calculer le mouvement d'une petite région même réduite à un pixel puisque les N équations permettent de contourner le "problème de l'ouverture".

Cette technique a aussi l'avantage d'être une méthode rapide et robuste pour estimer le déplacement d'une région, comme le montreront les résultats expérimentaux présentés dans la section suivante.

2.5 Résultats expérimentaux

Plusieurs expériences ont été menées sur des images synthétiques et réelles pour valider la méthode. Dans un premier temps, nous allons montrer l'intérêt de l'utilisation d'un M-estimateur robuste pour l'estimation du mouvement. Pour cela, nous évaluons le comportement de l'algorithme sur des images animées de mouvements synthétiques. Puis nous présenterons les résultats obtenus sur des séquences d'images réelles. Dans l'ensemble des tests, nous avons utilisé des ondelettes analytiques (Cf. Chapitre 1) définies à partir des ondelettes réelles de Deslauriers-Dubuc de filtre m_0 :

$$[-1 \quad 0 \quad 9 \quad 16 \quad 9 \quad 0 \quad -1]/16. \tag{2.60}$$

2.5.1 Comportement de l'algorithme sur des images animées de mouvements synthétiques

Pour évaluer quantitativement l'algorithme d'estimation du mouvement, nous reprenons les expériences menées par Odobez [Odo94]. Ces expériences consistent à appliquer à une image de référence, un mouvement synthétique afin d'obtenir une seconde image. Pour vérifier le comportement de l'algorithme, le mouvement construit est composé de deux mouvements affines différents, l'un sur une fenêtre carrée au centre de l'image (zone 2, Z_2) et l'autre sur le reste de l'image (zone 1, Z_1) (Fig 2.3(a)). Les expériences diffèrent les unes des autres par les mouvements affines appliqués aux deux zones. Chaque coefficient des mouvements affines est tiré suivant une loi uniforme dans l'intervalle $[-3; 3]$ pour les coefficients constants et dans l'intervalle $[-0,05; 0,05]$ pour les coefficients linéaires. Un modèle de champ de déplacement ainsi généré est présenté figure 2.3(b). Étant donné que la qualité de l'estimation du mouvement dépend beaucoup de la variation du gradient dans l'image, nous avons utilisé comme images de référence, quatre images différentes (Fig 2.3) pour effectuer les tests. Ces images sont issues de la base de séquences tests de Barron

et *al.*.
Pour chaque expérience et pour chaque image, l'évaluation de la méthode consiste à étudier la qualité de l'estimation du mouvement de la zone 2 en fonction de la fenêtre d'estimation $\mathcal{R}$ sur laquelle le mouvement est calculé. L'erreur d'adéquation entre le mouvement affine estimé $\overrightarrow{\tilde{v}}$ et le mouvement affine réel de la zone 2, $\overrightarrow{v}_{Z_2}$, choisie est l'erreur angulaire relative moyenne (2.44)

$$Erreur(t) = \frac{\sum_{p \in Z_2} E_{ang}(\overrightarrow{\tilde{v}}, \overrightarrow{v}_{Z_2})}{\sum_{p \in Z_2} E_{ang}(\overrightarrow{v}_{Z_1}, \overrightarrow{v}_{Z_2})}, \tag{2.61}$$

où t est le taux d'occupation de Z_2 dans la fenêtre d'estimation $\mathcal{R}$

$$t = \frac{\mathrm{Card}(\mathcal{R} \cap Z_2)}{\mathrm{Card}(\mathcal{R})}. \tag{2.62}$$

Cette erreur doit être proche de zéro lorsque la vitesse calculée est proche du mouvement réel de la zone 2 et tend vers 1 quand t tend vers 1. En effet, dans ce dernier cas, la zone 1 devient majoritaire dans le support d'estimation $\mathcal{R}$ et le mouvement dominant dans l'image devient $\overrightarrow{v}_{Z_1}$.

Les expériences menées sur les quatre images sont présentées figures 2.4 et 2.5 . Sur celles-ci, nous comparons la qualité de l'estimation de notre méthode utilisant un M-estimateur robuste avec une simple résolution au sens des moindres carrés. Comme prévu, l'estimation du mouvement grâce au M-estimateur robuste permet de mieux calculer le mouvement prépondérant dans la région $\mathcal{R}$. Le saut d'un mouvement à l'autre est en effet beaucoup plus prononcé par l'estimation robuste que par l'estimation au sens des moindres carrés. La transition se produit dès $\sum_{p \in \mathcal{R} \cap Z_1} ||\overrightarrow{\nabla} I(p)|| = \sum_{p \in \mathcal{R} \cap Z_2} ||\overrightarrow{\nabla} I(p)||$. Le gradient de l'image joue effectivement un rôle important dans l'estimation du mouvement comme nous le montrent les résultats sur l'image Nasa. Dans ce cas, il est difficile d'estimer correctement le mouvement de la zone périphérique en particulier sans estimation robuste car celle-ci est peu texturée par rapport à la zone centrale de l'image. C'est ce que nous montre la figure 2.5.1 où nous avons comparé le rapport sur la fenêtre $\mathcal{R}$ entre la norme du gradient de l'image sur la zone 2 et la zone 1,

$$Rap(t) = \frac{\sum_{p \in \mathcal{R} \cap Z_1} ||\overrightarrow{\nabla} I(p)||}{\sum_{p \in \mathcal{R} \cap Z_2} ||\overrightarrow{\nabla} I(p)||}, \tag{2.63}$$

sur les images Nasa et Tree. Pour l'image Tree, dès que $t < 0.3$, nous avons $Rap(t) > 2$ et une erreur d'estimation $Erreur > 0.9$, ce qui se produit seulement si $t < 0.2$ dans le cas de l'image Nasa.

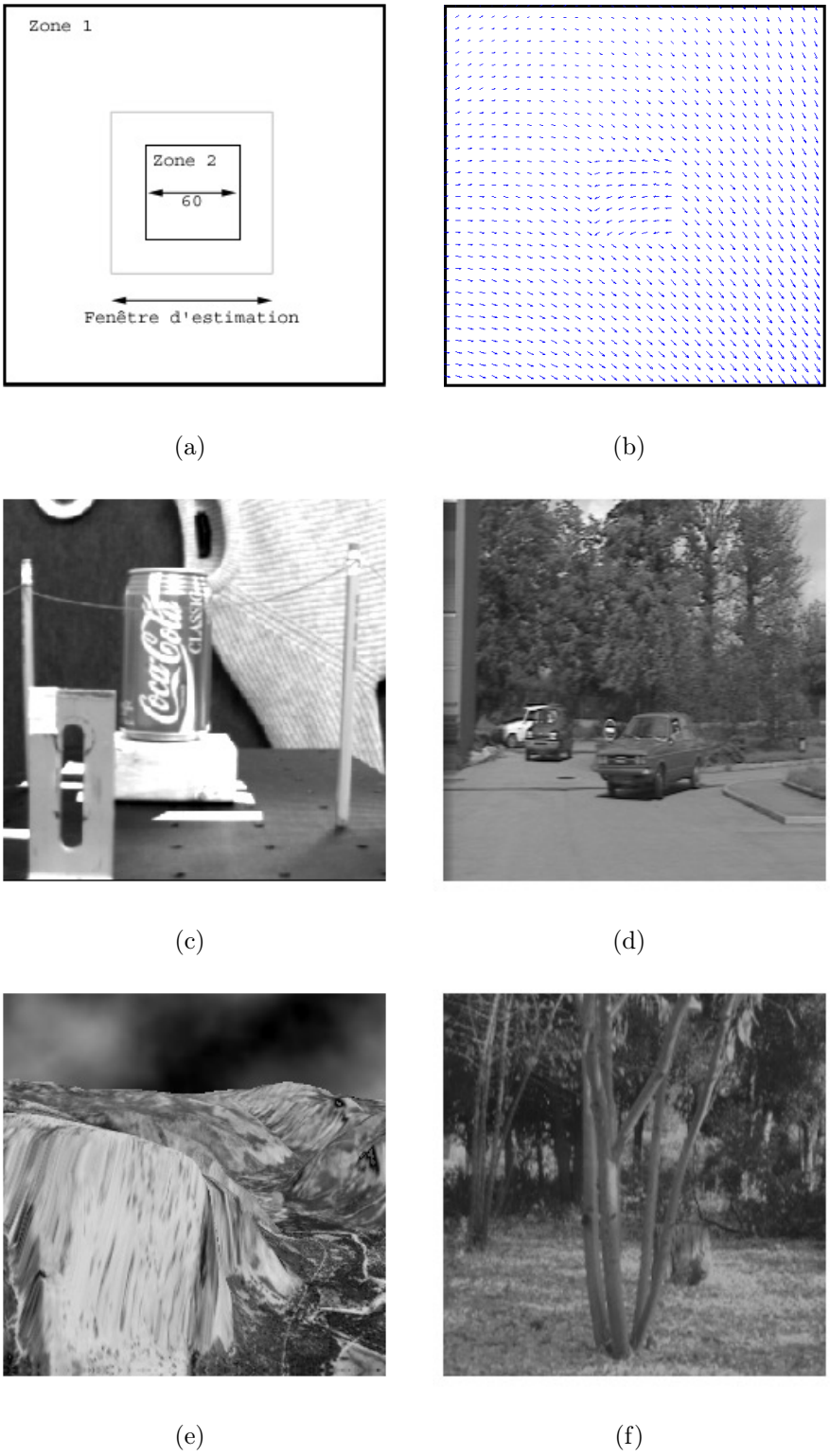
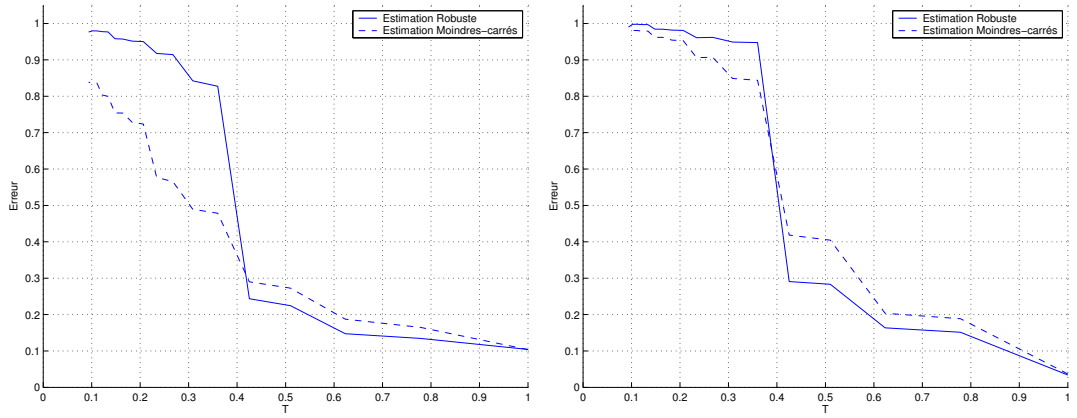
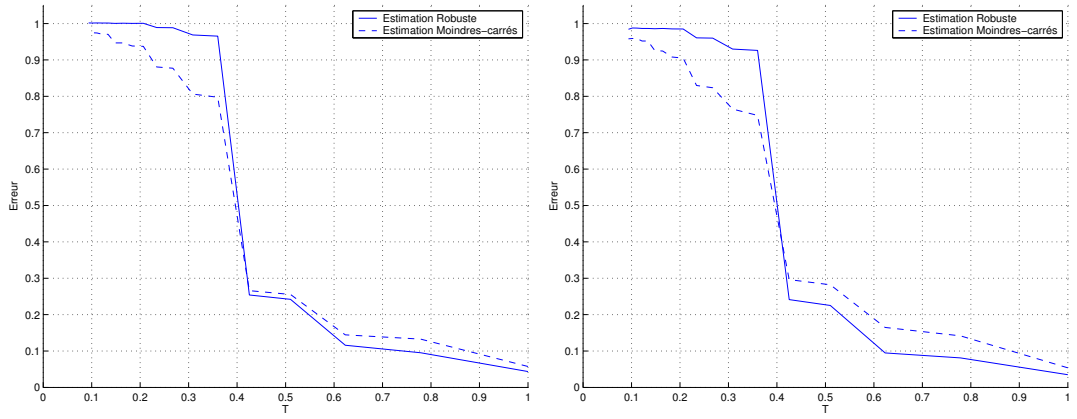


FIGURE 2.3 – Expériences de validation de la méthode. (a) Schéma de construction des expériences, (b) champ synthétique des vitesses, (c) Séquence Nasa, (d) Séquence Parking, (e) Séquence Yosemite (f) Séquence Tree.



(a) (b)

FIGURE 2.4 – Erreur moyenne de l'estimation en fonction de la taille du support. (a) Séquence Nasa, (b) Séquence Parking



(a) (b)

FIGURE 2.5 – Erreur moyenne de l'estimation en fonction de la taille du support. (a) Séquence Yosemite (b) Séquence Tree.

Nous avons aussi testé la robustesse de notre méthode en fonction du bruit dans l'image. Pour cela, nous avons effectué les mêmes expériences que précédemment à la différence que la seconde image, obtenue à partir des deux mouvements affines sur l'image de référence, a été bruitée par un bruit blanc gaussien centré d'écart type σ. Nous montrons dans la figure 2.7 l'erreur moyenne obtenue sur l'ensemble des expériences menées sur les quatre images de référence et $\sigma = 5, 10, 15$. Nous remarquons que même avec un bruit élevé ($\sigma = 15$), notre méthode est peu sensible au bruit.

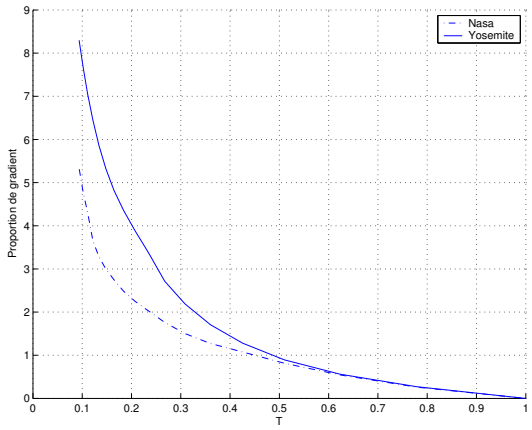
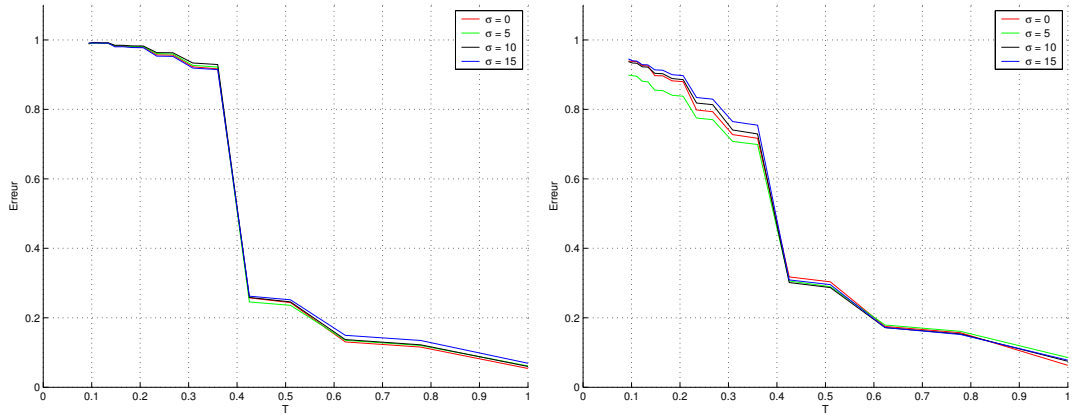


FIGURE 2.6 – Rapport entre le gradient de chaque zone en fonction de la taille de la fenêtre d'estimation sur l'image Nasa et l'image Tree.



(a) (b)

FIGURE 2.7 – Erreur moyenne sur les quatre images de références en fonction du bruit blanc introduit ($\sigma = 0, 5, 10, 15$). (a) Estimation par un M-estimateur de Tukey, (b) Estimation par moindres carrés.

2.5.2 Expérimentations sur des séquences d'images réelles

Nous avons évalué notre méthode sur plusieurs séquences d'images réelles. Nous présentons ici le résultat sur trois séquences très utilisées pour valider les algorithmes d'analyse de mouvement : la séquence "Mobile and Calendar", "Flower Garden" et "Coastguard". Nous nous proposons d'estimer le mouvement dominant dans l'image, dû au mouvement de la caméra. Cela nous permettra dans la suite de cette thèse, de compenser le mouvement de la caméra dans l'image pour ensuite détecter les objets mobiles dans la scène. Pour ce faire, nous prenons comme support d'estimation du mouvement $\mathcal{R}$ l'image entière.

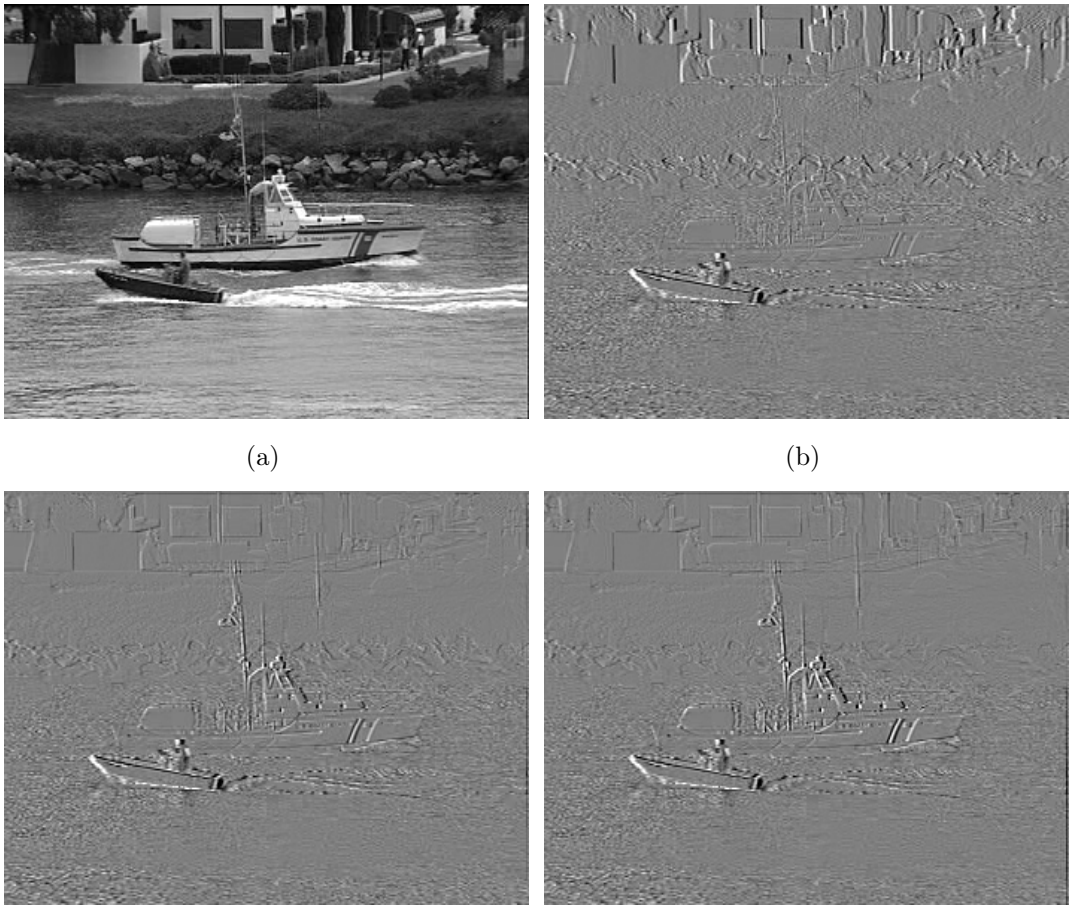
Séquence "Coastguard" : La figure 2.8(a) présente une image de la séquence "Coastguard". Le mouvement dominant dans la séquence est dû au mouvement de la caméra qui suit le bateau des garde-côtes croisant une petite embarcation. La figure 2.8(b) met en évidence la différence temporelle entre les deux images $I((x,y),t+1) - I((x,y),t)$ à laquelle nous avons ajouté un offset de 128 (un point gris correspond à une valeur nulle et plus le point est sombre ou clair, plus la différence est importante). Nous avons une nouvelle fois testé notre méthode avec une estimation au sens des moindres carrés et une estimation robuste. La différence compensée entre les deux images

$$DFD(\overrightarrow{v}) = I((x,y),t+1) - I((x,y)+\overrightarrow{v},t) \tag{2.64}$$

(où une interpolation bilinéaire a été utilisé pour calculer l'image compensée $I((x,y)+\overrightarrow{v},t)$) est présentée en figure 2.9(c) dans le cas de l'estimation au sens des moindres carrés et figure 2.9(d) dans le cas du M-estimateur robuste. Le mouvement de la caméra a été relativement bien estimé dans les deux cas. L'estimation robuste apporte peu d'avantages dans ce cas car la zone ne subissant pas le mouvement dans l'image (les deux bateaux) est peu importante relativement à l'ensemble de l'image.

Séquence "Flower Garden" : Dans cette séquence, le mouvement dans la scène est exclusivement dû au déplacement de la caméra (Fig 2.9(a)). Cependant, ce mouvement se répercute différemment dans l'image en fonction de la profondeur des objets dans la scène. On distingue trois plans différents, l'arbre au premier plan, les fleurs puis les maisons. La différence temporelle entre les deux images est montrée figure 2.9(b). Le résultat de l'estimation du mouvement dominant est présenté figure 2.9(c) au sens des moindres carrés et figure 2.9(c) dans le cas de l'estimation à partir du M-estimateur robuste. L'estimation au sens des moindres carrés estime le mouvement moyen dans l'image. En revanche, l'utilisation du M-estimateur permet de ne prendre en compte que les points ayant le mouvement dominant dans l'image à savoir la zone du deuxième plan.

Séquence "Mobile and Calendar" : Cette séquence plus complexe est composée de plusieurs objets en mouvement, un train électrique qui avance de la droite vers la gauche tout en poussant un ballon, un mobile en rotation et un calendrier assujetti d'un mouvement vertical. De plus, l'ensemble de l'image subit un mouvement de translation apparent de la gauche vers la droite dû au mouvement de la caméra (Fig 2.10(a)). La figure 2.10(b) montre la dynamique temporelle entre les deux images. On remarque comme prévu que l'estimation au sens des moindres carrés tend à déterminer le champ des déplacements qui correspond au mieux à l'ensemble des points de l'image (Fig 2.10(e)). En effet, sur



(a) (b)

(c) (d)

FIGURE 2.8 – Estimation du mouvement dominant dans la séquence "Coastguard", (a) première image, (b) différences temporelles entre les deux images, (c) différences compensées avec une estimation au sens des moindres carrés, (d) différences compensées à partir d'un M-estimateur.

la gauche de l'image, le mouvement du fond de la scène qui correspond au mouvement de la caméra est correctement estimé et sur la droite le champ des vitesses se rapproche du mouvement du calendrier. Au contraire, notre méthode grâce au M-estimateur calcule correctement le mouvement dominant dans l'image (Fig 2.10(f)).

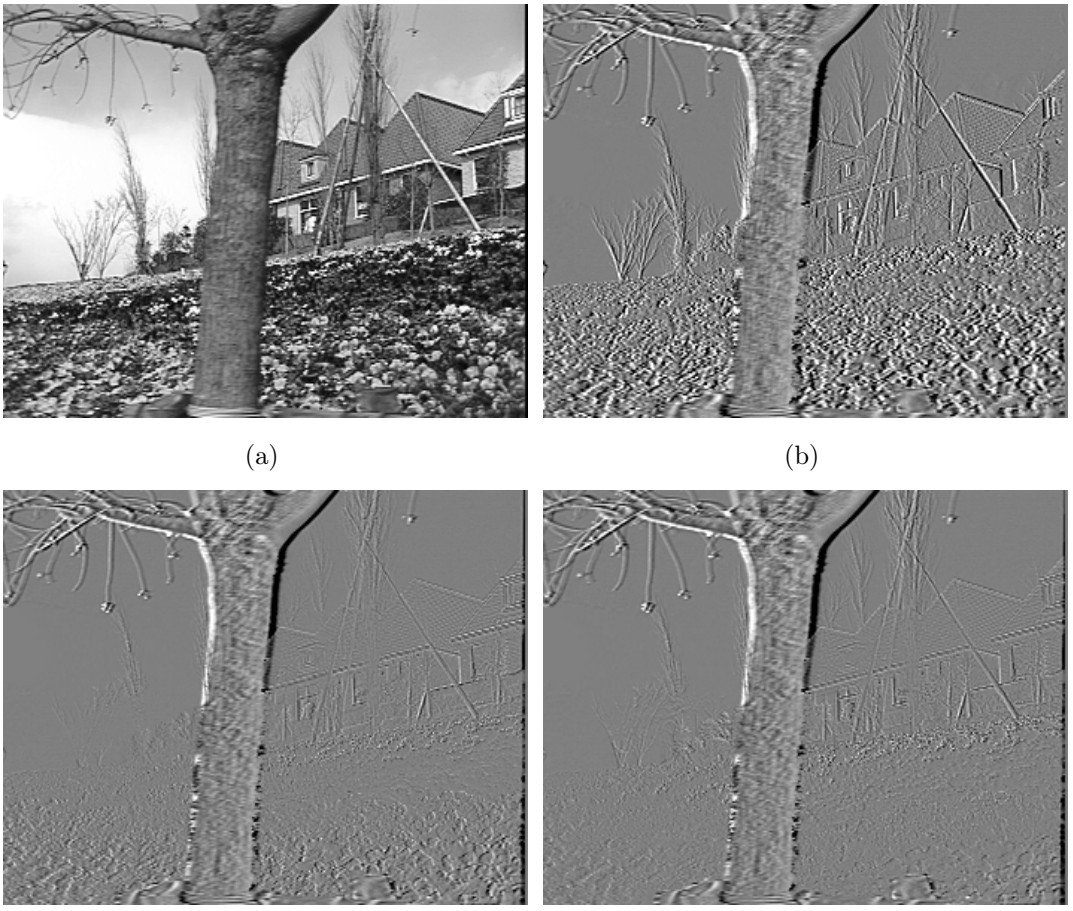


FIGURE 2.9 – Estimation du mouvement dominant dans la séquence "Flower Garden", (a) première image, (b) différences temporelles entre les deux images, (c) différences compensées avec une estimation au sens des moindres carrés, (d) différences compensées à partir d'un M-estimateur.

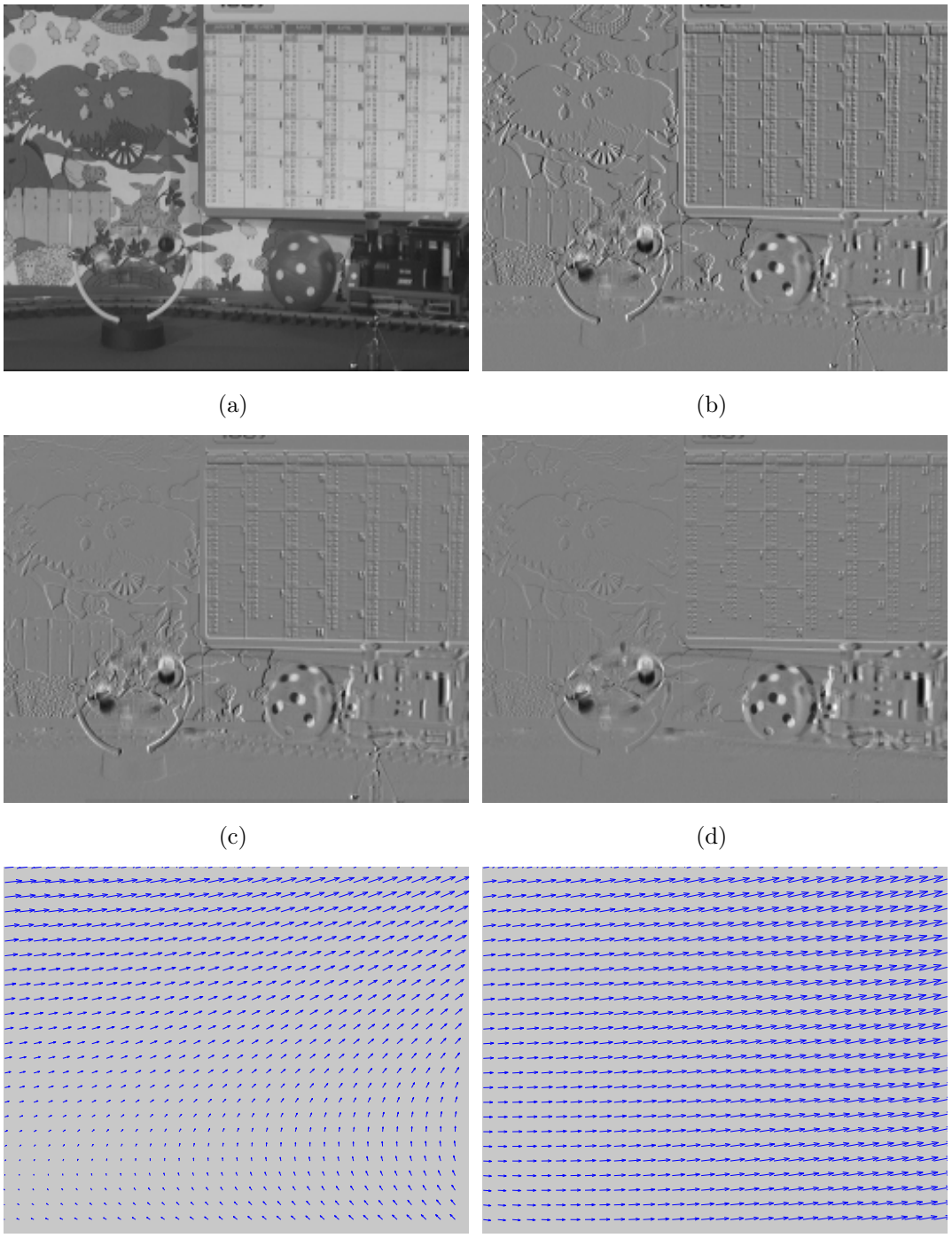


FIGURE 2.10 – Estimation du mouvement dominant dans la séquence "Mobile and Calendar", (a) première image, (b) différences temporelles entre les deux images, (c)-(e) différences compensées et champ de vitesses avec une estimation au sens des moindres carrés, (d)-(f) différences compensées et champ des vitesses à partir d'un M-estimateur.

2.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème de l'estimation du mouvement dans des séquences d'images. Étant donné le champ important de recherche sur ce sujet, nous nous sommes contentés de discuter les méthodes les plus couramment utilisées pour extraire une information essentielle sur les séquences d'images : le mouvement apparent. Après un bref rappel des problèmes rencontrés pour estimer le flot optique, nous avons discuté brièvement des trois grands groupes de méthodes d'estimation : les méthodes de mise en correspondance, les méthodes fréquentielles et les méthodes différentielles. Nous avons ensuite exposé les différentes méthodes d'estimation du mouvement par modélisation paramétrique. Ces méthodes se sont avérées très intéressantes dans des domaines d'application tels que la segmentation au sens du mouvement, la détection de mouvement, la reconnaissance de mouvement ou encore l'indexation de vidéos.

Nous avons ensuite développé une méthode d'estimation du mouvement dans les séquences d'images inspirée des travaux de Bernard. Cette méthode estime directement le mouvement d'une région $\mathcal{R}$ à partir de l'équation de contrainte du flot optique et grâce à la projection de celle-ci sur une base d'ondelettes. L'utilisation de telles fonctions a deux avantages : elles permettent dans le cas où la région est réduite à un seul pixel de contourner le problème de l'ouverture et elles permettent de calculer des grands déplacements par multirésolution. Notre méthode présente plusieurs intérêts par rapport aux méthodes cherchant à minimiser la différence d'image déplacée (équation (2.20)) :

- La détermination du mouvement se fait par la résolution d'un simple système linéaire et non par une descente de gradient.
- Le calcul du gradient spatial de l'image est évité grâce à l'intégration par parties.
- Notre méthode est capable d'estimer le mouvement d'une région même réduite à un pixel.

De surcroît, la méthode définie dans ce chapitre est une méthode relativement rapide pour estimer le mouvement dans l'image. En effet, l'algorithme programmé en Matlab calcule le mouvement affine global d'une image de taille 256×256 en moins de 30 secondes sur un PC Pentium III 933 mhz, ce qui laisse espérer une implantation temps réel à l'aide de langage de programmation adéquat. Les résultats à la fois sur des séquences synthétiques et réelles ont montré que notre méthode était capable d'estimer le mouvement dominant dans l'image ou d'une région particulière sans être perturbée par des mouvements secondaires. Cette méthode sera par la suite la base des méthodes de détection de mouvement et de segmentation au sens du mouvement détaillées dans la suite de ce mémoire.

Chapitre 3

Détection de mouvement par une approche hiérarchique

Nous traitons ici le problème de détection du mouvement dans des séquences d'images. Ce problème suscite beaucoup d'intérêt car il offre de nombreuses applications : la surveillance vidéo, la compression d'images, le suivi de cibles, la détection d'obstacles en robotique mobile... C'est pourquoi ce sujet a fait l'objet de nombreuses recherches. Dans notre étude, nous nous plaçons dans le cas tout à fait général où la caméra est mobile. Nous cherchons à détecter tout objet non statique dans la scène ou toute région qui ne peut être modélisée par le mouvement apparent dû au mouvement de la caméra qui sera représenté par un modèle paramétrique.

Dans ce chapitre, nous allons faire un bref état de l'art des méthodes de détection de mouvement dans des séquences d'images. Dans une deuxième partie, nous introduirons une nouvelle méthode pour résoudre ce problème entre deux images à partir d'une modélisation hiérarchique. Puis, nous généraliserons cette méthode sur une séquence d'images en utilisant un processus de Markov temporel-hiérarchique.

3.1 Éléments d'état de l'art des méthodes de détection de mouvement

Les méthodes de détection de mouvement cherchent à obtenir des masques binaires sur la séquence d'images. Ces masques caractérisent les zones mobiles dans le cas d'une caméra fixe ou les zones non conformes au mouvement dominant dans le cas d'une caméra mobile.

Dans le cas d'une caméra fixe, une simple détection des changements temporels de l'intensité lumineuse dans l'image peut suffire à obtenir une bi-partition de l'image en zone

fixe et zone en mouvement. En effet, les éléments mobiles dans la scène génèrent une modification de l'intensité entre deux images consécutives de la séquence. Cependant, ce type de méthode est très sensible aux changements des conditions d'illumination. De nombreux auteurs ont donc proposé une modification de cette méthode pour diminuer cette sensibilité.
Une méthode consiste à utiliser des tests statistiques fondés sur une fonction de vraisemblance. Pour cela, Hsu et *al* [HNR84] utilise une modélisation locale polynomiale de la fonction d'intensité. Rosin [Ros02] modélise le bruit dans la séquence d'images par une distribution normale. Aach et *al* [AKM93], [ADMT01], Sifakis et Tziritas [ST01], Paragios et Deriche [PD97] modélisent la différence inter-image par des distributions de probabilité. Une approche différente qui consiste à régulariser les masques de détection de mouvement par un modèle markovien, a par ailleurs été proposée [BL90], [LRB93], [LCL99]. Cette modélisation permet de réduire les fausses alarmes de détection dans les régions uniformes grâce à une dépendance à la fois spatiale et temporelle du champ d'étiquettes.
Ces deux types de méthodes permettent en général d'obtenir de bon résultats. Notons cependant que la plupart ne permettent pas de localiser les objets en mouvement dans la scène. En effet, prenons l'exemple simple d'un objet en translation (Fig 3.1). La carte de détection de mouvements entre deux images ne correspond pas à la détection de l'objet mais à la zone comprenant à la fois l'objet et les zones couvertes et découvertes entre les images. Pour résoudre ce problème, Karmann et von Brandt [KvB90], Donohoe et al. [DHA88] utilisent une image de référence qui correspond au fond de la scène en l'absence d'objets mobiles. Ces méthodes sont néanmoins sensibles aux conditions d'illumination et ne permettent pas de prendre en compte d'éventuels changements dans la composition du fond sans mise à jour régulière de l'image de référence.

