



UNIVERSITE IBN ZOHR

CENTRE DES ETUDES DOCTORALES IBN ZOHR

Formation doctorale :

Mathématiques, Informatique et applications

ETABLISSEMENT :

FACULTE DES SCIENCES - AGADIR

THESE

Présentée par

Khalid HOUSNI

pour l'obtention de grade de
DOCTEUR de l'Université Ibn Zohr

Spécialité : Informatique

Analyse d'événements dans une vidéo par les coupes de graphes

Soutenu Le 22 décembre 2012

Devant la commission d'examen composée de :

M. Fattehallah GHADI,	Vice Président de l'Université Ibn Zohr, Président de jury
M. Driss MAMMASS,	Directeur de l'EST d'Agadir, Directeur de thèse
M. Youssef CHAHIR,	HDR à l'Université de Caen, France, Co-directeur de thèse
M. Elhassane IBN-ELHAJ,	Professeur à l'Ecole Nationale des Postes et Télécommunications, Rapporteur
M. Hassan DOUZI,	Professeur à la Faculté des Sciences d'Agadir, Rapporteur
M. Ahmed EL OIRRAK,	Professeur Habilité à la Faculté des Sciences Semlalia, Marrakech, Rapporteur
M. Mohamed WAKRIM,	Professeur à la Faculté des Sciences d'Agadir, Examineur

Titre : Analyse d'événements dans une vidéo par Coupes de graphes

Résumé

Cette thèse s'intéresse à l'analyse de vidéo et plus particulièrement :

- à la détection de mouvement
- au suivi d'objets en mouvement
- détection de mouvements cohérents et extraction de couches de même mouvement.

Dans la première partie, deux nouvelles approches ont été proposées. Nous avons proposé une nouvelle méthode automatique pour la détection d'objets en mouvement basée sur une modélisation des interactions spatio-temporelles entre étiquettes (arrière plan et objets mobiles) par un champ de Markov. Puis, une nouvelle approche de suivi d'objets en mouvement a été proposée. Cette méthode se décompose en trois étapes principales. La première étape consiste à déterminer pour chaque objet le modèle de mouvement affine caractérisant son déplacement. La deuxième étape consiste à prédire les objets à suivre dans l'image suivante en utilisant les modèles de mouvement calculés dans l'étape précédente. La dernière étape consiste à raffiner les frontières des objets suivis par une méthode de segmentation dynamique. Une quatrième étape est ajoutée dans le cas de chevauchement des objets suivis. Cette dernière étape est basée sur la classification des pixels des objets qui se chevauchent par la méthode des coupes de graphes (algorithme alpha-expansion).

La deuxième partie du document a traité le problème de détection de mouvements cohérents et d'extraction de couche de même mouvement. Dans un premier temps, nous avons présenté une nouvelle approche de détection de mouvements cohérents. Cette méthode se décompose en trois étapes principales.

- La première étape consiste à détecter et à mettre en correspondance les points d'intérêt.
- La deuxième étape consiste à grouper les points d'intérêt spatio-temporellement connexes. Cette dernière étape permet de regrouper les points d'intérêt en fonction de leurs vecteurs de déplacement locaux, donc incapable de regrouper les points distants (non connexes) ayant le même mouvement, ainsi incapable de regrouper les points dont les vecteurs de déplacement locaux sont différents et qui peuvent être modélisés par un seul modèle de mouvement paramétrique.
- Dans la troisième étape, afin de résoudre le problème de sur-groupement, nous avons utilisé un algorithme permettant de fusionner les groupes dont le modèle du mouvement calculé à partir de l'ensemble des points de ces groupes supporte plus que 85% des points de chaque groupe via un processus itératif.

L'approche précédente de détection de mouvements cohérents permet d'obtenir le nombre de mouvements cohérents ainsi que leurs modèles de mouvement. Ces informations sont ensuite utilisées afin de segmenter la scène en couches de même mouvement

Pour réaliser cette recherche, nous avons utilisé les descripteurs intra-images (tels que la géométrie, le mouvement, la forme, la couleur ...) ainsi les descripteurs inter-images (tels que la trajectoire). Nous avons utilisé aussi des techniques basées sur les coupes de graphes.

Mots clés : Détection mouvements, suivi d'objets en mouvement, mouvements cohérents, couches de même mouvement, vidéo, coupes de graphes.

Title: Analysis of events in a video by Graph Cuts

Abstract

This thesis focuses on the analysis of video and more particularly:

- motion detection
- moving objects tracking
- coherent motion detection and motion layers extraction of similar motion.

In the first part, two new approaches have been proposed. We have proposed a new method for automatic detection of moving objects based on modelling of spatio-temporal interactions between labels (background and moving objects) by a Markov random field. Then a new approach of tracking of moving objects has been proposed. This method consists of three main steps. The first step consists in determining for each object the affine motion model characterizing its displacement. The second step consists to predict the objects to be tracked in the following frame using motion models calculated in the previous step. The last step consists to refine the boundaries of objects tracked by a method of dynamic segmentation. A fourth step is added in the case of overlapping of the tracked objects. This last step is based on the classification of the pixels of overlapping objects by the alpha-expansion algorithm.

The second part treated the problem of coherent motion detection and motion layer extraction of the similar motion. As a first step, we presented a new approach to coherent motion detection. This method consists of three main steps.

- The first step consists to detect and matching the points of interest.
- The second step consists to group points of interest spatio-temporally connected. This step allows us to group points of interest based on their local displacement vectors. However, the detection of coherent motion is not reliable except for continuous areas. Consequently, the detection of coherent motion is over-segmentation in some cases, especially in regions where the points of interest are spatiotemporally discontinuous.
- In the third step, in order to solve the problem of over-group, we used an algorithm to merge the groups whose movement model calculated from all points of these groups supports more than 85% points of each group via an iterative process.

The previous approach of temporal coherent motion detection provides the number of coherent movements and their motion models. This information is then used to segment the scene into layers of the similar motion.

To realize this research, we used intra-frame descriptors (such as geometry, motion, shape, color ...) and inter-frame descriptors (such as path). We have also used techniques based on graph cuts.

Keywords : Motion detection, moving objects tracking, coherent motion, layers of the similar motion, video, graph cuts.

Remerciements

Je tiens à exprimer ma reconnaissance la plus sincère à mes directeurs de thèse professeur Driss MAMMASS et professeur Youssef CHAHIR, pour la confiance qu'ils m'ont accordée et la qualité de leurs encadrements. Leurs soutiens et leurs disponibilités m'ont permis de m'épanouir sereinement tout au long de mes travaux. Ce fut une réelle chance de travailler avec eux.

J'exprime ma profonde reconnaissance à l'égard de tous les membres du jury. Je remercie en particulier le professeur Fattehallah GHADI de me faire l'honneur de présider ce jury.

Conscient de la charge que cela représente, j'adresse mes profonds remerciements aux professeurs Elhassane IBN-ELHAJ, Hassan DOUZI et Ahmed El OIRRAK d'avoir accepté de rapporter ces travaux ainsi que le professeur Mohamed WAKRIM qui a accepté de faire partie du jury de thèse.

Je tiens à remercier l'ensemble des secrétaires du C.E.D et tout particulièrement Madame la Directrice L. Mina IDRISSE HASSANI pour sa gentillesse et sa disponibilité.

Pour continuer dans l'environnement du travail, je tiens à remercier les thésards ainsi que les étudiants du laboratoire IRF – SIC qui m'ont chaleureusement accueilli au sein de l'équipe. Je remercie en particulier Nouh Izem, Mohamed EL Hajji, Youssef Essaady, Mustapha Amrouche, Taher Zaki, Mohamed Ben Adi, Ali Elmezouari ... qui m'ont accompagné et aidé avec leur esprit à la fois solidaire et critique. Leur soutien infini, leur amitié et pour les discussions sur plein de sujets.

Maintenant vient le tour de ceux qui me supportent depuis le début (et même avant), ma famille. Merci à mes parents de m'avoir toujours laissé libre dans mes choix tout en me soutenant et me conseillant constamment. Merci à mes frères, à mes sœurs ainsi que à mon beau frère.

Table des Matières

Introduction générale.....	13
Liste des contributions	14
<i>Première partie : Détection et suivi d'objets en mouvement</i>	
Introduction de la première partie	16
Chapitre 1 : Coupe minimale/flot maximal dans un graphe	18
1.1 Notations et définitions	18
1.2 Calcul du flot maximal (coupe minimale).....	21
1.2.1 Flot maximal par saturation de chemins	22
1.2.2 Flot maximal par poussage de flot	23
1.2.3 Algorithmes de « Dynamic Graph Cuts » et « Active Cuts »	24
1.3 Equivalence étiquetage/minimisation d'énergie	24
1.3.1 Champs aléatoires de Markov (<i>souvent abrégé par MRF</i>)	24
1.3.2 Maximum <i>A Posteriori</i> (MAP)	25
1.4 Minimisation d'énergie par les coupes de graphe.....	26
1.4.1 Cas d'une classification binaire.....	26
1.4.2 Cas d'une classification multi-étiquettes.....	29
(α,β)-swap.....	30
α -expansion.....	32
1.5 Autres approches de minimisation	33
1.5.1 Le recuit simulé.....	33
1.5.2 Les modes conditionnels itérés (ICM)	34
1.5.3 La propagation bouclée de croyances ou « Loopy Belief Propagation » (LBP)..	34
1.5.4 Tree-ReWeighted Message Passing (TRW)	34
1.6 Prise en compte des contraintes fortes par les coupes de graphe.....	34
1.7 Conclusion.....	38
Chapitre 2 : Détection de mouvement.....	40
2.1 Représentation des données vidéo.....	40
2.2 Etat de l'art des méthodes de détection basées sur la différence inter-images	41
2.2.1 Décision sur un pixel.....	41
2.2.2 Décision basée région.....	44

2.3	Conclusion et perspective sur les méthodes de l'état de l'art	45
2.4	Approche proposée.....	45
2.5	Conclusion.....	51
Chapitre 3 : Etat de l'art des méthodes de suivi d'objets en mouvement.....		52
3.1	Suivi de points.....	53
3.2	Suivi de silhouettes.....	54
3.2.1	Approches explicites	55
3.2.2	Approches implicites.....	55
3.3	Approches basées sur la correspondance de blocs.....	55
3.4	Suivi de noyaux.....	57
3.5	Conclusion.....	58
Chapitre 4 : Une nouvelle approche de suivi d'objets en mouvement basée sur les coupes de graphes		59
4.1	Principe.....	59
4.2	Sélection des objets à suivre.....	61
4.3	Modèles de mouvement	62
4.4	Raffinement des frontières des objets prédits	64
4.5	Gestion des problèmes d'occultation	66
4.6	Résultats Expérimentaux.....	67
4.6.1	Données.....	67
4.6.2	Résultats et Complexité.....	67
4.7	Conclusion et perspectives	71
<i>Deuxième partie : Détection de mouvements cohérents et extraction de couches de même mouvement</i>		
Introduction de la deuxième partie.....		73
chapitre 5 : Etat de l'art des méthodes de détection des points d'intérêt, d'estimation de mouvement et de détection de mouvements cohérents		74
5.1	Détection et mise en correspondance des points d'intérêt	75
5.1.1	Détection des points d'intérêt.....	75
5.1.1.1	Détection de coins	75
5.1.1.1.1	Le détecteur de Harris [Harris 88].....	75
5.1.1.1.2	Extraction et suivi de points caractéristiques KLT	77

5.1.1.2	Détection d'extrema locaux : descripteurs SIFT	78
5.1.2	Appariements	80
5.1.2.1	Descripteurs basés sur les intensités.....	80
5.1.2.2	Descripteurs basés sur l'orientation	80
5.1.3	Conclusion.....	81
5.2	Estimation du mouvement.....	81
5.2.1	Modèles de mouvement	82
5.2.1.1	Modélisation locale de mouvement (flot optique)	82
5.2.1.2	Modélisation de mouvement par région.....	84
5.2.1.2.1	Le modèle affine.....	84
5.2.1.2.2	Le modèle projectif	84
5.2.1.2.3	Modèle retenu.....	85
5.2.2	Estimation du mouvement paramétrique.....	85
5.2.2.1	Méthode utilisant une paire d'images distantes	85
5.2.2.2	Détails sur la méthode de Dupont et al. [Dupont 06].....	85
5.2.2.3	Estimation de mouvement paramétrique à partir des mesures de flot optique. 86	
5.2.2.4	Estimation de mouvement via les points d'intérêt et la régression plane 3D .. 87	
5.3	Détection de mouvements cohérents.....	88
5.3.1	Etat de l'art	88
5.3.2	Conclusion.....	90
Chapitre 6 : Nouvelle approche d'estimation de mouvements cohérents et d'extraction de couches de même mouvement		91
6.1	Détection de mouvements cohérents.....	91
6.1.1	Regroupement local.....	92
6.1.2	Fusion des groupes cohérents en sens de mouvements paramétriques	94
6.1.2.1	Calcul du modèle de mouvement	95
6.1.2.2	Algorithme de fusionnement de paires de groupes	95
6.2	Extraction de couches de même mouvement	97
6.2.1	Définition	97
6.2.2	Etat de l'art	98
6.2.3	Extraction des supports des couches	98
6.2.3.1	Formulation du problème	98

6.2.3.2	Minimisation d'énergie	100
6.3	Résultats et expérimentations.....	100
6.3.1	Données.....	100
6.3.2	Résultats et complexité.....	100
6.4	Conclusion.....	102
	Conclusion générale et perspectives	107
a)	Synthèse des travaux effectués.....	107
b)	Perspectives.....	108
	<i>ANNEXE : Régression plane 3D</i>	<i>110</i>
A.1	Introduction	111
A.2	Moindres carrés des distances vraies, méthode directe (non itérative)	112
A.3	Discussion sur le choix de l'équation $Ax+By+Cz+1=0$	117
	Bibliographie.....	118