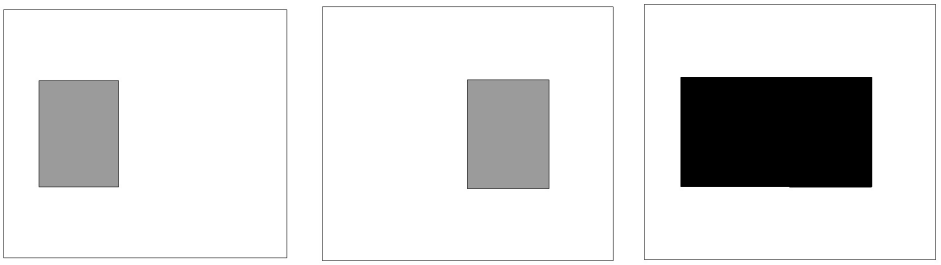


FIGURE 3.1 – Problème de la localisation d'objets.(a) Image 1, (b) Image 2, (c) Masque de détection sur l'image 1. Les regions découverte et recouverte par l'objet correspondent à des zones de changements temporels.

Dans le cas d'une caméra mobile, le problème est beaucoup plus complexe. Il existe essentiellement deux approches pour résoudre le problème de détection des objets en mou-

vement dans la scène.
Les premières méthodes consistent à utiliser des contraintes sur le champ des déplacements dues au mouvement de la caméra. Par exemple, si nous connaissons la composante rotationnelle du déplacement de la caméra, tout point de l'image ne vérifiant pas cette contrainte sera supposé mobile [TP90]. Les deuxièmes méthodes consistent, dans un premier temps, à estimer le mouvement dominant dans l'image puis à compenser les images par ce mouvement [IRP92],[OB94]. Ceci permet de se ramener au cas plus simple d'une caméra fixe. Des techniques classiques de seuillage [IRP92], ou des techniques bayésiennes [OB94], [TM93], [PPT+96] sont alors utilisées. On peut trouver aussi des méthodes qui utilisent une segmentation spatiale de l'image et qui regroupent ensuite les régions en zones statiques et mobiles. Ces méthodes permettent d'obtenir une meilleure localisation des objets dans la scène à partir des informations d'intensité, de textures ou de couleurs [FBG99], [NWD00]. Cependant, l'utilisation d'une segmentation spatiale préalable de l'image augmente le coût de calcul de l'algorithme.

Dans chaque technique, l'utilisation de champs de Markov pour modéliser le problème inverse de détection de mouvement s'est révélé très intéressante [OB94], [LRB93]. Ces champs permettent de transformer le problème de détection en un problème de minimisation d'énergie. Cependant, cette fonction d'énergie globale étant le plus souvent non convexe, déterminer le minimum global de cette énergie nécessite dans l'absolu l'utilisation d'algorithmes stochastiques beaucoup trop gourmands en temps de calcul et inutilisables pour les applications citées plus haut. C'est pourquoi on préfère souvent minimiser cette énergie à l'aide d'algorithmes déterministes. Néanmoins, comme nous l'avons vu, ces algorithmes sont très sensibles aux conditions initiales et convergent vers un minimum local de l'énergie. Pour surmonter ce problème, Odobez et Bouthemy [OB94] utilisent un algorithme multi-échelle beaucoup moins sensible aux conditions initiales afin de minimiser la fonction d'énergie considérée.
Dans ce chapitre, nous allons résoudre le problème de la détection du mouvement dans une séquence d'images à partir d'une modélisation hiérarchique du problème. Pour ce faire, nous compenserons la séquence d'images par le mouvement dominant estimé à chaque résolution.

3.2 Détection de mouvement entre deux images

Dans cette partie, nous nous restreignons au cas de la détection du mouvement entre deux images I_1 et I_2. Afin de contourner l'aliasage temporel et d'obtenir un algorithme peu coûteux en temps de calcul, nous utilisons une modélisation hiérarchique du problème de détection.

3.2.1 Modélisation hiérarchique

Supposons que la méthode décrite au chapitre 3 nous ait permis d'obtenir à partir d'une analyse multirésolution le mouvement dominant entre les deux images. Rappelons que ce déplacement $\overrightarrow{v}$ est définie grâce à une estimation du mouvement sur chaque échelle $j = 0 \cdots J$ de la façon suivante :

$$\begin{aligned}
&\overrightarrow{\widetilde{v}}^{\,j}(x,y) = B(x,y).\mathcal{P}^j(\Theta^j),\\
&\overrightarrow{v}^{\,j}(x,y) = B(x,y).\textstyle\sum_{n=j}^{J}\mathcal{P}^j(\Theta^j),\\
&\text{avec } \mathcal{P}^j(X) = \mathrm{diag}(1 \quad 1 \quad 2^j \quad 1 \quad 1 \quad 2^j).X,
\end{aligned}$$

où $\overrightarrow{\widetilde{v}}^{\,j}$ est le mouvement dominant calculé à l'échelle j entre l'image I_1 et l'image compensée $\widetilde{I}((x,y),t+1) = I((x+v_1^{j+1}(x,y), y+v_2^{j+1}(x,y)), t+1)$ et $\overrightarrow{v}^{\,j} = (v_1^j, v_2^j)$ est l'estimation du mouvement dominant entre $I(t)$ et $I(t+1)$.

Cette estimation du mouvement nous permet de construire de façon naturelle un graphe hiérarchique. Pour cela, reprenons les notations du chapitre précédent, $S = \cup_{j=0}^{J} S^j$ où S^j désigne le niveau de résolution j de l'image sur lequel nous avons estimé le mouvement, posons $\bar{i}$ le parent du site i, $\underline{i}$ l'ensemble des enfants de i et $\underline{\underline{i}}$ l'ensemble des sites formant l'arbre de racine i (Fig 3.2). Le problème de détection de mouvement consiste à déterminer le champ aléatoire inconnu des étiquettes $E = \{E_s, s \in S\}$ connaissant le champ aléatoire des observations $O = \{O_s, s \in S\}$. E_s pouvant prendre deux valeurs 0 ou 1, 1 correspondant à un site non conforme au mouvement dominant et 0 à un site conforme. Notons E^j l'ensemble des étiquettes des sites de niveau j, *i.e* $E^j = \{E_s, s \in S^j\}$ et de la même manière $O^j = \{O_s, s \in S^j\}$.

En reprenant les hypothèses de Chardin [Cha00b], sur les champs aléatoires E et O, e est solution au sens du MAP de :

$$\widehat{e} = \min_e [\sum_{<i,j>\in S^J} v_{i,j}(e_i, e_j) + \sum_{i \notin S^J} w_i(e_i, e_{\bar{i}}) + \sum_{i \in S} q_i(e_i, o_i)] \tag{3.1}$$

où les potentiels rentrant en jeu : v_{ij} le terme de régularisation spatiale des étiquettes (intervenant seulement au niveau le plus grossier S^J), w_i le terme d'attache en échelle et l_i le terme d'attache aux données, sont tous connus *a priori*.
Les potentiels spatiaux v_{ij} et hiérarchiques w_i étant définis à partir d'un modèle de Potts classique

$$v_{i,j}(e_i, e_j) = \alpha[1 - \delta(e_i, e_j)], \tag{3.2}$$

$$w_i(e_i, e_{\bar{i}}) = \beta[1 - \delta(e_i, e_{\bar{j}})]. \tag{3.3}$$

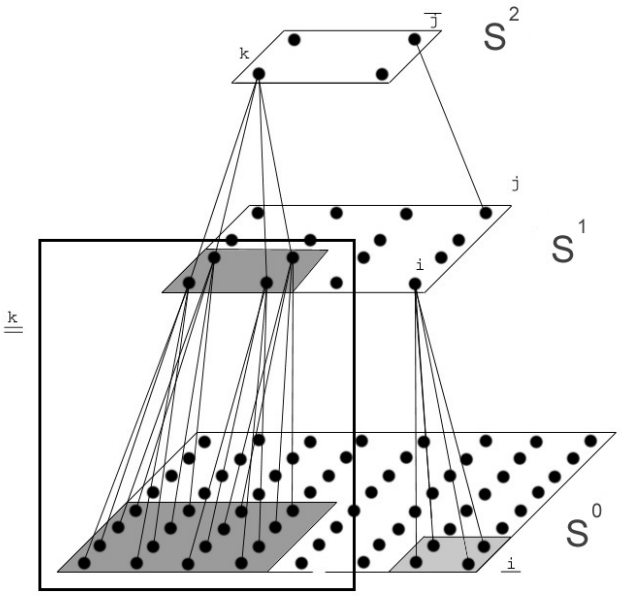


FIGURE 3.2 – Modélisation hiérarchique sur 3 niveaux.

Ces potentiels favorisent le champ d'étiquettes de détection à être homogènes à la fois en espace et en échelle.
Intéressons-nous désormais au terme d'attache aux données qui relie les observations à savoir le mouvement dominant au champ d'étiquettes e.

3.2.2 Mesure d'erreur d'estimation du mouvement

Le problème posé dans cette section est de trouver une mesure qui permette de quantifier l'erreur commise en chaque point p entre le mouvement dominant $\overrightarrow{v}$ et le mouvement apparent réel. De nombreuses mesures ont été définies pour caractériser cette erreur. La mesure la plus fréquemment employée est sans conteste la différence inter-images définie par :

$$DFD(p) = I(p + \overrightarrow{v}, t+1) - I(p,t) = I_{\overrightarrow{v}}(t+1) - I(t). \tag{3.4}$$

Par exemple, Haynes et Jain [HJ83] utilisent la quantité $|DFD| \times ||\overrightarrow{\nabla} I||$ pour détecter les contours en mouvement. Cependant, cette mesure se révèle mal adaptée dans le cas d'une caméra mobile. En effet, elle présente deux inconvénients majeurs :

- Le premier vient du fait que le calcul de (3.4) nécessite d'interpoler l'image I_2. Cette interpolation génère une erreur qui se répercute sur le calcul de $|DFD| \times ||\overrightarrow{\nabla} I||$.
- Le second est intrinsèque à la définition de la DFD. En effet, même si l'interpolation est parfaite, une faible erreur d'estimation sur la vitesse $\overrightarrow{v}$ en un point de fort

gradient d'intensité peut engendrer une DFD importante.

C'est pourquoi Irani et al. [IRP92], Odobez et Bouthemy [OB94] utilisent une mesure d'erreur définie à partir de la vitesse normale entre l'image $I(t)$ et l'image compensée $I_{\overrightarrow{v}}(t+1)$:

$$v_n = \left|\frac{\partial I}{\partial t}\right| . \frac{1}{||\overrightarrow{\nabla} I(q)||}. \tag{3.5}$$

Cette vitesse est moyennée sur un voisinage $\mathcal{V}(p)$ pour réduire sa grande sensibilité au bruit et est pondérée par le carré du module du gradient spatial d'intensité $||\overrightarrow{\nabla} I||$. Cette pondération est employée car elle constitue une bonne mesure de fiabilité de la précision du calcul de v_n [BFB94].

Irani et al. utilisent donc pour estimer l'erreur entre la vitesse estimée et la vitesse réelle la formule suivante

$$Er(p) = \frac{\sum\limits_{q \in \mathcal{V}(p)} (||\overrightarrow{\nabla} I(q)||^2 \times v_n(q))}{\zeta + \sum\limits_{q \in \mathcal{V}(p)} ||\overrightarrow{\nabla} I(q)||^2}, \quad \zeta > 0. \tag{3.6}$$

Alors que Odobez préfère modifier le dénominateur en posant

$$Er(p) = \frac{\sum\limits_{q \in \mathcal{V}(p)} (||\overrightarrow{\nabla} I(q)||^2 \times v_n(q))}{\max(G_{moy}, \sum\limits_{q \in \mathcal{V}(p)} ||\overrightarrow{\nabla} I(q)||^2)}, \quad G_{moy} > 0. \tag{3.7}$$

Cette dernière impose un gradient minimum moyen G_{moy}, ce qui a pour effet d'améliorer la robustesse de l'observation en chaque pixel [Odo94]. Cette mesure a depuis été utilisée par Fablet et al. pour détecter les mouvements dans des séquences d'images couleurs [FBG99], mais aussi pour caractériser les mouvements dans le domaine de l'indexation de séquences d'images [BF98], [FBP02].

Elle a cependant un inconvénient. Comme la séquence est échantillonnée en temps, elle nécessite de faire une approximation de la dérivée temporelle de l'image par une différence finie

$$\frac{\partial I}{\partial t}(p) \approx I_{\overrightarrow{v}}(p, t+1) - I(p, t). \tag{3.8}$$

Or cette approximation engendre une erreur non maîtrisable dépendant fortement des variations d'intensité dans l'image. En effet, supposons que $I(p,t) = I(p - t\overrightarrow{v})$ et qu'après calcul, le mouvement dominant $\overrightarrow{v}$ réel entre les deux images $I(t)$ et $I(t+1)$ ait été estimé par $\overrightarrow{v}_e \neq \overrightarrow{v}$. Le mouvement entre les images $I(t)$ et $I_{\overrightarrow{v}_e}(t+1)$ n'est donc pas nul mais vaut exactement $\overrightarrow{\tilde{v}} = \overrightarrow{v} - \overrightarrow{v}_e$. Ainsi, $\overrightarrow{\tilde{v}}$ vérifie l'E.C.M.A (2.5) entre les images $I(t)$ et $I_{\overrightarrow{v}_e}(t+1)$

$$\overrightarrow{\nabla} I . \overrightarrow{\tilde{v}} + \frac{\partial I}{\partial t} = 0; \tag{3.9}$$

donc faire l'approximation (3.8) revient à dire que :

$$\begin{aligned}\overrightarrow{\nabla} I(p).\overrightarrow{\widetilde{v}} &\approx -(I_{\overrightarrow{v}_e}(p,t+1) - I(p,t)) \\ &\approx -(I(p+\overrightarrow{v}_e, t+1) - I(p,t)) \\ &\approx -(I(p-\overrightarrow{\widetilde{v}}, t) - I(p,t)),\end{aligned} \tag{3.10}$$

et d'après la formule de Taylor-Lagrange, l'erreur commise en faisant l'approximation (3.10) est

$$\begin{gathered}\frac{||\overrightarrow{\widetilde{v}}||^2}{2}\Delta I(c), \quad c \in [x-\widetilde{v}_1, x] \times [y-\widetilde{v}_2, y], \\ \text{où} \quad p=(x,y), \quad \overrightarrow{\widetilde{v}} = (\widetilde{v}_1, \widetilde{v}_2).\end{gathered} \tag{3.11}$$

Ainsi, nous sommes une nouvelle fois en présence d'aliasage temporel. En effet, dans le cas où le Laplacien est élevé ou que la vitesse du point p n'a pas été correctement estimé, l'approximation (3.8) n'est pas valable ce qui entraîne une mauvaise estimation de l'erreur d'observation $Er(p)$. Remarquons que le deuxième cas se produit en pratique si la vitesse du point n'est pas celle du mouvement dominant même si celui-ci a été correctement estimé.

Nous avons opté ici pour une estimation de l'erreur différente. A une échelle j fixée, nous posons

$$\forall k=(k_1,k_2) \quad Er(j,k) = \frac{\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n\rangle||.|\langle \overrightarrow{\nabla} I.\overrightarrow{\widetilde{v}^j} + \frac{\partial I}{\partial t}, \psi_{jk}^n\rangle|}{\max(G, \sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n\rangle||^2)}. \tag{3.12}$$

Cette mesure a la propriété suivante

Propriété 3.1 Soit $\overrightarrow{v}_{jk}$ la vitesse réelle à l'échelle j au point $(2^j k_1, 2^j k_2)$ entre les images $I(t)$ et $I_{v^{j+1}}(t+1)$.
Si la vitesse $\overrightarrow{\widetilde{v}}^j(x,y)$ est constante sur le support des ondelettes ψ_{jk}^n, alors $\exists(\lambda_{jk}, \Lambda_{jk})$ tels que

$$\frac{\lambda_{jk}}{\max(G, \lambda_{jk}+\Lambda_{jk})}||\overrightarrow{\widetilde{v}}^j - \overrightarrow{v}_{jk}|| \leq Er(j,k) \leq \frac{\sqrt{\Lambda_{jk}(\lambda_{jk}+\Lambda_{jk})}}{\max(G, \lambda_{jk}+\Lambda_{jk})}||\overrightarrow{\widetilde{v}}^j - \overrightarrow{v}_{jk}||. \tag{3.13}$$

avec λ_{jk} et Λ_{jk} respectivement la plus petite et la plus grande valeur propre de

$$A_{jk} = \begin{pmatrix} \sum_{n=1}^{N} |\langle I, \frac{\partial \psi_{jk}^n}{\partial x}\rangle|^2 & \sum_{n=1}^{N} \overline{\langle I, \frac{\partial \psi_{jk}^n}{\partial x}\rangle}\langle I, \frac{\partial \psi_{jk}^n}{\partial y}\rangle \\ \sum_{n=1}^{N} \langle I, \frac{\partial \psi_{jk}^n}{\partial x}\rangle\overline{\langle I, \frac{\partial \psi_{jk}^n}{\partial y}\rangle} & \sum_{n=1}^{N} |\langle I, \frac{\partial \psi_{jk}^n}{\partial y}\rangle|^2 \end{pmatrix}. \tag{3.14}$$

<u>Démonstration :</u>
Remarquons tout d'abord que :

$$\begin{aligned}\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||^2 &= \sum_{n=1}^{N} [|\langle I, \tfrac{\partial \psi_{jk}^n}{\partial x}\rangle|^2 + |\langle I, \tfrac{\partial \psi_{jk}^n}{\partial y}\rangle|^2] \\ &= \sum_{n=1}^{N} |\langle I, \tfrac{\partial \psi_{jk}^n}{\partial x}\rangle|^2 + \sum_{n=1}^{N} |\langle I, \tfrac{\partial \psi_{jk}^n}{\partial y}\rangle|^2 \\ &= \mathrm{trace}(A_{jk}) \\ &= \lambda_{jk} + \Lambda_{jk}\end{aligned} \tag{3.15}$$

et que comme A_{jk} est symétrique positive,

$$\lambda_{jk}||X||^2 \leq X^T A_{jk} X \leq \Lambda_{jk}||X||^2, \quad \forall X. \tag{3.16}$$

Or par définition, $\overrightarrow{v}_{jk}$ vérifie

$$\overrightarrow{\nabla} I . \overrightarrow{v}_{jk} = -\frac{\partial I}{\partial t}. \tag{3.17}$$

Par conséquent,

$$Er(j,k) = \frac{\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||.|\langle \overrightarrow{\nabla} I.(\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk}), \psi_{jk}^n \rangle|}{\max(G, \sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||^2)}. \tag{3.18}$$

Par Cauchy-Schwarz,

$$Er(j,k) \leq \frac{\sqrt{\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||^2} . \sqrt{\sum_{n=1}^{N} |\langle \overrightarrow{\nabla} I.(\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk}), \psi_{jk}^n \rangle|^2}}{\max(G, \sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||^2)}. \tag{3.19}$$

De plus, comme par hypothèse $(\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk})$ est constant sur les supports des ondelettes, nous obtenons par intégration par parties :

$$\begin{aligned}\sum_{n=1}^{N} |\langle \overrightarrow{\nabla} I.(\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk}), \psi_{jk}^n \rangle|^2 &= \sum_{n=1}^{N} |\langle (\widetilde{v}_1^j - v_{jk_1}) \tfrac{\partial I}{\partial x}, \psi_{jk}^n \rangle + \langle (\widetilde{v}_2^j - v_{jk_2}) \tfrac{\partial I}{\partial y}, \psi_{jk}^n \rangle|^2 \\ &= \sum_{n=1}^{N} |(\widetilde{v}_1^j - v_{jk_1}) \langle I, \tfrac{\partial \psi_{jk}^n}{\partial x} \rangle + (\widetilde{v}_2^j - v_{jk_2}) \langle I, \tfrac{\partial \psi_{jk}^n}{\partial y} \rangle|^2 \\ &= (\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk})^T A_{jk} (\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk}) \\ &\leq \Lambda_{jk} ||\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk}||^2.\end{aligned} \tag{3.20}$$

Ainsi d'après (3.15), (3.19), (3.20) :

$$Er(j,k) \leq \frac{\sqrt{\Lambda_{jk}(\lambda_{jk} + \Lambda_{jk})}}{\max(G, \lambda_{jk} + \Lambda_{jk})} ||\overrightarrow{\widetilde{v}}^{\,j} - \overrightarrow{v}_{jk}||. \tag{3.21}$$

De plus, d'après l'inégalité de Minkowski,

$$\begin{aligned}
&\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n\rangle||.|\langle \overrightarrow{\nabla} I.(\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}), \psi_{jk}^n\rangle| = \sum_{n=1}^{N} |||\langle \overrightarrow{\nabla} I.(\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}), \psi_{jk}^n\rangle|.\langle \overrightarrow{\nabla} I, \psi_{jk}^n\rangle|| \\
&\geq \sqrt{(\sum_{n=1}^{N} \langle I, \tfrac{\partial \psi_{jk}^n}{\partial x}\rangle|\langle \overrightarrow{\nabla} I.(\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}), \psi_{jk}^n\rangle|)^2 + (\sum_{n=1}^{N} \langle I, \tfrac{\partial \psi_{jk}^n}{\partial y}\rangle|\langle \overrightarrow{\nabla} I.(\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}), \psi_{jk}^n\rangle|)^2} \\
&= ||A_{jk}.(\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk})|| \geq \lambda_{jk}||\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}||.
\end{aligned}$$

Par conséquent,

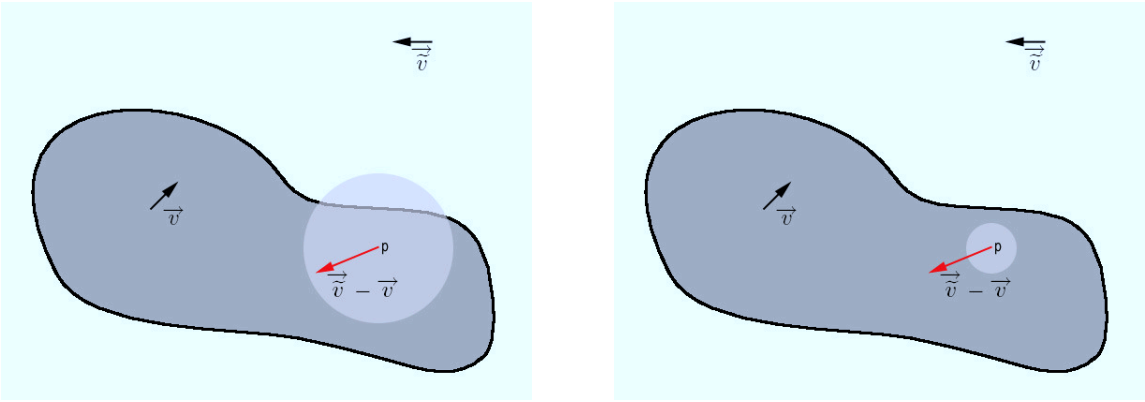
$$Er(j,k) = \frac{\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n\rangle||.|\langle \overrightarrow{\nabla} I.(\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}), \psi_{jk}^n\rangle|}{\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n\rangle||^2} \geq \frac{\lambda_{jk}}{\max(G, \lambda_{jk} + \Lambda_{jk})}||\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}||. \tag{3.22}$$

Cette propriété traduit donc l'équivalence entre la mesure Er et l'erreur $||\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}||$ commise en approchant la vitesse réelle $\overrightarrow{v}_{jk}$ au point $(x,y) = (2^j k_1, 2^j k_2)$ par $\overrightarrow{\tilde{v}}^j$ entre l'image $I(t)$ et l'image compensée $I_{\overrightarrow{v}^{j+1}}(t+1)$. Néanmoins, pour utiliser cette propriété sur la mesure Er, nous devons faire l'hypothèse que le vecteur affine $\overrightarrow{\tilde{v}}^j(x,y)$ peut être assimilé à un vecteur constant sur le support des ondelettes ψ_{jk}^n. Cette affirmation risque d'être mise en défaut aux échelles grossières (FIGFig 3.3(a)) c'est-à-dire là où les supports d'ondelettes sont grands. Nous devrions alors calculer cette erreur sur les résolutions fines de l'image. Or, une fois encore le problème de l'aliasage temporel nous oblige à faire le contraire. En effet, à cause de l'échantillonnage temporel de la séquence d'images, l'erreur Er est approchée par

$$Er(j,k) = \frac{\sum_{n=1}^{N} ||\langle I, \overrightarrow{\nabla}\psi_{jk}^n\rangle||.|\langle \overrightarrow{\nabla} I.\overrightarrow{\tilde{v}^j} + I_{\overrightarrow{v}^{j+1}}(t+1) - I(t), \psi_{jk}^n\rangle|}{\max(G, \sum_{n=1}^{N} ||\langle I, \overrightarrow{\nabla}\psi_{jk}^n\rangle||^2)}, \tag{3.23}$$

et cette approximation n'est fiable (comme nous l'avons vu au chapitre précédent, voir relation (2.40)) que si $||\overrightarrow{\tilde{v}}^j - \overrightarrow{v}_{jk}||$ est proportionnellement plus petit que les supports des ondelettes. C'est-à-dire qu'à une échelle fine, la mesure Er n'est pas valable si l'erreur entre l'estimation du mouvement et l'estimation réelle du point est importante. Ce dernier cas peut se produire notamment sur un point correspondant à un objet en déplacement dans la scène avec une vitesse importante relativement à la caméra (Fig 3.3(b)). En résumé, à une échelle grossière, l'approximation de la dérivée temporelle par une différence finie est plausible mais l'hypothèse que le vecteur $\overrightarrow{\tilde{v}}^j$ soit constant risque d'être mise en défaut. A l'inverse, à une échelle fine, $\overrightarrow{\tilde{v}}^j$ peut localement être approchée par un vecteur constant alors que calculer la dérivée temporelle par une différence finie n'est pas correct si le

point ne suit pas le mouvement dominant ou que celui-ci est mal estimé. Ce problème est en partie résolu grâce à la dépendance en échelle de la modélisation hiérarchique. A une échelle fine, l'aliasage temporel en un site sera atténué grâce à la dépendance de son étiquette e_s avec l'étiquette de son parent $e_{\overline{s}}$.



(a) (b)

FIGURE 3.3 – Deux cas où la mesure d'erreur Er n'est pas valable. La caméra se déplace en translation uniforme vers la gauche à une vitesse $\widetilde{v}$ et l'objet a un mouvement de translation uniforme v. Le cercle gris représente la taille des supports des ondelettes.(a) La vitesse $\widetilde{v} - v$ peut être estimée mais l'hypothèse de mouvement constant sur le support des ondelettes n'est pas vérifiée. (b) Le mouvement est constant sur le support mais l'erreur est trop grande pour être estimée par ces ondelettes.

3.2.3 Terme d'attache aux données

La propriété précédente permet de déduire le corollaire important suivant

Corollaire 3.1 $\forall (x, y) = (2^j k_1, 2^j k_2)$, si la vitesse $\overrightarrow{\widetilde{v}}^j(x, y)$ est constante sur le support des ondelettes ψ^n_{jk}, alors $\forall \epsilon > 0$,

$$Er(j,k) \leq \frac{\lambda_{jk}}{\max(G, \lambda_{jk} + \Lambda_{jk})}.\epsilon \Rightarrow ||\overrightarrow{\widetilde{v}}^j - \overrightarrow{v}_{jk}|| \leq \epsilon$$

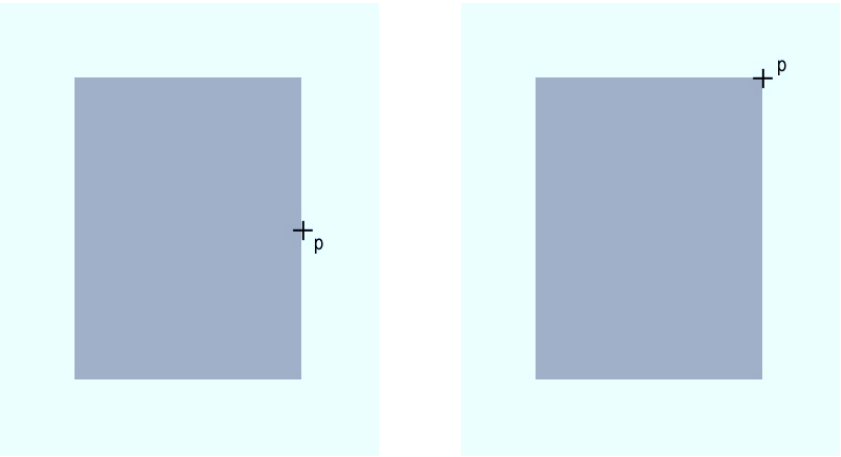
$$Er(j,k) \geq \frac{\sqrt{\Lambda_{jk}(\lambda_{jk} + \Lambda_{jk})}}{\max(G, \lambda_{jk} + \Lambda_{jk}}).\epsilon \Rightarrow ||\overrightarrow{\widetilde{v}}^j - \overrightarrow{v}_{jk}|| \geq \epsilon.$$

C'est à dire que $Er(j,k)$ permet de mesurer l'erreur commise à l'échelle j en substituant la vitesse réelle au point $(2^j k_1, 2^j k_2)$ entre les images $I(t)$ et $I_{\overrightarrow{v}^{j+1}}(t+1)$ par $\overrightarrow{\widetilde{v}}^j$.

Posons $\forall \epsilon > 0, j = 0 \cdots J, k = (k_1, k_2)$ les constantes suivantes,

$$l_{jk} = \frac{\lambda_{jk}}{\max(G, \lambda_{jk} + \Lambda_{jk})}.\epsilon \quad \text{et} \quad L_{jk} = \frac{\sqrt{\Lambda_{jk}(\lambda_{jk} + \Lambda_{jk})}}{\max(G, \lambda_{jk} + \Lambda_{jk})}.\epsilon.$$

Avec ces notations, on peut affirmer que si $Er(j,k) \leq l_{jk}$, l'erreur commise en approchant la vitesse est plus petite que ϵ et si $Er(j,k) \geq L_{jk}$, l'erreur est plus grande que ϵ. Cette erreur va nous permettre par la suite de juger si le point $(2^j k_1, 2^j k_2)$ suit le mouvement dominant de la scène à ϵ près ou non. Remarquons que dans le cas où $Er(j,k) \in]l_{jk}, L_{jk}[$, nous ne pouvons pas conclure. Cet intervalle est maximal lorsque le point se trouve sur un bord rectiligne d'un objet uniforme (3.4(a)) ; dans ce cas, $\lambda_{jk} = l_{jk} = 0$ et $L_{jk} = \epsilon$. Il est minimal lorsque le point se trouve sur le coin d'un objet rectangulaire uniforme (3.4(b)), on a alors $\lambda_{jk} = \Lambda_{jk} = 0$, $l_{jk} = \frac{\epsilon}{2}$ et $L_{jk} = \frac{\epsilon}{\sqrt{2}}$.



(a) (b)

FIGURE 3.4 – Deux situations extrêmes pour l'intervalle d'incertitude. (a) Le point p se situe sur un bord rectiligne d'un objet, l'intervalle est alors maximal ($l_{jk} = 0, L_{jk} = \epsilon$). (b) p se situe sur le coin d'un objet rectangulaire, l'intervalle est minimal ($l_{jk} = L_{jk} = \frac{\epsilon}{\sqrt{2}}$).

Le terme d'attache aux données est défini à partir de Er de la façon suivante :

$$\begin{aligned} &\forall i = (2^j k_1, 2^j k_2) \in S^j \\ &q_i(e_i, o_i) = \begin{cases} \frac{1}{1+\exp(-16(Er(j,k)-l_{jk}))} & \text{si } e_i = 0 \\ \\ 1 - \frac{1}{1+\exp(-16(Er(j,k)-L_{jk}))} & \text{si } e_i = 1 \end{cases} \end{aligned} \tag{3.24}$$

Ce potentiel incite les sites du graphe à prendre l'étiquette 0 (conforme au mouvement

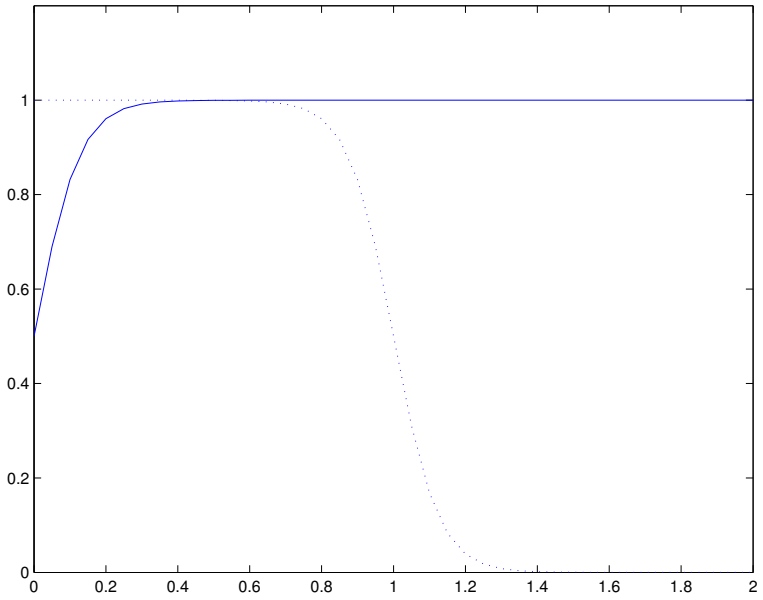


FIGURE 3.5 – Potentiel d'attache aux données dans le cas de la figure (3.4(a)) avec $\epsilon = 1$, la courbe continue correspond à un site conforme au mouvement dominant $e = 0$, la courbe en pointillé à un site non conforme $e = 1$.

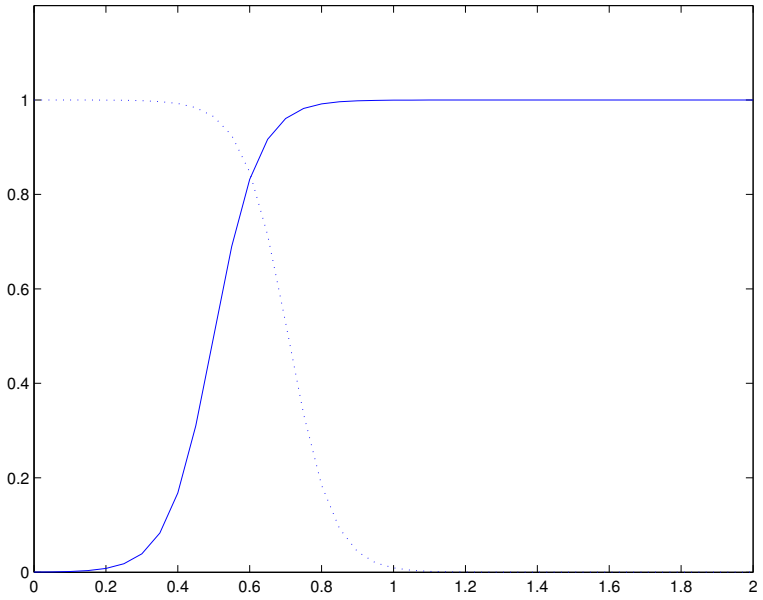


FIGURE 3.6 – Potentiel d'attache aux données dans le cas de la figure (3.4(b)) avec $\epsilon = 1$, la courbe continue correspond à un site conforme au mouvement dominant $e = 0$, la courbe en pointillé à un site non conforme $e = 1$.

dominant) dans le cas où $Er(j,k) \leq l_{jk}$ et à prendre l'étiquette 1 (non conforme au mouvement dominant) si $Er(j,k) \geq L_{jk}$. Nous avons représenté les deux situations extrêmes décrites précédemment (Fig 3.4) sur les figures (3.5) et (3.6). On remarque bien que la plage d'incertitude entre un mouvement conforme et non conforme d'un point situé sur un contour rectiligne d'un objet est beaucoup plus important que pour un point situé sur un coin. Dans cette situation, la modélisation hiérarchique favorisera l'étiquette la plus représentée sur les sites voisins si $j = J$ ou favorisera l'étiquette portée par le parent du site si celui-ci se trouve à une résolution plus fine.

3.3 Détection de mouvement dans une séquence d'images

Les déplacements des objets étant continus au cours du temps, la carte de détection du mouvement entre deux instants varie peu. Il est donc tout à fait naturel d'utiliser la carte de détection à l'instant précédent pour en déduire celle à l'instant t. Cette nouvelle information va nous permettre d'atténuer les ambiguïtés dues aux bruits tels que les erreurs d'interpolation, les artefacts d'acquisition, les variations d'illumination qui rendent les algorithmes fondés sur deux images instables. Pour ce faire, nous allons définir un nouveau modèle pour la détection du mouvement dans les séquences formées de plusieurs images.