Table des Figure

Figure 1.1 - Exemple de graphe et de coupe.....	21
Figure 1.2 - Généralisation au cas multi-étiquette.....	22
Figure 1.3 - Les coupes possibles dans le cas de la classification binaire de deux variables.....	28
Figure 1.4 - Construction du graphe pour une classification multi-étiquettes.....	30
Figure 1.5 - La mise en graphe de l'algorithme (α, β) -swap.....	31
Figure 1.6 - La mise en graphe de l'algorithme α -expansion.....	33
Figure 1.7 - Cette figure, histogrammes des germes d'objet et de fond, illustre comment les termes $-\ln\Pr(I_p \text{"fond"})$ et $-\ln\Pr(I_p \text{"objet"})$ seront calculés.....	36
Figure 1.8 - Exemple de segmentation d'une image 3x3 avec une connectivité V8.....	36
Figure 1.9 - Cas de l'extraction des objets d'intérêts avec la prise en compte des propriétés régionales $\lambda=1$. Les graines d'objet sont marquées en bleu et ceux de fond sont en rouge.....	37
Figure 1.10 - Résultats obtenus pour IRM médicale.....	37
Figure 1.11 - Amélioration des résultats de SVM par les coupes de graphes.....	39
Figure 2.1 - Exemple de séquence vidéo avec sa représentation en 3D.....	41
Figure 2.2 - Détection de mouvement basée sur la différence d'images.....	43
Figure 2.3 - Détection de changements par différence avec une image de fond.....	44
Figure 2.4 - Allures des fonctions à utiliser pour le calcul des poids des arêtes t-link.....	46
Figure 2.5 - Différents types d'informations qui peuvent être prises comme observation.....	47
Figure 2.6 - Détection de point de changement dans l'histogramme cumulé ainsi la valeur à partir de laquelle l'histogramme devient constant.....	47
Figure 2.7 - La probabilité d'appartenance à l'objet ou au fond en fonction de différence d'intensité.....	48
Figure 2.8 - Détection de mouvement par l'utilisation de la différence d'images successives.....	48
Figure 2.9 - Comparaison des résultats de la détection de mouvement par l'utilisation de la différence entre trois images successives avec ceux obtenus par l'utilisation de différence entre l'image de référence et l'image I_t	49
Figure 2.10 - Détection de mouvement dans les séquences de trafic routier.....	50
Figure 3.1 - Représentations d'objet [Yilmaz 06].....	53
Figure 3.2- Evolution du contour actif vers l'objet d'intérêt.....	56
Figure 3.3 - Dilatation du contour pour obtenir l'emplacement de la source et du puits. Cette figure provient de [Xu 02].....	56

Figure 3.4 - Exemples de deux histogrammes multi-échelles [Mikram 06].....	57
Figure 4.1 - Schéma général de notre approche.....	60
Figure 4.2 – L’interface utilisée pour la saisie de l’information initiale.....	61
Figure 4.3 - Connexité et voisinage	62
Figure 4.4 - Le suivi d’objets en mouvement par l’utilisation de modèle de mouvement.....	64
Figure 4.5 – Résultat d’une prédiction par le modèle de mouvement.....	64
Figure 4.6 – Génération automatique des grains d’objet et de fond.	65
Figure 4.7 - Détection optimale des contours par les coupes de graphe	65
Figure 4.8 - Comparaison de résultat obtenu en utilisant le modèle de mouvement uniquement avec celui obtenu après le raffinement par des coupes de graphe.....	66
Figure 4.9 – Résultats statistiques obtenus indiquant le nombre de pixels détectés dans la première étape (suivi à l'aide du modèle de mouvement uniquement), le nombre de pixels ajoutés et rejetés par l’étape de raffinement.....	68
Figure 4.10 - Cette figure montre le changement de la taille des objets suivis.....	68
Figure 4.11 - Détection de mouvement dans la séquence <i>rond-point</i>	69
Figure 4.12 - Détection de mouvement dans les séquence croisement, walk, croisement et Coastguarder.....	70
Figure 4.13 - Comparaison des résultats de suivi d'objet sur la séquence <i>walk</i>	70
Figure 5.1 – La recherche d’image par la détection et la mise en correspondance des points d’intérêt SIFT (résultat obtenu par le logiciel SiftDimo).....	74
Figure 5.2 - Cartes de Harris pour différentes échelles.....	77
Figure 5.3 – Le voisinage (3x3x3) utilisé pour la recherche des extrema.	78
Figure 5.4 – Exemple des points d’intérêt.....	79
Figure 5.5 – Exemple d'ambiguïté inhérente à l'estimation du mouvement	83
Figure 5.6 - Calcul des paramètres du modèle de mouvement affine par la régression plane 3D	88
Figure 5.7 - Exemple de limitation à la détection de mouvements cohérents dans les séquences acquises avec une caméra en mouvement.....	90
Figure 6.1- Cette figure illustre les différents paramètres utilisés pour la détermination du voisinage spatio-temporel et pour le calcul de la connectivité.....	93
Figure 6.2 - Détection de mouvements cohérents.....	96
Figure 6.3 - Exemple de décomposition en couches.....	97
Figure 6.4 - Cette figure donne les pourcentages de recouvrement de mouvement des pixels de chaque couche par les modèles de mouvement calculés.....	101
Figure 6.5 – Comparaison des résultats de segmentation sur la séquence <i>flower</i>	103

Figure 6.6 – Résultats de la détection de mouvements cohérents et d'extraction de couches de même mouvement. Cas de la séquence *flowers* images 1- 8-14 -29 et 35..... 104

Figure 6.7 – Résultats de la détection de mouvements cohérents et d'extraction de couches de même mouvement. Cas de la séquence *calendar* images 5, 12, 24 et 35. 105

Figure 6.8 – Résultats de la détection de mouvements cohérents et d'extraction de couches de même mouvement. Cas de la séquence *taxi* images 5, 6, 7 et 22..... 106

Introduction générale

La vidéo est un média qui pose des problèmes complexes en raison du volume important de données à traiter et de la difficulté de représenter et d'extraire des informations de son contenu. A l'heure actuelle, l'indexation du contenu des images et le découpage de la vidéo en unités élémentaires (plans) et en images représentatives ne répondent que partiellement à ce besoin croissant. Mais le contenu d'une image est souvent lui même trop riche et contient plusieurs objets d'intérêt (objets sémantiques).

L'analyse de la vidéo est un vaste sujet qui englobe un certain nombre de problématiques. On peut notamment citer :

- Détection d'objets en mouvement, ces objets correspondent généralement à des événements importants dans la scène.
- Suivi d'objets en mouvement qui consiste à localiser à chaque instant les objets d'intérêts dans la scène.
- Détection de mouvements cohérents qui consiste à déterminer les différents mouvements ainsi que leurs distribution dans la scène ce qui permet d'identifier les événements inhabituels dans le cas de la vidéo surveillance par exemple.
- L'estimation du mouvement qui consiste à estimer, à partir d'une séquence d'images, le mouvement apparent des objets composant une scène tridimensionnelle.
- Extraction de couches de même mouvement qui consiste à partitionner l'image en régions ayant chacune un mouvement différent.

Nos travaux de recherche s'articulent autour de ces cinq problèmes. Nous avons utilisé pour réaliser cette recherche les descripteurs intra-images (tels que la géométrie, le mouvement, la forme,...), les descripteurs inter-images (tels que la trajectoire) ainsi les techniques basées sur les coupes de graphes.

Un découpage en deux parties a été adopté pour la présentation de cette thèse. La première partie, est dédiée à la détection et au suivi d'objet en mouvement. La deuxième partie aborde le problème de la détection de mouvements cohérents et d'extraction de couches de même mouvement.

Liste des contributions

Mes travaux de recherche présentent trois contributions :

- Une variante approche des algorithmes de détection de mouvement basés sur la différence d'images (section 2.4 chapitre 2).
- Une nouvelle approche de suivi d'objets en mouvement basée sur les Coupes de graphes et le modèle de mouvement paramétrique (chapitre 4).
- Une nouvelle approche automatique de détection de mouvements cohérents et d'extraction de couches de même mouvement basée sur la détection et la mise en correspondance des points d'intérêt ainsi que sur l'algorithme des coupes de graphes (chapitre 6).

Première partie :

*Détection et suivi d'objets en
mouvement*

Introduction de la première partie

La détection et le suivi d'objets en mouvement sont parmi les problèmes les plus étudiés ces dernières années. Ils sont à la base de tout système de vision.

La détection d'objets en mouvement consiste à classifier différents pixels des images en pixels statiques ou mobiles. Il s'agit en effet d'un problème binaire qui est difficile à cause des fluctuations des intensités dues au bruit ou au contenu dynamique de la scène. Dans cette première partie, nous étudions des séquences vidéo de surveillance acquises à l'aide d'une caméra fixe. Le cas de la caméra mobile sera traité dans la deuxième partie de ce document.

Nous abordons aussi dans cette partie le suivi d'objets en mouvement dans des séquences d'images. Une méthode de suivi d'objets en mouvement doit permettre la localisation spatiotemporelle d'objets d'intérêts, et doit aussi fournir une réponse dans un temps très réduit voire en temps réel avec la prise en compte des contraintes de changement de topologie et de larges déplacements.

Dans le chapitre 1, nous présentons, en premier lieu, la théorie de coupe minimale/flot maximal dans un graphe. Puis nous montrons comment les graphes permettent de minimiser une fonction d'énergie donnée. En fin de chapitre, nous étudions l'application de segmentation interactive d'une image fixe.

Le chapitre 2 présente l'état de l'art des méthodes de détection de mouvement. Nous distinguons celles où les décisions sont faites sur les pixels (pixel par pixel) de celles où les décisions sont faites sur des régions de petite taille. Après une conclusion sur les méthodes de l'état de l'art, nous présentons de façon détaillée le principe de notre approche proposée. Quelques résultats expérimentaux obtenus par notre approche seront présentés à la fin de chapitre.

Le chapitre 3 présente l'état de l'art des méthodes de suivi d'objet en mouvement. Nous avons choisi de classer les différentes approches en quatre grandes catégories :

- Approches basées sur le suivi de points
- Approches basées sur le suivi de silhouettes
- Approches basées sur la correspondance de blocs
- Approches basées sur le suivi de noyaux

Nous citons pour chaque approche leurs avantages et leurs limites.

Le chapitre 4 présente une nouvelle approche de suivi d'objets en mouvement. Cette méthode se décompose en trois étapes principales. La première étape consiste à déterminer pour chaque objet le modèle de mouvement affine caractérisant son déplacement. La deuxième étape consiste à prédire les objets à suivre dans l'image suivante en utilisant les modèles de mouvement calculés dans l'étape précédente. L'utilisation des modèles de mouvement pour la prédiction des objets permet de résoudre le problème de large déplacement. La dernière étape consiste à raffiner les frontières des objets suivis par une méthode de segmentation dynamique afin de prendre en compte des éventuels changements

de topologie. Une quatrième étape est ajoutée dans le cas de chevauchement des objets suivis. Cette dernière étape est basée sur la classification des pixels des objets qui se chevauchent par la méthode des coupes de graphes (algorithme alpha-expansion).

CHAPITRE 1

Coupe minimale/flot maximal dans un graphe

Nombreux sont les problèmes en vision par ordinateur qui peuvent être traités comme un problème d'étiquetage. C'est le cas de la segmentation où segmenter une image en deux régions (objet/fond) revient à assigner une étiquette unique à chaque pixel. Ces problèmes d'étiquetage peuvent être représentés en termes de minimisation d'énergie qui peut être résolu par les coupes de graphes sous certaines conditions [Kolmogorov 04]. Cette approche de minimisation d'énergie par les coupes de graphes a été introduite en vision par D.M. Greig [Greig 86][Greig 89]. Une dizaine d'années après, l'approche a connu un succès réel pour une large variété d'applications à savoir la restauration d'images, la segmentation, la synthèse de texture etc. L'utilisation des algorithmes de coupes de graphes (en anglais "Graph cuts") est justifiée par leur temps d'exécution qui est très réduit et la qualité des résultats.

On va adopter dans la suite les notations anglaises classiques en théorie des graphes :

V (Vertices) est l'ensemble des sommets du graphe,

E (Edges) $E \in V * V$ l'ensemble des arcs du graphe,

W (Weight) est une fonction $E \rightarrow \mathbb{R}^+$, qui à toute arcs associe son poids (sa capacité).

s et t deux sommets terminaux (appelés respectivement la source et le puits).

1.1 Notations et définitions

Graphe

Un graphe pondéré $G = (V, E)$ se compose d'un ensemble des nœuds (appelés aussi sommets ou sites) V et d'un ensemble des arcs (appelés aussi arêtes) $E \subset V * V$.

Graphe pondéré

On appelle un graphe pondéré, un graphe, orienté ou non, dont les arrêtes (ou les arcs) $(p,q) \in E$ possèdent un poids $W(p,q)$.

Sommets adjacents

Deux sommets p et q sont adjacents s'ils sont reliés par une arête. On qualifie de voisins deux sommets adjacents.

Voisinage

Le voisinage d'un sommet est l'ensemble de tous ses sommets adjacents.

Soient $p \in V$ est un sommet et N_p son voisinage on a:

$$\begin{aligned} (i) \quad & p \notin N_p \\ (ii) \quad & \text{Si } q \in N_p \text{ alors } p \in N_q \end{aligned} \tag{1.1}$$

Clique

Une clique C est un ensemble de sommets qui vérifie la condition suivante:

$$\forall (p, q) \in C, p \in N_q \tag{1.2}$$

Coupe

On appelle coupe dans un graphe, une partition de l'ensemble des sommets V en deux ensembles V_1 et V_2 (notés aussi S et T) tels que :

$$\begin{aligned} V_1 \cup V_2 &= V \\ V_1 \cap V_2 &= \phi \\ s \in V_1 \quad \text{et} \quad t \in V_2 \end{aligned} \tag{1.3}$$

On appelle les arcs d'une coupe le sous-ensemble de E défini par :

$$\{(p, q) \in E / p \in V_1 \text{ et } q \in V_2\} \tag{1.4}$$

Poids d'une coupe

On appelle poids d'une coupe la somme des poids de ses arcs.

$$\sum_{\substack{p \in V_1 \\ q \in V_2 \\ (p,q) \in C}} W(p, q) \tag{1.5}$$

Coupe minimale

On appelle coupe minimale pour un graphe G , une coupe de poids minimal.