3.3.1 Principe de la méthode

La méthode utilisée pour détecter les régions mobiles dans la scène est résumée figure 3.7. Entre les deux premières images de la séquence, nous détectons ces régions grâce à la méthode décrite précédemment en calculant le mouvement sur l'image entière. Puis, pour détecter les zones mobiles dans la scène à l'instant t, nous utilisons la carte de détection $E(t-1)$ à l'instant $t-1$ projetée dans le sens du mouvement dominant $\overrightarrow{v}(t-1)$ que l'on note $\widetilde{E}(t-1)$. Ce champ d'étiquettes correspond en fait à la prédiction de la carte de détection à l'instant t à partir de l'instant précédent.

$$\begin{gathered} \forall j, \forall i = (2^j k_1, 2^j k_2) \in S^j \\ \widetilde{E}_i(t-1) = E_{i+\overrightarrow{v}^j(t-1)}(t-1). \end{gathered} \tag{3.25}$$

Pour calculer le mouvement dominant entre les instants t et $t+1$, nous utilisons comme support d'estimation la région $R(t)$ considérée conforme à ce mouvement soit

$$R(t) = \{i = (2^j k_1, 2^j k_2) \in S^j \forall j | \widetilde{E}_i(t-1) = 0\}. \tag{3.26}$$

En effet, cette zone est une bonne estimation de la région conforme au mouvement dominant puisque les objets mobiles se déplacent peu entre deux images consécutives. Pour cette même raison, le champ d'étiquettes à l'instant t dépend de l'instant $t-1$. C'est pourquoi nous introduisons une modélisation temporelle-hiérarchique du champ d'étiquettes $E(t)$.

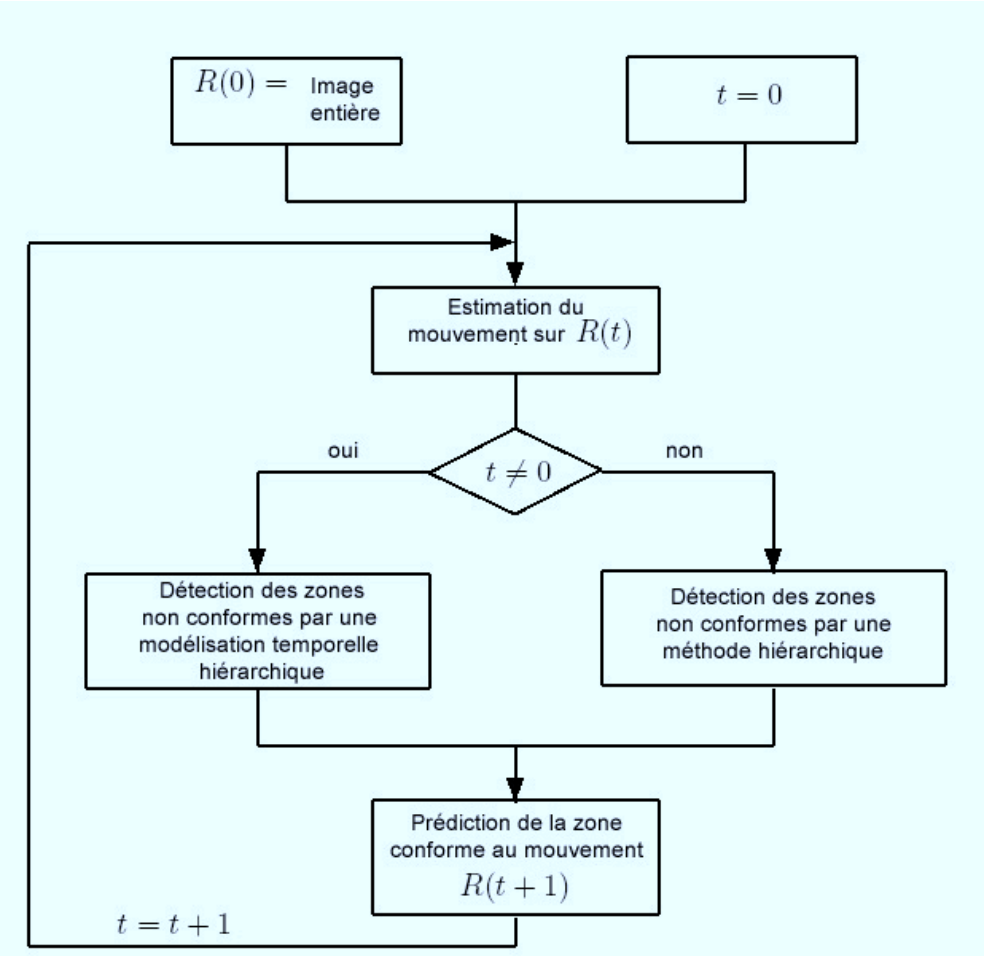


FIGURE 3.7 – Schéma du principe de détection des zones non conformes au mouvement dominant sur une séquence d'images complète.

3.3.2 Modélisation temporelle-hiérarchique

Considérons l'ensemble des champs d'étiquettes $(E(1), E(2), \cdots, E(t), \cdots)$ où $E(t)$ est la configuration du champ d'étiquettes à l'instant t. Supposons que celui-ci forme une chaîne de Markov d'ordre 1 c'est-à-dire que :

$$P(E(t) = e(t)|E(r) = e(r), \quad 0 < r < t) = P(E(t) = e(t)|E(t-1) = e(t-1)) \quad (3.27)$$

Remarquons qu'une hypothèse de Markov d'ordre supérieur permettrait de considérer un "effet mémoire" du champ d'étiquettes plus important et pourrait peut-être diminuer encore plus les erreurs de détection. Mais la minimisation de l'énergie globale ainsi définie serait beaucoup plus coûteuse en temps de calcul. De plus, d'autres effets indésirables peuvent apparaître. En effet, la dépendance du champ d'étiquettes à l'instant t avec les précédents peut obliger un point d'un fond uniforme à conserver une étiquette non conforme au mouvement de la caméra si celui-ci correspondait à un point d'un objet mobile dans les images précédentes.
A partir de l'hypothèse (3.27), la distribution conjointe devient

$$\begin{aligned} P(E(t)|(E(r) \quad 0 < r < t, O(t))) &= P(E(t)|(E(t-1), O(t))) \\ &= P((E(t-1), O(t))|E(t))P(E(t)), \end{aligned} \tag{3.28}$$

en supposant $O(t)$ indépendant de $E(t-1)$,

$$P(E(t)|(E(r) \quad 0 < r < t, O(t))) = P(E(t-1)|E(t))P(O(t)|E(t))P(E(t)). \tag{3.29}$$

De plus, si $E(t)$ est conditionnellement indépendant à $E(t-1)$,

$$P(E(t)|E(t-1)) = \prod_{i \in S} P(E_i(t)|E(t-1)). \tag{3.30}$$

Si $\forall i \in S^j$, $E_i(t)$ est conditionnellement à $E(t-1)$ indépendant de $E_{p \neq i + \overrightarrow{v}^j(t-1)}(t-1)$,

$$P(E_i(t)|E(t-1)) = P(E_i(t)|\widetilde{E}_i(t-1)). \tag{3.31}$$

Par conséquent, en notant

$$P(E_i(t)|\widetilde{E}_i(t-1))) = \exp(-h(e_i, \widetilde{e}_i)) \tag{3.32}$$

et en connaissant les processus $E(t-1)$ et $O(t)$, la meilleure configuration $e(t)$ de $E(t)$ au sens du MAP est solution de

$$\widehat{e}(t) = \min_e [\sum_{<i,j> \in S^J} v_{i,j}(e_i, e_j) + \sum_{i \notin S^J} w_i(e_i, e_{\bar{i}}) + \sum_{i \in S} q_i(e_i, o_i) + \sum_{i \in S} h(e_i, \widetilde{e}_i)], \tag{3.33}$$

où h exprime la dépendance temporelle entre deux cartes de détection consécutives. Notons que les déplacements $\overrightarrow{v}^j$ ne sont pas nécessairement entiers, il nous est donc impossible de connaître de façon exacte la carte de détection $\widetilde{e}(t-1)$. Même si cette indétermination ne se présente qu'aux frontières des objets mobiles et n'a donc qu'une incidence très locale, nous préférons remédier à ce problème en prenant en compte l'étiquette des sites voisins de i. Nous définissons le voisinage $\mathcal{V}_i$ d'un site $i \in S^j$ les 4 sites les plus proches de i à la

résolution j, le parent $\bar{i}$ et ses quatre enfants $\underline{i}$ (Fig 3.8). Ainsi, nous définissons h de la façon suivante

$$\forall i \in S^j \quad h(e_i, \widetilde{e}_i) = \Gamma(\widetilde{e}_i) \sum_{p \in \{i\} \cup \mathcal{V}_i} [1 - \delta(e_i, \widetilde{e}_p)], \tag{3.34}$$

avec

$$\Gamma(\widetilde{e}_i) = \gamma \frac{1}{\sharp(\mathcal{V}_i)} \sum_{p \in \mathcal{V}_i} \delta(\widetilde{e}_i, \widetilde{e}_p).$$

Le facteur de pondération $\Gamma(\widetilde{e}_i)$ permet de ne pas avoir un effet mémoire trop important et ainsi d'éviter de conserver une étiquette non conforme au mouvement sur un site immobile qui avait été auparavant considéré comme mobile. Il permet de pondérer l'information temporelle d'un site i en fonction du nombre d'étiquettes identiques sur le voisinage $\mathcal{V}_i$. $\Gamma(\widetilde{e}_i) = \gamma$ si tous les voisins du site i portent la même étiquette. Dans ce cas il est donc "fort probable" que i ait la même étiquette à l'instant t. Au contraire, si tous les voisins de i n'ont pas l'étiquette de i, il est "peu probable" que ce point garde la même étiquette. Dans ce cas, $\Gamma(\widetilde{e}_i) = 0$ et l'information temporelle n'est pas prise en compte. Ceci se produit sur un site frontière d'un objet mobile ou encore un site mal étiquetté.

3.4 Résultats expérimentaux

Le but de ce paragraphe est de valider notre méthode de détection de mouvement à partir de séquences d'images tests. Dans un premier temps, nous évaluerons la méthode sur deux séquences synthétiques, la première avec un fond mobile et l'autre avec un fond fixe. Celles-ci nous permettront de vérifier l'influence des différents paramètres introduits. Puis nous analyserons les résultats obtenus sur des séquences réelles : la séquence Ping-Pong et la séquence CoastGuard. Dans les résultats présentés, les zones blanches de l'image correspondent aux zones conformes au mouvement dominant.

Remarquons que d'un point de vue expérimental, l'utilisation d'ondelettes analytiques ne permet pas d'avoir d'observations sur les deux premières résolutions de l'image. Par conséquent, la détection de mouvement la plus fine que l'on puisse obtenir sur une image de taille $2^n \times 2^m$ est de taille $2^{n-2} \times 2^{m-2}$. Par conséquent, la modélisation markovienne hiérarchique est construite sur le graphe $S = \cup_{j=2}^{J} S^j$.

Une valeur trop importante du paramètre G introduit dans la formule (4.6) conduit à ne considérer les informations de mouvement que sur les zones à variations d'intensité importante et une valeur trop faible augmente les mauvaises détections en cas de bruit dans la séquence. Sur l'ensemble des expérimentations, ce paramètre sera fixé à 50.

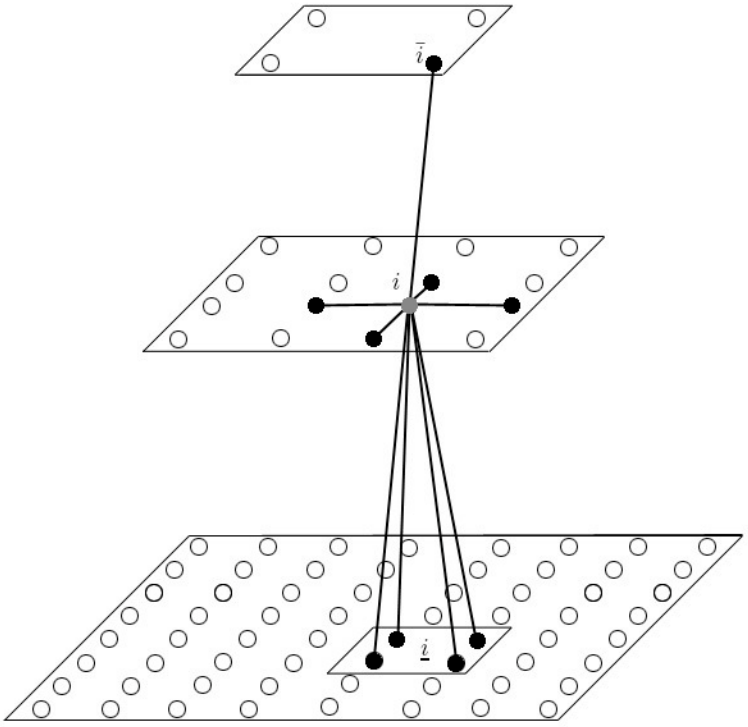


FIGURE 3.8 – Voisinage $\mathcal{V}_i$ d'un site i.

3.4.1 Séquences synthétiques

Séquence Tree

Cette séquence synthétique n'est composée que de deux images. Elle va nous permettre de montrer l'intérêt de la modélisation hiérarchique. Les deux images sont générées à partir d'une image de la séquence "Tree" de Barron [1] en translation uniforme à la vitesse $\overrightarrow{v} = (-5, 0)$ sur laquelle nous avons ajouté deux cercles texturés, le cercle en haut à droite en mouvement à la vitesse $\overrightarrow{v}_{C_1} = (0, 5)$ et le cercle en bas à gauche à la vitesse $\overrightarrow{v}_{C_2} = (10, 0)$ (Fig 3.9). Une différence de 10 niveaux de gris est introduit entre les deux

1. séquence disponible sur le site ftp :csd.uwo.ca/pub/vision

images pour vérifier la robustesse de notre méthode dans le cas où l'E.C.M.A. n'est pas vérifiée. Puis, les deux images sont bruitées par un bruit blanc gaussien d'écart type 15.

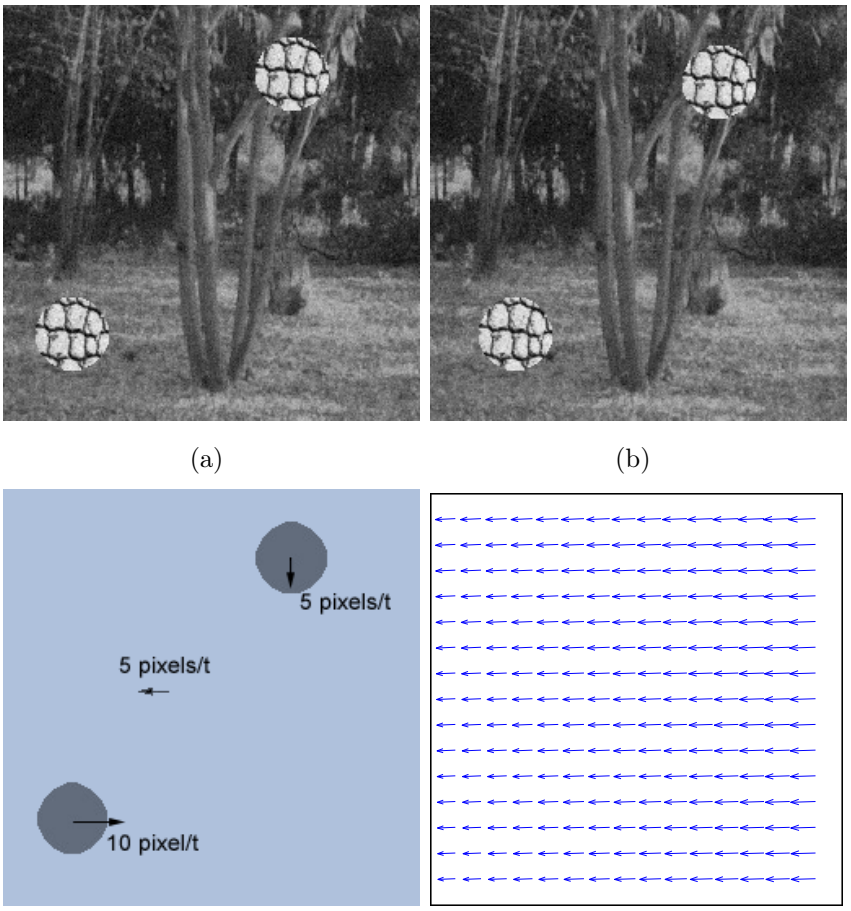


FIGURE 3.9 – Séquence Tree. (a) Première image, (b) deuxième image, (c) mouvement réel, (d) mouvement dominant estimé.

Nous avons estimé le mouvement dominant sur 5 niveaux de résolution et obtenu les paramètres suivants

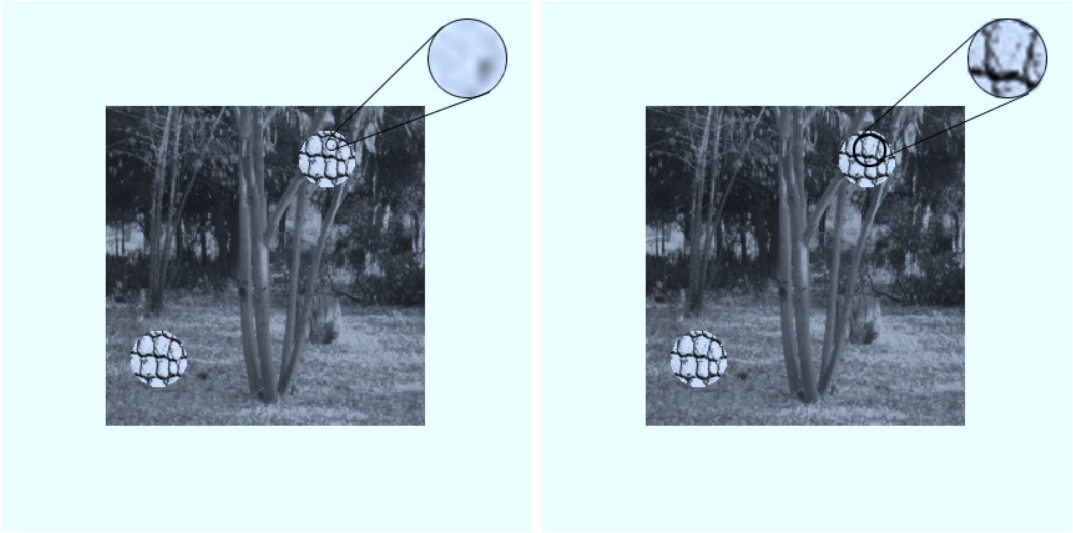
$$\Theta = (a_1, a_2, a_3, a_4, a_5, a_6)^T = (-0.0001, 0, -5.012, 0.0001, 0, 0.2125)^T.$$

Le résultat du champ de vitesses obtenu est représenté figure 3.9(d) où l'origine du repère est le coin en haut à gauche de l'image, l'axe des x est l'axe horizontal orienté de gauche à droite et l'axe y l'axe vertical orienté de haut en bas.

Dans un premier temps, nous comparons notre méthode de détection avec une méthode qui s'appuie simplement sur la compensation du mouvement dominant entre les deux images à la résolution la plus fine. La figure 3.11(a) présente la détection obtenue à partir

du maximum de vraisemblance local sur les observations q_i c'est à dire sans tenir compte du terme de régularisation v_{ij} sur le voisinage et en prenant $\epsilon = 0.6$. On remarque que le résultat est très bruité. En effet, de nombreuses fausses alarmes sont présentes alors que les deux cercles ne sont pas complètement détectés. Ceci est dû à plusieurs raisons :

1. Le bruit introduit entre les deux images, nous oblige à utiliser une méthode robuste pour la détection de mouvement, ce qui n'est pas le cas du maximum de vraisemblance.
2. Certaines zones de l'image sont faiblement texturées et rendent la matrice A (formule (3.14)) numériquement instable comme le montre la figure 3.10.
3. La différence entre la vitesse $\overrightarrow{v}_{C_i}$ d'un point d'un des cercles et le mouvement dominant $\overrightarrow{v}$ est trop importante pour être mesuré sur des ondelettes à supports petits à cause de l'aliasage temporel.



(a) (b)

FIGURE 3.10 – Influence de la taille du support des ondelettes sur la mesure d'erreur. Utiliser des supports d'ondelettes petits peut aboutir à une matrice A instable si la zone est localement peu texturée(a). Il faut donc considérer des ondelettes à plus grands supports (b).

Le premier point peut être appréhendé en modélisant le problème par un modèle markovien. C'est-à-dire par la minimisation de

$$\begin{aligned} \widehat{e}(t) &= \min_e [U_1(e,o) + U_2(e)] \\ &= \min_e [\textstyle\sum_{i \in S^0} q_i(e_i, o_i) + \sum_{<i,j> \in S^2} v_{i,j}(e_i, e_j)]. \end{aligned} \tag{3.35}$$

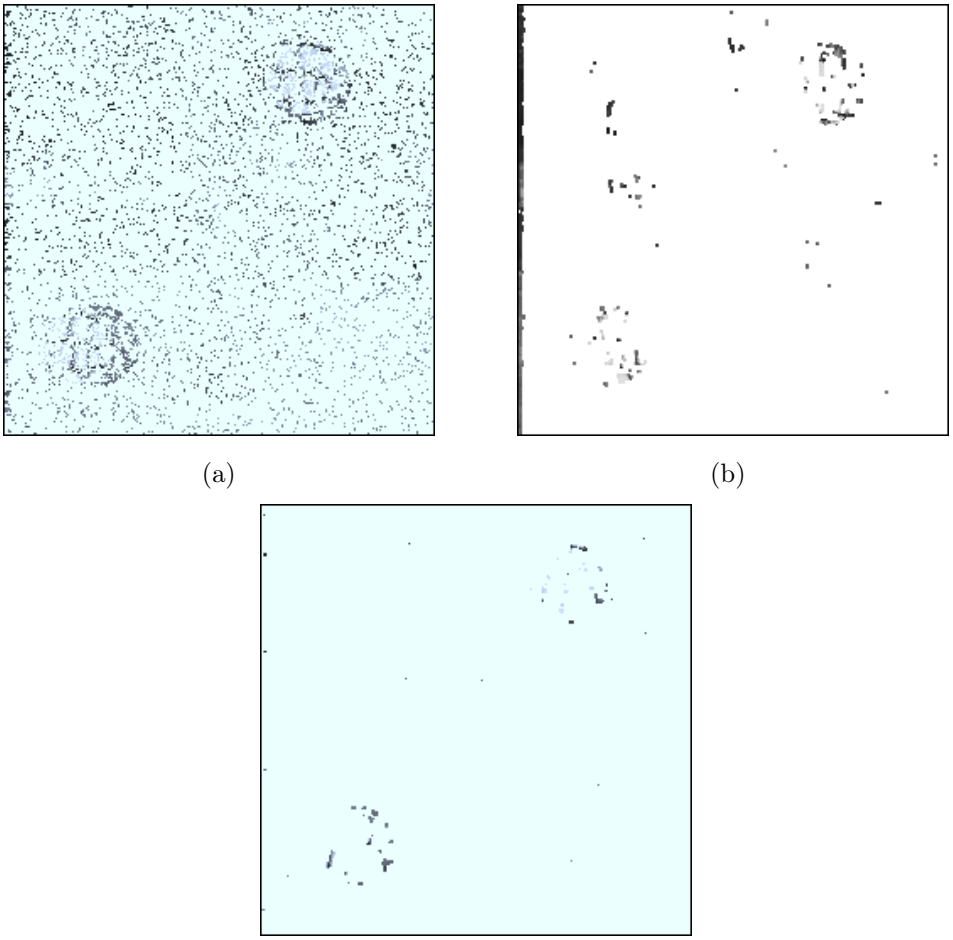


FIGURE 3.11 – Séquence Tree. (a) Maximum de Vraisemblance ($\epsilon = 0.6$), (b) ICM mono-résolution, (c) Recuit Simulé.

C'est ce qui est fait figure 3.11(b) et figure 3.11(c) grâce respectivement à un algorithme ICM et à un Recuit Simulé. Cependant, bien que ces méthodes améliorent sensiblement la détection en réduisant fortement le bruit, elles ne peuvent pas résoudre les problèmes soulevés en (2) et en (3) car dans ces deux cas, la propriété **3.1** n'est pas utilisable et donc le terme d'attache aux données U_1 n'est pas valable.

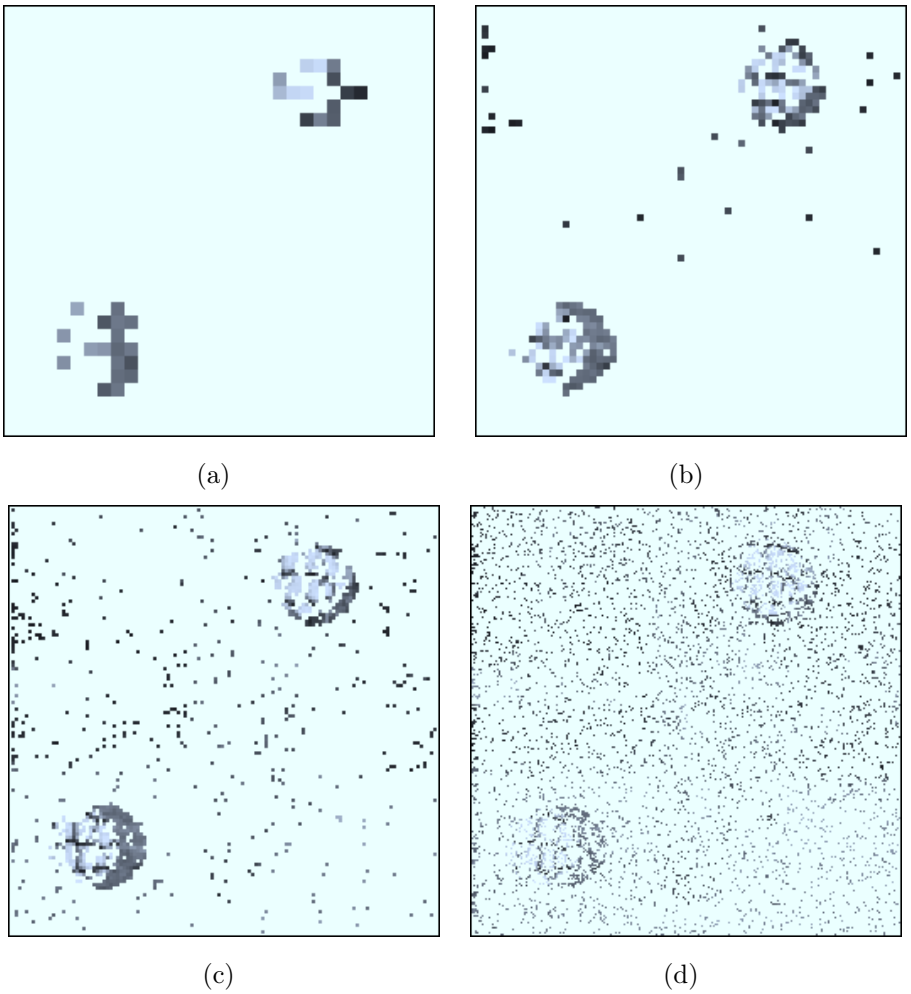


FIGURE 3.12 – Maximum de vraisemblance du terme d'attache aux données sur 4 niveaux avec $\epsilon = 0.6$, (a) $J = 5$, (b) $J = 4$, (c) $J = 3$, (d) $J = 2$.

La figure 3.12 représente le maximum de vraisemblance du potentiel d'attaches aux données $\sum_{i \in S^j} q_i(e_i, o_i)$ sur quatre niveaux de résolutions ($J = 2, 3, 4, 5$). Ces quatre résultats mettent en évidence les difficultés d'application de la propriété **3.1**. A une échelle fine, à cause de l'aliasage temporel, la mesure $Er(j, k)$ n'est valable que si la différence entre le mouvement dominant estimé et le flot réel au point $(2^j k_1, 2^j k_2)$ est faible. Alors qu'à une échelle grossière l'hypothèse de déplacement constant sur le support des ondelettes risque d'être mise en défaut et la propriété **3.1** n'est plus applicable. C'est pour ces raisons que les résultats sont plus ou moins corrects. Pour améliorer la détection, il faut donc faire un juste compromis entre la taille des supports des ondelettes (niveau de résolution de l'image) et l'erreur admissible entre le mouvement dominant et le mouvement réel des points. N'ayant aucune connaissance *a priori* sur les images, ce compromis est réalisé grâce à la dépendance entre deux niveaux de résolution dans la modélisation markovienne hiérarchique. La figure 3.13 montre l'évolution des résultats de la détection par la modélisation hiérarchique sur 4 niveaux ($J = 4$) en fonction du paramètre de dépendance entre les échelles avec $\epsilon = 0.6$ et $\alpha = 0.2$. Pour $\beta = 0$, on ne considère aucune dépendance entre les échelles, le résultat correspond donc au maximum de vraisemblance sur les observations (terme U_1 de la formule (3.35)). Le bruit de détection s'estompe au fur et à mesure que β augmente au détriment d'un effet de bloc sur le résultat dû à la forte dépendance entre un site et son parent. Le choix de la valeur du paramètre β dépend du niveau de bruit dans la séquence d'images. Plus le bruit est élevé, plus β doit être important. Un comparatif des résultats entre une modélisation hiérarchique sur 2 et 3 niveaux est présenté figure 3.14. Avec $J = 3$, certaines zones ne sont pas détectées mais l'effet de bloc est moins présent que pour $J = 4$. Le tableau 3.1 récapitule le temps de calcul de notre méthode sur un PC Pentium III 933Ghz.

	Temps de calcul
Estimation du mouvement dominant	1,8s
Détection sur 3 niveaux de résolution	1,2s
Détection sur 2 niveaux de résolution	2,9s
I.C.M mono-resolution	10,8s

TABLE 3.1 – Temps de calcul de la méthode proposée sur un PC Pentium III 933 Ghz.

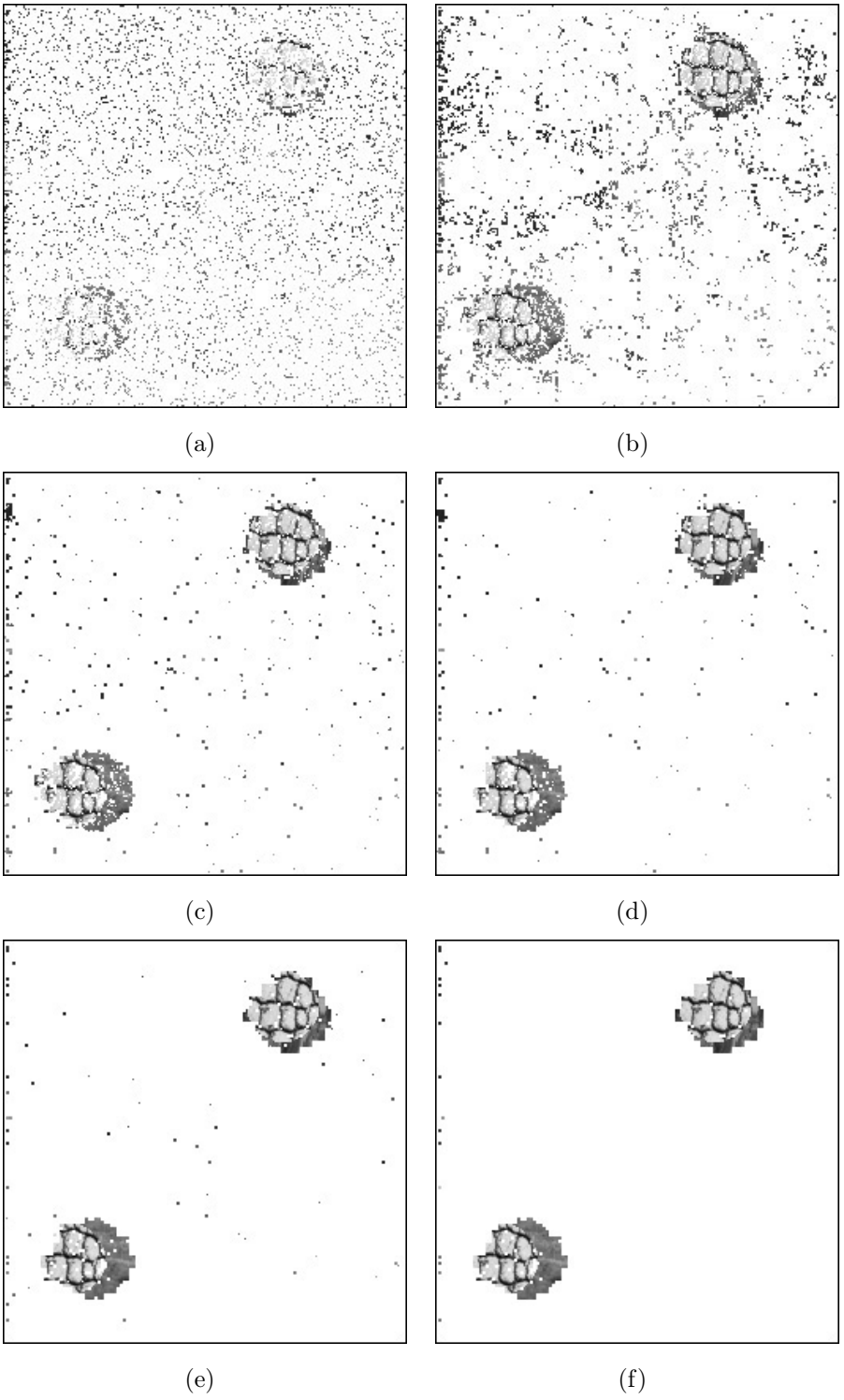
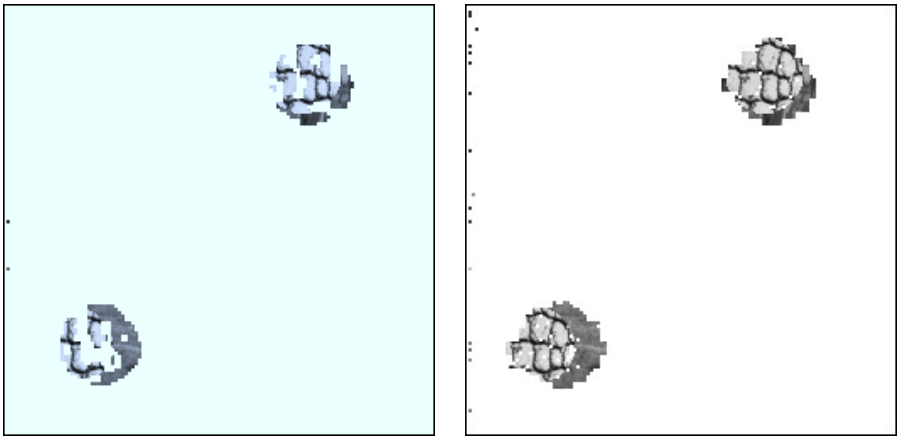


FIGURE 3.13 – Influence du paramètre β avec $\alpha = 0.2$, $\epsilon = 0.6$ et $J = 4$, (a) $\beta = 0$, (b) $\beta = 0.2$, (c) $\beta = 0.4$, (d) $\beta = 0.6$, (e) $\beta = 0.8$, (f) $\beta = 1$.



(a) (b)

FIGURE 3.14 – Résultats en fonction du niveau de résolution avec $\alpha = 0.2$, $\epsilon = 0.6$ et $\beta = 1$. (a) $J = 3$, (b) $J = 4$

Séquence Damier

Cette séquence de 15 images (Fig 3.15) est composée d'un damier immobile sur lequel trois formes géométriques se déplacent en translation uniforme : un rectangle à la vitesse $\overrightarrow{v} = (-2, 0)$, un cercle à la vitesse $\overrightarrow{v} = (-1, -1)$ et enfin un carré à la vitesse $\overrightarrow{v} = (0, 2)$. Les images étant de taille 1024×1024, les résultats obtenus sont présentés sur des images de taille 256×256. Chaque image de la séquence est bruitée avec un bruit blanc gaussien d'écart type 15.

Cette séquence montre l'intérêt du terme temporel pour la détection de mouvement. La figure 3.15 présente les résultats de détection avec $\epsilon = 0.8$, $J = 4$, $\beta = 1$, $\alpha = 0.2$ avec (fig 3.15(g)-(i)) et sans (fig 3.15(d)-(f)) l'intégration de l'information temporelle. On remarque que celle-ci permet comme prévu de réduire les fausses alarmes. Cependant, une dépendance temporelle trop importante contraint souvent un site d'une zone uniforme détecté comme non conforme à l'instant t à le rester à l'instant $t + 1$.

Dans la suite des expérimentations, nous utiliserons (sauf mention contraire) les paramètres suivants :

J	α	β	γ	G
4	0.2	1	0.2	50

seul le paramètre ϵ qui détermine la marge d'erreur admissible entre le mouvement apparent réel et le mouvement estimé variera en fonction de la séquence traitée.