Flot

Si on considère les arcs comme des canalisations, un flot φ est alors défini comme une fonction $\varphi : V \times V \rightarrow \mathbb{R}$ qui à toute arête (p, q) associe une valeur qui est égale à la quantité de liquide qui s'écoule de p vers q . On a les propriétés suivantes:

$$- \text{ pour tout } e = (p, q) \in E \quad \varphi(p, q) = -\varphi(q, p) \tag{1.6}$$

– pour tout sommet p autre que s ou t , on a

$$\sum_{\substack{e=(p, \cdot) \\ e \in E}} \varphi(e) = 0 \quad \text{Contrainte de conservation du flot} \quad (1.7)$$

– pour toute arête $e \in E$, on a : $\varphi(e) \leq W(e)$. (1.8)

On a le résultat suivant : $\sum_{e=(s, \cdot)} \varphi(e) = \sum_{e=(\cdot, t)} \varphi(e)$ (1.9)

Le flot sortant de p : $\varphi^+(p) = \sum_{(p, q) \in E} \varphi(p, q)$ (1.10)

Le flot entrant à p : $\varphi^-(p) = \sum_{(q, p) \in E} \varphi(q, p)$ (1.11)

Valeur d'un flot

La valeur d'un flot $|\varphi|$ est égale à la quantité de liquide qui attend le puits à partir de la source.

$$\sum_{e=(s, \cdot)} \varphi(e) = \sum_{e=(\cdot, t)} \varphi(e) = |\varphi| \quad (1.12)$$

Flot maximal

Le flot maximal $|\varphi_{max}|$ est la quantité maximale de liquide qui peut atteindre le puits à partir de la source.

$$\forall \text{ flot } \varphi \text{ de } s \text{ à } t, |\varphi| < |\varphi_{max}| \quad (1.13)$$

Graphe résiduel

Soit $G = (V, E)$ un graphe et φ un flot, le graphe résiduel de G induit par φ est le graphe $G_r = (V, E_r)$ où $E_r = \{(p, q) \in E \mid C_r(p, q) = W(p, q) - \varphi(p, q) > 0\}$ (1.14)

On appelle $C_r(p, q)$ la capacité résiduelle de l'arête (p, q) . La capacité résiduelle d'un chemin est la plus petite capacité résiduelle de ses arcs.

Chemin améliorant

Soit $G = (V, E)$ un graphe et φ un flot dans G , un chemin améliorant k est un chemin simple de s à t dans $G_r = (V, E_r)$.

Saturation

Un arc (p, q) est dit saturé par un flot f si la valeur du flot sur l'arc est égale à sa capacité :

$$\varphi(p, q) = W(p, q).$$

Un chemin est saturé si l'un de ses arcs est saturé.

Saturer un chemin k entre le nœud s et le nœud t consiste à augmenter le flot sur les arcs du chemin de la capacité résiduelle du chemin : $\varphi(p, q) \leftarrow \varphi(p, q) + C_r(k)$ pour tout $(p, q) \in k$.

Théorème de Ford-Fulkerson [Ford 62]

Le flot maximal qui passe du noeud s vers le noeud t est égal au poids d'une coupe minimale qui sépare s et t .

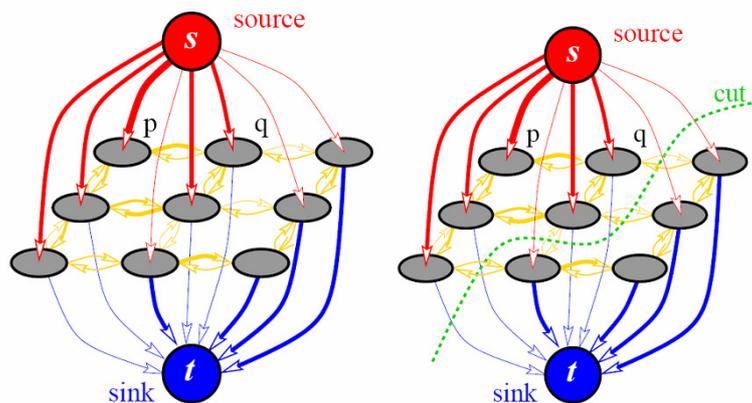
Donc la coupe minimale et le flot maximal sont deux problèmes équivalents et la valeur de flot maximal est égale au coût de la coupe minimale. Un fait important en optimisation combinatoire, est que les algorithmes de coupe minimale/flot maximal sont en polynomiale.

1.2 Calcul du flot maximal (coupe minimale)

La recherche de la coupe minimale dans un graphe est un problème classique. Plusieurs algorithmes ont été développés que l'on peut classifier en deux grandes catégories :

- Les algorithmes dits à chemins augmentant, dont les plus connus sont ceux de Ford et Fulkerson [Ford 62] et l'algorithme itératifs [Boykov 01a] qui est très utilisé pour son faible temps de calcul observé;
- Les algorithmes dits avec préflots dont les plus connus sont Push-Relabel (Poussage-reéiquetage) de Goldberg-Tarjan [Goldberg 88] et l'algorithme de Boykov – Kolmogrov [Boykov 04].

Nous présentons ici les grandes lignes de ces deux catégories. Pour plus de détails voir [Cook 98].



(a) Un graphe orienté pondéré G à deux

(b) Une coupe dans G

Terminaux

Figure 1.1 - Exemple de graphe et de coupe. Le graphe est composé de deux nœuds terminaux s et t et 9 nœuds intermédiaires. En rouge, la source s et les t -links connectés à s . En bleu, le puits t et les t -links connectés à t . En jaune, les n -links entre nœuds intermédiaires. En vert, une coupe. Le poids des arcs est proportionnel à leur épaisseur. Par souci de clarté, certains arcs n'ont pas été représentés. Par ailleurs, dans cet exemple construit sur une grille régulière à deux dimensions, la coupe devrait être représentée par une surface et non par une courbe. Elle coupe effectivement certains t -links afin que pour tout nœud intermédiaire p , un (et un seul) des deux arcs (s, p) et (p, t) soit coupé. Cette figure provient de [Boykov 04].

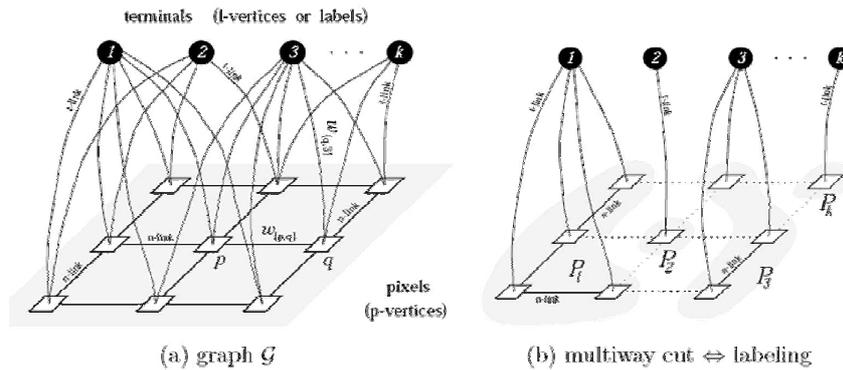


Figure 1.2 - Généralisation au cas multi-étiquette. (a) Un graphe $G = (V, E)$, avec plusieurs terminaux $1 \dots k$ (b) Un multi-way coupe sur G .

1.2.1 Flot maximal par saturation de chemins

Les algorithmes de calcul du flot maximal par saturation de chemins tels que [Ford 62], [Dinic 70], [Edmonds 72], [Goldberg 88], [Boykov 04], visent à trouver, à chaque itération, un chemin non saturé de s à t (s'il existe) puis ajouter autant de flot possible à ce chemin. Ces algorithmes itèrent tant qu'il y a des chemins non saturés reliant s à t .

Algorithme 1.1 : Algorithme de Ford & Fulkerson

Initialisation:

Poser $\varphi(p, q) = 0$ pour toute arête $(p, q) \in E$.

Tant qu'il existe un chemin K de s à t dans G_r (graphe résiduel) faire

Saturer le chemin K dans le graphe résiduel G_r :

$a \leftarrow \min(C_r(p, q) \text{ tel que } (p, q) \in K)$

pour tout $(p, q) \in K$ faire

$\varphi(p, q) \leftarrow \varphi(p, q) + a$

fin pour

fin tant que

Les algorithmes présentés dans [Dinic 70] [Edmonds 72] reprennent ce principe, mais cherchent à chaque itération un chemin de longueur minimale dans le graphe résiduel.

L'algorithme de Boykov – Kolmogorov [Boykov 04] appartient à l'ensemble des algorithmes basés sur l'augmentation de chemins, avec une construction de deux arbres de recherche S -arbre et T -arbre (un arbre à partir de la source et l'autre à partir du puits).

Cet algorithme s'exécute itérativement sur trois étapes :

- Etape de Croissance : d'une itération à une autre, les deux arbres S -arbre et T -arbre s'accroissent. La rencontre de ces deux arbres permet la détection de chemins non saturés
- Etape d'Augmentation : à chaque itération on pousse autant de flots possibles dans les chemins trouvés de telle sorte qu'ils soient saturés.

- Etape d'Adoption : dans cette étape les deux arbres *S-arbre* et *T-arbre* sont reconstitués.

L'inconvénient de cette méthode réside dans le fait que le chemin à augmenter (à saturer) trouvé n'est pas toujours le plus court.

Une autre extension, qui existe dans la littérature [Edmonds 72], introduit l'usage à chaque étape de chemins qui pourraient apporter une plus grande quantité de courant de *s* à *t* (augmentation maximale) au lieu du plus court chemin.

1.2.2 Flot maximal par poussage de flot

Les algorithmes de calcul de flot maximal par poussage de flot considèrent des pré-flots, qui ne respectent pas la contrainte de conservation du flot.

Définition (pré-flot) On appelle pré-flot une fonction $f : E^* \rightarrow \mathbb{R}$ vérifiant les mêmes hypothèses qu'un flot, sauf la contrainte de conservation de flot qui est substituée par :

$$\sum_{\substack{e=(p, \cdot) \\ e \in E^*}} f(e) \geq 0 \quad (1.15)$$

Définition (sommet actif) On appelle sommet actif dans un graphe, un sommet où la contrainte de conservation de flot n'est pas respectée.

Définition (excès de flot) On appelle excès de flot en un sommet *p* la quantité

$$e(p) = \sum_{\substack{e=(p, \cdot) \\ e \in E^*}} f(e) \quad (1.16)$$

Le principe du poussage de flot est alors d'envoyer autant de flot que possible à partir de la source. Bien entendu, certains noeuds reçoivent un excès de flot. Le flot excessif dans ces noeuds est alors poussé vers d'autres noeuds disponibles. Une étiquette, évoluant au cours des itérations, est associée à chaque noeud. On notera $d(p)$ l'étiquette associée au sommet *p*. Le principe des méthodes par poussage de flot est détaillé dans l'algorithme 1.2.

Algorithme 1.2 : Flot maximal par poussage de flot

Initialisation:

Étiqueter les noeuds de la façon suivante :

- *s* reçoit l'étiquette $d(s) = |V|$. // *d* mesure une sorte de proximité au puits.
- *t* reçoit l'étiquette $d(t) = 0$.
- Tout autre noeud *p* dans $V - \{s, t\}$ reçoit l'étiquette $d(p) = 1$.

Poser $f(s, p) = W(s, p)$ pour toute arête (s, p) .

Tant qu'il y a des sommets actifs faire

Choisir un sommet actif *i*

Tant que *i* est actif faire

si il existe $j \in V$ tel que $(i, j) \in E$, $d(i) = d(j) + 1$ et $C_r(i, j) > 0$ alors

pousser autant d'excès que l'on peut vers *j*

POUSSAGE

sinon

ré-étiqueter i de la façon suivante :

RÉ-ÉTIQUETAGE

$d(i) \leftarrow \min(d(j) + 1 / (i, j) \in E \ \& \ C_r(i, j) > 0)$

fin si

fin tant que

fin tant que

1.2.3 Algorithmes de « Dynamic Graph Cuts » et « Active Cuts »

Dans les applications où l'on souhaite faire une suite de coupes minimales dans des graphes similaires, où seul les poids de certains arcs changent, il est intéressant d'exploiter le résultat de la coupe précédente en vue d'accélérer le temps de calcul de la coupe courante. C'est le cas dans les travaux de Kohli et Torr [Kohli 05],[Kohli 07] «Dynamic Graph Cuts» où les auteurs calculent le flot maximal dans un graphe connaissant un flot maximal dans un autre graphe approchant.

Dans [Juan 06], Juan et Boykov ont développé une méthode appelée « Active Cuts » qui permet de calculer une coupe minimale en partant d'une autre coupe minimale. Les Active Cuts vont donc être applicables en changeant la topologie du graphe par exemple dans un cadre multi-échelle.

1.3 Equivalence étiquetage/minimisation d'énergie

La plupart des applications en vision par ordinateur peuvent être vues comme des problèmes d'étiquetage. Un problème d'étiquetage est défini par deux ensembles P et L où P est l'ensemble des variables à étiqueter et L l'ensemble des étiquettes à attribuer.

Le problème que nous allons étudier est celui de la classification (étiquetage) de pixels $p \in P$, où P est l'ensemble de tous les pixels de l'image, pour un ensemble de classes L (étiquettes) donné, via une observation faite sur ces pixels.

Dans cette section, nous allons montrer l'équivalence entre l'étiquetage et la minimisation d'une fonction d'énergie puis nous illustrerons comment les Coupes de graphes peuvent minimiser cette fonction d'énergie.

1.3.1 Champs aléatoires de Markov (souvent abrégé par MRF¹)

Soit un ensemble $X = (X_1, X_2, \dots, X_n)$ de variables aléatoires définies sur P , où chaque variable X_i prend une valeur dans L . Une réalisation (ou une configuration) de champ X est notée $x = (x_1, x_2, \dots, x_n)$.

X est un champ aléatoire de Markov si et seulement si :

$$(i) \ P(X = x) > 0, \ \forall x \in X \tag{1.17}$$

$$(ii) \ P(X_p / x_p) = P(X_p / x_{Np})$$

¹ MRF : Markov Random Field

Le premier point assure que chaque configuration est probable et le deuxième signifie que l'étiquetage d'une variable p ne dépend que de son voisinage x_{N_p} .

Le rapport entre chaque site et un autre est défini en utilisant les fonctions de potentiel sur l'ensemble des cliques C . Pour une clique c donnée, la fonction de potentiel $V_c(x)$ représente la probabilité *a priori* d'une réalisation particulière x des éléments de la clique c .

En utilisant la notion de champs de Gibbs, la distribution de probabilité sur les états possibles est en effet dépendante d'une fonction énergie. Cela permet, notamment pour les images, de partir de modèles physiques concrets pour obtenir une distribution connue et voulue.

$$P(x) = P(X = x) = \frac{1}{Z} \text{Exp} \left(- \sum_{c \in C} V_c(x) \right) \quad (1.18)$$

$$Z : \text{est une constante de normalisation } Z = \sum_{x \in X} \text{Exp} \left(- \sum_{c \in C} V_c(x) \right) \quad (1.19)$$

appelée fonction de partition de Gibbs.

Dans le cas où C est réduit à une arête, la fonction de potentiel $V_c(x)$ peut être assimilée à une "canalisation" laissant circuler une quantité de flot énergétique maximale $u\{p,q\}$ entre les deux nœuds p et q voisins si p et q ont des étiquettes différentes. $V_c(x)$ peut être écrite comme dans [Boykov 01b], [Rother 04] comme suit : $V_{p,q}(x_p, x_q) = u\{p,q\} * \delta(x_p \neq x_q)$ où $\delta(p)$ est une fonction qui envoie 1 si p est vrai et 0 sinon.

Le théorème de Hammersley-Clifford prouve l'équivalence entre le champ aléatoire de Gibbs (CAG) et MRF. Ce résultat permet la simulation de MRF en utilisant la distribution de Gibbs. Pour une restriction sur des cliques d'ordre deux uniquement nous avons:

$$P(x) = \text{Pr}(X = x) = \frac{1}{Z} \text{Exp} \left(- \sum_{p,q \in N} V_{p,q}(x_p, x_q) \right) \quad (1.20)$$

Une fonction de potentiel classique liée au *modèle de Potts* est donnée par:

$$V_{p,q}(x_p, x_q) = \delta(x_p \neq x_q) = \begin{cases} 1, & \text{si } x_p \neq x_q \\ 0, & \text{sinon} \end{cases} \quad (1.21)$$

Ce modèle est largement utilisé dans plusieurs applications.

1.3.2 Maximum A Posteriori (MAP)

L'estimation d'une réalisation x à partir d'une observation Obs est donnée par la probabilité conditionnelle $P(x/Obs)$. La méthode classique pour estimer la réalisation x consiste à utiliser le maximum *a posteriori* $P(x/Obs)$:

$$P(x/Obs) = \frac{P(Obs/x)P(x)}{P(Obs)} \quad (1.22)$$

La maximisation de la probabilité est donc équivalente à:

$$\operatorname{argmax}_{x \in X} P(obs / x)P(x) \quad (1.23)$$

Le terme $P(x)$ est donné par le modèle de MRF (1.20), il reste à estimer $P(Obs / x)$. Ce terme donne une relation entre le modèle et l'observation. En supposant que l'observation Obs est indépendante pour chaque variable $x_{i \in P}$ on a :

$$P(Obs/x) = \prod_{p \in P} P(Obs_p / x_p) \quad (1.24)$$

Pour un modèle de données D^{Obs} qui lie l'observation à la réalisation définie par:

$$P(Obs_p / x_p) = \alpha \operatorname{Exp}(-D_p^{Obs}(x_p)) \quad (1.25)$$

Avec α : une constante de normalisation.

L'estimation complète de $P(Obs / x)$ donc est donnée par :

$$P(Obs / x) \text{ est proportionnelle, à une constante près, à } \operatorname{Exp}(-\sum_{p \in P} D_p^{Obs}(x_p)) \quad (1.26)$$

Et le MAP-MRF est réécrit donc comme suit :

$$\operatorname{argmax}_{x \in X} \operatorname{Exp}\left(-\sum_{p \in X} D_p^{Obs}(x_p) - \sum_{p,q \in N} V_{p,q}(x_p, x_q)\right) \quad (1.27)$$

Ce qui est équivalent à la minimisation de l'énergie suivante :

$$E(x) = \underbrace{\sum_{p \in X} D_p^{Obs}(x_p)}_{\text{terme 1}} + \underbrace{\sum_{p,q \in N} V_{p,q}(x_p, x_q)}_{\text{terme 2}} \quad (1.28)$$

$$E(x) = E(x)_{\text{données}} + E(x)_{\text{lissage}}$$

Nous venons donc de déterminer la fonction d'énergie à minimiser pour résoudre notre problème d'étiquetage. $E_{\text{données}}$ représente l'énergie d'attache aux données qui incite le résultat à être proche des données initiales, et E_{lissage} est un terme de régularité (l'énergie de régularisation) portant sur des pixels (variables x_i) voisins qui incite ces pixels à prendre les mêmes valeurs.

1.4 Minimisation d'énergie par les coupes de graphes

Après avoir défini la fonction d'énergie E à minimiser, nous allons montrer que minimiser celle-ci est équivalent à rechercher une coupe minimale dans un graphe donné.

1.4.1 Cas d'une classification binaire

La construction du graphe $G(V,E)$ pour la minimisation d'une fonction d'énergie binaire se fait comme suit :

- $V = P \cup L$
 - P est ensemble des nœuds correspondant aux variables à étiqueter. Nous appellerons ces nœuds des p-nœuds.
 - $L = \{0, 1\}$ ensemble des nœuds correspondant aux labels (étiquettes à attribuer). Nous appellerons ces nœuds des nœuds terminaux (source et puits).
- $E = E_N \cup E_T$: E_N est l'ensemble des arcs qui relient les nœuds voisins appartenant à P et E_T est l'ensemble des arcs reliant les nœuds P aux terminaux.
- $W(p, q) = V_{p,q}(x_p, x_q)$: capacité des arcs E_N .
- $W(p, l) = D_p^{Obs}(l) / l \in \{0, 1\}$: capacité des arcs E_T .

Puisque une coupe sépare les nœuds terminaux, un p-nœud ne peut être connecté qu'à un seul nœud terminal. A chaque coupe C correspond une configuration x^C dans L de la manière suivante : chaque p-nœud p se voit assigner un label $lj \in L$ représenté par un nœud terminal lj auquel p est connecté dans le graphe induit $G(C)$.

Théorème Si C est une coupe minimale sur un graphe G , alors le poids de la coupe est égal à $E(x^C)$.

Preuve

$$\begin{aligned}
 |C| &= \sum_{p \in T \setminus \{t\}} D_p(1) + \sum_{p \in S \setminus \{s\}} D_p(0) + \sum_{\substack{(p,q) \in N \\ p \in S \setminus \{s\}, q \in T \setminus \{t\}}} V_{p,q}(x_p, x_q) \\
 &= \underbrace{\sum_{p \in T \setminus \{t\} \cup S \setminus \{s\}} D_p(x_p)}_{= \sum_{p \in P} D_p(x_p)} + \sum_{p,q \in N} V_{p,q}(x_p, x_q) - \underbrace{\sum_{\substack{p,q \in N \\ x_p = x_q}} V_{p,q}(x_p, x_q)}_{=0} \\
 &= \sum_{p \in P} D_p(x_p) + \sum_{p,q \in N} V_{p,q}(x_p, x_q) \\
 &= E(x^c)
 \end{aligned}$$

Les fonctions à minimiser par les coupes de graphes doivent vérifier un ensemble de propriétés. Pour déterminer ces propriétés, nous considérons le cas très simple de deux variables (avec interaction) x_1 et x_2 .

L'énergie à minimiser dans ce cas est $D_1(x_1) + D_2(x_2) + V_{12}(x_1, x_2)$

Le graphe $G(V,E)$ sera construit comme suit :

- $V = \{s, t, x_1, x_2\}$
- $E = \cup_{i=1,2} \{s_i = (s, x_i), t_i = (x_i, t)\} \cup \{n_{12} = (x_1, x_2)\}$

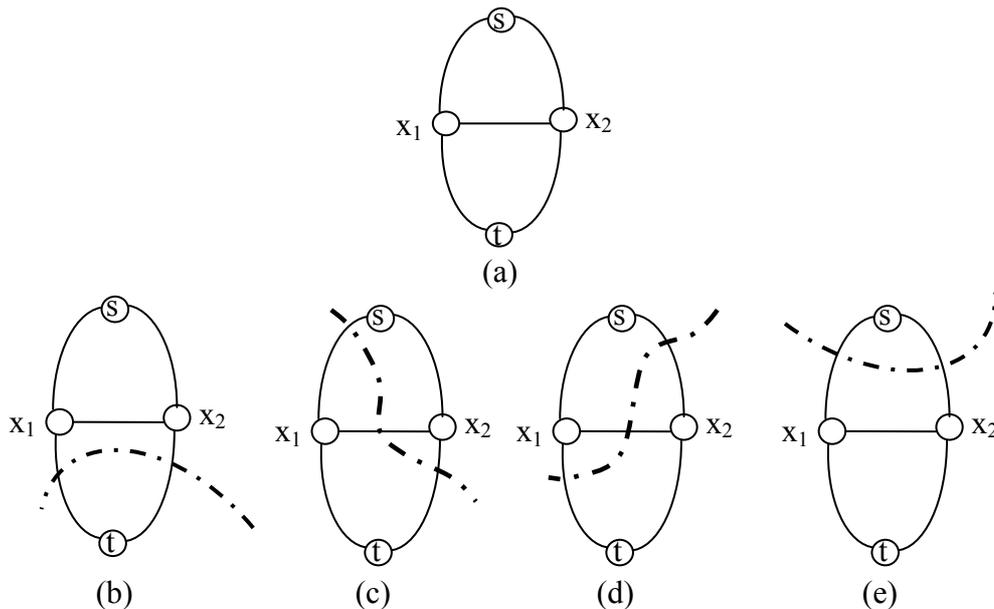


Figure 1.3 - (a) Exemple d'un graphe de deux variables avec interaction, (b), (c), (d) et (e) sont les coupes possibles dans le cas de la classification binaire de deux variables x_1 et x_2 .

Supposons que les poids des arcs sont fixés de telle sorte que la valeur de la coupe soit égale à l'énergie de la configuration associée.

Ceci impose un certain nombre de relations :

$$\left\{ \begin{array}{l} \text{(b)} \rightarrow x_1=0 \quad x_2=0 \rightarrow W(t_1)+W(t_2) = D_1(0)+D_2(0)+V_{12}(0,0) \quad (1) \\ \text{(c)} \rightarrow x_1=1 \quad x_2=0 \rightarrow W(s_1)+W(t_2)+W(n_{12}) = D_1(1)+D_2(0)+V_{12}(1,0) \quad (2) \\ \text{(d)} \rightarrow x_1=0 \quad x_2=1 \rightarrow W(t_1)+W(s_2)+W(n_{12}) = D_1(0)+D_2(1)+V_{12}(0,1) \quad (3) \\ \text{(e)} \rightarrow x_1=1 \quad x_2=1 \rightarrow W(s_1)+W(s_2) = D_1(1)+D_2(1)+V_{12}(1,1) \quad (4) \end{array} \right.$$

$$\underbrace{\hspace{10em}}_{\text{La valeur de s-t coupe}} \quad \underbrace{\hspace{10em}}_{\text{La valeur de l'énergie}}$$

$$(- (1) + (2) + (3) - (4)) \rightarrow 2 * W(n_{12}) = V_{12}(1, 0) + V_{12}(0, 1) - V_{12}(0, 0) - V_{12}(1, 1)$$

Or, les algorithmes de coupe minimale imposent que les poids des arcs soient positifs. Donc l'arc qui dépend de deux variables x_1 et x_2 doit satisfaire une condition dite contrainte de sous modularité.

Définition (sous modulaire) On dit que V_{ij} est régulière (ou sous-modulaire) si on a

$$V_{ij}(0, 0) + V_{ij}(1, 1) \leq V_{ij}(0, 1) + V_{ij}(1, 0). \quad (1.29)$$

La minimisation d'une fonction d'énergie par les Coupes de graphes est donc assujettie à la sous-modularité du terme V_{ij} . On aboutit alors au théorème suivant :

Théorème L'énergie précédente (1.28) peut être minimisée par les Coupes de graphes si et seulement si chacun de ses termes dépendant de deux variables est sous-modulaire. Si c'est le cas, trouver une coupe minimale dans le graphe construit ci-dessus permet d'obtenir un minimum global de l'énergie.

Remarquons que si V_{ij} est un terme de voisinage et qu'il s'écrit sous forme $V_{p,q}(x_p, x_q) = u\{p,q\} * \delta(x_p \neq x_q)$ avec $u\{p,q\} \geq 0$, on aura $V_{ij}(0, 0) = V_{ij}(1, 1) = 0$. Le terme correspondant sera sous-modulaire.

Théoriquement, toute fonction binaire sous-modulaire de classe F^n peut être minimisée par les Coupes de graphes en temps polynomial. En pratique, l'optimisation a été limitée à la classe F^3 [Kolmogorov 04]. Ce n'est qu'en 2009, qu'Ishikawa a développé une méthode générale de réduction des fonctions de la classe F^n à la classe F^2 [Ishikawa 09].

Une fonction de classe F^2 est de la forme

$$E(x_1, x_2, \dots, x_n) = \sum_i E^i(x_i) + \sum_{i < j} E^{i,j}(x_i, x_j)$$

Une fonction de classe F^3 est de la forme

$$E(x_1, x_2, \dots, x_n) = \sum_i E^i(x_i) + \sum_{i < j} E^{i,j}(x_i, x_j) + \sum_{i < j < k} E^{i,j,k}(x_i, x_j, x_k)$$

Pour le cas d'une fonction non sous modulaire, Rother et *al.* [Rother 05] proposent de remplacer la fonction à minimiser par une approximation sous modulaire, puis minimiser cette dernière.

En 2007, Kolmogorov et *al.* [Kolmogorov 07] ont proposé une technique pour optimiser les fonctions binaires non sous-modulaires appelée QPBO (Quadratic Pseudo-Boolean Optimization). Sans rentrer dans les détails, cette approche produit une solution dont seulement la partie correspond aux potentiels de cliques sous-modulaires a été optimisée, et pour cette partie optimisée les étiquettes trouvées sont globalement optimales. Les performances de QPBO dépendent du nombre de termes non sous-modulaires.

1.4.2 Cas d'une classification multi-étiquettes

Dans la plupart des applications en vision par ordinateur la classification n'est pas binaire. A titre d'exemple, la segmentation en couches de même mouvement où le nombre des étiquettes est égal au nombre de couches, peut être supérieur à deux. Dans ce cas l'ensemble des étiquettes possibles s'écrit $L = \{1, \dots, n\}$ et pour tout $p \in P$ on a $x_p \in \{1, \dots, n\}$.

La construction du graphe pour la minimisation de l'énergie multi-étiquettes dépend de l'algorithme utilisé et du problème à résoudre.

Dans [Roy 98] et [Ishikawa 03] le graphe $G(V,E)$ est construit comme indiqué dans la figure 1.4. Chaque variable est représentée par une colonne et l'étiquette de niveau où la colonne est coupée représente l'étiquette qui va être prise par la variable.