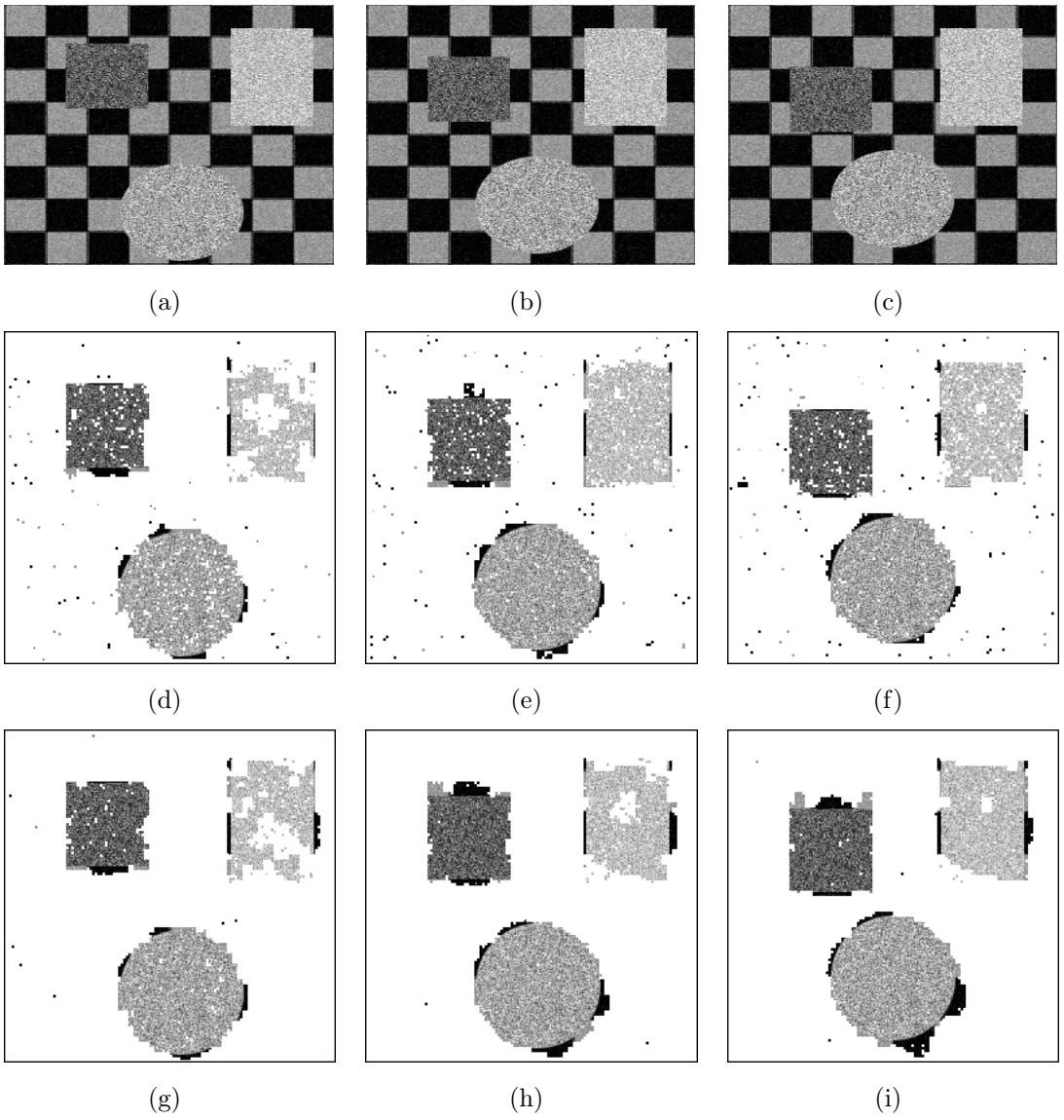


FIGURE 3.15 – Prise en compte de la dépendance temporelle, (a) Image à $t = 2$,(b) $t = 6$, (c)$t = 11$,(d)-(e)-(f) résultat sans intégration temporelle à $t = 2, 6, 11$, (g)-(h)-(i) résultat avec intégration temporelle : $\gamma = 0.3$.

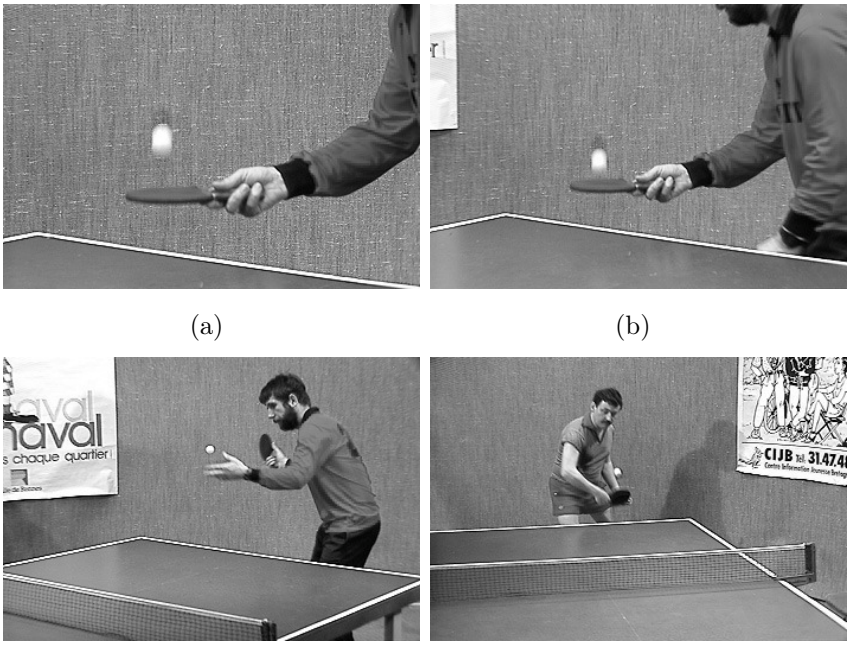


FIGURE 3.16 – Images de la séquence Ping-Pong. (a) $t = 10$, (b) $t = 40$, (c) $t = 70$, (d) $t = 90$.

3.4.2 Séquences réelles

Séquence Ping-Pong

Cette séquence de 150 images de taille 352×240 est souvent utilisée en analyse de séquence d'images. Elle peut être décomposée en trois parties distinctes dans lesquelles le mouvement de la caméra diffère. Dans les premiers instants (de $t = 0$ à $t = 14$), la caméra est fixe et se focalise sur la main droite du premier joueur dans laquelle il tient une raquette. La main, la raquette et la balle de ping-pong ont un mouvement de va et vient de haut en bas (Fig 3.16(a)). Entre les instants $t = 15$ et $t = 88$ la caméra fait un zoom arrière pendant que le joueur envoie la balle à son adversaire (Fig 3.16(b) et Fig 3.16(c)). Enfin, à partir de $t = 89$, la caméra est de nouveau fixe et filme le deuxième joueur qui renvoie la balle (Fig 3.16(d)).

Les résultats de cette séquence sont présentés figure 3.17 avec $\epsilon = 0.9$. Lorsque la caméra est fixe (entre les instants $t = 0$ à 14 et $t = 89$ à 150), la détection de mouvement ne pose pas de réel problème. La main et la raquette du premier joueur sont correctement détectées ainsi que la balle de ping-pong (Fig 3.17(a)). Le deuxième joueur l'est aussi (Fig 3.17(e) et Fig 3.17(f)). Par contre lorsque la caméra fait le zoom arrière, plusieurs fausses alarmes sont présentes. En effet, les contours de la table de ping-pong et une partie de

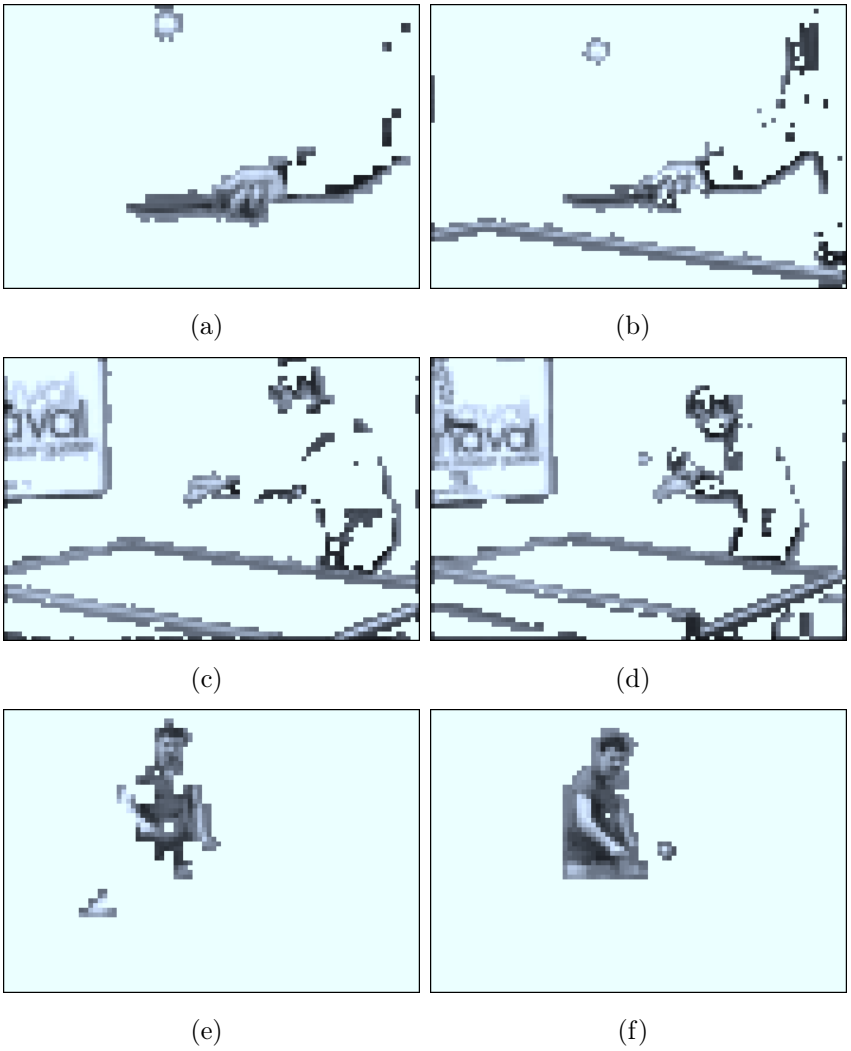
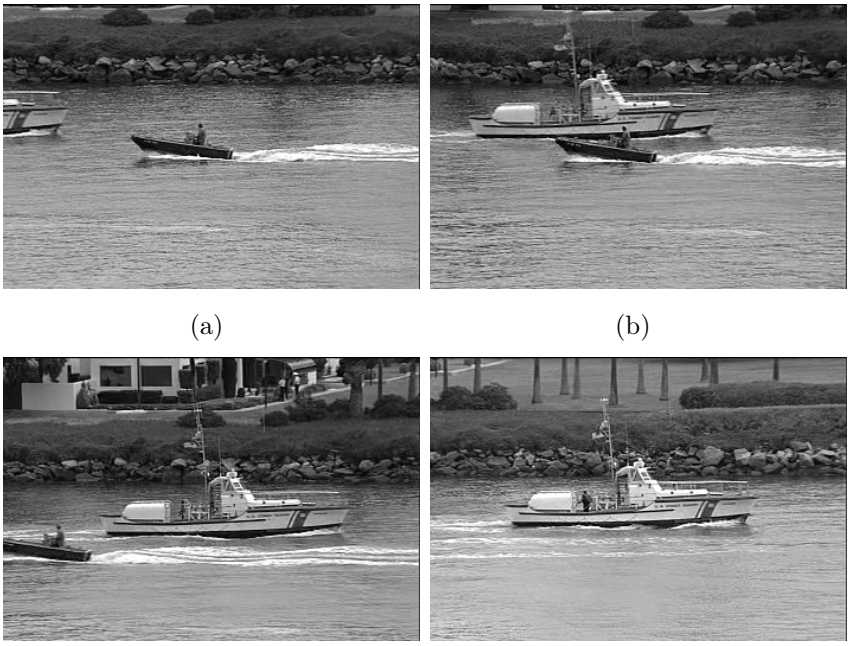


FIGURE 3.17 – Résultats de la détection de mouvement sur la séquence Ping-Pong. (a) $t = 19$, (b) $t = 33$, (c) $t = 63$, (d) $t = 73$, (e) $t = 120$, (f) $t = 129$.

l'affiche sont considérés en déplacement. Ceci est dû à deux raisons. La première est que l'échantillonnage spatial et temporel de la séquence engendre en effet une différence spatio-temporelle sur les bandes blanches de la table de ping-pong. La seconde est que l'hypothèse de mouvement affine du mouvement apparent de la caméra ne permet pas de décrire correctement le mouvement réel de celle-ci. Par conséquent, le champ de déplacement affine $\overrightarrow{v}$ calculé ne permet pas de modéliser le déplacement apparent de l'ensemble des régions statiques de l'image et la détection de mouvement considère alors ces régions comme non conformes au mouvement affine estimé.



(a) (b)

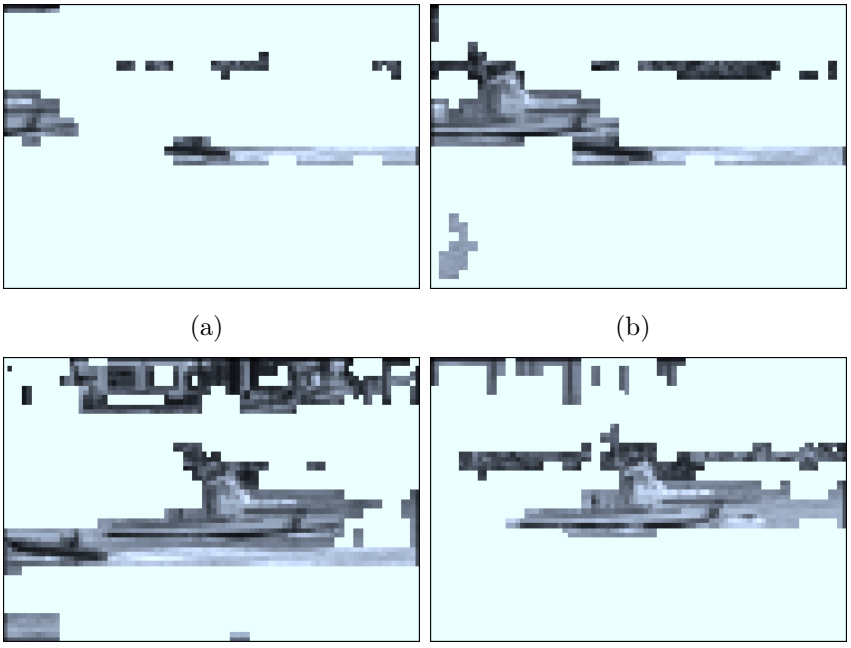
(c) (d)

FIGURE 3.18 – Quatre images de la séquence CoastGuard, (a) $t = 10$, (b) $t = 60$, (c) $t = 104$, (d) $t = 260$.

Séquence CoastGuard

Cette séquence a déjà été utilisée dans le chapitre 3 pour estimer le mouvement dominant. Dans un premier temps, la caméra suit une barque par un mouvement panoramique de la droite vers la gauche. Puis une fois que la barque a croisée le bateau des gardes côtes, la caméra se met à suivre celui-ci en se déplaçant de la gauche vers la droite (Fig 3.18).

Les résultats sur cette séquence sont présentées figure 3.19 en prenant comme marge d'erreur entre le mouvement estimé et le mouvement réel $\epsilon = 1.5$. Les deux bateaux ainsi que les vagues qu'ils produisent sont détectés comme mobiles. Quelques zones de l'eau sont parfois aussi détectées (Fig 3.19(b) et Fig 3.19(c)) à cause de leur mouvement complexe qui ne peut être modélisable par un simple mouvement affine. De plus, à cause des différences de profondeur dans la scène, la modélisation affine du mouvement dominant ne permet pas de modéliser correctement l'ensemble des zones statiques de la scène. C'est pourquoi, le fond de l'image est considéré comme mobile.



(a) (b)

(c) (d)

FIGURE 3.19 – Détection de mouvement sur la séquence CoastGuard avec $\epsilon = 1.5$, $t = 13, 40, 100, 283$

3.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème de la détection de mouvement dans les séquences d'images à partir d'une caméra mobile. Nous avons vu que l'estimation multirésolution du mouvement dominant par analyse d'ondelettes présentée au chapitre précédent se prêtait très bien à une modélisation markovienne hiérarchique de type arbre tronqué proposé par Chardin et Pérez. Cette modélisation a deux avantages. En effet, grâce à l'algorithme d'optimisation semi-itératif déduit de cette modélisation, cette méthode est plus rapide que les méthodes markoviennes classiques à une seule échelle et même que la méthode d'optimisation multirésolution proposée par Pérez [Pér93]. Elle permet de plus de contourner le problème de l'aliasage temporel en considérant une mesure d'erreur non pas uniquement sur l'échelle la plus fine de l'image mais sur l'ensemble de la résolution.

Notre méthode a néanmoins deux inconvénients qui induisent des effets de bloc dans les résultats. L'algorithme de transformée en ondelettes analytiques (Fig **??**) ne permet de calculer les coefficients d'ondelettes qu'à partir de la résolution $j = 2$. Nous ne disposons donc d'aucune information aux résolutions supérieures. C'est pourquoi, pour une image de taille $2^m \times 2^p$, le résultat est de taille $2^{m-2} \times 2^{p-2}$. De plus, étant donné que dans la

modélisation markovienne hiérarchique, nous ne considérons pas de dépendance spatiale sur un même niveau de résolution, l'interaction entre deux sites voisins d'une même échelle intervient au niveau de leur premier ancêtre en commun. Il y a donc une forte corrélation entre les quatre enfants d'un même site, ce qui a pour conséquence de créer des effets de bloc. Ces effets de bloc sont d'autant plus importants que la dépendance hiérarchique *a priori* est importante. Le premier inconvénient peut être résolu en interpolant l'image originale afin d'obtenir une image à traiter de taille $2^{m+2} \times 2^{p+2}$. Pour le deuxième, nous aurions pu introduire des dépendances spatiales sur chaque résolution de l'image tel que le proposent Kato et al. [KBZ93]. Mais cette modélisation a l'inconvénient de casser la causalité du graphe, le problème d'optimisation devenant alors plus complexe et plus coûteux en temps de calcul.

Les résultats présentés sont malgré tout satisfaisants et ces effets de bloc ne constitueront pas un handicap dans l'application recherchée de cette thèse à savoir la détection d'obstacles dans un milieu routier. Il est à noter que le résultat de cette méthode ne permet pas de conclure que chaque zone connexe détectée constitue un seul et même objet se déplaçant dans l'image. Si l'on souhaite distinguer les différents objets en déplacement dans la scène, cela ne peut se faire qu'avec une étape de segmentation au sens du mouvement.

Chapitre 4

Segmentation d'images au sens du mouvement par multirésolution

Dans ce chapitre, nous présentons une méthode de segmentation au sens du mouvement entre deux images consécutives d'une séquence qui s'appuie sur une étude multirésolution des images. L'estimation du mouvement se fera conjointement avec la segmentation sur chaque résolution de l'image. Nous commencerons ce chapitre par un état de l'art non exhaustif des méthodes de segmentation au sens du mouvement. Puis, dans une deuxième partie, nous expliciterons notre méthode d'estimation et de segmentation du mouvement mono-résolution. L'estimation du mouvement étant déduite de l'analyse en ondelettes de l'équation de contrainte du flot optique (Chapitre 3), nous sommes une nouvelle fois confrontés au problème de l'aliasage temporel. C'est pourquoi nous définirons dans la troisième partie, une méthode multirésolution. Enfin, nous proposerons un moyen d'initialiser la méthode sans connaissance *a priori* sur la scène et sans supposer que l'on puisse extraire un mouvement dominant. Des résultats de notre méthode sur des séquences réelles et synthétiques seront présentés dans la cinquième partie.

4.1 Éléments de l'état de l'art

Le but de la segmentation au sens du mouvement dans les séquences d'images est de partitionner l'image en régions ayant des déplacements différents. Ce problème se révèle très important pour l'interprétation et l'analyse de scènes à partir d'une caméra. C'est en effet une étape nécessaire dans le cadre de la compression vidéo, de la détection d'obstacles en robotique mobile, de la vidéo-surveillance, de la création automatique de résumé de vidéo, de recherche dans une base de séquence d'images.

Ainsi, une des difficultés majeures de la segmentation au sens du mouvement est qu'il

y a une interdépendance entre deux problèmes difficiles en traitement d'images : l'estimation du mouvement et la segmentation d'une image en zones homogènes. En effet, une mauvaise estimation du mouvement affecte la qualité de la segmentation qui en découle. Réciproquement, l'estimation du mouvement sur chaque région de la partition sera d'autant plus précise que le découpage de l'image en zones homogènes au sens du mouvement est correct. Il existe différentes méthodes pour segmenter l'image au sens du mouvement, nous les classons en trois catégories : les méthodes séquentielles, les méthodes de regroupement des régions et les méthodes conjointes.

4.1.1 Les méthodes de segmentation séquentielle

Ces approches consistent à découper l'image de façon successive en estimant le mouvement apparent dominant des zones non étiquetées puis en détectant les zones non conformes à ce modèle de mouvement. Le principe est le suivant : On part de l'image globale sur laquelle on estime le mouvement dominant, les zones conformes à ce mouvement formant une première région de la partition. Ce procédé d'estimation et de détection est itéré sur les zones non conformes jusqu'à ce que l'ensemble des pixels soient classés [IRP92][ASB94]. Le défaut majeur de ces méthodes est que la convergence n'est assurée que sous l'hypothèse forte de la présence de mouvement dominant à chaque itération. De plus, certaines zones de faible gradient d'intensité peuvent se retrouver conformes au mouvement dominant dès la première étape et être ainsi mal étiquetées.

4.1.2 Les méthodes de regroupement de région

Ces méthodes partent d'une segmentation préalable de l'image au sens d'un critère autre que celui du mouvement. Ce premier partitionnement de l'image peut tout simplement être un découpage arbitraire de l'image en blocs de taille fixe [WA94]. De façon plus sophistiquée, si l'on considère que les frontières des régions en mouvement sont aussi des frontières de région au sens de l'intensité, ce premier découpage peut être fourni par une segmentation spatiale de l'image au sens de l'intensité, de la texture ou de la couleur [GB00], [MBK98], [AET98].

Le mouvement de chaque région de la partition initiale est souvent caractérisé par une modélisation paramétrique $2D$ estimée par des méthodes décrites au chapitre 3. Les méthodes diffèrent surtout sur la façon dont le regroupement des régions élémentaires est mené. Il peut être réalisé par une simple comparaison des modèles par une classification du type *k-means* [WA94], par des techniques de clustering [AET98], de maximum de vraisemblance [MBK98] ou une modélisation markovienne [GB00].

Ces méthodes ont souvent l'avantage de fournir un découpage plus précis des frontières des régions au sens du mouvement que les méthodes séquentielles grâce à l'étape préliminaire de segmentation spatiale. Néanmoins, l'utilisation d'une carte de segmentation initiale n'est pas sans poser de problème. En effet, cette segmentation se doit d'être suffisamment fine pour que les régions initiales ne soient animées que d'un seul modèle de mouvement sans quoi nous obtiendrions une sous-segmentation au sens du mouvement. Or, si la taille d'une région est trop petite, l'estimation du modèle paramétrique sur celle-ci risque d'être peu fiable (surtout si les régions initiales ont été obtenues par un critère d'homogénéité au sens de l'intensité sur lesquelles l'information mouvement est difficilement exploitable). Dans ce cas, nous sommes en présence de régions dont le mouvement est mal estimé qui ne peuvent être regroupées avec d'autres par la comparaison de leurs modèles paramétriques respectifs. Le résultat risque donc d'être une sur-segmentation au sens du mouvement. Pour amoindrir ce problème, Konrad et Dang [KD96] proposent d'estimer le modèle de mouvement sur l'union des deux régions susceptibles d'être réunies et de vérifier sa conformité sur chacune d'entre elles.

4.1.3 Les méthodes d'estimation du mouvement et de segmentation conjointes

Les méthodes d'estimation du mouvement et de segmentation conjointes cherchent à obtenir simultanément l'estimation et la segmentation du mouvement. Elles reposent sur la définition d'une énergie modélisant l'adéquation entre les modèles de mouvement et les observations dans l'image. Le calcul du mouvement peut être effectué hors du cadre de l'énergie qui modélise le problème de segmentation [OB98] ou faire partie des variables à minimiser [MP98]. Ces énergies sont définies à partir d'approche variationnelle [Cre03] ou par modélisation bayésienne [CTS97],[OB98]. L'optimisation se fait généralement en deux étapes qui sont itérées jusqu'à convergence. La première consiste à étiqueter les pixels de l'image en fonction des différents modèles de mouvement à disposition, puis la seconde réévalue les paramètres de mouvement à l'aide de la nouvelle segmentation. Ces méthodes donnent des résultats très intéressants, mais les problèmes d'optimisation sous-jacents sont souvent difficiles à résoudre et coûteux en temps de calcul. La méthode que nous développons dans ce chapitre se place dans cette dernière catégorie où nous allons tenter de réduire la complexité algorithmique grâce à une résolution multi-échelle du problème.

4.2 Segmentation mono-résolution du mouvement entre deux images

Dans cette partie, nous nous intéressons au problème de la segmentation au sens du mouvement entre deux images I_1 et I_2 sur la résolution la plus fine de l'image. L'estimation du mouvement dans l'image se fera toujours par projection sur une base de fonctions, puis l'étape de segmentation sera obtenue par une modélisation markovienne. Nous verrons une fois de plus qu'il sera judicieux d'utiliser des ondelettes. Celles-ci permettront en effet à la fois de contourner l'effet d'aliasage mais aussi d'initialiser la segmentation.

Considérons une partition $\mathcal{P}$ de l'image I_1 en p régions $\mathcal{P} = \{\mathcal{R}_1, \cdots, \mathcal{R}_p\}$ (ce découpage peut par exemple être obtenu à partir de la projection dans le sens du mouvement du résultat de la segmentation à l'instant précédent). Le problème se résume à calculer le mouvement de chaque région $\mathcal{R}_i$, puis de vérifier si chaque point de chaque région est bien en conformité avec ce mouvement, ceci afin d'obtenir un meilleur partitionnement de l'image au sens du mouvement.

L'estimation du mouvement d'une région $\mathcal{R}_i$ de l'image se fait de nouveau à partir d'une projection sur une base de fonctions f^n et en supposant celui-ci affine. C'est-à-dire que pour chaque $i = 1 \cdots p$, nous cherchons le vecteur de paramètres $\Theta_i = (a_i, b_i, c_i, d_i, e_i, f_i)^T$ solution du système linéaire sur-déterminé suivant (formule (2.53))

$$\begin{aligned} &\forall s = (x, y) \in \mathcal{R}_i^f \\ &M_s \Theta_i = P_s, \end{aligned} \tag{4.1}$$

où $\mathcal{R}_i^f = \{s = (x, y) | \quad \text{supp}(f_s^n) \subset \mathcal{R}_i, \forall n = 1..N \ \}$ et

$$M_s = \begin{pmatrix} \left[\langle xI, \frac{\partial f_s^1}{\partial x}\rangle + \langle I, f_s^1\rangle\right] & \langle yI, \frac{\partial f_s^1}{\partial x}\rangle & \langle I, \frac{\partial f_s^1}{\partial x}\rangle & \langle xI, \frac{\partial f_s^1}{\partial y}\rangle & \left[\langle yI, \frac{\partial f_s^1}{\partial y}\rangle + \langle I, f_u^1\rangle\right] & \langle I, \frac{\partial f_s^1}{\partial y}\rangle \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \left[\langle xI, \frac{\partial f_s^N}{\partial x}\rangle + \langle I, f_s^N\rangle\right] & \langle yI, \frac{\partial f_s^N}{\partial x}\rangle & \langle I, \frac{\partial f_s^N}{\partial x}\rangle & \langle xI, \frac{\partial f_s^N}{\partial y}\rangle & \left[\langle yI, \frac{\partial f_s^N}{\partial y}\rangle + \langle I, f_s^N\rangle\right] & \langle I, \frac{\partial f_s^N}{\partial y}\rangle \end{pmatrix},$$

$$P_s = \begin{pmatrix} \langle \frac{\partial I}{\partial t}, f_s^1\rangle \\ \vdots \\ \langle \frac{\partial I}{\partial t}, f_s^N\rangle \end{pmatrix}.$$

Nous obtenons ainsi le mouvement de chaque région $\mathcal{R}_i$, $i = 1 \cdots p$. Nous souhaitons maintenant affecter à chaque pixel s de l'image une étiquette $e_s \in \{1, \cdots, p\}$ afin d'obtenir une mise à jour de la carte de segmentation au sens du mouvement $E = \{e_s, s \in S\}$.

4.2.1 Formulation markovienne du problème de segmentation

Une formulation markovienne du problème à partir d'*a priori* spatiaux du champ d'étiquettes, nous amène à rechercher $\widehat{e}$ solution du problème d'optimisation

$$\widehat{e} = \min_{e}[U_1(e,o) + U_2(e))] \tag{4.2}$$

où U_1 caractérise l'adéquation entre le champ d'étiquettes e et les observations o, U_2 confère une propriété spatiale à e.

Terme de régularisation spatiale

D'après le théorème de Hammersley-Clifford (Théorème 1.1), l'hypothèse markovienne sur le champ d'étiquettes E permet de définir le terme de régularisation comme une somme de potentiels relativement au système de voisinage. L'*a priori* spatial que nous considérons ici sur le champ d'étiquettes est défini à partir d'un modèle de Potts classique sur un voisinage d'ordre 1,

$$U_2(e) = \sum_{<s,s'>\in\mathcal{C}} V_2(e_s, e_{s'}), \tag{4.3}$$

avec

$$V_2(e_s, e_{s'}) = \lambda(1 - \delta(e_s, e_{s'})), \quad \lambda > 0. \tag{4.4}$$

Ce terme favorise l'homogénéité spatiale du champ d'étiquettes en pénalisant des sites voisins ayant des étiquettes différentes.

Terme d'attache aux données

Le terme d'attache aux données U_1 est décomposé en une somme de potentiels locaux

$$U_1(e) = \sum_{s\in S} V_1(e_s, o_s) \tag{4.5}$$

et traduit dans quelle mesure le modèle de mouvement Θ_i, $i = \{1, \cdots, p\}$ décrit le mouvement d'un site s en fonction d'une marge d'erreur ϵ. Pour cela, nous reprenons l'erreur $Er(e_s)$ définie dans le chapitre précédent.

$$\forall s \in S \quad Er(e_s) = \frac{\sum_{n=1}^{N} ||\langle\overrightarrow{\nabla} I, f_s^n\rangle||.|\langle\overrightarrow{\nabla} I.\overrightarrow{v}_{\Theta_{e_s}} + \frac{\partial I}{\partial t}, f_s^n\rangle|}{\max(G, \sum_{n=1}^{N} ||\langle\overrightarrow{\nabla} I, f_s^n\rangle||^2)}. \tag{4.6}$$

D'après le Corollaire 3.1, si cette mesure est inférieure à l_{0s}, Θ_{e_s} est une bonne approximation de la vitesse réelle de s à ϵ près et si $Er(e_s)$ est supérieure à L_{0s}, l'erreur entre le

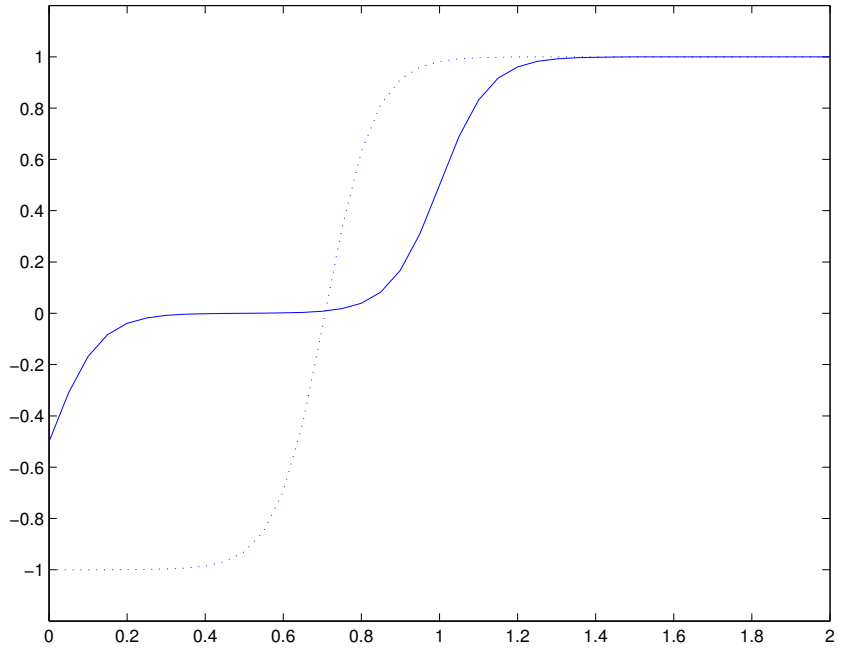


FIGURE 4.1 – Potentiel V_1 avec $\epsilon = 1$ dans le cas d'un site situé sur un coin (courbe en pointillé), et dans le cas d'un site situé sur un contour rectiligne (courbe continue).

mouvement réel et $\overrightarrow{v}_{\Theta_{e_s}}$ est supérieure à ϵ.
A partir de cette erreur, nous définissons V_1 comme suit :

$$\begin{aligned} &\forall s \in S, \\ &V_1(e_s, o_s) = \frac{1}{1+\exp(-16(Er(e_s)-l_{0s}))} + \frac{1}{1+\exp(-16(Er(e_s)-L_{0s}))} - 1. \end{aligned} \tag{4.7}$$

Ce potentiel est faible si $Er(e_s) \leq l_{0s}$, grand si $Er(e_s) \geq L_{0s}$. La figure 4.2.1 montre le potentiel V_1 dans les deux cas extrêmes. Lorsque le site s est situé sur un coin, la zone d'indétermination entre les deux valeurs l_{0s} et L_{0s} est nulle. Quand le site est sur un contour rectiligne, nous avons une zone importante dans laquelle on ne peut rien déduire de l'erreur commise en approchant la vitesse réelle de s par $\overrightarrow{v}_{\Theta_{e_s}}$.

4.2.2 Détection de nouvelles régions

Une fois la nouvelle carte de segmentation obtenue par la minimisation de (4.2), il se peut que certains sites de l'image ne soient pas en adéquation avec le modèle de mouvement qui leur a été affecté. Ceci peut, par exemple, se produire lorsque le mouvement d'un objet devient plus complexe et qu'un seul modèle affine ne suffise plus pour décrire celui-ci ou lorsqu'un nouvel objet apparaît dans la scène. C'est pourquoi une phase de détection de nouvelles régions est nécessaire. Cette étape est effectuée de façon similaire

au chapitre précédent (formule 3.33) mais à partir d'une modélisation markovienne mono-échelle d'un champ d'étiquettes e à valeurs dans $\{0, 1\}$ ($0 =$ conforme au mouvement, $1 =$ non conforme)

$$\widehat{e} = \min_{e}[\sum_{s \in S} q_s(e_s, o_s) + \sum_{s \in S} h_s(e_s, \widetilde{e}_s) + \sum_{<s,s'> \in \mathcal{C}} v_{ss'}(e_s, e_{s'})]. \tag{4.8}$$

Une fois cette optimisation réalisée, les zones connexes étiquetées non conformes, de taille suffisamment importante, forment de nouvelles régions de l'image. Le nombre d'étiquettes $p(t)$ est alors modifié en fonction et une étape d'estimation puis de segmentation à partir de ce nouvel étiquetage est nécessaire. Cette procédure est ensuite exécutée de façon successive (estimation/segmentation/détection) jusqu'à stabilisation du nombre d'étiquettes dans l'image comme le proposent Odobez et Bouthemy [OB98].

Par conséquent, à partir d'une résolution d'un système linéaire (4.1) et de deux problèmes d'optimisation ((4.2) et (4.8)) résolus par un algorithme déterministe I.C.M. [Bes74], nous obtenons une méthode simple et rapide de segmentation au sens du mouvement.