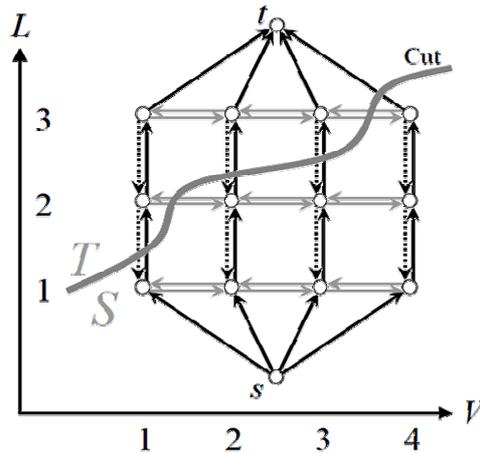


Figure 1.4 - Construction du graphe pour une classification multi-étiquettes. La figure illustre le cas de quatre variables avec trois étiquettes possibles. La coupe correspond aux assignations suivantes : $f(p1) = 1$, $f(p2) = 2$, $f(p3)=2$, $f(p4) = 3$. Cette figure provient de [Ishikawa 03].

Ishikawa a montré que pour appliquer les algorithmes de recherche de coupes minimales dans ce type de graphe, la fonction de régularisation doit être convexe. Ce cas particulier a été appliqué dans de nombreuses applications comme la stéréovision où L représente un ensemble de profondeurs.

Cette forme de graphe ne peut pas donc être utilisée dans le cas où le terme de régularisation n'est pas convexe telle que l'interaction de Potts où $V(x_p, x_q) = \delta(x_p \neq x_q)$. Dans ce cas, il n'existe pas d'algorithme permettant de trouver une solution exacte en temps polynomial. Néanmoins, les deux algorithmes (α, β) -swap et α -expansion développés par Boykov et al. [Boykov 01a] permettent de minimiser approximativement l'énergie (1.28) sur un ensemble fini d'étiquettes L et pour une classe de pénalité V dite métrique dans le cas de α -expansion et pour une classe de pénalité V dite semi-métrique pour le cas de (α, β) -swap. V est une métrique sur l'espace des étiquettes L s'il satisfait

1. $V(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$
 2. $V(\alpha, \beta) = V(\beta, \alpha) \geq 0$
 3. $V(\alpha, \beta) \leq V(\beta, \gamma) + V(\gamma, \beta)$
- (1.30)

pour tout α, β et $\gamma \in L$. Si V vérifie seulement 1 et 2 elle est dite semi-métrique.

Exemple :

$$V(\alpha, \beta) = \min(K, |\alpha - \beta|^2) \text{ est semi métrique}$$

$$V(\alpha, \beta) = \min(K, |\alpha - \beta|) \text{ est métrique}$$

(α, β) -swap

L'idée principale de l'algorithme (α, β) -swap est de segmenter successivement tous les nœuds ayant l'étiquette α des nœuds ayant l'étiquette β avec la méthode des coupes de

graphes. L'algorithme change la combinaison (α, β) à chaque itération. À chaque étape, on ne considère que les variables ayant les étiquettes α ou β et on change leur valeur en α ou β uniquement. On applique donc l'algorithme du cas binaire $(|L|^2 - |L|)$ fois pour faire une série complète de (α, β) -swap.

Algorithme 1.3 : (α, β) -swap [Boykov 01a]

Initialisation:

Commencer avec un étiquetage x arbitraire

$E1 \leftarrow E(x)$

baisse $\leftarrow 1$

tant que baisse = 1 **faire**

baisse $\leftarrow 0$

pour tout couple d'étiquette (α, β) **faire**

soit $I = \{i / x_i = \alpha \text{ ou } \beta\}$

appliquer une minimisation d'énergie binaire à x_I pour les étiquettes α et β

$E2 \leftarrow E(x^c)$ // x^c est la configuration obtenue après la coupe

si $(E2 < E1)$ **alors**

baisse $\leftarrow 1$

$E1 \leftarrow E2$

$x \leftarrow x^c$

fin si

fin pour

fin tant que

retourner x

La structure du graphe est déterminée à chaque itération en fonction de l'étiquetage f courant et des étiquettes α et β . La figure 1.5 montre le cas d'une image 1D. L'ensemble des sommets inclut α et β et aussi tous les pixels $p \in P$ de l'image tels que $f_p \in \{\alpha, \beta\}$.

$$V_\alpha = \{\alpha, \beta, p \in P / f_p \in \{\alpha, \beta\}\} \quad (1.31)$$

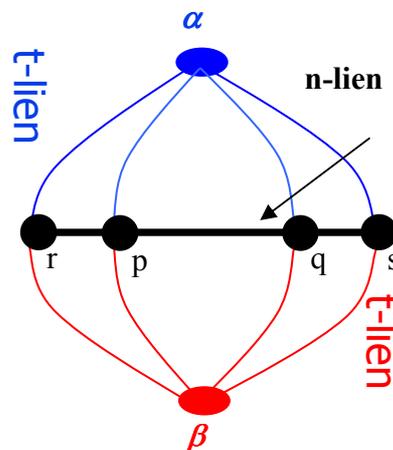


Figure 1.5 - La mise en graphe de l'algorithme (α, β) -swap. $I = \{r, p, q, s\}$.

α -expansion

L'idée principale de l'algorithme α -expansion est de segmenter successivement tous les nœuds α et non- α avec des coupes de graphes et l'algorithme change la valeur de α à chaque itération. L'algorithme réitérera pour chaque étiquette possible de α jusqu'à ce qu'aucune transformation ne change la configuration actuelle.

Algorithme 1.4 : α -expansion [Boykov 01a]

Initialisation:

Commencer avec un étiquetage x arbitraire

$E1 \leftarrow E(x)$

baisse $\leftarrow 1$

tant que baisse = 1 **faire**

 baisse $\leftarrow 0$

pour tout étiquette α **faire**

 appliquer une minimisation d'énergie binaire à X pour les étiquettes α et $I =$ le reste des étiquettes

$E2 \leftarrow E(x^c)$ // x^c est la configuration obtenue après la coupe

si ($E2 < E1$) **alors**

 baisse $\leftarrow 1$

$E1 \leftarrow E2$

$x \leftarrow x^c$

fin si

fin pour

fin tant que

retourner x

La structure du graphe est déterminée à chaque itération en fonction de l'étiquetage f courant et de l'étiquette α . La figure 1.6 montre le cas d'une image 1D. L'ensemble des sommets inclut α et $\bar{\alpha}$ et aussi tous les pixels $p \in P$ de l'image. En outre, pour chaque paire de pixels voisins $\{p, q\} \in N$ dont les étiquettes courantes sont différentes ($f_p \neq f_q$), nous créons un noeud auxiliaire $a_{\{p, q\}}$. Ainsi, l'ensemble de sommets est :

$$V_\alpha = \left\{ \alpha, \bar{\alpha}, P, \bigcup_{\substack{\{p, q\} \in N \\ f_p \neq f_q}} a_{\{p, q\}} \right\} \quad (1.32)$$

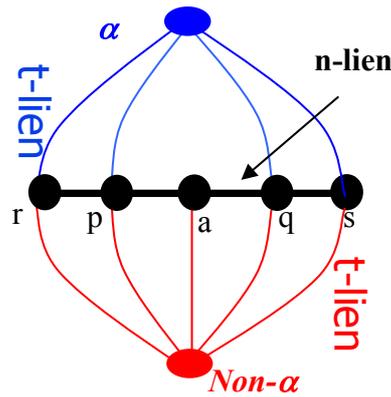


Figure 1.6 - Un exemple de G pour une image 1D. L'ensemble des Pixels de l'image est $P=\{p, q, r, s\}$. Dans le partitionnement courant $f_p \neq f_q$. Une noeuds auxiliaire $a = a_{\{p, q\}}$ est créé entre les Pixel p et q .

Chaque fois que l'algorithme α -expansion est utilisé, il garanti que le minimum local calculé est plus ou moins un facteur du minimum global [Boykov 01a]. Ce facteur est :

$$C = \frac{\max_{\alpha \neq \beta \in l} V(\alpha, \beta)}{\min_{\alpha \neq \beta \in l} V(\alpha, \beta)} \quad (1.33)$$

1.5 Autres approches de minimisation

Le problème de minimisation est un problème fréquemment rencontré en traitement numérique. Plusieurs approches ont été développées. Nous présentons ici quelques alternatives des algorithmes de coupe minimale/flot maximal pour la minimisation d'une fonction d'énergie d'un champ de Markov.

1.5.1 Le recuit simulé

En métallurgie, afin d'améliorer la qualité d'un solide on minimise son énergie par un processus qui alterne des cycles de refroidissement lent et de réchauffage (*recuit*). Le recuit simulé [Kirkpatrick 83] est une méthode empirique (métaheuristique) inspirée de ce processus physique pour trouver les extrema d'une fonction en faisant appel à un paramètre fictif T qui va jouer le rôle de la température du système. L'idée générale est qu'à partir d'une initialisation donnée, on modifie l'état d'une variable sélectionnée aléatoirement à chaque itération. Si la nouvelle configuration obtenue après ce changement améliore le critère que l'on cherche à optimiser alors elle est directement acceptée. Sinon, elle n'est acceptée qu'avec une probabilité égale à $\exp(\Delta E / T)$, avec E =énergie, cette règle est appelée critère de Metropolis.

L'acceptation d'une « mauvaise » solution permet d'explorer une plus grande partie de l'espace de solution et tend à éviter de s'enfermer trop vite dans la recherche d'un optimum local.

La minimisation par la méthode de recuit simulé permet d'obtenir de bons résultats mais

au prix de longs calculs avec une température que l'on diminue très lentement.

1.5.2 Les modes conditionnels itérés (ICM)

L'algorithme des modes conditionnels itérés développé par Besag [Besag 86] est beaucoup plus rapide que le recuit simulé. Le principe général de cet algorithme est qu'à partir d'une configuration donnée, on modifie à chaque itération, de manière déterministe, l'état de chaque variable à partir des états probables des variables voisines de telle sorte que l'on maximise la probabilité conditionnelle locale. Le processus s'arrête lorsque le nombre de changements d'une itération à l'autre devient suffisamment faible.

Cette méthode est très sensible à l'initialisation et elle converge vers le premier minimum local rencontré.

1.5.3 La propagation bouclée de croyances ou « Loopy Belief Propagation » (LBP)

L'algorithme de propagation bouclée de croyances [Felzenszwalb 06] est un algorithme développé pour le calcul approximatif des statistiques marginales sur des graphes. L'algorithme de propagation de croyances original a été proposé par Pearl en 1982 [Pearl 82] pour trouver des marginaux exacts sur les arbres qui ne contiennent pas de boucles. L'algorithme est interprété comme le fait de faire remonter des messages des feuilles vers la racine. Dans le cas où le graphe contient des cycles, les messages sont passés soit en parallèle, soit dans un ordre aléatoire sur les nœuds.

Une étude comparative de LPB aux *coupes de graphes* est donnée dans [Szeliski 06], [Tappen 03]. Les auteurs ont montré que cet algorithme est moins efficace que les coupes de graphes.

1.5.4 Tree-ReWeighted Message Passing (TRW)

L'algorithme Tree-ReWeighted Message Passing a été proposé par Wainwright et *al.* en 2005 [Wainwright 05] et amélioré par Kolmogorov [Kolmogorov 06b] en 2006. L'idée principale de TRW est de trouver un ensemble d'arbres couvrants du graphe et d'appliquer successivement à chacun l'algorithme de propagation de croyances [Pearl 82]. Cette approche permet d'obtenir des résultats similaires à ceux obtenus par les coupes de graphes pour des graphes faiblement connectés mais moins bons lorsque ceux-ci sont hautement connectés [Szeliski 06].

1.6 Prise en compte des contraintes fortes par les coupes de graphes

Nous avons montré dans la section 1.3 qu'un problème d'étiquetage peut être reformulé en termes de minimisation d'énergie. Ainsi dans la section 1.4 on a montré que cette énergie peut être minimisée par les coupes de graphes sous certaines conditions. Nous proposons dans

cette section, pour montrer la possibilité de la prise en compte des contraintes fortes par le formalisme de coupe de graphe, d'étudier le cas de la segmentation interactive [Boykov 01b] [Housni 09a].

La segmentation interactive consiste à étiqueter les pixels de l'image en fonction de leur appartenance aux zones d'intérêt ou non en utilisant certaines contraintes topologiques dites "contraintes fortes" qui peuvent indiquer que certains germes de l'image sont connus *a priori* comme faisant partie de « l'Objet » ou du « Fond ». La segmentation doit satisfaire les contraintes suivantes :

$$\begin{cases} \forall p \in O : x_p = 1 \\ \forall p \in F : x_p = 0 \end{cases} \quad (1.34)$$

Autrement dit, la minimisation de la fonction d'énergie doit se faire sans que les nœuds du graphe correspondant aux graines $\{O, F\}$ ne puissent changer d'étiquette.

La construction du graphe $G(P \cup \{s, t\}, E)$ se fait comme suit :

Chaque pixel p de l'image correspond à un nœud P_p du graphe. Deux autres nœuds supplémentaires forment la source s et le puits t , représentant respectivement l'objet et le fond (les étiquettes à attribuer). Chaque nœud p-nœud est relié à ses voisins par des arcs n-liens (neighbor), avec une connectivité choisie, et dont les capacités dépendent des différences de l'intensité. Chaque nœud (pixel) est aussi relié par des arcs t-liens aux terminaux (source et puits).

Les deux termes de la fonction d'énergie (1.28) sont calculés par :

$$\begin{aligned} D_p^{Obs} (I) &= -\lambda * \ln(P(I_p / Obs)) \\ V_{p,q} (x_p, x_q) &= \exp\left(-\frac{\|I_p - I_q\|^2}{2 * \sigma^2} \right) \frac{1}{dist(p, q)} \end{aligned} \quad (1.35)$$

λ : spécifie une importance relative des propriétés de région $\Sigma D(.)$ contre les propriétés de frontière $\Sigma B(.)$.

σ est un paramètre permettant de plus ou moins lisser les résultats.

I_p est l'intensité du pixel p .

$dist$ est une fonction de distance.