Cependant, cette solution engendre trois problèmes importants :

- Les fonctionnelles définies par (4.2) et (4.8) étant non convexes, leur minimisation par un algorithme I.C.M. dépend fortement du champ d'étiquettes initial. C'est pourquoi il est très important d'avoir une bonne initialisation du champ d'étiquettes.
- Les étapes successives d'estimation, de segmentation puis de détection jusqu'à ce que l'ensemble des points de l'image soit bien affectés d'un mouvement affine en adéquation avec leur mouvement réel peut s'avérer très lourdes en temps de calcul si la scène contient de nombreux mouvements complexes.
- De plus, nous sommes une nouvelle fois en présence d'aliasage temporel. En effet, nous avons vu (formule (2.40)) que la précision de l'approximation de la dérivée temporelle de l'image par une différence finie dépendait de la vitesse du point et de la taille du support des fonctions f^n. C'est-à-dire que l'estimation numérique de $Er(e_s)$ (formule (4.6)) n'est valable que si les vitesses vérifient

  $$\forall i = \{1, \cdots, p\} \quad ||\overrightarrow{v}_{\Theta_i}|| < C. \text{ taille du support de } f^n. \tag{4.9}$$

  Par conséquent, si le support des fonctions est petit, seules des images composées d'objets ayant de faibles mouvements peuvent être segmentées. Pour étudier des scènes présentant de grands mouvements, l'utilisation de fonctions à grand support est nécessaire. Mais dans ce cas, nous réduisons le nombre d'équations dans (4.1) et nous sommes donc plus sensibles au bruit dans les images.

Pour contourner ces problèmes, nous proposons une approche multirésolution. Le problème de la sensibilité de l'I.C.M. par rapport à l'initialisation va être résolu par projection des champs d'étiquettes d'une échelle à l'autre, l'alternance successive des étapes de détection et de segmentation sera évitée en considérant autant d'étiquettes que de pixels à l'échelle la plus grossière et enfin, l'aliasage temporel sera contourné par compensation de mouvement entre chaque échelle.

4.3 Segmentation du mouvement par une approche multirésolution

Dans ce paragraphe, nous allons construire une méthode de segmentation au sens du mouvement par une approche multi-résolution afin de résoudre les différents problèmes rencontrés dans le cas mono-résolution. Cette méthode va nous permettre d'estimer conjointement le mouvement dans l'image et une segmentation au sens du mouvement par projection successive des résultats d'une échelle grossière J à l'échelle la plus fine (Fig 4.2).

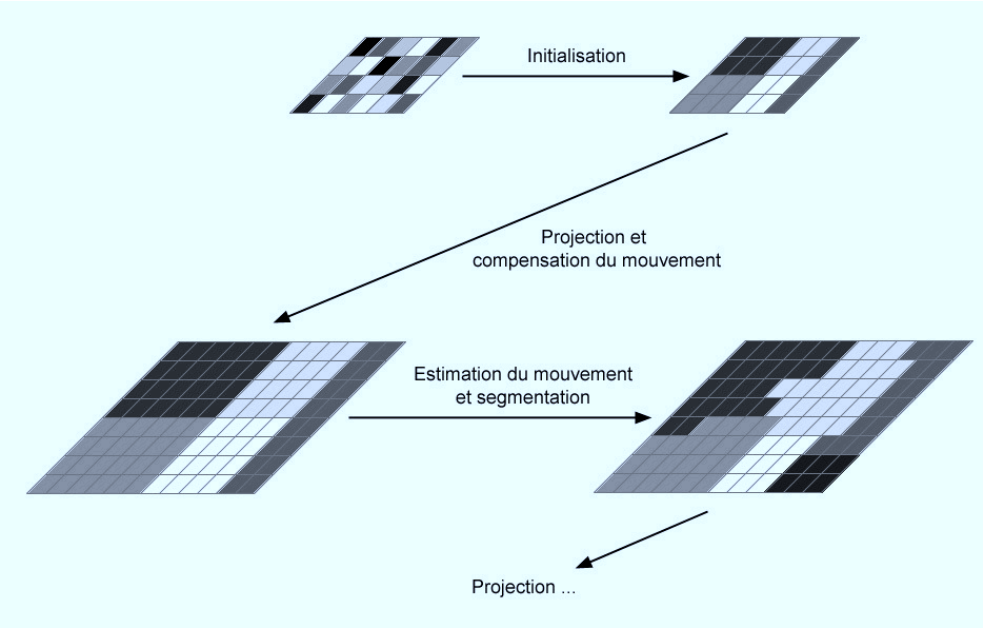


FIGURE 4.2 – Principe de la méthode de segmentation au sens du mouvement par multirésolution.

4.3.1 Estimation du mouvement des régions par compensation

Supposons qu'à une échelle grossière $j+1$, nous ayons obtenu une partition $\mathcal{P}^{j+1}$ de l'image en p_{j+1} régions $\mathcal{R}_i^{j+1}, i = 1 \cdots p_{j+1}$ de mouvement respectif Θ_i^{j+1}

$$\mathcal{P}^{j+1} : \begin{array}{lcl} S^0 & \rightarrow & \{1, \cdots, p_{j+1}\} \\ (x,y) & \mapsto & i, \quad (x,y) \in \mathcal{R}_i^{j+1}. \end{array}$$

Nous décomposons alors la vitesse réelle de chaque point (x,y) de l'image de la façon suivante

$$\forall (x,y), \quad \overrightarrow{v}(x,y) = \overrightarrow{v}_{\Theta^{j+1}_{\mathcal{P}^{j+1}(x,y)}}(x,y) + \overrightarrow{v}_{\epsilon_{\mathcal{P}^{j+1}(x,y)}}(x,y), \tag{4.10}$$

et introduisons l'image compensée $\widetilde{I}_2$,

$$\widetilde{I}_2((x,y)) = I_2((x,y) + \overrightarrow{v}_{\Theta^{j+1}_{\mathcal{P}^{j+1}(x,y)}}).$$

Le déplacement de chaque point (x,y) entre les images I_1 et $\widetilde{I}_2$ est exactement de $\overrightarrow{v}_{\epsilon_{\mathcal{P}^{j+1}(x,y)}}(x,y)$. Comme $\overrightarrow{v}_{\Theta^{j+1}_{\mathcal{P}^{j+1}(x,y)}}(x,y)$ est une approximation de $\overrightarrow{v}(x,y)$, on peut donc supposer le résiduel $\overrightarrow{v}_{\epsilon_{\mathcal{P}^{j+1}(x,y)}}(x,y)$ suffisamment petit pour pouvoir être calculé à l'échelle j à partir de la résolution du système suivant

$$\begin{array}{c} \forall (x,y) = 2^j k = 2^j (k_1, k_2) \in \mathcal{R}_i^{j+1}, \\ \langle \overrightarrow{\nabla} I . \overrightarrow{v}_{\epsilon_{\mathcal{P}^{j+1}(x,y)}} + \frac{\partial \widetilde{I}}{\partial t}, \psi_{jk}^n \rangle = 0 \qquad \forall n = 1 \cdots N; \end{array} \tag{4.11}$$

soit en posant [1]

$$\overrightarrow{v}_{\epsilon_i} = B.\Theta_i^j.$$

Nous en déduisons donc que $\forall i = \{1, \cdots, p_{j+1}\}$, Θ_i^j est solution du système (formule (2.59) :

$$\begin{array}{c} \forall (x,y) = 2^j (k_1, k_2) \in \mathcal{N}_i^{j+1}, \\ M_{jk} \Theta_i^j = \widetilde{P}_{jk} \end{array} \tag{4.12}$$

résolu par moindres carrés [2].

4.3.2 Segmentation par une formulation markovienne multi-modèles

L'étape précédente fournit une information supplémentaire sur le mouvement de chaque région. Chaque région R_i^{j+1} est animée d'un mouvement :

$$\begin{array}{c} \forall i = \{1, \cdots, p^{j+1}\}, \quad \forall (x,y) = 2^j (k_1, k_2) \in R_i^{j+1}, \\ \overrightarrow{v}_i^j (x,y) = B(x,y). \sum_{n=j}^{J} \mathcal{P}^n (\Theta_i^n), \end{array}$$

1. Le mouvement réel de chaque région étant supposé affine, la relation (4.10) montre que $\overrightarrow{v}_{\epsilon_i}$ l'est aussi.

2. Grâce à la multi-résolution, l'utilisation d'un M-estimateur n'est pas nécessaire car on peut supposer que la partition $\mathcal{P}^{j+1}$ réalise déjà un découpage correct de la scène au sens du mouvement.

où $\mathcal{P}^n$ est l'opérateur de projection de l'échelle n à l'échelle 0,

$$\mathcal{P}_n(\Theta) = (a, b, 2^n c, d, e, 2^n f)^T, \quad \Theta = (a, b, c, d, e, f)^T.$$

Il s'agit maintenant de vérifier que chaque point de S^j est bien étiqueté et d'affiner la segmentation au sens du mouvement obtenue à l'échelle $j+1$. Pour ce faire, nous utilisons une modélisation markovienne multi-échelle. Les segmentations de l'image à chaque résolution sont obtenues de façon indépendante et avec une énergie markovienne différente.

A l'échelle j fixée, nous cherchons le champ d'étiquettes $E^j = \{e_s^j, s \in S^j\}$ où $e_s^j \in \{1, \cdots, p^{j+1}\}$ qui partage "au mieux" l'image à la résolution j en fonction des observations $O^j = \{o_s^j, s \in S^j\}$. En posant $\mathcal{C}^j$ l'ensemble des cliques associées au système de voisinage d'ordre 1, $\mathcal{V}^j = \{\mathcal{V}_s^j \quad | s \in S^j\}$ sur le niveau de résolution S^j, l'énergie définie pour déterminer le champ d'étiquettes est similaire au cas mono-échelle (4.2),

$$\widehat{e}^j = \min_{e^j} U^j(e^j, o^j) = \min_{e^j} [U_1^j(e^j, o^j) + U_2^j(e^j)], \tag{4.13}$$

avec,

$$U_2^j(e^j) = \sum_{<s,s'> \in \mathcal{C}^j} \lambda^j (1 - \delta(e_s^j, e_{s'}^j)), \quad \lambda^j > 0.$$

Le terme d'attache aux données U_1^j est défini de la même façon que dans le cas mono-résolution (formules (4.5) et (4.7)) avec la mesure d'erreur suivante

$$\forall s = 2^j(k_1, k_2) \in S^j \quad Er(e_s) = \frac{\sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||.|\langle \overrightarrow{\nabla} I. \overrightarrow{v}_{\Theta_{e_s}^j} + \frac{\partial}{\partial t} I_{\overrightarrow{v}_{e_s}^{j+1}}, \psi_{jk}^n \rangle|}{\max(G, \sum_{n=1}^{N} ||\langle \overrightarrow{\nabla} I, \psi_{jk}^n \rangle||^2)}. \tag{4.14}$$

En théorie, à cause de l'aliasage temporel, $Er(e_s)$ n'est pas correctement mesurable si l'erreur entre la vitesse $\overrightarrow{v}_{\Theta_{e_s}^j} + \overrightarrow{v}_{e_s}^{j+1}$ et la vitesse réelle au site s est trop grande. Mais en pratique, comme l'initialisation du champ d'étiquettes dans l'algorithme I.C.M. est proche de la solution cherchée, les erreurs de mesure sur Er n'ont peu de conséquences sur le résultat.

A l'échelle j, il se peut que certaines zones de l'image qui étaient trop petites voire inexistantes à l'échelle $j+1$ apparaissent pour former de nouvelles régions dont le mouvement ne peut pas être décrit par l'un des p^{j+1} modèles. Il faut donc ajouter à l'ensemble des étiquettes $\{1, \cdots, p^{j+1}\}$ une étiquette d'attente ρ d'énergie $V_1^j(\rho) = \mathcal{V}_\rho$ permettant la création de nouvelle régions à l'échelle j comme le proposent Bouthemy et Francois [BF93]. Remarquons qu'une étape plus complexe de détection de mouvement comme nous le proposions dans le cas mono-résolution n'est pas utile. En effet, dans le cas mono-résolution, nous ne connaissons pas le nombre de régions nécessaires pour partitionner l'image à

l'instant t au sens du mouvement mais seulement celui à l'instant $t-1$. Une détection plus complexe (formule 4.8) semble nécessaire pour détecter l'apparition éventuelle d'un nouvel objet dans la scène entre deux images consécutives. Par contre, dans le cas multi-résolution, le temps étant fixé, le nombre d'objets dans la scène n'évolue pas entre la segmentation au sens du mouvement à l'échelle $j+1$ et celle à l'échelle j.

La minimisation de (4.13) est effectuée par un algorithme I.C.M. initialisé par la segmentation obtenue à l'échelle $j+1$. Les zones connexes étiquetées ρ forment de nouvelle régions dans la partition et le nombre d'étiquette p_j est mis à jour en fonction.

Cette méthode multiéchelle nous permet donc de résoudre le problème d'aliasage temporel, d'être moins sensible à l'initialisation pour l'algorithme I.C.M. et d'accélérer la méthode. Il nous reste à partitionner la séquence à l'échelle la plus grossière pour initialiser la méthode.

4.4 Initialisation à l'échelle grossière

A l'échelle la plus grossière J, comme nous ne connaissons pas a priori le nombre de régions dans l'image, trois choix s'offrent à nous :

- Utiliser la segmentation au sens du mouvement à l'instant $t-1$ pour initialiser l'algorithme à l'instant t.
- Initialiser l'algorithme en utilisant une seule région dans l'image et utiliser une méthode de détection de mouvement du type (4.8) de façon récursive pour obtenir le nombre de régions.
- Initialiser la méthode avec autant de régions que de pixels.

La première méthode ne résout pas le problème lors de la segmentation de la scène entre les deux premières images de la séquence. La deuxième méthode revient en fait à un problème de détection de mouvement à l'échelle la plus grossière. Cette méthode n'est valable que lorsque la scène est régie par un mouvement dominant. Dans notre cas, comme la méthode d'estimation du mouvement permet de calculer le déplacement affine d'une région même réduite à un pixel, nous choisissons la dernière solution.

Ainsi, à l'échelle J, nous avons autant de régions R_i^J que de pixels. La vitesse de chaque pixel et donc de chaque région est calculée à partir de la résolution du système d'inconnue Θ_i^J

$$\begin{array}{l} \forall s = 2^J(k_1,k_2) \\ \left\{ \begin{array}{ll} \langle \overrightarrow{\nabla} I . \overrightarrow{v}_{R_s^J} + \frac{\partial I}{\partial t}, \psi_{Jk}^n \rangle = 0 & \forall n = 1 \cdots N, \\ \overrightarrow{v}_{R_s^J} = B\Theta_s^J. & \end{array} \right. \end{array} \tag{4.15}$$

Une fois le mouvement affine de chaque pixel calculé, une segmentation est réalisée à partir de la minimisation de U^J (4.13) donnant p régions. Même si à l'échelle grossière le nombre

de sites est faible et donc l'initialisation de l'algorithme I.C.M. pour la minimisation de (4.13) est moins importante, les résultats montrent que les images restent sur-segmentées comme le montre l'exemple de la figure 4.3. Pour résoudre ce problème, une segmentation au sens du mouvement est effectuée non plus site par site mais régions par régions ([GB00], [HRB96], [XG94]).

FIGURE 4.3 – Cas de deux régions ayant le même mouvement. Que le site s ait l'étiquette de la région R_1 ou R_2, le terme d'attache aux données $V_1^J(e_s^J, o_s^J) = Cst$ est identique.(a) configuration initiale du champ d'étiquettes, l'énergie locale au site s est $Cst + \lambda^J$. (b) modification du champ d'étiquettes au site s, son énergie locale est alors de $Cst + 3\lambda^J$. Un algorithme de descente I.C.M ne pourra donc pas réunir ces deux régions.

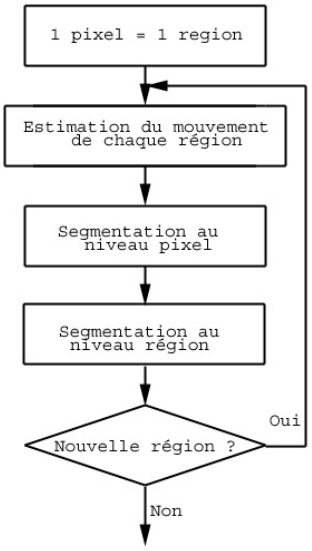


FIGURE 4.4 – Initialisation.

Considérons donc le graphe déduit de la segmentation $P = \{R_i, i = 1, \cdots, p\}$ contenant les p régions. Notons $S' = \{s'_1, \cdots, s'_p\}$ l'ensemble des sites de ce graphe, $\mathcal{V}'$ le système de voisinage correspondant aux régions adjacentes du graphe et $\mathcal{C}'$ l'ensemble des cliques associées. Le champ d'étiquettes $e' = \{e'_1, \cdots, e'_p\}$, $e'_1 \in \{1, \cdots, p\}$, définissant la nouvelle segmentation, est obtenu par la minimisation de la fonctionnelle suivante

$$\begin{aligned} U'(e') &= U'_1(e', o^J) + U'_2(e') \\ &= \sum_{n=1}^{p} V'(e'_{s'_n}, o^J_{s'_n}) + \lambda^J \sum_{<s'_1, s'_2> \in \mathcal{C}'} (1 - \delta(e'_{s'_1}, e'_{s'_2})), \end{aligned} \tag{4.16}$$

avec,

$$V'(e'_{s'_n}, o^J_{s'_n}) = \frac{1}{\sharp(R_{s'_n})} \sum_{i \in R_{s'_n}} V^J_1(e'_{s'_n}, o^J_i).$$

Ensuite, une nouvelle estimation du mouvement de chaque région est calculée, suivi d'une étape de segmentation pixel et segmentation région, et ceci de façon récursive, jusqu'à ce qu'il n'y ait plus de réunion de régions (Fig 4.4). Nous obtenons ainsi une segmentation à l'échelle grossière J qui permet d'initialiser la méthode à l'échelle $J - 1$.

4.5 Résultats

Nous avons testé notre méthode de segmentation sur quatre paires d'images extraites respectivement des séquences Taxi, Yosemite, "Mobile and Calendar" et "Flower Garden". Dans l'ensemble de ces tests, nous avons fixé le paramètre G à 50 et $\mathcal{V}_\rho$ à $0,8$.

Remarquons une nouvelle fois que l'utilisation d'ondelettes analytiques nous contraint à ne traiter que les résolutions $j \geq 2$ de l'image. Par conséquent, le résultat le plus fin de la segmentation au sens du mouvement que l'on puisse obtenir pour une image de taille $2^n \times 2^m$ est de taille $2^{n-2} \times 2^{m-2}$.

4.5.1 Séquence Taxi

La séquence Taxi de Hambourg est une séquence composée de quatre objets en mouvement filmés par une caméra fixe. On y voit un piéton en haut à gauche qui marche à une vitesse d'environ $0,3$ pixels/images, un taxi tournant sur sa droite, une camionnette venant de droite partiellement cachée par un arbre et une voiture se déplaçant de la gauche vers la droite de l'image. Cette séquence est relativement bruitée et présente une grande variation temporelle de la luminosité. Nous avons considéré les images aux instants $t = 19$ et $t = 20$ pour tester notre méthode (Fig 4.5).

Les résultats sont montrés à la figure 4.6 où nous avons projeté les segmentations au sens du mouvement obtenues à chaque échelle sur la résolution la plus fine. Comme les

(a) (b)

FIGURE 4.5 – Images originales tirées de la séquence Taxi. (a) $t = 19$, (b) $t = 20$.

objets se déplacent à une vitesse très réduite entre les deux images, nous avons considéré une valeur pour ϵ relativement faible ($\epsilon = 0.4$) afin d'éviter tout regroupement entre régions fixes et mobiles. Le paramètre de dépendance spatial λ^j du champ d'étiquettes est quant à lui fixé à 0.1 sur l'ensemble des résolutions de l'image. Après un premier regroupement au niveau pixel à l'échelle grossière, le résultat est très sur-segmenté (Fig 4.6(a)). La phase de regroupements successifs au niveau pixel et au niveau région permet d'obtenir une meilleure segmentation et d'initialiser la segmentation à l'échelle $j = 3$. Le résultat le plus fin obtenu est présenté figure 4.6(d). Notons que le piéton n'est pas correctement segmenté à cause de sa faible vitesse.

Comme nous ne disposons pas d'informations aux échelles $j = 0$ et $j = 1$, la segmentation n'est pas très précise. C'est pourquoi, dans les deux prochaines séquences, nous avons choisi d'interpoler d'un facteur 4 les images pour obtenir un résultat à la taille $2^n \times 2^m$ (taille de l'image originale).

4.5.2 Séquence Yosemite

La séquence synthétique Yosemite est une séquence souvent utilisée pour valider les méthodes d'estimation du flot optique. Elle représente le survol par un avion d'une chaîne de montagne. Les nuages dans le ciel ne sont pas fixes mais se déplacent de la gauche vers la droite et le mouvement de la caméra sur les montagnes ne peut pas être décrit par un seul mouvement affine à cause de la variation de profondeur.

Les résultats sont présentés figure 4.7, sur 3 niveaux avec $\epsilon = 1$ et $\lambda^j = 0.1$. L'image est segmentée en cinq régions, le ciel, la montagne au premier plan, celle au second plan, et la plaine est décomposée en deux parties. L'estimation du mouvement à partir de cette partition multirésolution au sens du mouvement est présentée figure 4.7(d).

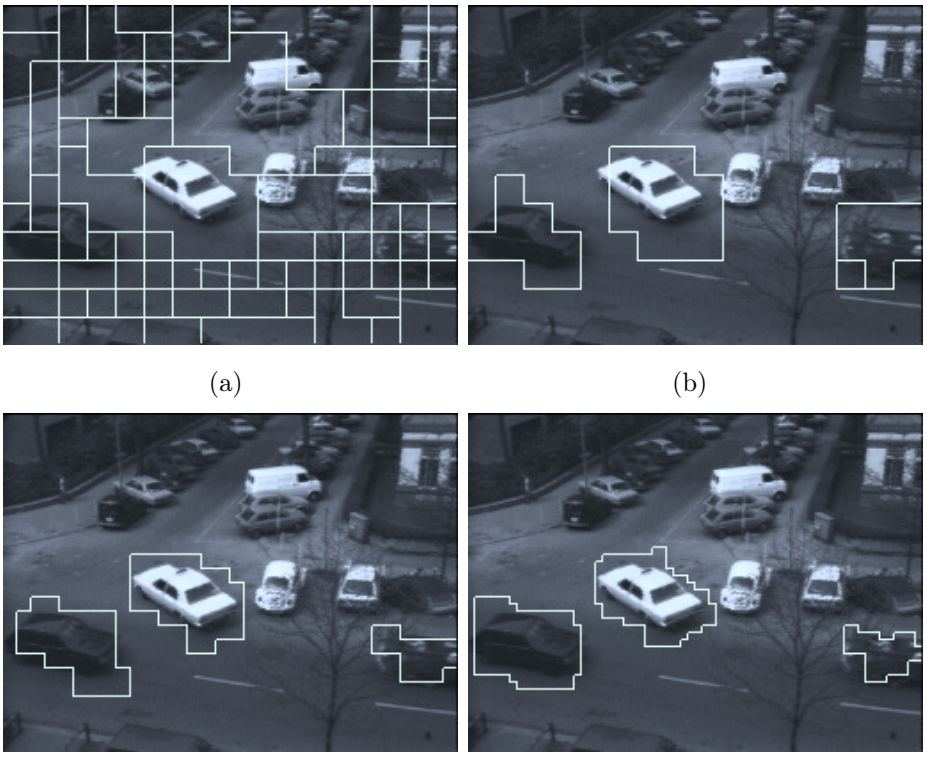


FIGURE 4.6 – Résultat de la segmentation au sens du mouvement sur la séquence Taxi. (a) Premier regroupement entre pixels à l'échelle grossière $J = 4$, (b) résultat à l'échelle $J = 4$ après les phases successives de regroupement aux niveaux pixel et région, (c) résultat à l'échelle $j = 3$, (d) résultat à l'échelle $j = 2$.

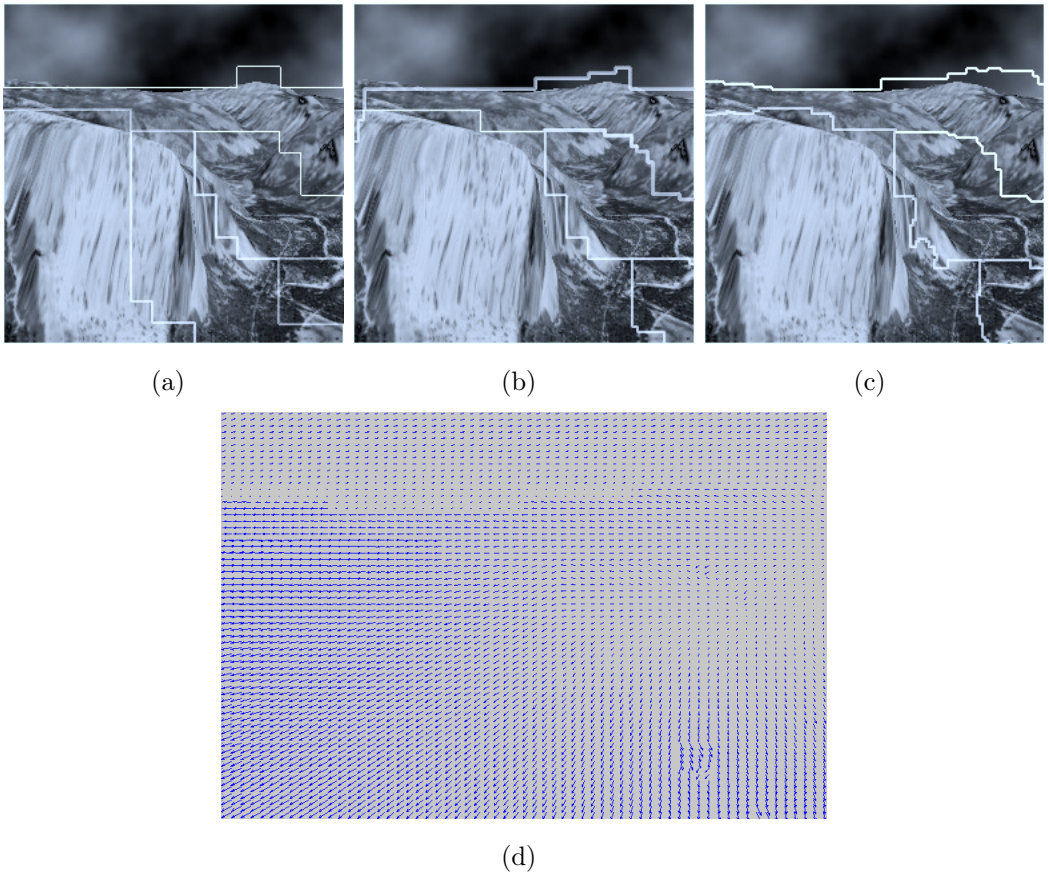


FIGURE 4.7 – Résultat sur la séquence Yosemite (a) Segmentation au sens du mouvement à l'échelle $J = 4$, (b) segmentation à $j = 3$ (c) segmentation à $j = 2$,(d) estimation du mouvement.

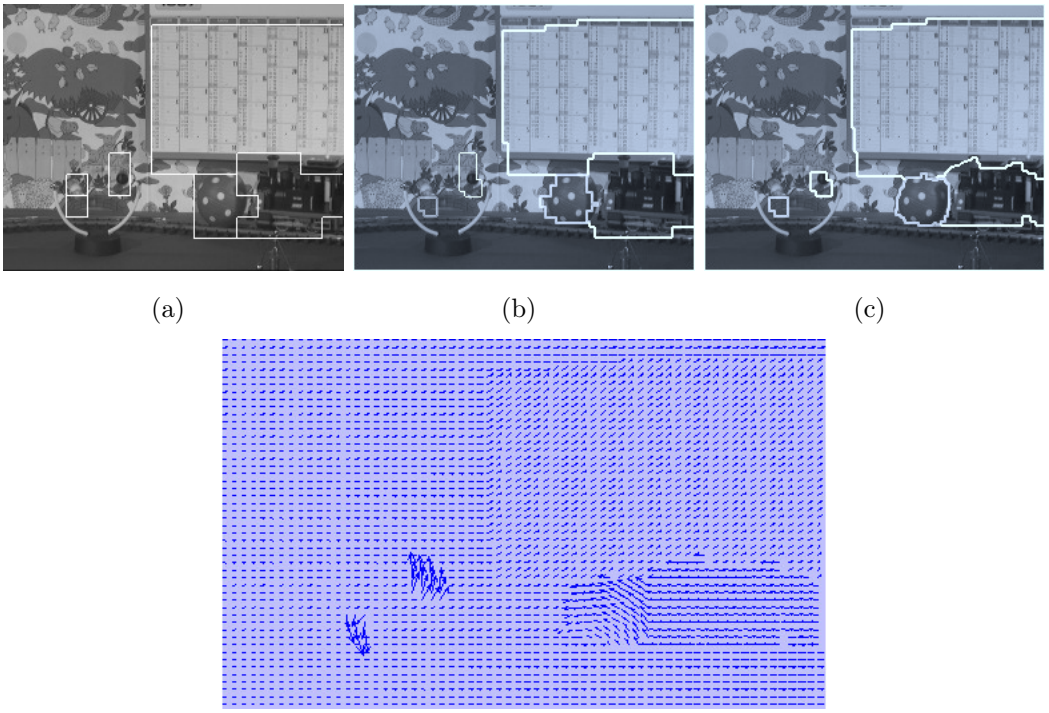
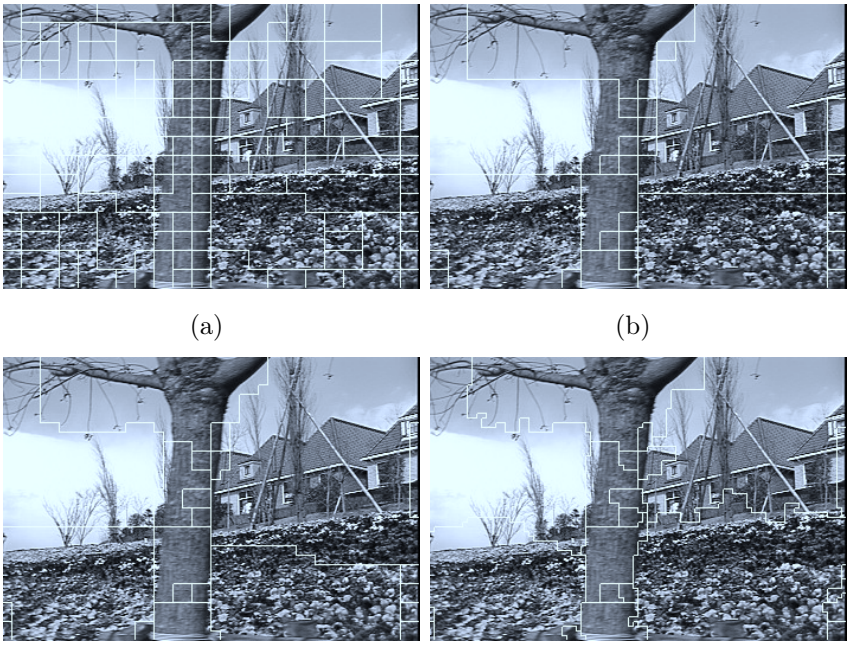


FIGURE 4.8 – Résultat sur la séquence "Mobile and Calendar" (a) Segmentation au sens du mouvement à l'échelle $J = 4$, (b) segmentation à $j = 3$ (c) segmentation à $j = 2$,(d) estimation du mouvement.

4.5.3 Séquence "Mobile and Calendar"

Cette séquence a déjà été traitée dans le chapitre 3, pour estimer son mouvement dominant. La caméra a un mouvement de translation de la gauche vers la droite pendant qu'un train électrique arrive de la droite en poussant un ballon alors que le mobile est en rotation et le calendrier est assujetti à un mouvement vertical. Nous avons testé notre méthode sur 4 niveaux de résolutions avec $\epsilon = 1$ et $\lambda^j = 0.1$. Les résultats sont présentés figure 4.8. Dès l'échelle grossière ($J = 4$), après le regroupement successif en pixels et en régions l'image est partitionnée en 7 régions, deux régions du mobile sont ensuite fusionnées au fil des résolutions. Le résultat de la segmentation au sens du mouvement le plus fin est présenté figure 4.8(c) et le flot obtenu figure 4.8(d).

Dans ces trois exemples, les résultats sont relativement satisfaisants avec un temps de calcul relativement peu élevé (en moyenne moins de une minute de traitement pour une image de taille 256×256 sous Matlab sur un Pentium $2,4$ Ghz). Par contre, elle a cependant ses limites comme le montre le résultat suivant.



(a) (b)

(c) (d)

FIGURE 4.9 – Résultat sur la séquence "Flower Garden" (a) Premier regroupement au niveau pixel à l'échelle grossière, (b) Segmentation au sens du mouvement à l'échelle $J = 4$, (c) segmentation à $j = 3$ (d) segmentation à $j = 2$.

4.5.4 Séquence "Flower Garden"

Cette séquence a été utilisée dans le chapitre 3 pour y estimer le mouvement dominant où seule la caméra est mobile. A cause des différences de profondeur dans la scène, un seul modèle affine ne suffit pas pour modéliser le mouvement de l'ensemble de l'image.

Les résultats de notre méthode de segmentation au sens du mouvement sont présentés figure 4.9 avec 4 niveaux de résolution, $\epsilon = 1$ et $\lambda^j = 0.1$. On remarque que la phase d'initialisation du nombre de régions distinctes au sens du mouvement à l'échelle grossière aboutit à un partitionnement de l'image sur-segmenté (Fig 4.9(b)). Malgré les étapes de raffinement aux échelles $j = 3$ et $j = 2$, le résultat reste très sur-segmenté même si le nombre de régions a diminué (Fig 4.9(d)).

4.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à la segmentation au sens du mouvement entre deux images consécutives d'une séquence. La méthode introduite est fondée sur une technique multirésolution qui procède de façon successive entre une estimation du

mouvement et une segmentation sur chaque résolution de l'image. Cette technique nous a permis de contourner une fois de plus le problème de l'aliasage temporel engendré par l'estimation du mouvement à partir de l'équation de contrainte du mouvement apparent (E.C.M.A.).

Notre méthode présente plusieurs avantages. Elle ne requiert pas de segmentation spatiale de l'image, ni une estimation d'un champ dense de déplacements préalable et utilise uniquement l'information de mouvement obtenue par une modélisation paramétrique. De plus, elle ne nécessite pas une hypothèse de présence de mouvement dominant entre les deux images. Elle évite ainsi les schémas traditionnels itératifs entre des phases d'estimation de mouvement, de détection de mouvement non conforme et de segmentation trop gourmands en temps de calcul si il y a dans la scène de nombreux objets aux mouvements distincts. Le caractère multirésolution de notre méthode permet d'utiliser un algorithme d'optimisation déterministe à chaque échelle pour minimiser l'énergie markovienne. En effet, dans notre cas, comme l'initialisation de l'algorithme d'optimisation provient du résultat de la segmentation à l'échelle supérieure, elle est proche de la solution à l'échelle courante et justifie donc l'utilisation d'un algorithme I.C.M.. Ainsi, l'estimation du mouvement par analyse d'ondelettes et la segmentation au sens du mouvement par un algorithme déterministe permettent d'obtenir une méthode peu coûteuse en temps de calcul.

Par contre, comme dans le cas de la détection du mouvement dans le chapitre précédent, l'utilisation d'ondelettes analytiques ne nous permet pas d'obtenir d'informations aux deux échelles les plus fines de l'image. Par conséquent, il est nécessaire d'interpoler l'image d'un facteur 4 pour obtenir une segmentation à la taille de l'image originale comme nous l'avons fait pour les tests sur la séquence Yosemite et "Mobile and Calendar". De plus, le résultat de la segmentation dépend fortement de l'étape d'initialisation à l'échelle grossière. Si celle-ci est sur-segmentée, le résultat final risque de l'être aussi. Pour éviter cela, il serait intéressant de généraliser notre méthode à la segmentation de séquence d'images en introduisant un terme de dépendance temporelle dans la modélisation markovienne.

Troisième partie

Application à la détection d'obstacles dans un milieu routier

Chapitre 5

Les méthodes de détection d'obstacles dans un milieu routier

5.1 Introduction

Le traitement d'images est devenu un outil important dans le domaine de la sécurité routière. Au départ, les caméras étaient utilisées pour la surveillance du trafic routier [HB96]. Avec l'amélioration des techniques d'analyse de séquences d'images, la vision artificielle est devenue un moyen d'aide à la conduite [BB98] très attractif, car elle est beaucoup moins coûteuse, plus facile à installer et nécessite moins de maintenance que l'utilisation d'autres capteurs tels que les radars ou les lasers.

Par contre, étant donné la grande diversité des environnements, l'analyse de scène routière reste un problème difficile. En effet, la perception de la route est souvent perturbée par de grandes variations de contraste dues par exemple à l'ombre d'un pont, à la présence d'arbres ou de parties mouillées de la chaussée. La grande complexité des conditions d'éclairage de la route rend l'analyse des images plus délicate. Les méthodes se doivent d'être extrêmement robustes aux changements d'illumination, aux changements de conditions climatiques, à la présence d'ombres... De plus, les solutions proposées doivent être rapides pour une implantation en temps réel.

Dans ce chapitre, nous proposons un survol des méthodes de détection d'obstacles dans un milieu routier à partir de caméras montées sur le véhicule (un état de l'art exhaustif peut être obtenu dans [KZK03]). Étant donné que de nombreuses méthodes utilisent une détection préalable de la route afin de restreindre la zone de recherche des obstacles, nous discuterons d'abord des différentes techniques de détection de la chaussée. Puis, nous nous pencherons sur les méthodes de détection d'obstacles.

5.2 Détection de la chaussée

La détection automatique de la chaussée permet d'estimer la position et l'orientation du véhicule mais elle est surtout, dans de nombreux systèmes d'aide à la conduite, la première étape de la détection d'obstacles. En effet, une fois que la route est localisée dans l'image, on peut restreindre la zone d'intérêt de recherche (Region of Interest ROI) des obstacles et ainsi réduire le temps de calcul des algorithmes de détection. On distingue deux grandes approches différentes permettant de détecter la route : les méthodes basées segmentation, les méthodes qui recherchent la délimitation entre la route et le bas-coté.

5.2.1 Les méthodes basées sur la segmentation en régions

Ces méthodes cherchent à classer les pixels de l'image en deux catégories : les pixels appartenant à la route et le reste de l'image. Pour cela, des méthodes de segmentation sur la couleur ont été introduites :

- Turk et al.[TMGM88] proposent un simple seuillage dans l'espace RGB où seules les composantes rouge et bleue sont nécessaires pour partitionner l'image.
- Buluswar et Draper [BD98] classent les pixels de l'image à partir d'une fonction d'apprentissage tenant compte des différentes conditions d'illumination.
- Crisman et Thorpe [CT93] utilisent deux caméras aux propriétés différentes et localisent la route grâce aux 6 composantes de couleur en chaque point ($[(R_1, G_1, B_1), (R_2, G_2, B_2)]$) et d'une fonction de vraisemblance définie avec 12 modèles de route et 12 modèles de "non-route".

On trouve aussi des méthodes d'analyse de texture [ACC00]. En supposant la route d'intensité uniforme, Zhang et Nagel [ZN94] caractérisent les pixels de la route comme étant des points de faible gradient et classent les pixels par une méthode bayésienne. Rasmussen [Ras04] estime dans un premier temps le point de fuite de la route (point de l'image où les bords de la route convergent sous l'effet de la perspective) en calculant l'orientation de la texture en chaque point de l'image par une transformée de Gabor. Puis, il localise les frontières de la route en optimisant une fonction de vraisemblance sur la couleur et la texture des pixels. De son coté, Beucher et Bilodeau [BB94] proposent d'utiliser les techniques de morphologie mathématique. Ils détectent les routes marquées grâce à une détection de contour et une méthode de lignes de partage des eaux sur le gradient de l'image. Cette méthode repose sur la présence de marquage au sol. Cette hypothèse est souvent la clé des méthodes de détection de route.

5.2.2 Les méthodes de délimitation de la route

Sous ce titre, nous englobons toutes les méthodes qui recherchent les limites de la voie de circulation dans l'image. Ces méthodes s'appuient pour la plupart sur la présence de marquage au sol ou sur une différence d'intensité dans l'image entre la chaussée et le bas-coté. De nombreuses méthodes existent.

Les méthodes de détection de marquages au sol

Ces méthodes comportent généralement deux étapes. Dans un premier temps, une détection de contours dans l'image est réalisée par un filtre de Canny-Deriche ou une méthode de morphologie mathématique. Les méthodes diffèrent ensuite sur le moyen de localiser les limites de la chaussée. Les premières méthodes supposaient que les routes étaient planes avec un marquage au sol rectiligne et détectaient celui-ci par une méthode de vote du type transformée de Hough [SYNK92]. Pour détecter les marquages courbés, Bertozzi et Broggi [BB98] proposent d'abord d'appliquer une transformation perspective sur l'image afin d'obtenir une vue aérienne de la route. Ensuite, la détection des marquages se fait grâce à la construction d'un histogramme sur la distance séparant les points horizontaux. La distance d la plus fréquente est choisie comme étant la largeur de la route et tous les points séparés de cette distance forment les limites de la route.
Ces méthodes permettent une détection rapide de la chaussée, mais sont fortement dépendantes de la méthode d'extraction de contours. De plus, en pratique les zones de plus fort gradient dans l'image ne correspondent pas nécessairement aux limites de la route. En effet, les contours des ombres par exemple peuvent affecter la méthode de détection.

Les méthodes de modélisation de la route

Le principe de ces méthodes est de rechercher les paramètres d'un modèle de route préalablement défini qui se rapprochent aux mieux de la frontière de la route. La route est généralement supposée plane et ses frontières sont modélisées par des arcs de cercles. Beauvais et Lakshmanan [BL00] ou Kluge et Lakshmanan [KL95] représentent les frontières par une courbe parabolique. Wang et al. [WST00] préfèrent quant à eux décrire l'effet de perspective sur des lignes courbes et parallèles par des splines d'ordre 3. Yuille et Coughlan [YC00] proposent une approche fondée sur les contours actifs.
Ces méthodes permettent d'obtenir une détection de route robuste aux ombres et occlusions. Certaines peuvent grâce à la modélisation de la route reconstruire la géométrie 3D de la route et ainsi localiser les obstacles sur celle-ci [CAC00]. Cependant, l'estimation des paramètres des modèles requiert souvent des algorithmes complexes pour l'optimisation, ce qui est très coûteux en temps de calcul.

5.3 Détection d'obstacles

Les méthodes de détection d'obstacles sur la route à partir de caméra(s) fixée(s) sur le véhicule diffèrent suivant la définition même de l'obstacle. Lorsque la définition de l'obstacle est réduite à la recherche d'objets spécifiques dans les images, les méthodes sont fondées sur la reconnaissance de caractéristiques dans l'image propres aux obstacles. Dans le cas contraire, c'est-à-dire lorsque le problème de la détection se résume à la recherche de la zone sur la route dans laquelle le véhicule peut circuler en toute sécurité, la détection est alors beaucoup plus complexe et nécessite des techniques d'analyse du mouvement ou de stéréoscopie.

5.3.1 Détection d'obstacles prédéfinis

Dans le cas où nous cherchons un objet précis, la détection d'obstacles se limite à l'analyse d'une seule image dans laquelle nous recherchons des attributs permettant de constater la présence ou non des obstacles. Intéressons-nous aux deux obstacles les plus présents sur les routes : les véhicules et les piétons.

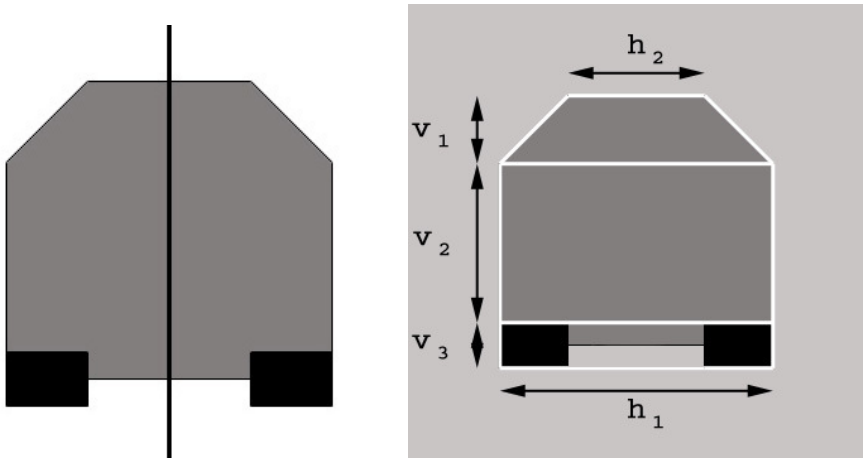
Les véhicules

On distingue trois types de méthodes de détection de véhicules :

• Les méthodes cherchant dans l'image des caractéristiques qui permettent de conclure sur la présence de véhicules. Ces caractéristiques peuvent être des propriétés de symétrie dans l'image [Kue98][BBF$^+$00] (Fig 5.1(a)), la présence d'ombres ou la présence de frontières horizontales et verticales [BHD00]. Ces méthodes ont l'avantage d'être simples et rapides mais peu robustes. En effet, les attributs recherchés par ces méthodes ne permettent pas de conclure de façon certaine à la présence de véhicules. Pour éviter de détecter de fausses alarmes, on peut rechercher plusieurs attributs [BBF$^+$00] et ainsi augmenter la probabilité qu'une détection corresponde bien à un véhicule. Un autre moyen de rendre plus robuste ces méthodes est d'ajouter une phase de vérification.

• Les méthodes qui s'appuient sur une modélisation géométrique des véhicules (Fig 5.1(b)). Ces méthodes consistent à rechercher les véhicules dans l'image à partir d'une description complète d'un modèle géométrique déformable [DJLJ96] [HCAdlE04] décrivant les caractéristiques du véhicule. Les paramètres de ces modèles sont estimés par une mesure de corrélation entre les images et les modèles grâce à des algorithmes génétiques [HCAdlE04] ou par analyse de données [FWSB95]. Ces méthodes sont beaucoup plus robustes que les précédentes mais sont plus lentes. De plus, ces approches sont mises en défaut dès que les véhicules ne correspondent pas au modèle prédéfini.

• Les méthodes d'apprentissage consistent à apprendre à détecter les véhicules à partir d'une base de données composée d'images de véhicules et d'images quelconques qui ont été préalablement classées manuellement. Des caractéristiques globales ou locales sont extraites de ces images par filtrage de Gabor [SBM02], par une méthode dite LOC (Local Orientation Coding) [GDW96] ou par un modèle statistique [SK98]. La classification véhicule/non-véhicule est ensuite générée par réseaux de neurones [GDW96] ou par SVM [SBM02]. Ces méthodes sont souvent utilisées pour confirmer la détection d'un véhicule et ainsi réduire les fausses alarmes des premières méthodes citées ci-dessus.



(a) (b)

FIGURE 5.1 – Caractérisation d'un véhicule (a) axe de symétrie, (b) un exemple de modèle de véhicule.

Les piétons

Les méthodes de détection de piétons à partir d'une caméra montée sur un véhicule s'appuient sur les mêmes principes que les méthodes de détection de véhicules. Néanmoins, comme les personnes portent des vêtements de couleurs très différentes se confondant parfois avec d'autres objets de la scène, changent parfois de direction de façon imprévisible, peuvent se déplacer latéralement comme longitudalement par rapport à la caméra, se trouvent dans des environnements très complexes, ces situations très diverses font du problème de détection de piétons un problème plus difficile que celui de la détection de véhicules.

De façon générale, les méthodes de détection de piétons se décomposent en deux temps : une détection de zones où il y a de très forte chance de trouver des piétons et une étape de vérifications. La première phase n'est nécessaire que pour limiter la zone de recherche de

l'étape de vérification qui est très coûteuse en temps de calcul. Elle recherche dans l'image les régions candidates par analyse de forme [Gav00], de textures [CEK+00], de symétries [BBFS00]. Pour la phase de reconnaissance, il existe essentiellement deux approches : les méthodes d'analyse de geste humain ou les méthodes de modélisation 2D et 3D de piétons. Les premières sont fondées sur la détection de mouvement périodique [CD00], les secondes soit sur une simple modélisation d'une partie du corps [BBFS00] ou de façon plus complexe par apprentissage de type réseaux de neurones ou SVM.

5.3.2 Détection de tous types d'obstacles

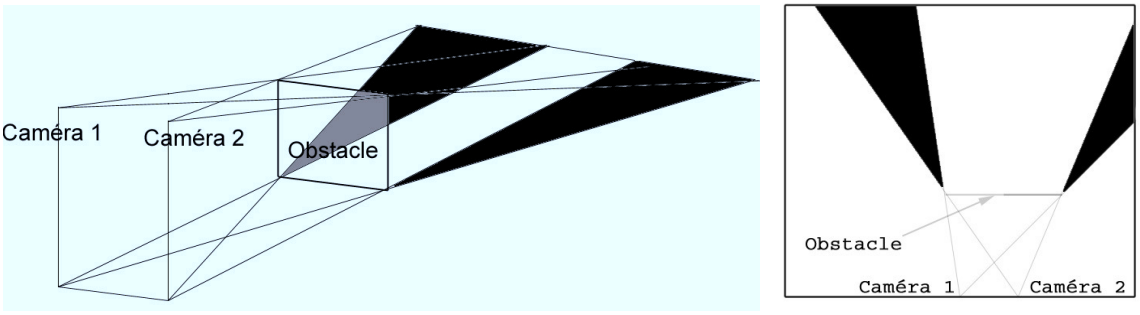
Dans le cas d'obstacles prédéfinis, la détection se réduit à l'analyse d'une seule image dans laquelle nous recherchons des caractéristiques propres aux objets. Par contre, dans le cas où l'obstacle recherché n'est pas connu, l'analyse d'une seule image ne peut suffire. Dans ce cas, on a recours aux méthodes de stéréoscopie ou d'analyse du mouvement.

Les méthodes de stéréoscopie

De nombreux auteurs proposent, pour détecter les obstacles sur la route, d'utiliser deux caméras. Ces systèmes de vision ont en effet un avantage indéniable puisqu'ils permettent à partir d'une mesure de disparité (différence de position des points mise en correspondance dans chacune des deux images) de retrouver la position 3D des points. De plus, les problèmes de mises en correspondance sont grandement simplifiés dès que la route est supposée plane grâce aux équations de Helmholtz [LWKM95]. Dans ce cas, la détection d'obstacles se fait par un simple seuillage d'une mesure de disparité, tous les points de la route plane correspondant à une disparité nulle alors que les points de disparité non nulle indiquant la présence d'objets sur la route [LWKM95]. Pour détecter les obstacles sur route non plane, Labayrade et al. [LAT02] suivent la même idée mais supposent la route plane par morceaux. Ils estiment ces différents plans par une mesure de disparité et considèrent tous les points qui ne suivent pas cette modélisation comme des obstacles.
Une approche différente consiste à déterminer les zones d'occlusion entre les deux images [BB98] (Fig 5.2). Lorsqu'un obstacle sur une route plane est devant le système de stéréoscopie, deux zones triangulaires diffèrent dans les deux images. La détection de ces zones permet d'en conclure la présence d'un obstacle.

Les méthodes d'analyse du mouvement

Les méthodes d'analyse du mouvement consistent à détecter les obstacles à partir d'une seule caméra en calculant les déplacements des points entre deux images consécutives. Les



(a) (b)

FIGURE 5.2 – Occlusion lors de la présence d'un obstacle sur une route plane (a) vue perspective, (b) vue aérienne.

obstacles sont les points de l'image dont le mouvement apparent n'est pas conforme à celui induit par le mouvement de la caméra fixée sur le véhicule.

D'un point de vue industriel, ces méthodes sont préférées aux méthodes stéréoscopiques puisqu'elles nécessitent une seule caméra et ont donc un coût moindre. Par contre d'un point de vue technique, ces méthodes doivent surmonter trois difficultés qui les rendent moins fiables et moins rapides :

- Les méthodes d'estimation du flot optique utilisées doivent être capables d'estimer de grands mouvements. En effet, la vitesse importante des véhicules et donc des caméras engendrent de grands déplacements entre deux images consécutives.
- Le manque de textures de la route rend l'estimation du flot optique sur celle-ci peu fiable.
- Les chocs et les vibrations de la caméra ne permettent pas de calculer le véritable mouvement apparent des points de l'image.

Si la caméra est en translation dans un environnement fixe, le champ des vitesses converge vers un point appelé foyer d'expansion (F.O.E.). Tout point de l'image dont le vecteur de vitesse ne converge pas vers ce point est alors un obstacle [WGK04]. De façon générale, les méthodes d'analyse du mouvement se décomposent en deux étapes. Le flot est calculé par des méthodes vues dans le chapitre 3, puis les obstacles mobiles ou immobiles sont détectés par étude de la différence entre le flot calculé et le flot réel. Ce flot réel peut être obtenu par un capteur odométrique [KER95] ou approché par une modélisation paramétrique du mouvement 3D [GCT94]. Une approche différente consiste à calculer le mouvement dominant dans la scène (supposé correspondre au mouvement de la caméra) par une méthode robuste et à identifier les "*outliers*" comme étant des obstacles sur la route [LMB02] .

5.4 Conclusion

Nous venons de faire un état de l'art des méthodes de détection d'obstacles dans un milieu routier à partir d'un système de vision fixé sur le véhicule. En résumé, la recherche d'une zone d'intérêt dans l'image où se trouvent les obstacles est souvent réalisée pour réduire le temps de calcul de l'étape de détection d'obstacles. Cette recherche consiste soit en une segmentation de l'image en deux régions (route/non route), soit en la détection de la délimitation de la chaussée. Les premières sont peu robustes à la présence d'ombres dans les images mais peuvent suffire pour détecter les obstacles sans aucune autre étape supplémentaire [BB94]. Les secondes sont beaucoup plus robustes mais échouent dès qu'une hypothèse nécessaire à la résolution est mise en défaut (ex : route plane).

Une fois la zone d'intérêt localisée dans l'image, l'étape de détection d'obstacles est effectuée. Les méthodes diffèrent en fonction de la définition de l'obstacle. Si l'obstacle est un objet particulier (véhicule, piéton...), l'étude d'une seule image suffit pour le détecter par recherche de forme, modélisation de l'objet ou méthode d'apprentissage à partir d'une base de données. Par compte, lorsque l'obstacle est défini comme étant tout objet qui entrave la voie de circulation du véhicule, des algorithmes plus complexes d'analyse de plusieurs images sont nécessaires. Il existe deux techniques différentes, les systèmes de stéréoscopie et les méthodes d'analyse du mouvement. Les premières recherchent les obstacles à partir de la détection de plans verticaux. Ils ont l'avantage de permettre la reconstruction 3D de la scène mais nécessitent une calibration très précise. De leur coté, les méthodes d'analyse du mouvement sont sensibles aux vibrations de la caméra et beaucoup plus complexes à mettre en oeuvre mais peuvent estimer la vitesse relative de chaque obstacle. Dans le cadre de cette thèse, nous ne disposons que d'une seule caméra fixée sur le véhicule, nous devons donc opter pour une méthode d'analyse du mouvement que nous décrirons dans le prochain chapitre.

Chapitre 6

Détection d'obstacles par une approche hiérarchique

6.1 Introduction

Dans ce chapitre, nous proposons une méthode de détection d'obstacles dans un milieu routier à partir d'une seule caméra montée sur le véhicule. Cette méthode doit être capable de détecter tous les types d'obstacles sur tous les types de routes. Comme nous l'avons vu dans le chapitre précédent, seule une méthode d'analyse de mouvement dans la séquence d'images peut résoudre ce problème. Cette méthode se doit d'être bien sûr rapide, mais aussi pour surmonter les problèmes inhérents qu'introduisent les méthodes d'analyse du mouvement, la méthode doit

- être capable d'estimer les grands déplacements dans l'image ;
- être efficace même sur des zones peu texturées ;
- être robuste aux chocs et vibrations de la caméra.

La méthode décrite ici s'inspire de la méthode de détection du mouvement introduite dans le chapitre 4. Les grands mouvements seront donc calculés grâce à une approche par multirésolution, les vibrations de la caméra ne nous poseront pas de problème majeur puisqu'ils seront pris en compte lors de l'estimation du mouvement dominant. Seul le cas d'une route non texturée nous posera un véritable problème mais nous verrons néanmoins comment l'atténuer.

La méthode proposée se décompose en quatre étapes (Fig 6.1). Dans un premier temps, le mouvement apparent de la route est estimé à partir de la méthode présentée dans le chapitre 3 que nous adaptons au cas précis d'une caméra fixée sur un véhicule. La détection d'obstacles revient ensuite à rechercher dans l'image les points qui ne suivent pas ce mouvement. C'est ce qui est fait dans un deuxième temps grâce à la méthode de

détection décrite dans le chapitre 4. Les mouvements des différents obstacles détectés sont calculés par modélisation paramétrique. Enfin, une étape de prédiction de position de la route à l'instant $t+1$ est réalisée afin d'initialiser la méthode à l'instant suivant. Le plan de ce chapitre suit ces quatre étapes. Nous présenterons les résultats dans une cinquième partie.

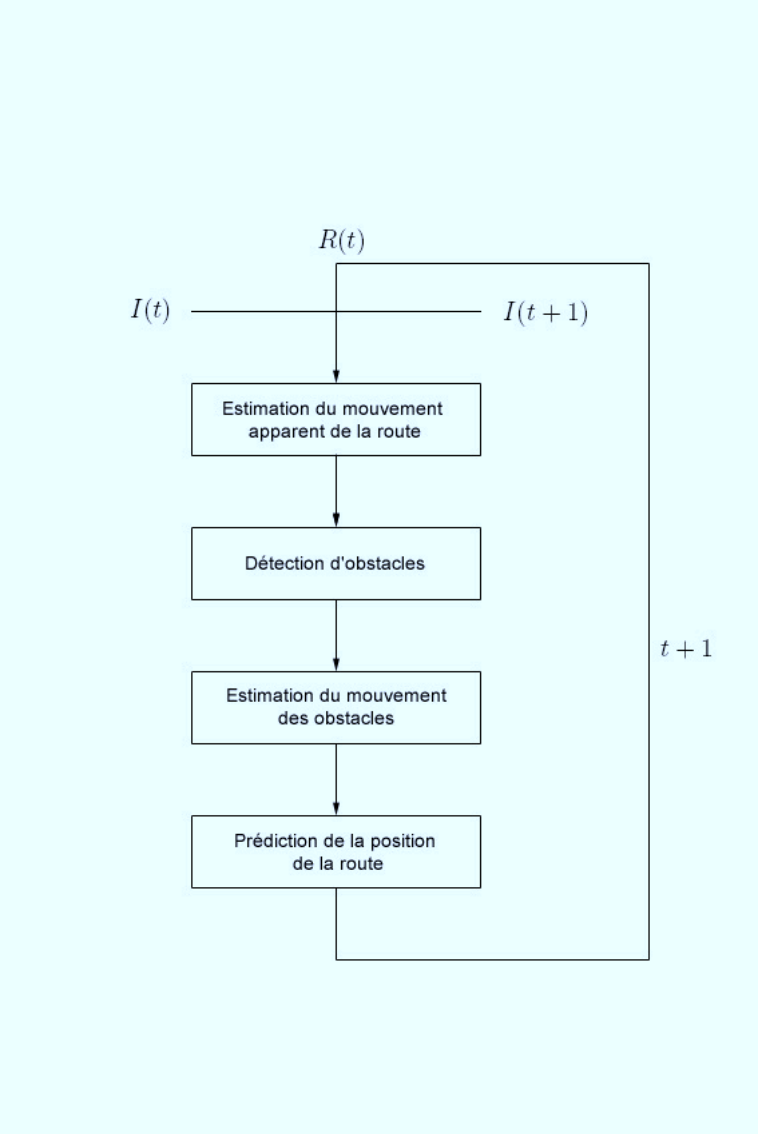


FIGURE 6.1 – Synopsis de la méthode de détection d'obstacles.

6.2 Estimation du mouvement apparent de la route

Notons $\widetilde{R}(t)$ une approximation de la zone de l'image correspondant à la surface de la route sur laquelle le véhicule peut rouler en toute sécurité à l'instant t. La première étape de la méthode consiste à estimer le mouvement apparent $\overrightarrow{v}$ de tout point de $\widetilde{R}(t)$ par la résolution du système :

$$\begin{array}{l} \forall (x,y) \in \widetilde{R}(t), \\ \overrightarrow{\nabla} I((x,y),t).\overrightarrow{v}((x,y),t) + \frac{\partial I}{\partial t}((x,y),t) = 0, \end{array} \tag{6.1}$$

et ce, grâce à un algorithme capable d'estimer les grands déplacements. Pour cela, nous adaptons la méthode d'estimation multirésolution du mouvement du chapitre 3 au cas bien précis d'une caméra fixée sur un véhicule.
Considérons donc une base d'ondelettes $(\psi^n)_{i=1\cdots N}$ de $L^2(\mathbf{R}^2)$ centrées autour de l'origine $(0,0)$, et la famille d'ondelettes discrètes

$$\psi^n_{jk}(x,y) = 2^{-j}\psi^n(2^{-j}x - k_1, 2^{-j}y - k_2), \quad k = (k_1,k_2).$$

En tout point $(x,y) = 2^j(k_1,k_2) \in \widetilde{R}(t)$, $\overrightarrow{v}((x,y),t) = (v_1((x,y),t), v_2((x,y),t))$ est solution de

$$\langle \frac{\partial I}{\partial x} v_1, \psi^n_{jk}\rangle + \langle \frac{\partial I}{\partial y} v_2, \psi^n_{jk}\rangle + \langle \frac{\partial I}{\partial t}, \psi^n_{jk}\rangle = 0, \quad \forall n = 1\cdots N. \tag{6.2}$$

A ce stade, dans le chapitre 3 nous avions fait l'hypothèse d'un mouvement affine de la caméra. Or ici dans le cas précis d'une caméra fixée sur un véhicule, on peut montrer (cf. Annexe 1) que si la route est plane, le mouvement apparent est de la forme

$$\begin{array}{rcl} \forall (x,y) \in \widetilde{R}(t) & & \\ \overrightarrow{v}_{\Xi}((x,y),t) & = & \begin{pmatrix} a_0 + a_2x + a_3y + a_6x^2 + a_7xy \\ a_1 + a_4x + a_5y + a_6xy + a_7y^2 \end{pmatrix} \\ & = & T(x,y)\Xi, \end{array} \tag{6.3}$$

où,

$$T(x,y) = \begin{pmatrix} 1 & 0 & x & y & 0 & 0 & x^2 & xy \\ 0 & 1 & 0 & 0 & x & y & xy & y^2 \end{pmatrix} \quad \text{et} \quad \Xi = (a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7)^T.$$

Ainsi, en substituant (6.3) dans (6.2) et par intégration par parties, nous obtenons à j fixé le système suivant :

$$\begin{array}{r} \forall (2^jk_1, 2^jk_2) \text{ tel que } \mathrm{supp}\psi_{jk} \subset \widetilde{R}(t), \forall n = 1..N, \\ a_0\langle I, \frac{\partial \psi^n_{jk}}{\partial x}\rangle + a_1\langle I, \frac{\partial \psi^n_{jk}}{\partial y}\rangle + a_2\left[\langle I, \psi^n_{jk}\rangle + \langle xI, \frac{\partial \psi^n_{jk}}{\partial x}\rangle\right] + a_3\langle yI, \frac{\partial \psi^n_{jk}}{\partial x}\rangle + a_4\langle xI, \frac{\partial \psi^n_{jk}}{\partial y}\rangle + \cdots \\ a_5\left[\langle I, \psi^n_{jk}\rangle + \langle yI, \frac{\partial \psi^n_{jk}}{\partial y}\rangle\right] + a_6\left[\langle 3xI, \psi^n_{jk}\rangle + \langle x^2I, \frac{\partial \psi^n_{jk}}{\partial x}\rangle + \langle xyI, \frac{\partial \psi^n_{jk}}{\partial y}\rangle\right] + \cdots \\ a_7\left[\langle 3yI, \psi^n_{jk}\rangle + \langle y^2I, \frac{\partial \psi^n_{jk}}{\partial y}\rangle + \langle xyI, \frac{\partial \psi^n_{jk}}{\partial x}\rangle\right] = \langle \frac{\partial I}{\partial t}, \psi^n_{jk}\rangle. \end{array} \tag{6.4}$$

Nous obtenons donc de nouveau l'estimation du mouvement de $R(t)$ à partir de la résolution d'un simple système sur-déterminé à $N \times \text{card}\{(k = (k_1, k_2)|\text{supp}\psi_{jk} \subset \widetilde{R}(t)\}$ équations et 8 inconnues que l'on note

$$M^j \Xi = P^j. \tag{6.5}$$

A $t = 0$, ne disposant d'aucune information sur la région correspondant à la route dans l'image, on définit $\widetilde{R}(0)$ comme étant l'image entière. (6.5) est alors résolu à l'aide d'un M-estimateur robuste. Ceci permettant, sous l'hypothèse que les obstacles dans l'image forment une zone négligeable par rapport à celle de la route, de calculer le mouvement dominant et donc le mouvement de la caméra.
A $t \neq 0$, la prédiction $\widetilde{R}(t)$ de la région de $R(t)$ étant en pratique suffisamment précise, l'utilisation d'un M-estimateur n'est pas nécessaire et une estimation de (6.5) au sens de moindres-carrés qui est plus rapide suffit.
Par conséquent, si $(M^j)^T M^j$ est inversible, Ξ est solution de :

$$\Xi = ((M^j)^T M^j)^{-1} (M^j)^T P^j. \tag{6.6}$$

Sinon, nous sommes dans le cas où la route est peu texturée pour estimer son mouvement. On pose alors $\Xi = (0, 0, 0, 0, 0, 0, 0, 0)^T$ et l'étape de détection d'obstacles consistera à détecter les zones texturées et en mouvement dans l'image.

Pour contourner l'aliasage temporel et donc pour estimer de grands déplacements de la route entre deux images consécutives, l'estimation du mouvement est obtenue par multirésolution et compensation de mouvement de la même façon qu'au chapitre 3.

6.3 Détection d'obstacles

Nous avons donc une estimation paramétrique du mouvement de la surface plane $\widetilde{R}(t)$. Il s'agit maintenant de trouver tous les points non conformes à ce mouvement, les zones connexes suffisamment importantes seront considérées comme des obstacles. Pour ce faire, nous reprenons la modélisation temporelle hiérarchique définie au chapitre 4. C'est-à-dire que nous cherchons le champ d'étiquettes $\widehat{e} = \{\widehat{e}_s, s \in S = \cup_{j=0}^{J} S^j\}$ à valeurs dans $\{0, 1\}$ (0 = étiquette conforme au mouvement, 1 = étiquette non conforme) solution de (3.33)

$$\min_e [\textstyle\sum_{<i,j> \in S^J} v_{i,j}(e_i, e_j) + \sum_{i \notin S^J} w_i(e_i, e_{\bar{i}}) + \sum_{i \in S} q_i(e_i, o_i) + \sum_{i \in S} h_i(e_i, \widetilde{e}_i)]. \tag{6.7}$$

On rappelle que les potentiels spatiaux v_{ij} et hiérarchiques w_i sont définis de la façon suivante

$$v_{i,j}(e_i, e_j) = \alpha[1 - \delta(e_i, e_j)],$$

$$w_i(e_i, e_{\bar{i}}) = \beta[1 - \delta(e_i, e_{\bar{j}})].$$

Ils favorisent le champ d'étiquettes à être homogène en espace et en échelle. La carte de prédiction $\widetilde{e}$ est définie comme suit

$$\forall s \in S^j \quad \widetilde{e}_s = \begin{cases} 0 \text{ si } 2^j s \in \widetilde{R}(t) \\ 1 \text{ sinon,} \end{cases} \tag{6.8}$$

h qui exprime la dépendance temporelle entre deux cartes de détection consécutives est défini par la formule (3.34) et le terme d'attache aux données qui relie l'observation sur le mouvement à l'étiquette de chaque point est détaillé au chapitre 4 par la formule (3.24).

Les zones connexes de la carte de détection $\widehat{e}$ solution de (6.7) ayant la valeur 1 sont les régions de l'image qui ne suivent pas le mouvement de $\widetilde{R}(t)$. Les zones suffisamment importantes sont alors les positions des obstacles à l'instant t. Posons $Ob_i(t)$, $i = 1 \cdots p$ les obstacles, la région de l'image $R(t)$ dans laquelle le véhicule peut circuler sans danger est donc,

$$R(t) = \{(x, y) \notin \bigcup_{k=1}^{p} Ob_i(t)\}. \tag{6.9}$$

6.4 Estimation du mouvement des obstacles et prédiction de la route

Le mouvement apparent de chaque obstacle est calculé grâce à une paramétrisation de celui-ci par un modèle affine. Bien que ce modèle ne représente pas précisément le mouvement de l'obstacle, il est préféré à un modèle plus complexe pour plusieurs raisons :

- Dans notre méthode de détection d'obstacles, cette étape n'est nécessaire que pour obtenir une prédiction de $R(t+1)$, or cette prédiction ne requiert pas une précision au pixel près car la carte de détection est affinée entre chaque image de la séquence.
- Étant donné que le nombre de pixels d'un obstacle dans l'image est faible, le nombre d'équations pour déterminer le mouvement paramétrique de celui-ci l'est aussi. Par conséquent, utiliser un modèle paramétrique trop complexe serait beaucoup plus sensible au bruit.
- De plus, dans le cas d'une caméra calibrée, l'hypothèse affine suffit pour estimer le temps avant collision [Mey94] (temps restant avant contact entre l'obstacle et la caméra).

On décrit donc chaque mouvement des obstacles par les formules suivantes :

$$(x, y) \in Ob_i(t) \quad \overrightarrow{v}_{\Theta_i}(x, y) = B(x, y).\Theta_i, \tag{6.10}$$

avec

$$B(x,y) = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{bmatrix}$$
$$\Theta_i = (a_i, b_i, c_i, d_i, e_i, f_i)^T .$$

Où chaque Θ_i est calculé par la méthode présentée au chapitre 3, où le système linéaire (2.53) est résolu par moindres-carrés et non à l'aide d'un M-estimateur robuste afin de réduire le temps de calcul [1].

Une fois le mouvement paramétrique de chaque obstacle $Ob_i(t)$, $i = 1 \cdots p$, la prédiction de la position des obstacles à l'instant $t+1$ est donnée par

$$\forall i = 1 \cdots p, \quad \widetilde{Ob}_i(t+1) = \{(x,y) + \overrightarrow{v}_{\Theta_i}(x,y) | (x,y) \in Ob_i(t)\}. \tag{6.11}$$

La prédiction de la zone dans laquelle le véhicule peut rouler en toute sécurité est définie par

$$\widetilde{R}(t+1) = \{(x,y) \notin \bigcup_{k=1}^{p} \widetilde{Ob}_i(t+1)\}.$$

Cette zone permet d'initialiser l'algorithme à l'instant $t+1$.

6.5 Expérimentations

Nous avons testé notre méthode sur trois séquences routières bien différentes. La première est une séquence où le véhicule sur lequel est fixée la caméra est immobile. La deuxième séquence est une séquence prise en milieu autoroutier, et enfin, la troisième plus complexe est une séquence prise à partir d'un véhicule évoluant dans un rond-point.

Dans l'ensemble des expérimentations, les paramètres ont pour valeurs,

J	α	β	γ	G
4	0.3	0.6	0.3	50

6.5.1 Séquence Van

Cette séquence est constituée de 64 images de taille 256×256. La caméra est placée sur un véhicule immobile situé au carrefour de deux rues. Une voiture blanche venant de la droite traverse le carrefour avant de disparaître derrière une camionnette blanche qui circule en sens inverse. Cette camionnette est suivie d'une voiture noire roulant à la même vitesse (Fig 6.2).

1. Remarquons que plusieurs obstacles de mouvement différent peuvent se toucher et former une seule région Ob_i. Dans ce cas, Θ_i représente le mouvement moyen de la région Ob_i et non le véritable déplacement de chaque obstacle.

Nous présentons les résultats de deux façons différentes. La première et la troisième lignes de la figure 6.2 correspondent aux résultats bruts de la détection de zones mobiles dans la séquence aux instants $t = 3, 6, 28, 45, 53, 60$ et les deuxième et quatrième lignes montrent les obstacles retenus à ces mêmes instants. Ces obstacles sont en fait représentés par les rectangles englobant des régions détectées.

Dans cette séquence, nous avons fixé ϵ à 1. Dans un premier temps, la voiture venant de la droite et la camionnette venant de la gauche sont correctement détectées (Fig 6.2(d) et Fig 6.2(e)). Lorsque ces deux véhicules se rapprochent avant même que la voiture blanche disparaisse derrière la camionnette, nous ne détectons plus qu'un seul obstacle (Fig 6.2(f)). Dans un deuxième temps, la voiture noire entre dans le champ de vision de la caméra. Étant donné que cette voiture est sombre, elle se confond avec le reste de la scène, c'est pourquoi celle-ci n'est pas entièrement localisée (Fig 6.2(j)). A $t = 53$, la camionnette n'étant pas suffisamment texturée, elle est décomposée en deux obstacles distincts (Fig 6.2(k)). Enfin, à la fin de la séquence, les trois véhicules sont correctement localisés dans l'image (Fig 6.2(l)).