Les poids des arcs sont calculés comme il est indiqué dans la section 1.4.1. sauf pour les liens entre les nœuds du graphe correspondant aux graines $\{O, F\}$ et les terminaux. Les grains d'objet sont reliés avec un poids infini (max) à la source, et avec un poids nul au puits (1.36). Le principe est inversé pour les graines de fond (1.37). Ceci permet de satisfaire les contraintes (1.34).

$$si \ p \in O \quad \begin{cases} W(p, s) = \infty \\ W(p, t) = 0 \end{cases} \quad (1.36) \quad si \ p \in F \quad \begin{cases} W(p, s) = 0 \\ W(p, t) = \infty \end{cases} \quad (1.37)$$

Les pénalités $\ln Pr(I_p | \text{fond})$ et $\ln Pr(I_p | \text{objet})$ sont justifiées par la formulation

fondamentale de MAP-MRF (section 1.3). Pour le calcul de ces termes, Boykov et Jolly [Boycov 01b] ont utilisé les intensités des pixels des régions graines pour estimer deux histogrammes, l'un pour l'objet et l'autre pour le fond (figure 1.7).

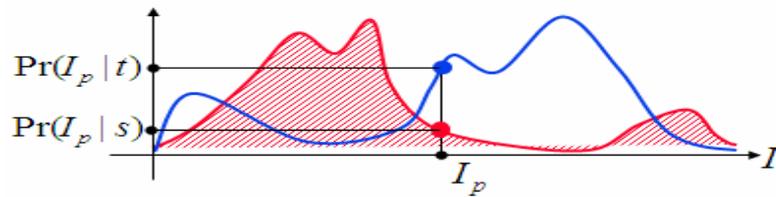


Figure 1.7 - Cette figure, histogrammes des germes d'objet et de fond, illustre comment les termes $-\ln\Pr(I_p|\text{"fond"})$ et $-\ln\Pr(I_p|\text{"objet"})$ seront calculés.

$W(p, q)$ est grand quand $|I_p - I_q| < \sigma$, c'est-à-dire quand les deux pixels p et q sont similaires. Il est plus petit et tend vers zéro quand $|I_p - I_q| > \sigma$. Le but est d'encourager la segmentation (la coupe) qui passe par des régions où le gradient de l'image est assez fort.

Les figures 1.9 et 1.10 donnent les résultats d'extraction d'objets d'intérêts à partir des contraintes fortes.

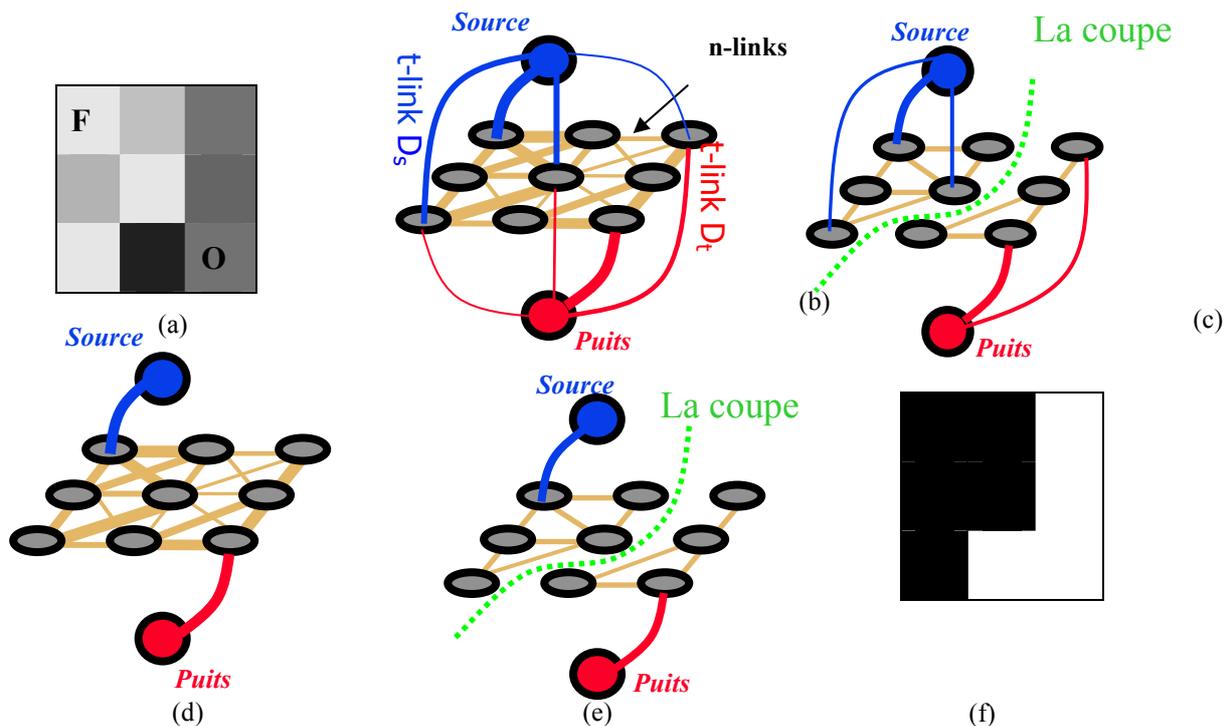


Figure 1.8 - Exemple de segmentation d'une image 3x3 avec une connectivité V8. Le poids de chaque arrête est reflété par l'épaisseur de cette dernière. Les poids de n-liens sont calculés par (1.35) pendant que les coûts des arrêtes t-liens sont calculés par (1.36) et (1.37). Les contraintes fortes seront rendues effectives par des arrêtes t-liens avec les coûts infinis. (a) image initiale avec contraintes fortes, (b) graphe correspondant pour $\lambda \neq 0$, (c) graphe résiduel après la convergence de l'algorithme de coupe minimale pour le cas où $\lambda \neq 0$, (d) graphe correspondant pour $\lambda = 0$, (e) graphe résiduel après la convergence de l'algorithme de coupe minimale pour le où $\lambda = 0$, (f) résultat de la segmentation.

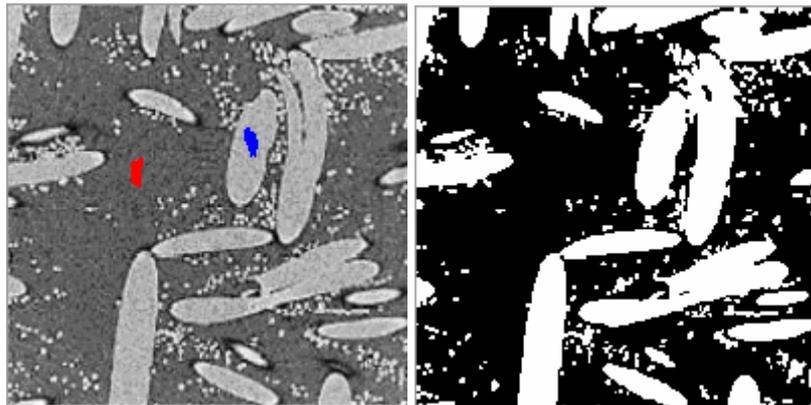


Figure 1.9 - Cas de l'extraction des objets d'intérêts avec la prise en compte des propriétés régionales $\lambda=1$. Les graines d'objet sont marquées en bleu et ceux de fond sont en rouge.

Image originale	Région d'intérêt	Image sans région d'intérêt

Figure 1.10 - Résultats obtenus pour les image IRM médicale avec $\lambda=0$ (Images de la mammographie et image du cerveau). L'utilisateur commence par placer des graines à l'intérieur et à l'extérieur de l'objet à segmenter. Les graines d'objet sont marquées en bleu et ceux de fond sont en rouge.

Les algorithmes de coupes de graphes peuvent être aussi utilisés pour améliorer et raffiner les résultats obtenus par d'autres algorithmes. La figure 1.11 donne les résultats de la combinaison SVM (Support Vector Machine) [Vapnik 95] et coupes de graphes pour la classification des images satellitaires [Bekkari 10] et la détection des tumeurs cérébrales [Housni 10].

1.7 Conclusion

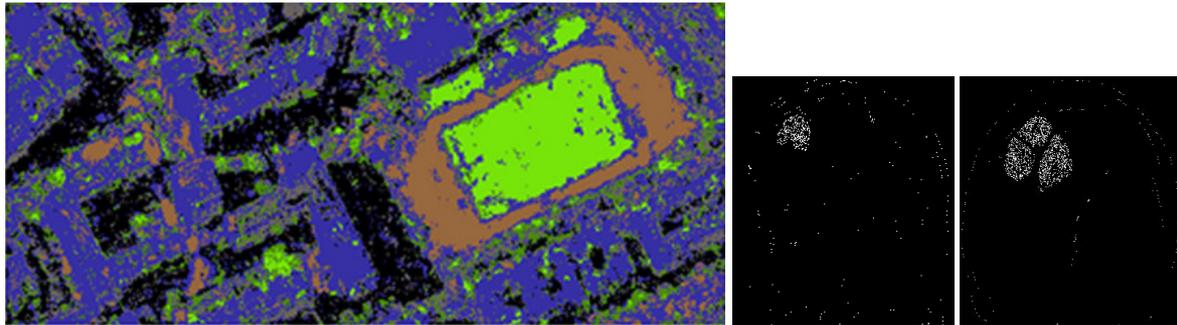
Nous avons présenté dans ce chapitre comment la méthode des coupes de graphes peut être appliquée en analyse d'image. En particulier, nous avons expérimenté cette méthode dans le cas de l'extraction d'objet. Cette approche présente plusieurs avantages:

- elle est flexible avec initialisation (la prise en compte des contraintes fortes) ;
- elle converge vers le minimum global plus rapidement pour le cas binaire ;
- à chaque itération de l'alpha-expansion le résultat de la coupe est optimal. Par conséquent, le risque de tomber dans un minima local est faible ;
- elle est très rapide par rapport à d'autres algorithmes de minimisation (pour une comparaison détaillée voir [Szeliski 06]).

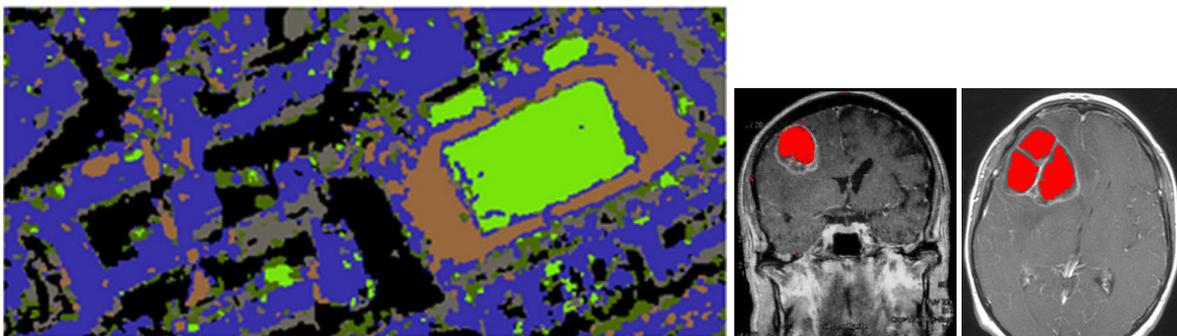
Dans la suite de ce document, nous allons exploiter ces algorithmes de coupe minimale/flot maximal pour la détection et le suivi d'objets en mouvement ainsi que pour la segmentation en couches de même mouvement.



(a)



(b)



(c)

Figure 1.11 – Cette figure donne des exemples d'utilisation des coupes de graphes pour l'amélioration des résultats obtenus par la méthode de classification SVM. (a) les images originales, (b) résultats par classification SVM, (c) résultats par SVM améliorés par Coupes de graphes.

CHAPITRE 2

Détection de mouvement

La détection des objets en mouvement dans une séquence vidéo, qui est la première étape dans l'analyse vidéo, est une tâche importante. Parmi ses applications, on peut citer la compression vidéo, la vidéosurveillance et le suivi et la reconnaissance d'objets.

Dans ce chapitre, nous proposons une approche pour segmenter automatiquement les objets en mouvement à partir de séquences d'images dans le cas d'une caméra fixe. Ce problème peut être résolu par une différence d'images seuillées [Jain 79]. Cependant, le choix de ce seuil résulte d'une analyse *a posteriori* des résultats par un observateur extérieur. Dans le cas d'une séquence fortement bruitée, le même seuil ne convient plus. Notre approche consiste à formuler le problème de détection de mouvement, dans un cadre bayésien, en terme de minimisation d'énergie. Cette énergie sera ensuite minimisée par l'algorithme de calcul de coupe minimale « coupes de graphes ». Pour cela, dans un premier temps, nous calculons la différence de deux ou trois images successives et l'image résultat sera ensuite utilisée pour la construction du graphe. Le résultat de la coupe est une carte binaire qui indique pour chaque pixel si sa valeur a changé ou non. Un objet en mouvement est donc défini comme toute partie (qui peut être un objet, ensemble d'objets ou encore une partie d'un objet) dont l'intensité entre deux images successives change.

Dans la première partie de ce chapitre, nous donnons l'état de l'art des méthodes de détection basées sur la différence inter-images. Puis nous détaillons notre approche et nous donnons quelques résultats expérimentaux.

2.1 Représentation des données vidéo

Une séquence vidéo est un ensemble d'images prises à des instants successifs avec un intervalle de temps assez court. Les données vidéo peuvent donc être représentées par une fonction définie dans un espace à trois dimensions : deux dimensions spatiales (x, y) et une dimension temporelle (t). Les deux premières dimensions sont connues et finies tandis que la dimension temporelle est *a priori* infinie. Chaque point de cet espace, noté $I(x, y, t)$,

représente un niveau de gris (ou un vecteur de composantes couleur).

Les différentes entités sémantiques (couches de mouvement, arrière-plan, objets mobiles) sont donc des sous-ensembles de points de cet espace.

Nous désignerons par I_t l'image à l'instant t .



Figure 2.1 - Exemple de séquence vidéo avec sa représentation en 3D.

2.2 Etat de l'art des méthodes de détection basées sur la différence inter-images

Un grand nombre d'algorithmes de détection du mouvement basés sur la différence d'images existe dans la littérature. Nous pouvons les classer en deux grandes catégories, en fonction des éléments sur lesquels les décisions sont prises. La première catégorie regroupe les méthodes où les décisions sont prises sur les pixels (pixel par pixel) tandis que la deuxième catégorie regroupe les méthodes où les décisions sont prises sur les régions.