6.5.2 Séquence Autoroute

Cette séquence est composée de 99 images de taille 360×280. Elle a été filmée à partir d'un véhicule circulant sur une autoroute. Dans les premiers instants de la séquence, aucun obstacle ne se trouve devant le véhicule. Puis, celui-ci se fait peu à peu doubler par quatre voitures (Fig 6.4).

Dans cette séquence, nous avons pris deux valeurs différentes pour l'erreur admissible entre le mouvement réel d'un pixel et le mouvement estimé de la route, $\epsilon = 1$ (Fig 6.3) et $\epsilon = 3$ (Fig 6.4). Nous présentons les résultats de la détection d'obstacles sur huit images de la séquence. Comme prévu, les zones considérées comme étant des obstacles sur la route (représentées en noir dans les résultats) sont plus importantes lorsque ϵ est plus faible. Cependant, lorsque ce paramètre est faible, les bandes blanches sont parfois considérées comme des obstacles notamment dans la figure 6.3(e) et la figure 6.3(f) tout comme l'ombre des panneaux figure 6.3(n). Ces fausses alarmes sont dues au fait que le mouvement de la route n'est pas correctement estimé. En effet, bien que la route représente la majeure partie de l'image, celle-ci est peu texturée et ne procure donc aucune information sur le mouvement de la caméra. Par conséquent, les zones texturées de l'image bien qu'elles soient de petites tailles ont un poids important dans l'estimation au sens des moindres carrés du système d'équations 6.4 et le résultat de l'estimation du mouvement se rapprochera plus du mouvement des zones texturées (en l'occurrence, les arbres situés à gauche et à droite de l'image) que du mouvement de la route. Ceci n'a pas de conséquence si l'on prend une

valeur pour ϵ relativement importante puisque l'estimation du mouvement la route reste convenable. En revanche, si ϵ est faible, toute partie texturée de la route sera considérée comme obstacle.

Néanmoins, pour $\epsilon = 3$ les résultats de la détection d'obstacles sont corrects. Il n'y a pas de fausses alarmes. Dans les premiers instants, seules les rambardes de sécurité sont détectées (Fig 6.4(e)), puis les quatre véhicules doublant la voiture sur laquelle est fixée la caméra sont détectés l'un après l'autre. Les panneaux de direction et le pont sont eux aussi considérés comme des obstacles potentiels.

6.5.3 Séquence Rond-Point

La séquence Rond-Point est quant à elle beaucoup plus complexe. Elle est composée de 150 images de taille 256×256. Elle est filmée à partir d'un véhicule se dirigeant vers un rond-point où à sa droite se trouve une voiture en stationnement et à sa gauche une voiture sortant du rond-point. Le véhicule entre ensuite dans le rond-point pour faire demi-tour sans marquer l'arrêt. Pendant que celui-ci est dans le rond-point, il croise plusieurs voitures qui lui cèdent le passage (Fig 6.5).

Nous présentons les résultats de notre méthode de détection d'obstacles en fixant ϵ à 3. Dans l'ensemble, les obstacles sur la route : le véhicule stationné et celui venant en sens opposé, les panneaux de signalisation, le terre-plein, les arbres et les véhicules cédant le passage sont correctement détectés. Il est à noter que pour les mêmes raisons que dans l'expérience précédente, la ligne blanche à l'instant $t = 25$ est considérée comme obstacle alors que le bâtiment à $t = 70$ qui n'étant pas suffisamment texturé n'est pas détecté. Notons que la méthode implantée sous Matlab, nécessite sur ce type d'images en moyenne moins de 30 secondes par image sur un PC Pentium IV $2,4$ Ghz, ce qui laisse présager que le temps réel avec cette méthode est accessible.

6.6 Conclusion

Dans ce chapitre, nous avons introduit une méthode de détection d'obstacles dans un milieu routier à partir d'une caméra fixée sur un véhicule. Nous avions comme ambition de détecter tous les types d'obstacles (véhicules, piétons, panneaux, arbres ...) sur tous les types de routes (marquées ou non). Pour ce faire, n'ayant qu'une seule caméra, nous avons vu dans le chapitre précédent, que seules les méthodes par estimation de mouvement étaient envisageables. Nous avons donc adapté notre méthode de détection de mouvement à partir de caméra mobile développée au chapitre 4 dans le cas bien précis d'une caméra fixée sur un véhicule.

En supposant la route plane, nous avons modélisé son mouvement non pas par un modèle affine comme nous l'avions fait au chapitre 4 mais par un modèle quadratique à 8 paramètres. Ces 8 paramètres ont été calculés par la projection de l'équation de contrainte du flot optique sur une base d'ondelettes. La détection d'obstacle consiste ensuite à rechercher les points de l'image qui ne suivent pas le mouvement apparent de la route modélisé par ces 8 paramètres. Ceci a été fait grâce à une modélisation markovienne hiérarchique. Enfin, nous avons calculé le mouvement de chaque obstacle. Cette dernière étape est utile pour deux raisons. La première est qu'elle permet d'obtenir une prédiction de la route à l'instant suivant et évite donc d'utiliser un estimateur robuste gourmand en temps de calcul pour calculer le mouvement de la route. La seconde est que la connaissance du mouvement de chaque obstacle permet dans la cas d'une caméra calibrée d'évaluer la dangerosité de chacun d'eux par le calcul du temps avant collision [Mey94].

La méthode que nous avons développée a été testée, dans ce chapitre, sur trois séquences différentes. Dans l'ensemble, les différents obstacles rencontrés dans ces séquences ont été correctement détectés par notre méthode. Cependant, comme toutes les méthodes fondées seulement sur l'information de mouvement dans les images, il est à noter que le choix de la valeur ϵ peut s'avérer sensible. Ceci est dû au fait que le mouvement de la route n'est pas toujours correctement estimé à cause des faibles variations spatiales d'intensité sur celle-ci, ce qui entraîne de faibles informations sur le mouvement.

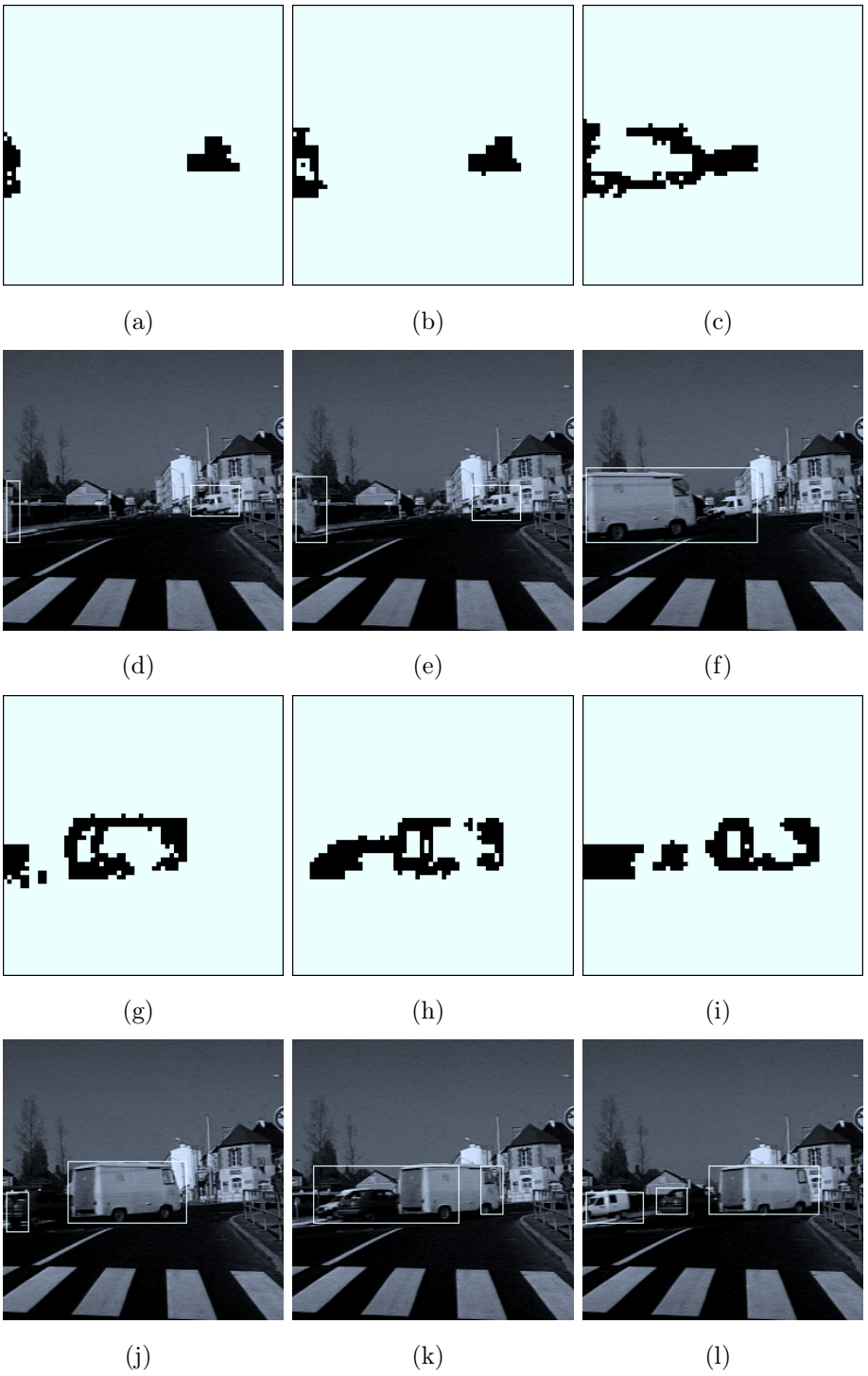


FIGURE 6.2 – Résultat de la détection d'obstacles sur la séquence Van. Première ligne, détection aux instants (a) $t = 3$, (b) $t = 6$, (c) $t = 28$. Deuxième ligne, obstacles retenus à ces mêmes instants. Troisième et quatrième lignes, détection aux instants (g)(j) $t = 45$, (h)(k) $t = 53$, (i)(l) $t = 60$.

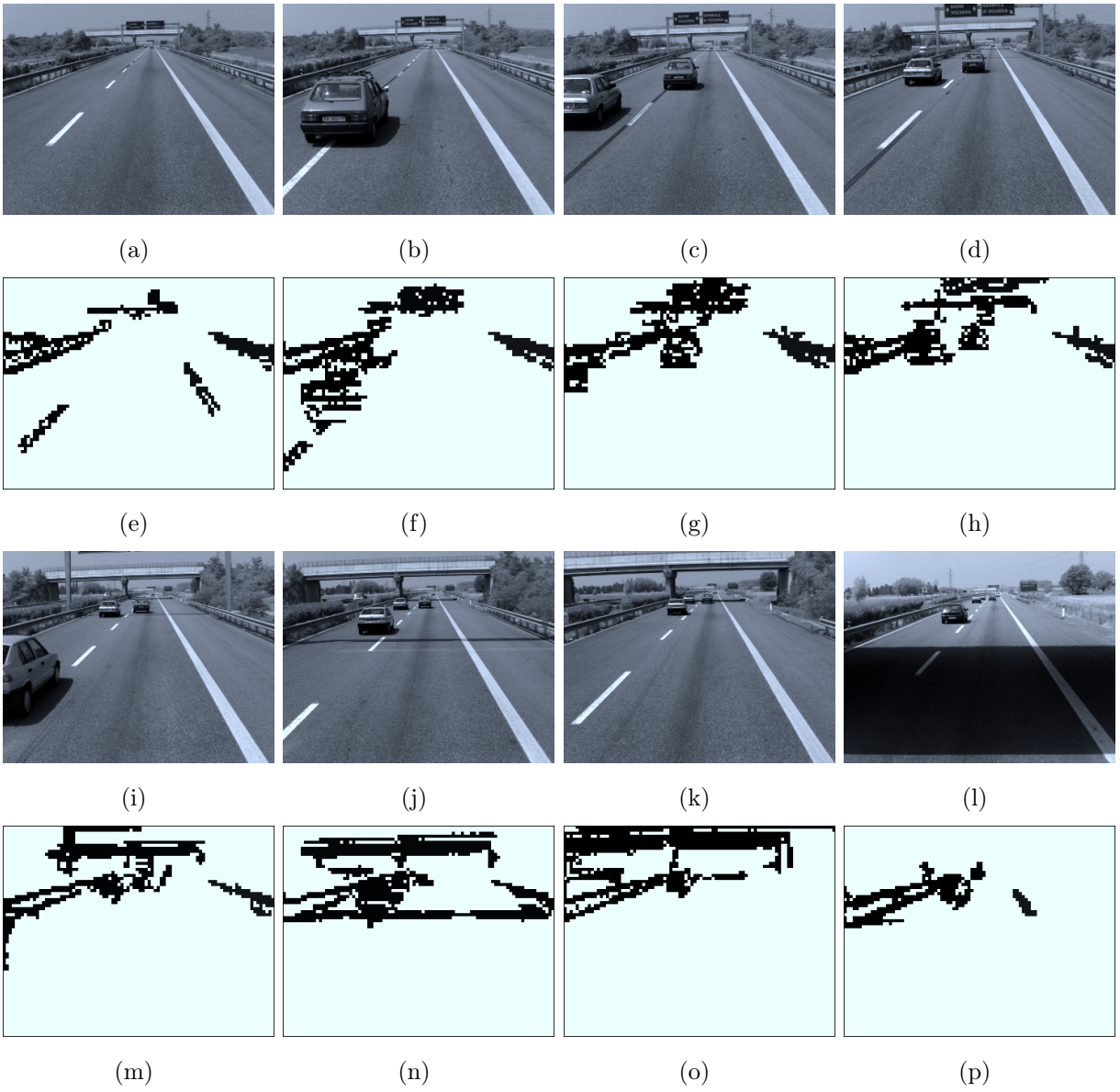


FIGURE 6.3 – Détection d'obstacles sur la séquence Autoroute avec $\epsilon = 1$. La première et la deuxième lignes correspondent respectivement aux images originales et aux résultats de la détection d'obstacles aux instants $t = 1, 7, 14, 20$. La troisième et la dernière lignes correspondent aux images originales et à la détection d'obstacles aux instants $t = 28, 36, 44, 69$.

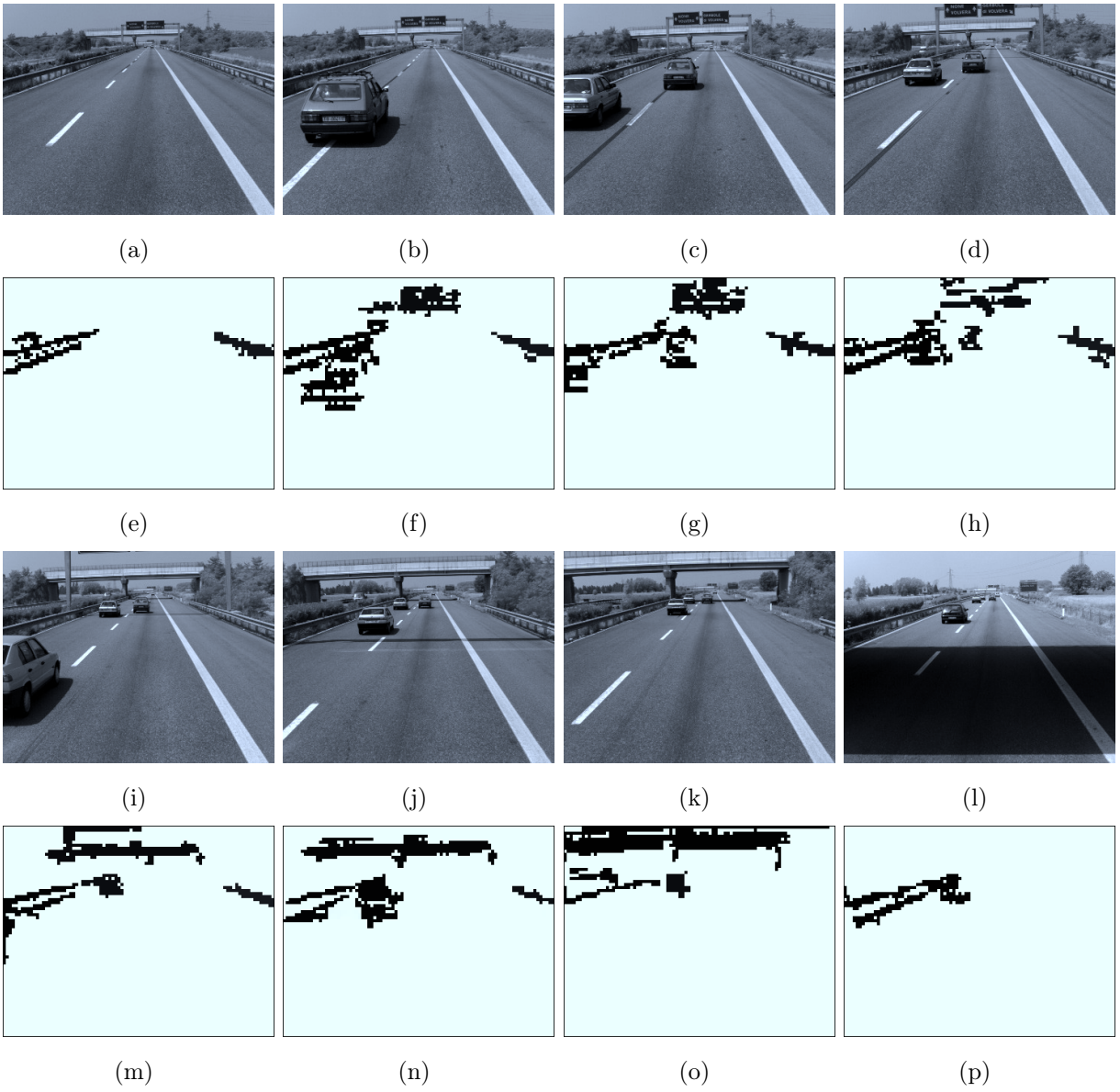


FIGURE 6.4 – Détection d'obstacles sur la séquence Autoroute avec $\epsilon = 3$. La première et la deuxième lignes correspondent respectivement aux images originales et aux résultats de la détection d'obstacles aux instants $t = 1, 7, 14, 20$. La troisième et la dernière lignes correspondent aux images originales et à la détection d'obstacles aux instants $t = 28, 36, 44, 69$.

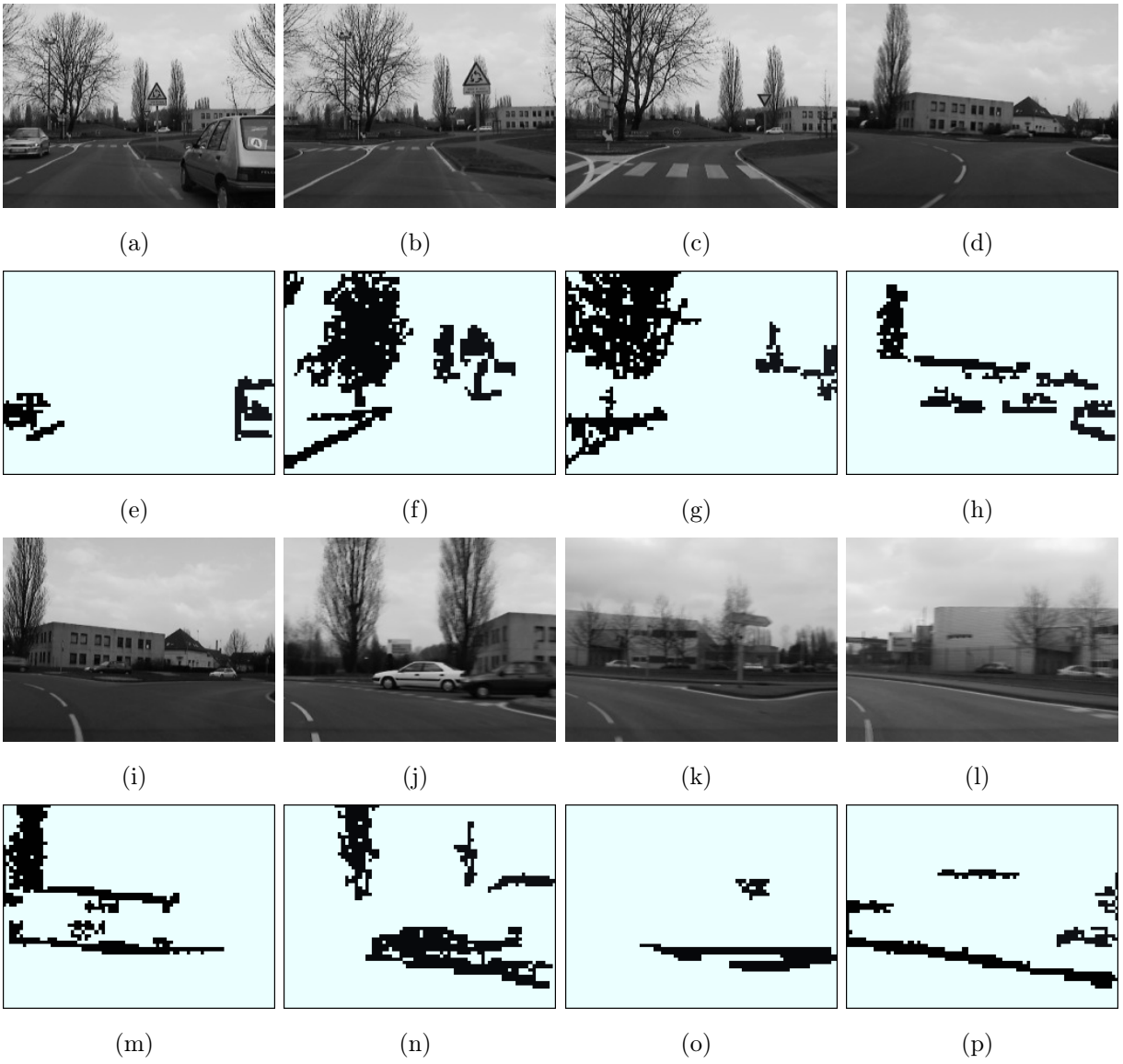


FIGURE 6.5 – Détection d'obstacles sur la séquence Rond-Point. La première et la deuxième lignes correspondent respectivement aux images originales et aux résultats de la détection d'obstacles aux instants $t = 5, 25, 45, 70$. La troisième et la dernière lignes correspondent aux images originales et à la détection d'obstacles aux instants $t = 80, 100, 127, 145$.

Conclusion et perspectives

Ce travail fait partie d'un projet de détection d'obstacles dans un milieu routier et de coopération inter-véhicules financé par la région Picardie intitulé : "Perception coopérative inter-véhicules pour la détection d'imprévus dans un milieu routier". La finalité de ce projet est de détecter les obstacles sur la route à partir d'un système mono-caméra fixé sur un véhicule. Ce système doit être capable de fonctionner de façon autonome ou par coopération avec d'autres véhicules munis du même système. Nous nous sommes uniquement intéressés au premier de ces deux cas. Après un état de l'art des méthodes de détection d'obstacles à partir de caméras, il s'est avéré que seules des méthodes d'analyse de mouvement pouvaient résoudre ce problème si nous ne disposions que d'une seule caméra. C'est donc de ce sujet que traite l'ensemble de ce mémoire.

Contributions

De nombreux chercheurs se sont penchés sur les problèmes d'analyse du mouvement aux multiples applications : en traitement d'images (compression ou indexation de vidéo, vidéo-surveillance...), en robotique mobile (détection d'obstacles, suivi de cibles...) ou encore en météorologie. L'application visée de ce travail étant la création d'un système embarqué fonctionnant en temps réel, nous étions contraints de développer des méthodes rapides d'estimation et de détection de mouvement à partir de caméras mobiles. Pour ce faire, nous avons introduit une méthode d'estimation de mouvement paramétrique par analyse d'ondelettes. Cette méthode nous a ensuite permis de modéliser le problème de détection d'objets en mouvement dans la scène grâce à un champ markovien hiérarchique. L'utilisation de ces deux outils mathématiques, les ondelettes et les champs markoviens hiérarchiques, nous ont permis de contourner le problème de l'aliasage temporel inhérent aux méthodes d'estimation du flot optique à l'aide de l'équation de contrainte du mouvement apparent (E.C.M.A) mais aussi d'obtenir une méthode relativement peu coûteuse en temps de calcul. Nous avons aussi introduit une méthode de segmentation au sens du mouvement entre deux images sans connaissance *a priori* et sans hypothèse de présence d'un mouvement dominant. L'ensemble de ces méthodes a été validé par des expérimentations à la fois sur des séquences synthétiques et des séquences réelles. Un inconvénient

de nos méthodes est que l'utilisation d'ondelettes analytiques pour l'estimation du mouvement ne nous permet pas d'obtenir d'information sur celui-ci aux deux résolutions les plus fines de l'image. Bien que ce problème puisse être un frein à nos méthodes pour des applications nécessitant une localisation précise des frontières de mouvement telles que la compression vidéo, il n'en est pas un dans le cas de la détection d'obstacles.

Perspectives

Dans nos travaux futurs, il serait tout de même intéressant d'utiliser des ondelettes qui permettent à la fois un bon conditionnement des systèmes linéaires définis dans l'estimation du mouvement (comme c'est le cas des ondelettes analytiques) et dont les coefficients peuvent être calculés par un algorithme rapide de transformée en ondelettes aux échelles plus fines. Ceci permettrait d'obtenir des informations sur le mouvement aux résolutions fines de l'image. Cet inconvénient pourrait aussi être surmonté en utilisant une mesure d'erreur différente aux échelles sur lesquelles les coefficients d'ondelettes analytiques ne sont pas calculés.

L'ajout d'une dépendance temporelle dans le cas de la détection de mouvement a permis d'améliorer les résultats en diminuant le nombre de fausses alarmes. Il nous semble donc important d'étendre notre méthode de segmentation au sens du mouvement entre deux images à la segmentation de séquences d'images en ajoutant une telle dépendance temporelle dans la modélisation markovienne et en utilisant la carte de segmentation précédente projetée dans le sens du mouvement pour initialiser l'algorithme de segmentation à l'instant courant. La méthode d'initialisation à l'échelle grossière décrite dans ce mémoire ne serait alors utilisée qu'entre les deux premières images de la séquence, ceci dans le but d'éviter les problèmes de sur-segmentation engendrés lorsque nous initialisons notre méthode à l'échelle grossière avec autant de régions que de pixels.

Dans le cas de la détection d'obstacles dans un milieu routier, les résultats expérimentaux ont montré le bon comportement de nos algorithmes. Il serait néanmoins important pour la suite du projet du pouvoir quantifier la dangerosité des différents obstacles détectées (obstacles mobiles sur la route, obstacles arrivant à vive allure...) afin de prévenir les autres véhicules par le système de coopération. De plus, étant donné la faible complexité algorithmique de notre méthode, on peut espérer une future implantation temps-réel sur des cartes spécifiques.

Quatrième partie

Annexes

Annexe A

Les modèles paramétriques du mouvement

Nous discutons ici de différents modèles paramétriques du mouvement apparent. Ces modèles proviennent d'une approximation plus ou moins précise de la projection du mouvement 3D de la scène sur la caméra.

A.1 Relation entre le mouvement réel et le mouvement dans l'image

Soit $P = (X, Y, Z)$ un point de l'espace, dans le cas d'une modélisation de la caméra par projection perspective, les coordonnées (x, y) de ce point dans le plan image sont données par

$$\begin{aligned} x &= f\tfrac{X}{Z}, \\ y &= f\tfrac{Y}{Z}. \end{aligned} \tag{A.1}$$

L'image obtenue a pour expression dans le repère propre aux pixels de l'image numérique

$$\begin{aligned} u &= x + sy + u_0, \\ v &= \tfrac{y}{r} + v_0, \end{aligned} \tag{A.2}$$

où (u, v) sont les coordonnées de l'image de P, f la focale (distance entre le point focal et le plan de l'image), s est le facteur de distortion, r est le rapport des distances horizontales et verticales et (u_0, v_0) sont les coordonnées du centre de l'image (projeté orthogonal du point focal). En pratique, ces différents paramètres peuvent être obtenus par calibration. Nous supposons pour ne pas surcharger les notations que nous sommes dans les conditions suivantes : (u_0, v_0) est le centre de l'image, r=1, s=0 et f=1. C'est-à-dire que $u = x$ et $v = y$.

Supposons que la caméra subit un déplacement V que l'on décompose sous forme d'une rotation $\omega = (\omega_1, \omega_2, \omega_3)^T$ et d'une translation $t = (t_x, t_y, t_z)^T$

$$V = (\dot{X}, \dot{Y}, \dot{Z})^T = -t - \omega \wedge (X, Y, Z)^T. \tag{A.3}$$

La vitesse $\overrightarrow{v} = (\dot{x}, \dot{y})$ d'un point de l'image qui correspond à un point fixe de la scène a alors pour expression

$$\begin{aligned} \dot{x} &= \tfrac{\dot{X}}{Z} - \tfrac{X\dot{Z}}{Z^2} = (-t_x + xt_z)g(x,y) + xy\omega_x - (1+x^2)\omega_y + y\omega_z \\ \dot{y} &= \tfrac{\dot{Y}}{Z} - \tfrac{Y\dot{Z}}{Z^2} = (-t_y + yt_z)g(x,y) + (1+y^2)\omega_x - xy\omega_y - x\omega_z, \end{aligned} \tag{A.4}$$

où $g(x,y) = \frac{1}{Z(x,y)}$ est l'inverse de la profondeur de la scène. Ainsi, plusieurs inconnues entrent en jeu pour estimer le déplacement exact des pixels dans l'image, les 6 degrés de liberté du mouvement de la caméra $(t_x, t_y, t_z, \omega_1, \omega_2, \omega_3)$ mais aussi la profondeur des objets $Z(x,y)$.
Une première méthode pour diminuer le nombre d'inconnues est de supposer le mouvement de translation de la caméra petit par rapport à la distance des objets. Dans ce cas, le déplacement dans l'image est dû à un mouvement de rotation pure de la caméra et s'exprime de la façon suivante

$$\begin{aligned} \dot{x} &= xy\omega_x - (1+x^2)\omega_y + y\omega_z, \\ \dot{y} &= (1+y^2)\omega_x - xy\omega_y - x\omega_z. \end{aligned} \tag{A.5}$$

Lorsque le champ de vue de la caméra est petit ($|x|, |y| \ll 1$) les termes de second ordre sont négligeables ce qui donne

$$\begin{aligned} \dot{x} &= -\omega_y + y\omega_z, \\ \dot{y} &= \omega_x - x\omega_z. \end{aligned} \tag{A.6}$$

Une autre méthode pour approcher (A.4), consiste à faire une hypothèse sur $g(x,y)$. Si l'on suppose que la scène est une surface plane, alors g est de la forme

$$g(x,y) = ax + by + c. \tag{A.7}$$

Dans ce cas, le mouvement de la caméra peut être calculée à partir de huit paramètres $(a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7)$ par

$$\begin{aligned} \dot{x} &= a_0 + a_1 x + a_2 y + a_6 x^2 + a_7 xy, \\ \dot{y} &= a_3 + a_4 x + a_5 y + a_6 xy + a_7 y^2. \end{aligned} \tag{A.8}$$

C'est ce modèle que nous utilisons pour estimer le mouvement d'une caméra montée sur un véhicule qui se déplace sur une route supposée plane (Chapitre 6).

A.2 Quelques exemples de modélisations paramétriques

D'autres approximations du mouvement 3D réel existent, nous citons ici les plus couramment utilisées. Nous supposons qu'une région de la scène se déplace dans le repère lié à la caméra à la vitesse $\overrightarrow{v}_{3D}$ et nous discutons des différents modèles paramétriques qui approchent la projection de ce mouvement sur le plan image :

• Modèle nul (0 paramètres) : $\Theta = \vec{0}$

$$\overrightarrow{v}(x,y) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Une région a pour modèle un modèle nul si la région est supposée statique au cours du temps.

• Modèle constant (2 paramètres) : $\Theta = (t_x, t_y)$

$$\overrightarrow{v}(x,y) = \begin{pmatrix} t_x \\ t_y \end{pmatrix}.$$

Ce modèle prend en compte les régions qui ont un mouvement de translation uniforme.

• Modèle affine simplifié (4 paramètres) : $\Theta = (t_x, t_y, k, \theta)$

$$\overrightarrow{v}(\mathrm{x},\mathrm{y}) = \begin{pmatrix} \mathrm{t_x} \\ \mathrm{t_y} \end{pmatrix} + \begin{pmatrix} k & -\theta \\ \theta & k \end{pmatrix} \begin{pmatrix} x - x_G \\ y - y_G \end{pmatrix}.$$

Ce modèle considère que la région où l'on étudie le flot optique est sujette à des mouvements de translation $2D$, de rotation $2D$ ou d'homothétie. Le paramètre θ est appelé vitesse de rotation, k est appelé paramètre de divergence et (x_G, y_G) est le centre de gravité de la région.

• Modèle affine (6 paramètres) : $\Theta = (t_x, t_y, a, b, c, d)$

$$\overrightarrow{v}(\mathrm{x},\mathrm{y}) = \begin{pmatrix} \mathrm{t_x} \\ \mathrm{t_y} \end{pmatrix} + \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} x - x_G \\ y - y_G \end{pmatrix}.$$

Ce modèle permet à la différence du précédent de modéliser les déformations comme les affinités et les cisaillements. Il est le plus couramment utilisé puisqu'il est un bon compromis entre richesse de la représentation du mouvement et complexité du calcul de l'estimation.

Annexe B

Les estimateurs robustes

Le but de l'estimation robuste est de déterminer le vecteur Θ de p paramètres qui ajuste au mieux le modèle $(M(X_i, \Theta))_{i=1\cdots n}$ relativement aux observations $(y_i)_{i=1\cdots n}$ $(n \gg p)$ sachant que certaines données, dites *outliers*, ne correspondent pas au modèle choisi. Dans notre cas, ces *outliers* peuvent correspondre à des points de l'image non conformes au mouvement ou à des points où l'hypothèse de conservation de l'intensité au cours du temps est mise en défaut.

L'efficacité d'un estimateur robuste est défini à partir de trois indices [Hub81] :

- *L'efficacité relative* qui est le rapport entre la variance la plus basse atteignable pour les paramètres à estimer et la variance obtenue par l'estimateur robuste.
- *Le point de rupture* qui représente la plus petite fraction de données erronées qui peut rendre l'estimation arbitrairement fausse.
- *La complexité algorithmique* de la méthode robuste qui doit être aussi faible que possible.

Il existe de nombreux estimateurs robustes [MMKR91], nous présentons ici les trois principaux : les moindres carrés tronqués, les moindres carrés médians et les M-estimateurs.

B.1 Estimateur des moindres carrés tronqués

Notons $r_i = M(X_i, \Theta) - y_i$, $i = 1 \cdots n$ et $r_\sigma(i)$ les résidus classés dans l'ordre croissant. L'estimateur des moindres carrés tronqués (Least Trimmed Squares LTS) est solution de

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{h} (r_\sigma(i))^2. \tag{B.1}$$

La constante h $(\frac{n}{2} \leq h \leq n)$ détermine le point de rupture de l'estimateur. La solution au sens des moindres carrés tronqués minimise la somme des carrés des h résidus les plus

petits, les $n-h$ plus grands résidus n'étant pas considérés dans l'estimation. Si $h=n$, la méthode correspond aux moindres carrés classiques et le point de rupture vaut $\frac{1}{n}$ (une seule erreur aberrante peu corrompre le résultat d'une valeur arbitrairement grande). La valeur de h pour laquelle le point de rupture maximum est atteint est $h = [\frac{n}{2}] + [\frac{p+1}{2}]$ ([RL87]).
La solution exacte d'un LTS est obtenue de manière exhaustive en considérant les C_n^h échantillons différents de taille h et en retenant le meilleur ajustement possible. Mais cette méthode est coûteuse en temps de calcul et il faut donc se contenter d'un échantillonnage aléatoire pour "visiter" grossièrement la plupart des régions de l'espace de recherche.