2.2.1 Décision sur un pixel

Nous présentons dans cette section quelques méthodes de détection par un seuillage global, ce type de détection est basé sur une analyse pixel par pixel des changements de luminance.

L'image de différence (appelée aussi différence d'image) est calculée par :

$$\zeta(s) = \left| \frac{dI(s)}{dt} \right| = \left| \Delta_{I_{t,t-1}}(s) \right| = \left| I_t(s) - I_{t-1}(s) \right| \quad (2.1)$$

où S représente l'ensemble des sites d'une image.

Théoriquement, en un pixel donné, $\zeta = 0$ indique l'absence de mouvement et $\zeta \neq 0$ une présence de mouvement. Le cas pratique est malheureusement très compliqué à cause du bruit

lié à la caméra. Résoudre ce problème revient donc à déterminer un seuil qui va permettre de classer les pixels en statique et dynamique.

Dans [Kon 00] Konrad a présenté un ensemble de méthodes classiques où souvent un seuil global est calculé par l'utilisation d'un rapport de vraisemblance et sous l'hypothèse que la différence d'intensité entre deux images successives peut être modélisée par une loi gaussienne dans le cas de l'absence de mouvement et par une loi uniforme sous l'hypothèse de mouvement. Ces méthodes dépendent de la vitesse et de la taille des objets en mouvement.

Plus tard, dans [Wu 05], au lieu de la différence inter-images, le rapport d'intensité entre deux images a été utilisé. Un seuil global est automatiquement déterminé par une analyse de l'histogramme cumulé des rapports d'intensité. Ce seuil correspond à un point de changement dans l'histogramme des valeurs cumulées.

Pour le cas des méthodes qui utilisent un triplé d'images. L'image à seuiller $O_t(s)$ est souvent calculée comme suit :

$$O_t(s) = \min \left(\left| \Delta_{I_{t,t-1}}(s) \right|, \left| \Delta_{I_{t+1,t}}(s) \right| \right) \quad (2.2)$$

Cette technique est très rapide et elle permet de gérer les occlusions, mais elle ne donne de bons résultats que dans le cas des mouvements suffisamment importants.

Au lieu d'utiliser une différence entre images successives, d'autres approches proposent d'utiliser des techniques de différence au fond [Jain 79] [Karmann 90] [Toyoma 99] [Stauffer 00] [Cheung 03] [Piccardi 04] où une image de référence (fond) représente la partie statique de la scène.

Le calcul de fond statique (sans aucun objet mobile) se fait généralement par une moyenne qui peut prendre différentes formes (moyenne arithmétique, moyenne récursive, max, min, etc...). Un bon effacement demande cependant un grand nombre d'images. Cette technique permet de contourner le problème de la vitesse et de la taille des objets. Cependant, elle ne peut pas s'adapter aux environnements dynamiques.

Dans [Jain 79], le calcul de l'image du fond à l'instant t est obtenu en remplaçant les zones mobiles dans l'image courante par le contenu statique de ces zones dans les images précédentes. Cela revient à remplacer les zones mobiles par ce qui est caché derrière ces zones.

Au lieu d'utiliser une seule image comme modèle de l'arrière-plan, les auteurs de [Toyoma 99] proposent d'utiliser k modèles. A l'instant t donné, la détection de mouvement est effectuée avec chacun des modèles, et celui qui détecte le moins de pixels en mouvement est retenu pour la décision finale. Ces modèles sont calculés par l'algorithme *k-means* à partir d'une séquence d'apprentissage. Afin de gérer les variations d'illuminations de la scène et les ajouts ou suppression des objets du fond, les auteurs proposent de mettre à jour ces modèles chaque fois que la majorité des pixels de l'image sont détectés comme étant en mouvement.

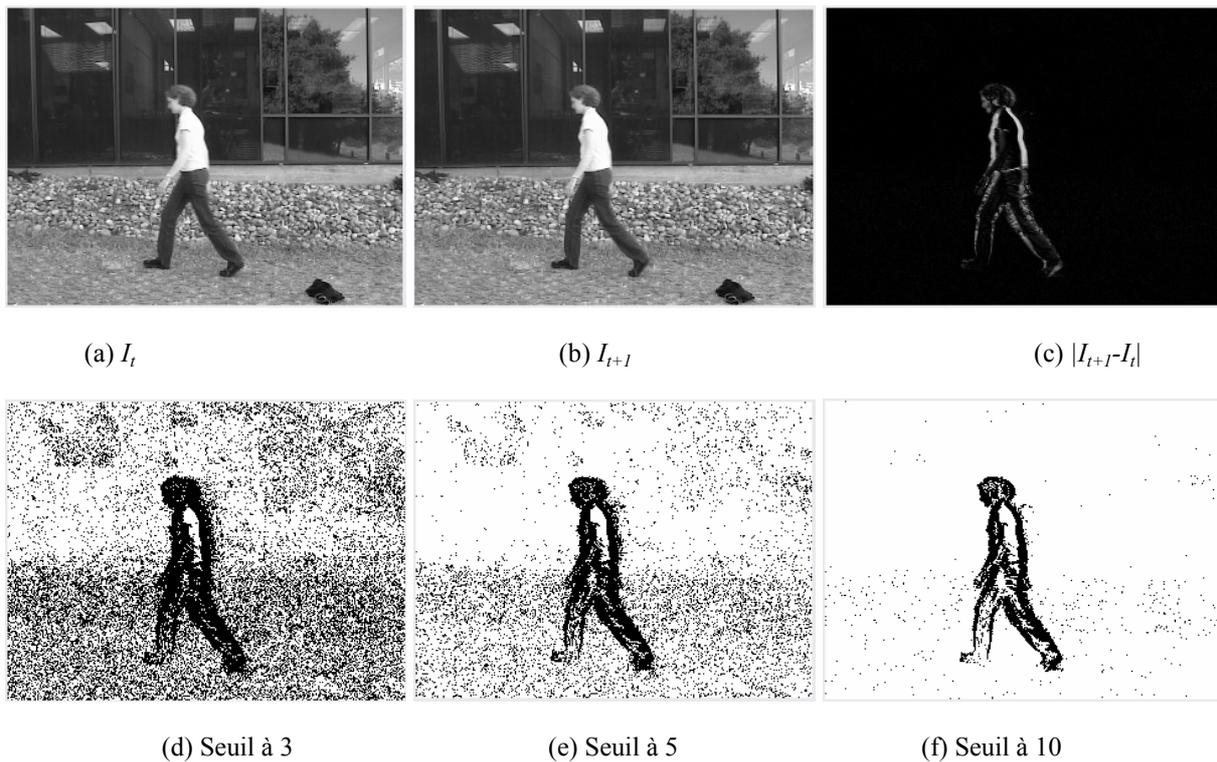


Figure 2.2 – La figure montre le cas d’une détection de mouvement basée sur la différence d’images. Pour un seuil faible (cas de 3 et 5), des changements d’intensité dus au bruit sont détectés sur le fond. Remarquons aussi que dans les images seuillées les zones homogènes d’objet en mouvement sont classées comme étant des parties de fond.

Afin d’assurer une cohérence spatiale, Hsu et *al.* [Hsu 84] utilisent une modélisation de l’intensité par un polynôme de deuxième degré sur des fenêtres de petites tailles 3 x 4. La décision s’il s’agit de zones mobiles ou non est faite par un test de Student sur les distributions d’intensité dans les deux images. Le niveau de confiance choisi influence les résultats.

Toutes les méthodes présentées dans cette section permettent d’obtenir des cartes binaires de mouvement. La classification en objets est une tâche *a posteriori* qui peut être faite par l’algorithme de détection des composantes connexes.

Pour une détection de mouvement avec classification des pixels en objets en mouvements distincts, Zhu [Zhu 05] propose de combiner la méthode de modélisation de fond avec une méthode basée sur la cohérence. La méthode proposée suit un processus en trois étapes. La première étape consiste à détecter les points caractéristiques à l’aide d’un détecteur de coins de Harris [Harris 88]. L’apprentissage dynamique du modèle de fond et la classification des points caractéristiques en fonction de l’appartenance à l’arrière-plan ou au premier plan sont faites dans la deuxième étape. Enfin, après le suivi des points caractéristiques n’appartenant pas au fond par la méthode KLT [Shi 94], les objets mobiles sont ainsi obtenus en groupant tous les points suivis ayant des trajectoires similaires et cohérentes.

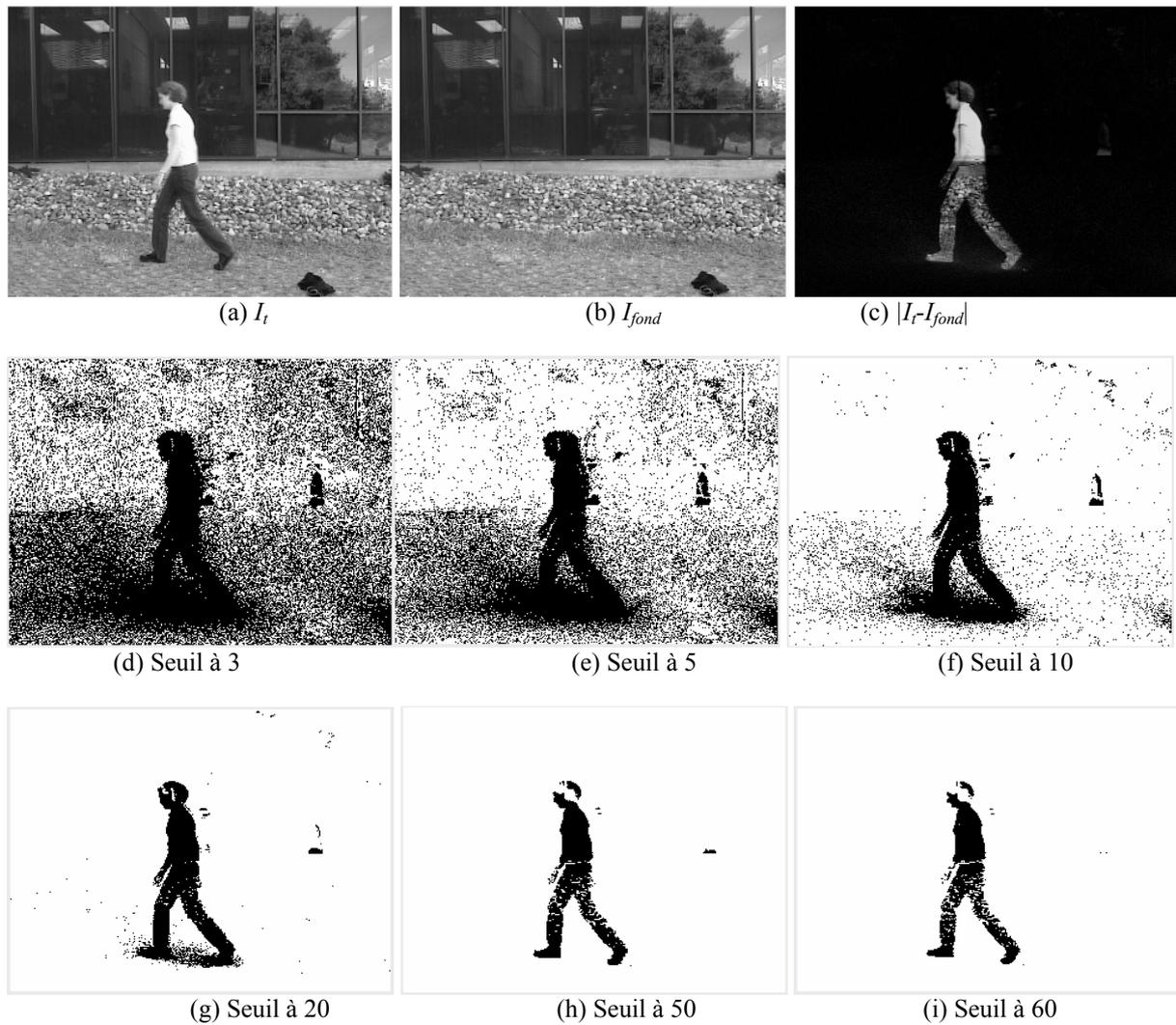


Figure 2.3 – Détection de changements par différence avec une image de fond suivi d'un seuillage. (a) Image à l'instant t , (b) Image de référence du fond statique construite, par exemple, avec un filtre médian temporel sur toute la séquence. (c) : Valeur absolue de la différence entre l'image de référence et l'image I_t . Les valeurs faibles apparaissent en clair, les valeurs fortes en foncé. (d), (e), et (f) : cartes binaires obtenues avec un seuil 3, 5 et 10. Le niveau de bruit important dans cette séquence ne permet pas de discerner les objets mobiles dans ces cartes. (g), (h), et (i) : cartes binaires obtenues avec un seuil plus élevé. Valeurs du seuil à 20, 50 et 60. Les objets mobiles se distinguent mieux du bruit avec un seuil plus élevé.

2.2.2 Décision basée région

Les algorithmes de détection de mouvement qui utilisent les régions sont peu nombreux car une classification erronée d'une région est plus coûteuse par rapport à une mauvaise classification d'un pixel. Parmi ces approches on trouve les méthodes de segmentation de mouvement ou d'extraction de couches de mouvement (voir chapitre 6). Sous l'hypothèse que le mouvement de la caméra (mouvement de l'arrière plan) correspond au mouvement dominant dans l'image. Les zones dont le mouvement n'appartient pas au mouvement

dominant sont ainsi classées comme des zones en mouvement.

Parmi les approches basées région, nous pouvons citer les approches de contours actifs qui sont largement employées [Paragios 00], [Ranchin 04], [Jehan-Besson 01b], [Jehan-Besson 01a]. De façon similaire aux approches bayésiennes, ces approches reposent sur la minimisation d'une fonction énergie pour faire évoluer un contour situé initialement près de l'objet d'intérêt. Ces techniques utilisent des connaissances *a priori* sur ce que nous cherchons, elles ne conviennent donc pas à des systèmes de détection automatique.

Dans ces dix dernières années, d'autres approches [Lisani 03] [Veit 04b] [Dibos 05] basées sur des modèles *a contrario* ont été utilisées pour la détection des zones de changement dans les images. Une courte présentation préliminaire du cadre *a contrario* appliquée à la détection de mouvement peut être trouvée dans [Veit 04a]. Dans [Lisani 03] [Dibos 05] le modèle *a contrario* est appliqué aux différences d'orientation du gradient spatial d'intensité pour détecter le changement. Les décisions sont faites sur des zones carrées de tailles variables déplacées sur toute l'image. Dans le cas de la méthode développée par Viet [Veit 04b], le modèle *a contrario* est utilisé pour faire une décision à partir des niveaux d'intensité de trois images successives. Les décisions sont faites sur des zones obtenues à partir d'un découpage en blocs, ou encore à partir d'une segmentation basée sur les niveaux d'intensité.