B.2 Estimateur des moindres carrés médians

L'estimateur des moindres carrés médians (LMS) [Rou84] est solution de :

$$\widehat{\Theta} = \arg\min_{\Theta} \mathrm{Med}_{i=1\cdots n}(M(X_i,\Theta) - y_i)^2. \tag{B.2}$$

$\widehat{\Theta}$ est la solution qui minimise la valeur médiane de l'ensemble des carrés des résiduels. Cet estimateur a l'avantage d'être très robuste puisqu'il atteint son point de rupture à 50% (valeur maximale possible pour un estimateur) mais il requiert cependant un temps de calcul très important. Pour obtenir une solution moins coûteuse, on utilise un algorithme de type RANSAC [FB81] :

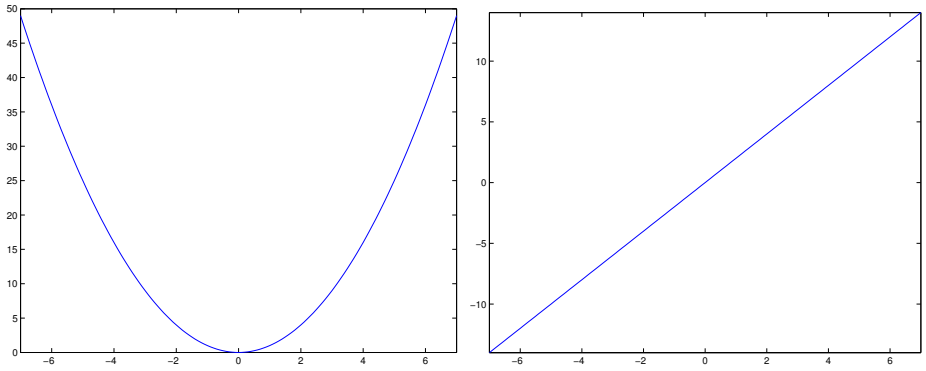
1. Choisir au hasard des ensembles de p points de l'ensemble de n points.
2. Calculer l'ajustement du modèle aux p points pour chaque ensemble.
3. Calculer la médiane des résidus au carré et prendre le modèle qui donne la médiane minimale.

B.3 Les M-estimateurs

Le principe de cet estimateur est de minimiser la somme de résiduels suivante

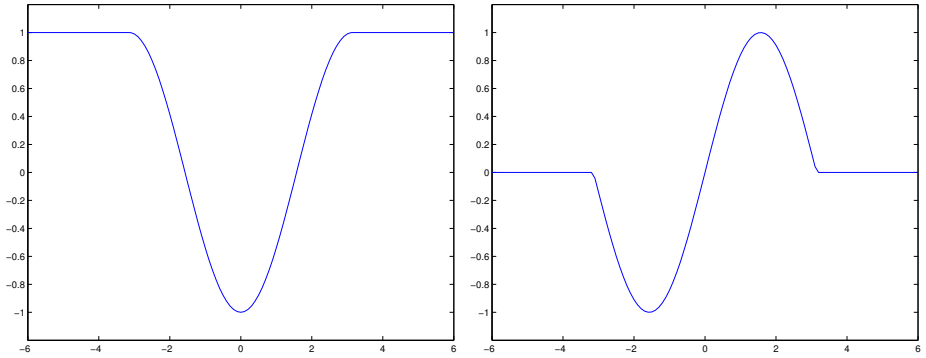
$$\widehat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{n} \rho(M(X_i,\Theta) - y_i, \sigma), \tag{B.3}$$

où σ correspond à un facteur d'échelle et ρ est appelé M-estimateur.
Dans le cas d'un estimateur continu, la robustesse d'un M-estimateur est caractérisée par sa dérivée Ψ. Elle définit l'influence que peut produire une erreur sur l'estimation. Par exemple, si l'on considère la norme L^2, l'influence des erreurs augmentent linéairement (Fig B.1(b)) ce qui explique la faible robustesse de cet estimateur.



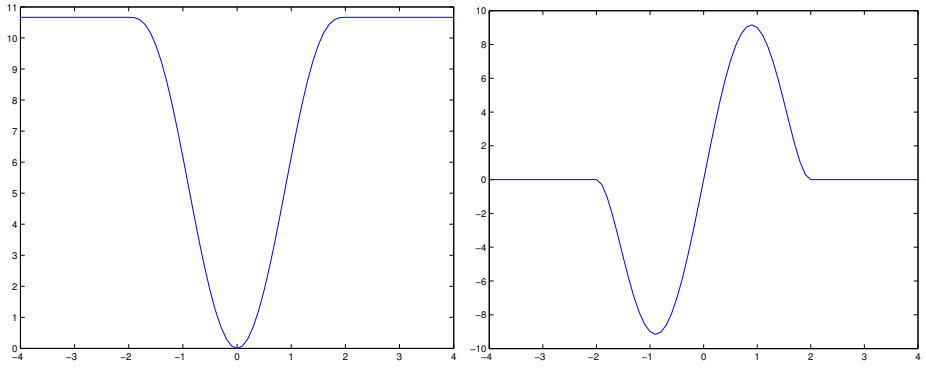
(a) (b)

FIGURE B.1 – Estimateur quadratique (a) et sa fonction d'influence (b).



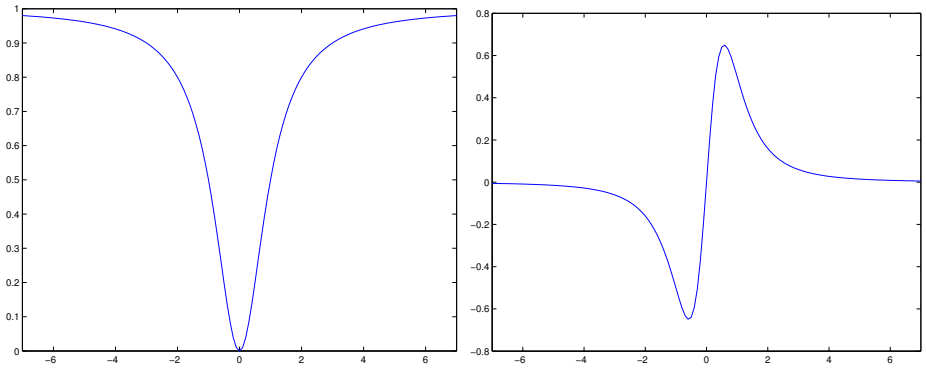
(a) (b)

FIGURE B.2 – Estimateur de Andrew (a) et sa fonction d'influence (b).



(a) (b)

FIGURE B.3 – Estimateur de Tukey (a) et sa fonction d'influence (b).



(a) (b)

FIGURE B.4 – Estimateur de Geman et Mc Clure (a) et sa fonction d'influence (b).

Il existe de nombreux estimateurs, citons par exemple :
L'estimateur d'Andrew (Fig B.2),

$$\rho(x,\sigma) = \begin{cases} -\sigma\cos(\frac{x}{r}) & \text{si} \quad |x| < \pi\sigma \\ \sigma & \text{sinon,} \end{cases} \qquad \Psi(x,\sigma) = \begin{cases} -\sin(\frac{x}{r}) & \text{si} \quad |x| < \pi\sigma \\ 0 & \text{sinon.} \end{cases} \tag{B.4}$$

L'estimateur "*biweight*" de Tukey (Fig B.3),

$$\rho(x,\sigma) = \begin{cases} \frac{x^6}{6} - \frac{\sigma^2 x^4}{2} + \frac{\sigma^4 x^2}{2} & \text{si} \quad |x| < \sigma \\ \frac{\sigma^6}{6} & \text{sinon,} \end{cases} \qquad \Psi(x,\sigma) = \begin{cases} x(x^2-\sigma^2)^2 & \text{si} \quad |x| < \sigma \\ 0 & \text{sinon.} \end{cases} \tag{B.5}$$

L'estimateur de Geman et Mc Clure (Fig B.4),

$$\rho(x,\sigma) = \frac{x^2}{\sigma + x^2}, \quad \Psi(x,\sigma) = \frac{2x\sigma}{(\sigma+x^2)^2}. \tag{B.6}$$

La technique des moindres carrés pondérés itérés (MCPI) est une méthode classique pour minimiser (B.3). L'idée de cette méthode est d'attribuer un poids w_i à chaque résiduel r_i permettant de contrôler l'influence de chaque observation (y_i). Des poids élevés sont attribués aux "bonnes" données et des poids faibles aux "*outliers*". Pour cela nous reformulons le problème (B.3) en

$$\widehat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{n} \tfrac{1}{2} w_i r_i^2 \quad \text{avec} \quad w_i(r,\sigma) = \frac{\Psi(r,\sigma)}{r}. \tag{B.7}$$

La minimisation est effectuée de manière alternée : estimation des poids w_i (initialement égaux à 1), puis estimation de $\widehat{\Theta}$ par moindres carrés pondérés à partir des valeurs courantes des poids, jusqu'à convergence.

Annexe C

Publications de l'auteur

Revue à comité de lecture

- C. Demonceaux, A. Potelle, D. Kachi-Akkouche : "Obstacle detection in road scene based on motion analysis" in IEEE Transactions on Vehicular Technology (accepté pour publication).

Conférences internationales

- C. Demonceaux, D. Kachi-Akkouche : "Fast motion estimation and motion segmentation using multi-scale approach" IEEE International Conference on Image Processing (ICIP 04), Singapour, 24-27 Octobre 2004.
- C. Demonceaux, A. Potelle, D. Kachi-Akkouche :"Obstacle detection in road scene based on road detection and motion analysis" Intelligent Autonomous Vehicles 2004, IAV04, Lisbonne, 5-7 Juillet 2004.
- C. Demonceaux, D. Kachi-Akkouche : "Robust obstacle detection with monocular vision based on motion analysis" IEEE Intelligent Vehicles Symposium, Parma, Italy, 14-17 Juin, 2004.

Workshops internationaux

- C. Demonceaux, D. Kachi-Akkouche : "Motion detection using wavelet analysis and hierarchical Markov model" in Proc. of the Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA 04), 15 mai 2004, Prague, République Tchèque.
- C. Demonceaux, D. Kachi-Akkouche : "Optical flow estimation in omnidirectional images using wavelet analysis" in Proc. of the Workshop on Omnidirectional Vision and Camera Networks (OMNIVIS 03), 21 Juin 2003, Madison, Wisconsin,USA.

Bibliographie

[AB85] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America-A*, 2(2) :284–299, 1985.

[ACC00] R. Aufrère, R. Chapuis, and F. Chausse. A dynamic vision algorithm to locate a vehicle on a non-structured road. *Int. Journal of Robotics Research*, 19(5) :411–423, Mai 2000.

[ADMT01] T. Aach, L. Duembgen, R. Mester, and D. Toth. Bayesian illumination-invariant motion detection. In *Proc. Int. Conf. on Image Processing*, pages III : 640–643, Thessalonique, Grèce, October 2001.

[AET98] Y. Altunbasak, P.E. Eren, and A.M. Tekalp. Region-based parametric motion segmentation using color information. *CVGIP : Graphical Models and Image Processing*, 60(1) :13–23, Janvier 1998.

[AG92] R. Azencott and C. Graffigne. Non supervised segmentation using multi-level markov random fields. In *Proc. Int. Conf. on Pattern Recognition*, volume 3, pages 201–204, La Hague, Pays-Bas, Septembre 1992.

[AKM93] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal Processing*, 31(2) :165–180, 1993.

[Ana89] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Int. Journal on Computer Vision*, 2(3) :283–310, Janvier 1989.

[AP95] N. Ancona and T. Poggio. Optical flow from 1d correlation : Application to a simple time-to-crash detector. *Int. Journal on Computer Vision*, 14(2) :131–146, Mars 1995.

[ASB94] S. Ayer, P. Schroeter, and J. Bigun. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In *Proc. European Conference on Computer Vision*, pages 316–327, Stockholm, Suède, Mai 1994.

[BA93] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. Int. Conf. on Computer Vision*, pages 231–236, Berlin, Allemagne, 1993.

[BA96] M.J. Black and P. Anandan. The robust estimation of multiple motions : Parametric and piecewise-smooth flow-fields. *Computer Vision and Image Understanding*, 63(1) :75–104, Janvier 1996.

[BB94] S. Beucher and M. Bilodeau. Road segmentation and obstacle detection by a fast watershed transform. In *Proc. IEEE Intelligent Vehicles*, pages 296–301, Paris, France, Octobre 1994.

[BB98] M. Bertozzi and A. Broggi. Gold : A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. on Image Processing*, 7(1) :62–81, Janvier 1998.

[BBC+92] M. Basseville, A. Benveniste, K.C. Chou, Golden, S. Nikoukhah, and A.S. R. Willsky. Modeling and estimation of multiresolution processes. *IEEE Trans. on Information Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis*, 38(2) :766–784, Mars 1992.

[BBF+00] A. Broggi, M. Bertozzi, A. Fascioli, C. Guarino Lo Bianco, and A. Piazzi. Visual perception of obstacles and vehicles for platooning. *IEEE Trans. on Intelligent Transportation Systems*, 1(3) :164–176, Septembre 2000.

[BBFC92] M. Barlaud, L. Blanc-Féraud, and P. Charbonnier. Image coding using multiresolution markov random fields. In *Proc. SPIE, Image Processing Algorithms and Techniques III*, San Jose, CA, Février 1992.

[BBFS00] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. Shape-based pedestrian detection. In *Proc. IEEE Intelligent Vehicles*, pages 196–201, Dearborn, MI, Octobre 2000.

[BD98] S.D. Buluswar and B.A. Draper. Color machine vision for autonomous vehicles. *Engineering Applications of Artificial Intelligence*, 11(2) :245–256, Avril 1998.

[Ber98] C. Bernard. Discrete wavelet analysis : A new framework for fast optic flow computation. In *Proc. European Conference on Computer Vision*, Freiburg, Allemagne, Juin 1998.

[Ber99] C. Bernard. *Ondelettes et Problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière.* PhD thesis, École Polytechnique, Novembre 1999.

[Bes74] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *RoyalStat*, B-36(2) :192–236, 1974.

[BF93] P. Bouthemy and E. Francois. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. Journal of Computer Vision*, 10(2) :157–182, Avril 1993.

[BF98] P. Bouthemy and R. Fablet. Motion characterization from temporal cooccurrences of local motion-based measures for video indexing. In *Proc. Int. Conf. on Pattern Recognition*, volume 1, pages 905–908, Brisbane, Australie, Août 1998.

[BFB94] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *Int. Journal on Computer Vision*, 12(1) :43–77, Février 1994.

[BHD00] M. Betke, E. Haritaoglu, and L.S. Davis. Real-time multiple vehicle detection and tracking from a moving vehicle. In *Machine Vision and Applications*, volume 12, pages 69–83, 2000.

[BK96] M.R. Banham and A.K. Katsaggelos. Spatially adaptive wavelet-based multiscale image restoration. *IEEE Trans. on Image Processing*, 5(4) :619–634, Avril 1996.

[BL90] P. Bouthemy and P. Lalande. Detection and tracking of moving objects based on a statistical regularization method in space and time. In *Proc. European Conference on Computer Vision*, pages 307–311, Antibes, France, Avril 1990.

[BL91] C. Bouman and B. Liu. Multiple resolution segmentation of textured images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2) :99–113, Février 1991.

[BL93] P. Bouthemy and P. Lalande. Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6) :1205–1212, Juin 1993.

[BL00] M. Beauvais and S. Lakshmanan. Clark : a heterogeneous sensor fusion method for finding lanes and obstacles. *Image and Vision Computing*, 18(5) :397–413, Avril 2000.

[Bla89] A. Blake. Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Trans. on Pattern Analanylsis and Machine Intelligence*, 11(1) :2–12, Janvier 1989.

[BP02a] E. Bruno and D. Pellerin. Robust motion estimation using spatial Gabor-like filters. *Signal Processing*, 82(2) :297–309, Février 2002.

[BP02b] E. Bruno and D. Pellerin. Video structuring, indexing and retrieval based on global motion wavelet coefficients. In *Proc. Int. Conf. on Pattern Recognition*, pages III : 287–290, 2002.

[Bér99] D. Béréziat. *Détection et suivi de structures déformables en mouvement, application à la météorologie.* PhD thesis, Université de Paris XI Orsay, Octobre 1999.

[BS94] C. Bouman and M. Shapiro. A multiscale random field model for bayesian image segmentation. *IEEE Trans. on Image Processing*, 3(2) :162–177, Mars 1994.

[CAC00] F. Chausse, R. Aufrere, and R. Chapuis. Recovering the 3D shape of a road by on-board monocular vision. In *Int. Conf. on Pattern Recognition*, pages Vol I : 325–328, Barcelone, Espagne, 2000.

[CB88] P.B. Chou and C.M. Brown. Multimodal reconstruction and segmentation with markov random fields and hcf optimization. In *Proc. Image Understanding Workshop*, pages 214–221, Cambridge Massachusetts, Avril 1988.

[CB90] P.B. Chou and C.M. Brown. The theory and practice of bayesian image labeling. *Int. Journal Computer Vision*, 4(3) :185–210, 1990.

[CD00] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8) :781–796, Août 2000.

[CEK+00] C. Curio, J. Edelbrunner, T. Kalinke, C. Tzomakas, and W. von Seelen. Walking pedestrian recognition. *IEEE Trans. Intelligent Transportation Systems*, 1(3) :155–163, Septembre 2000.

[Cha00a] B. Chalmond. *Eléments de modélisation pour l'analyse d'images.* Springer-Verlag, 2000.

[Cha00b] A. Chardin. *Modèles énergétiques hiérarchiques pour la résolution des problèmes inverses en analyse d'images-Application à la détection.* PhD thesis, Université de Rennes I, Janvier 2000.

[CM02] M. Clerc and S. Mallat. The texture gradient equation for recovering shape from texture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4) :536–549, Avril 2002.

[CMP02] T. Corpetti, E. Memin, and P. Pérez. Dense estimation of fluid flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3) :365–380, Mars 2002.

[Coh93] I. Cohen. Nonlinear variational method for optical flow computation. In *proc. Scandinavian Conference on Image Analysis*, pages 523–530, Tromsö, Norvège, Mai 1993.

[CP98] A. Chardin and P. Pérez. Semi-iterative inference with hierarchical models. In *Proc.Int. Conf. Image Processing*, volume 1, pages 630–634, Chicago, USA, Octobre 1998.

[Cre03] D. Cremers. A variational framework for image segmentation combining motion estimation and shape regularization. In *Proc Computer Vision and Pattern Recognition*, pages 53–58, Madison, Wisconsin, 2003.

[CT93] J.D. Crisman and C.E. Thorpe. SCARF : A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automation*, 9 :49–58, Février 1993.

[CTPPB98] C. Collet, Thourel, P. P. Pérez, and P. Bouthemy. Une nouvelle approche en traitement d'images sonar haute résolution : la segmentation markovienne hiérarchique multimodèle. *Traitement du signal*, 15(3) :231–250, 1998.

[CTS97] M.M. Chang, A.M. Tekalp, and M.I. Sezan. Simultaneous motion estimation and segmentation. *IEEE Trans. Image Processing*, 6(9) :1326–1333, Septembre 1997.

[Dau92] I. Daubechies. Ten lectures on wavelets. In *SIAM*, 1992.

[DHA88] G.W. Donohoe, D.R. Hush, and N. Ahmed. Change detection for target detection and classification in video images. In *Proc. Int. Conf. Acoust., Speech, Signal Processing*, New York, 1988.

[DJL92] R.A DeVore, B. Jawerth, and B.J. Lucier. Image compression through wavelet transform coding. *IEEE Trans. on Information Theory*, 38 :719–746, Mars 1992.

[DJLJ96] M.P. Dubuisson-Jolly, S. Lakshmanan, and A.K. Jain. Vehicle segmentation and classification using deformable templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(3) :293–308, Mars 1996.

[DKA95] R. Deriche, P. Kornprobst, and G. Aubert. Optical flow estimation while preserving its discontinuities : A variational approach. In *Proc. Asian Conf. Computer Vision*, Singapour, Décembre 1995.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications for the Association for Computing Machinery*, 24(6) :381–395, Juin 1981.

[FBG99] R. Fablet, P. Bouthemy, and M. Gelgon. Moving object detection in color image sequences using region-level graph labeling. In *Proc. Int Conf. on Image Processing*, Kobe, Japon, Octobre 1999.

[FBP02] R. Fablet, P. Bouthemy, and P. Pérez. Nonparametric motion characterization using causal probabilistic models for video indexing and retrieval. *IEEE Trans. Image Processing*, 11(4) :393–407, Avril 2002.

[FJ90] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *Int. Journal Computer Vision*, 5(1) :77–104, Août 1990.

[FWSB95] J. Ferryman, A. Worall, G. Sullivan, and K. Baker. A generic deformable model for vehicle recognition. In *British Machine Vision Conference*, Birmingham, 1995.

[Gav00] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proc. European Conference on Computer Vision*, pages II : 37–49, Dublin, Irlande, Juin 2000.

[GB00] M. Gelgon and P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33(4) :725–740, Mars 2000.

[GCT94] A. Giachetti, M. Campani, and V. Torre. The use of optical flow for the autonomous navigation. In *Europ. Conf. on Computer Vision*, volume 1, pages 146–151, Stockholm, Suède, Mai 1994.

[GDW96] C. Goreick, N. Detlev, and M. Werner. Artificial neural networks in real-time car detection and tracking applications. *Pattern Recognition Letters*, 17 :335–343, 1996.

[Gel98] M. Gelgon. *Segmentation spatio-temporelle et suivi dans une séquence d'images : application à la structuration et à l'indexation vidéo*. PhD thesis, Université de Rennes 1, Décembre 1998.

[GG84] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, Novembre 1984.

[GHP+97] C. Graffigne, F. Heitz, P. Pérez, F. Prêteux, M. Sigelle, and J. Zerubia. Hierarchical and statistical models applied to image analysis : a review. *rapport GDR-ISIS*, 1997.

[Gid89] B. Gidas. A renormalization group approach to image processing problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(2) :164–180, Février 1989.

[GJ95] S. N. Gupta and Prince J. On variable brightness optical flow for tagged mri. In *Information Processing in Medical Imaging*, pages 323–334, Juin 1995.

[Hac85] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, 1985.

[HB93] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(12) :1217–1232, Décembre 1993.

[HB96] R.J. Howarth and H. Buxton. Visual surveillance monitoring and watching. In *Proc. European Conference on Computer Vision*, volume 2, pages 321–334, Cambridge, UK, Avril 1996.

[HCAdlE04] C. Hilario, J.M. Collado, J.M. Armingol, and A. de la Escalera. Driver assistance system based on computer vision for vehicle detection. In *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisbonne, Portugal, Juillet 2004.

[Hee88] D.J. Heeger. Optical flow from spatiotemporal filters. *Int. Journal Computer Vision*, 1(4) :279–302, Janvier 1988.

[HJ83] S.M. Haynes and R. Jain. Detection of moving edges. *Computer Vision Graphics and Image Processing*, 21 :345–367, 1983.

[HNR84] Y.Z. Hsu, H.H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision Graphics and Image Processing*, 26(1) :73–106, Avril 1984.

[HPB94] F. Heitz, P. Pérez, and P. Bouthemy. Multiscale minimization of global energy functions in some visual recovery problems. *Computer Vision Graphics and Image Processing : Image Understanding*, 59(1) :125–134, Janvier 1994.

[HRB96] C. Hennebert, V. Rebuffel, and P. Bouthemy. A hierarchical approach for scene segmentation based on 2D motion. In *Proc. Int. Conf. Pattern Recognition*, pages 218–222, Vienne, Août 1996.

[HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[Hub81] P.J. Huber. *Robust Statistics*. J.Wiley and Sons, 1981.

[HW77] P.W. Holland and R.E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics : Theory and Methods*, A6 :813–827, 1977.

[IRP92] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. European Conference on Computer Vision*, pages 282–287, Santa Margherita Ligure, Italy, Mai 1992.

[Joh73] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Physics*, 14 :201–211, 1973.

[KBZ93] Z. Kato, M. Berthod, and J. Zerubia. Multiscale markov random field models for parallel image classification. In *Proc. Int. Conf. on Computer Vision*, pages 253–257, Mai 1993.

[KD92] J. Konrad and E. Dubois. Bayesian estimation of motion vector fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(9) :910–927, Septembre 1992.

[KD96] J. Konrad and V.N. Dang. Coding-oriented video segmentation inspired by MRF models. In *Proc. Int. Conf. Image Processing*, pages 909–912, Lausanne, Suisse, Septembre 1996.

[KER95] W. Kruger, W. Enkelmann, and S. Rossle. Real-time estimation and tracking of optical flow vectors for obstacle detection. In *Proc. IEEE Intelligent Vehicles*, pages 304–309, Détroit, Septembre 1995.

[KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13*, 220 :671–680, Mai 1983.

[KL95] K.C. Kluge and S. Lakshmanan. A deformable template approach to lane detection. In *Proc. IEEE Intelligent Vehicles*, Détroit, Septembre 1995.

[Kue98] A. Kuehnle. Symmetry-based vehicle location for AHS. In *Proc. SPIE - Transportation Sensors and Controls : Collision Avoidance, Traffic Management, and ITS*, volume 2902, pages 19–27, Boston, MA, Novembre 1998.

[KvB90] K.P. Karmann and A. von Brandt. Moving object recognition using an adaptive background memory. *Time-Varying Image Processing and Moving Object Recognition 2,V. Cappellini, editor, Elsevier Science Publishers*, pages 289–296, 1990.

[KZK03] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis. A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21(4) :359–381, Avril 2003.

[LAT02] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection on non flat road geometry through 'v-disparity' representation. In *Proc. IEEE Intelligent Vehicles*, Versailles, France, 2002.

[LCL99] F. Luthon, A. Caplier, and M. Lievin. Spatiotemporal MRF approach to video segmentation : Application to motion detection and lip segmentation. *IEEE Trans. Signal Processing*, 76(1) :61–80, Juillet 1999.

[LHPF95] J.M. Laferté, F. Heitz, P. Pérez, and E. Fabre. Hierarchical statistical models for the fusion of multiresolution image data. In *Proc. Int. Conf. Computer Vision*, pages 908–913, Cambridge, Massachusetts, Juin 1995.

[LK81] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc IJCAI*, pages 674–679, Vancouver, B.C., Canada, 1981.

[LKW94] M.R. Luettgen, W.C. Karl, and A.S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Trans. Image Processing*, 3(1) :41–64, 1994.

[LMB02] G. Lefaix, E. Marchand, and P. Bouthemy. Motion-based obstacle detection and tracking for car driving assistance. In *Proc. Int. Conf. on Pattern Recognition*, volume 4, pages 74–77, Québec, Canada, Août 2002.

[LPH00] J.M. Laferte, P. Pérez, and F. Heitz. Discrete markov image modeling and inference on the quadtree. *IEEE Trans. Image Processing*, 9(3) :390, Mars 2000.

[LRB93] J.M. Létang, V. Rebuffel, and P. Bouthemy. Motion detection robust to perturbations : A statistical regularization and temporal integration framework. In *Proc Int. Conf. Computer Vision*, pages 21–30, Berlin, Allemagne, 1993.

[Lue93] M.R. Luettgen. *Image processing with multiscale stochastic models.* PhD thesis, MIT Laboratory of Information and Decision Systems, Mai 1993.

[LWKM95] Q.T. Luong, J. Weber, D. Koller, and J. Malik. An integrated stereo-based approach to automatic vehicle guidance. In *Int. Conf. on Computer Vision*, pages 52–57, Cambridge, USA, Juin 1995.

[Mal89] S.G. Mallat. A theory for multiresolution signal decomposition : The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7) :674–693, Juillet 1989.

[MB94] F.G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP : Image Understanding*, 60(2) :119–140, Septembre 1994.

[MBK98] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatiotemporal segmentation based on region merging. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(9) :897–915, September 1998.

[Mey94] F. Meyer. Time-to-collision from first-order models of the motion field. *IEEE Trans on Robotics and Automation*, 10(6) :792–798, Décembre 1994.

[MK98] J. Magarey and N. Kingsbury. Motion estimation using a complex-valued wavelet transform. *IEEE Trans. on Signal Processing*, 46(4) :1069–1084, Avril 1998.

[MMKR91] P. Meer, D. Mintz, D.Y. Kim, and A. Rosenfeld. Robust regression methods for computer vision : A review. *Int. Journal on Computer Vision*, 6(1) :59–70, Avril 1991.

[MN94] M. Mattavelli and A. Nicoulin. Motion estimation relaxing the constant brightness constraint. In *Proc. Int. Conf. Image Processing*, pages II : 770–774, Novembre 1994.

[Mor81] J. Morlet. Sampling theory and wave propagation. In *Proc. 51st. Annual Meeting Society Exploration Geophysicists*, Los Angeles, 1981.

[MP98] E. Mémin and P. Pérez. Joint estimation-segmentation of optic flow. In *Proc. European Conference on Computer Vision*, pages 563–580, Freiburg, Germany, Juin 1998.

[MP02] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *Int Journal Computer Vision*, 46(2) :129–155, February 2002.

[Nag83] H.H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21(1) :85–117, January 1983.

[NWD00] H.T. Nguyen, M. Worring, and A. Dev. Detection of moving objects in video using a robust motion similarity measure. *IEEE Trans. on Image Processing*, 9(1) :137–141, Janvier 2000.

[NY93] S. Negahdaripour and C.H. Yu. A generalized brightness change model for computing optical flow. In *Proc. Int. Conf. Computer Vision*, pages 2–11, Berlin, Germany, 1993.

[OB94] J.M. Odobez and P. Bouthemy. Detection of multiple moving objects using multiscale Markov random fields with camera compensation. In *Proc. Int. Conf. Image Processing*, pages 257–261, Austin, Texas, USA, Novembre 1994.

[OB95] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4) :348–365, Decembre 1995.

[OB98] J.M. Odobez and P. Bouthemy. Direct incremental model based image motion segmentation for video analysis. *Signal Processing*, 66(2) :143–155, Avril 1998.

[Odo94] J.M. Odobez. *Estimation, détection et segmentation du mouvement : une approche robuste et markovienne.* PhD thesis, Université de Rennes I, Décembre 1994.

[PCL00] P. Pérez, A. Chardin, and J.M. Laferte. Noniterative manipulation of discrete energy-based models for image analysis. *Pattern Recognition*, 33(4) :573–586, Avril 2000.

[PCP99] J.N. Provost, C. Collet, and P. Pérez, P. Bouthemy. A hierarchical unsupervised multispectral model to segment spot images for ocean cartography. In *Proc. Int. Conf. Image Processing*, Octobre 1999.

[PD97] N. Paragios and R. Deriche. Detecting multiple moving targets using deformable contours. In *IEEE International Conference on Image Processing*, volume I, pages 183–186, Octobre 1997.

[PPT+96] N. Paragios, P. Pérez, G. Tziritas, C. Labit, and P. Bouthemy. Adaptative detection of moving objects using multiscale techniques. In *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, Septembre 1996.

[Pér93] P. Pérez. *Champs markoviens et analyse multirésolution de l'image : application à la détection du mouvement.* PhD thesis, Université de Rennes I, Juillet 1993.

[RA97] S. Rakshit and C.H. Anderson. Computation of optical-flow using basis functions. *IEEE Trans. Image Processing*, 6(9) :1246–1254, Septembre 1997.

[Ras04] C. Rasmussen. Texture-based vanishing point voting for road shape estimation. In *British Machine Vision Conference*, London, UK, Septembre 2004.

[RL87] P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection.* J.Wiley and Sons, 1987.

[Ros02] P.L. Rosin. Thresholding for change detection. *CVIU*, 86(2) :79–95, Mai 2002.

[Rou84] P.J. Rousseeuw. Least median of squares regression. *Journal of American Statistics Association*, 79 :871–880, 1984.

[SBM02] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection using Gabor filters and support vector machines. In *IEEE International Conference on Digital Signal Processing*, Santorini, Greece, Juillet 2002.

[Sin90] A. Singh. An estimation-theoretic framework for image-flow computation. In *Proc. Int. Conf. Computer Vision*, Osaka, Japan,, 1990.

[SK98] H. Schneiderman and T. Kanade. Probabilistic formulation for object recognition. In *Proc. Computer Vision and Pattern Recognition*, pages 45–51, Santa Barbara, Juin 1998.

[ST01] E. Sifakis and G. Tziritas. Moving object localisation using a multi-label fast marching algorithm. *Signal Processing : Image Communication*, 16(10) :963–976, 2001.

[SYNK92] A. Suzuki, N. Yasui, N. Nakano, and M. Kaneko. Lane recognition system for guiding of autonomous vehicle. In *Proc. IEEE Intelligent Vehicles*, Detroit, USA, 1992.

[TM93] P.H.S. Torr and D.W. Murray. Statistical detection of independent movement from a moving camera. *IVC*, 11 :180–187, 1993.

[TMGM88] M.A. Turk, D.G. Morgenthaler, K.D. Gremban, and M. Marra. VITS–a vision system for autonomous land vehicle navigation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(3) :342–361, Mai 1988.

[TP90] W.B. Thompson and T.C. Pong. Detecting moving objects. *Int. Journal on Computer Vision*, 4(1) :39–58, Janvier 1990.

[Uns95] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Trans. Image Processing*, 4(11) :1549–1560, Novembre 1995.

[WA94] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3(5) :625–638, Septembre 1994.

[WGK04] F. Woelk, S. Gehrig, and R. Koch. A monocular image based intersection assistant. In *IEEE Intelligent Vehicles Symposium*, Parme, Italie, Juin 2004.

[WM95] J. Weber and J. Malik. Robust computation of optical-flow in a multiscale differential framework. *Int. Journal Computer Vision*, 14(1) :67–81, January 1995.

[WST00] Y. Wang, D. Shen, and E.K. Teoh. Lane detection using spline model. *Pattern Recognition Letters*, 21(6-7) :677–689, Juin 2000.

[XG94] W. Xiong and C. Graffigne. A hierarchical method for detection of moving objects. In *IEEE International Conference on Image Processing*, pages 795–799, Austin, Texas, Novembre 1994.

[YC00] A.L. Yuille and J.M. Coughlan. Fundamental limits of Bayesian inference : Order parameters and phase transitions for road tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(2) :160–173, Février 2000.

[ZN94] J. Zhang and H.H. Nagel. Texture-based segmentation of road images. In *IEEE Intelligent Vehicles*, pages 260–265, Paris, Octobre 1994.

Etude du mouvement dans les séquences d'images par analyse d'ondelettes et modélisation markovienne hiérarchique. Application à la détection d'obstacles dans un milieu routier.

Résumé : Ce travail a pour objectif de détecter les obstacles sur la route à partir d'une caméra montée sur un véhicule. Étant donné les contraintes que nous nous sommes fixées, un état de l'art des différentes méthodes existantes sur ce sujet montre que seule une analyse du mouvement dans les séquences d'images peut résoudre ce problème. En effet, cette méthode doit permettre de détecter tout type d'obstacles sur tout type de routes avec une seule caméra.
Pour ce faire, nous avons développé une nouvelle méthode d'estimation de mouvement paramétrique par analyse d'ondelettes de l'équation de contrainte du flot optique (E.C.M.A.). Cette méthode multi-résolution nous a permis de contourner le problème de l'aliasage temporel inhérent à l'E.C.M.A.. Nous avons ensuite proposé de résoudre le problème de la détection de mouvement dans une scène filmée à partir d'une caméra mobile à l'aide d'une modélisation markovienne hiérarchique déduite de façon naturelle de l'estimation multi-résolution du mouvement. Puis, nous avons introduit une méthode de segmentation au sens du mouvement entre deux images sans connaissance *a priori* et sans hypothèse de présence de mouvement dominant grâce à un raffinement successif de la segmentation d'une échelle grossière de l'image à l'échelle la plus fine. Chaque méthode (estimation, détection et segmentation) a été validée expérimentalement sur des séquences synthétiques et réelles. Enfin, celles-ci ont été adaptées au problème concret visé par cette thèse : la détection d'obstacles dans un milieu routier. L'utilisation d'ondelettes et de champs de Markov hiérarchiques aboutit à des solutions peu coûteuses en temps de calcul.

Mots Clés : Estimation du mouvement/ Flot optique/ Détection de mouvement/ Segmentation au sens du mouvement/ Analyse d'ondelettes/ Modèles markoviens hiérarchiques.

Motion analysis in image sequence using wavelet analysis and hierarchical Markovian model. Application to obstacle detection in road sequences.

Abstract : The purpose of this work is to detect the obstacles on the road thanks to a camera mounted on a vehicle. Due to fixed constraints, an overview of various existing methods has shown that only motion analysis in the image sequence can solve this problem. Indeed, this method must permit to detect all types of obstacles on all kinds of roads with only one camera.
For that, we have developed a method of parametric motion estimation by wavelet analysis. This multi-resolution method enabled us to overcome the aliasing problem. Then, we have proposed to solve the motion detection problem in a scene filmed with a mobile camera using a hierarchical Markovian modeling deduced in a natural way of the multi-resolution motion estimation. Next, we have introduced a motion segmentation method between two images without *a priori* knowledge and without hypothesis of dominant motion thanks to a successive refinement of the segmentation from coarse scale to fine scale of the image. All methods (estimation, detection and segmentation) have been validated in experiments on synthetic and real sequences. Lastly, those were adapted to the concrete problem aimed by this thesis : the obstacle detection in road sequences. The use of wavelet and hierarchical Markovian model leads to inexpensive solutions in computing times.

Keywords : Motion estimation/ Optical Flow/ Motion detection/ Motion segmentation/ Wavelet analysis/ Hierarchical Markovian models.