2.3 Conclusion et perspective sur les méthodes de l'état de l'art

La plupart des méthodes présentées jusqu'ici, tentent de faire une décision soit sur chaque pixel en utilisant un seuil global déterminé d'une manière *a posteriori* ou par une analyse statistique des intensités de l'image de référence, soit sur des régions de petite taille. Ces méthodes sont très simples à implémenter. Cependant, comme il est illustré dans les figures 2.2 et 2.3, l'utilisation d'un seuil plus élevé pour réduire le bruit dû à la caméra, ne permet pas de détecter les zones homogènes d'un objet en mouvement, autrement dit, elle ne donne pas une détection complète des objets mobiles ce que rend l'interprétation des zones de la scène difficile.

Il ressort de ce qui précède que même la valeur de l'intensité d'un pixel donné ne change pas d'une image à l'autre, il est probable qu'il appartienne à un objet en mouvement. Ce qui revient à dire que chaque configuration est probable. Ceci d'une part, d'autre part, la classification des pixels doit se faire avec la prise en compte de leurs voisinages. Cela nous amène directement à une modélisation par champs de Markov.

2.4 Approche proposée

Dans cette section nous détaillons notre approche automatique de détection de mouvement [Housni 09b] basée sur une modélisation des interactions spatio-temporelles entre étiquettes par un champ de Markov faisant intervenir trois images successives.

Le problème de détection de mouvement est un problème binaire, la résolution de ce problème consiste à déterminer le masque de changement Q . Les étiquettes pertinentes dans ce cas sont les suivantes :

$$Q(x, y) = \begin{cases} 0 & \text{Fixe} \\ 1 & \text{Mobile} \end{cases} \quad (2.3)$$

Si on considère que la caméra est fixe et que l'éclairage est quasi constant dans la scène, les changements temporels des intensités des pixels ne peuvent donc être dus qu'au mouvement des objets ou au bruit généré par le capteur. Cela conduit naturellement à prendre comme observation la valeur absolue de la dérivée temporelle de la fonction de luminance $I(x, y, t)$ qui est approchée numériquement par une différence entre les instants t et $t+1$ (2.1).

Comme un cas particulier d'une estimation bayésienne, nous essayons d'estimer le masque de changement Q tel que sa probabilité *a posteriori* compte tenu de l'image de différence $\zeta(s)$ est maximisée. Pour cela, nous considérons la variation de l'intensité des pixels dans l'image comme un phénomène aléatoire.

Après une modélisation par champs de Markov aléatoire et une simulation par l'utilisation de champs de Gibbs (chapitre 1 section 1.3), le problème de la détection de mouvement revient donc à minimiser une fonction d'énergie de la forme :

$$E(x) = \sum_{p \in X} D_p^{Obs}(x_p) + \sum_{p, q \in N} V_{p, q}(x_p, x_q)$$

La minimisation de cette fonction d'énergie, comme nous l'avons déjà montré dans le chapitre précédent, peut se faire par la méthode des coupes de graphes. Pour cela, un graphe est construit à partir de différence d'images (2.1) ou à partir de résultat de (2.2) sur un triplé d'images.

Afin de prendre en compte le fait que le changement d'intensité peut être dû au bruit et sous l'hypothèse qu'une zone homogène peut appartenir à un objet en mouvement, les fonctions à utiliser pour le calcul des poids des arcs terminales "t-link" doivent avoir l'aspect suivant:

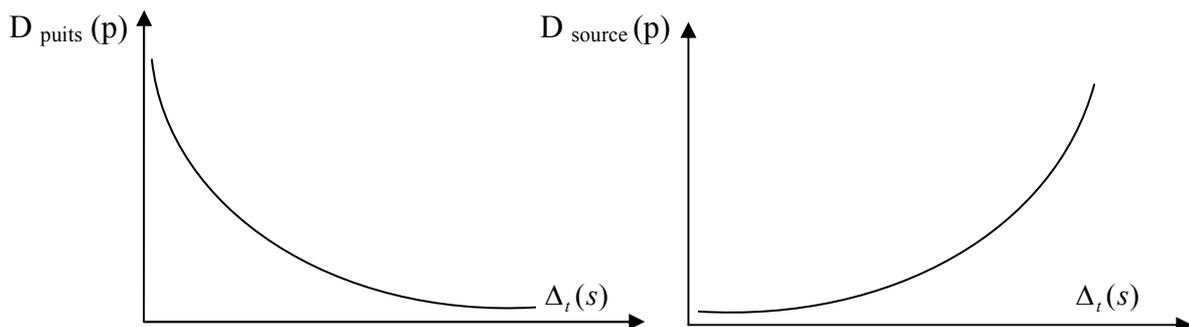


Figure 2.4 – Allures des fonctions à utiliser pour le calcul des poids des arcs t-link.

Pour le calcul du poids des arcs t-link, nous avons utilisé les fonctions suivantes:

$$W(p, source) = D_p^{Obs}(source) = |\Delta_t^p(s)|^2 / cste_1 \quad (2.4)$$

$$W(p, puits) = D_p^{Obs}(puits) = \exp(-|\Delta_t^p(s)|) / cste_2 \quad (2.5)$$

où $\Delta_t(s)$ est la différence entre deux (2.1) ou trois images successives (2.2).

Les constantes $cste_1$ et $cste_2$ sont choisies en fonction du lissage voulu.

Une arête entre deux noeuds voisins aura un poids :

$$W(p, q) = \exp\left(-\frac{|\Delta_t^p(s) - \Delta_t^q(s)|^2}{2 * \sigma^2}\right) \frac{1}{dist(p, q)} \quad (2.6)$$

où $dist(.)$ est la distance euclidienne et le paramètre σ contrôle le poids de l'arc.

Le terme de lissage (2.6) favorise la segmentation en régions d'intensités homogènes.

Nous avons testé notre méthode sur trois types d'informations prises comme observation : différence absolue entre deux images successives (2.1), entre trois images successives (2.2) et entre l'image de référence et l'image courante I_t (voir figure 2.5). Les figures 2.8 et 2.9 montre quelques résultats obtenus pour chaque cas.

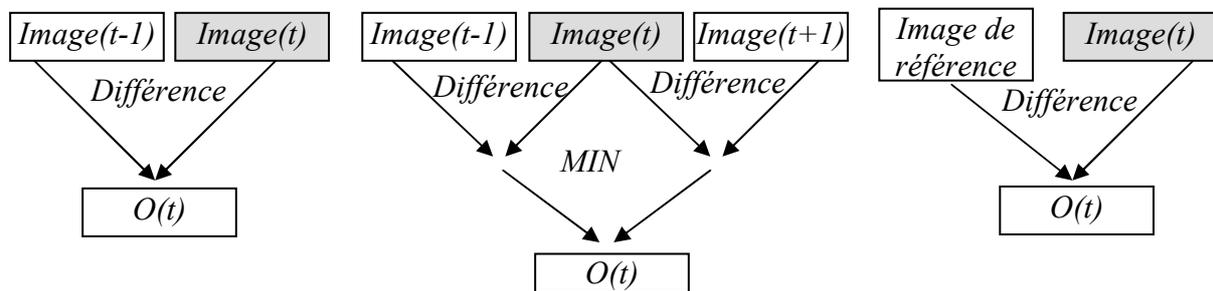


Figure 2.5 - Différents types d'informations qui peuvent être prises comme observation.

Les constantes $cste_1$ et $cste_2$ peuvent être aussi fixées automatiquement par une analyse de l'histogramme cumulé des différences d'intensité.

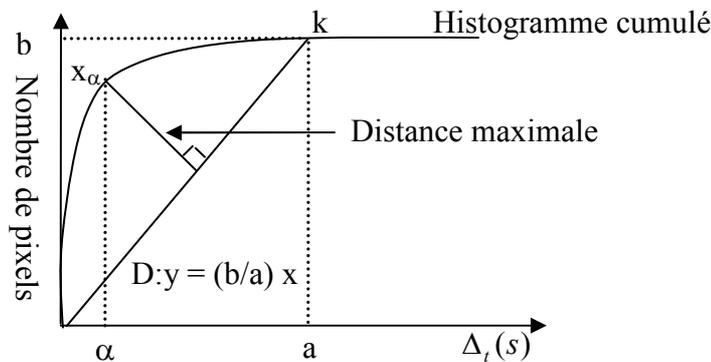


Figure 2.6 - Détection de point de changement dans l'histogramme cumulé ainsi la valeur à partir de laquelle l'histogramme devient constant.

Le point k correspond au point à partir duquel l'histogramme devient constant. Au niveau de ce point, on a $P(p \in \text{Objet en mouvement}) = \max$ ce que peut être traduit par $W(p, source)=1$ (valeur maximale) d'où on déduit que $cste_1 = a^2$.

Dans la figure 2.6, α correspond au point de changement dans l'histogramme des valeurs cumulées. Ce point est sélectionné tel que la distance de x_α à l'axe D est maximisée. Ce point représente le point d'équiprobabilité ; c'est-à-dire

$$P(p \in \text{Objet en mouvement}) = P(p \in \text{Arrière plan}) \rightarrow W(p, \text{source}) = W(p, \text{puits}).$$

Graphiquement on a :

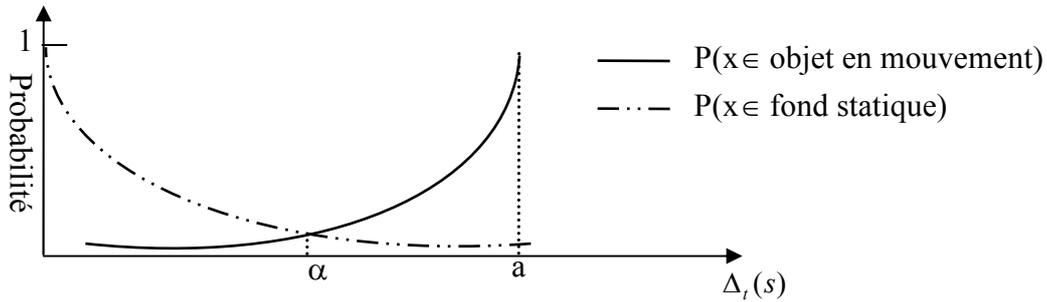


Figure 2.7 - La probabilité d'appartenance à l'objet ou au fond en fonction de différence d'intensité.

L'équation $W(p/\Delta_t(s) = \alpha, \text{source}) = W(p, \text{puits})$ permet de déterminer la constante $cste_2$.

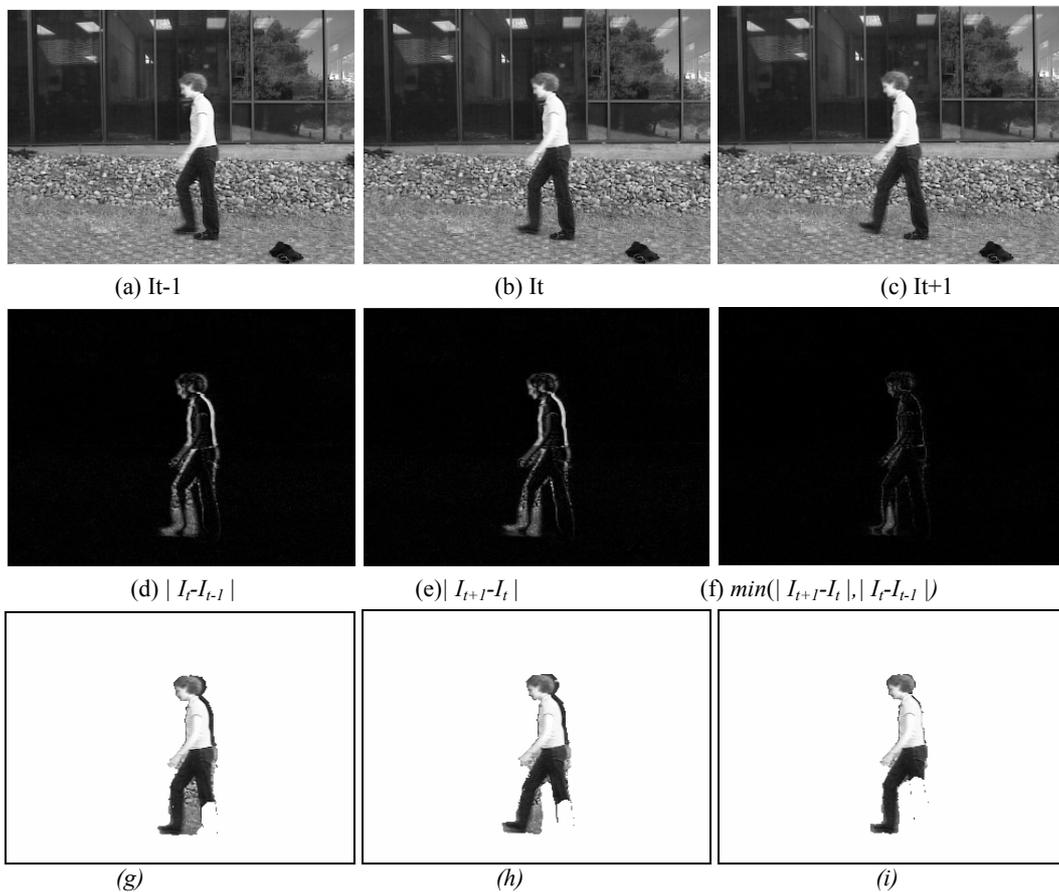


Figure 2.8 – (a) (b) et (c) trois images successives de la séquence walkstraight, (d) et (e) images de différence calculées par 2.1 (f) image de différence calculée par 2.2 (g) et (h) résultats obtenus par l'utilisation de (d) et (e). (i) résultat obtenu par l'utilisation de (f). On peut remarquer que l'utilisation de la différence de trois images successives permet de réduire l'erreur due aux occlusions.



Figure 2.9 – Résultat de détection de mouvement dans la séquence *Walk*. La première colonne contient les images originales. La deuxième colonne contient les résultats obtenus par l'utilisation de la différence entre trois images successives (2.2). La troisième colonne contient les résultats obtenus par l'utilisation de différence entre l'image de référence et l'image I_t .

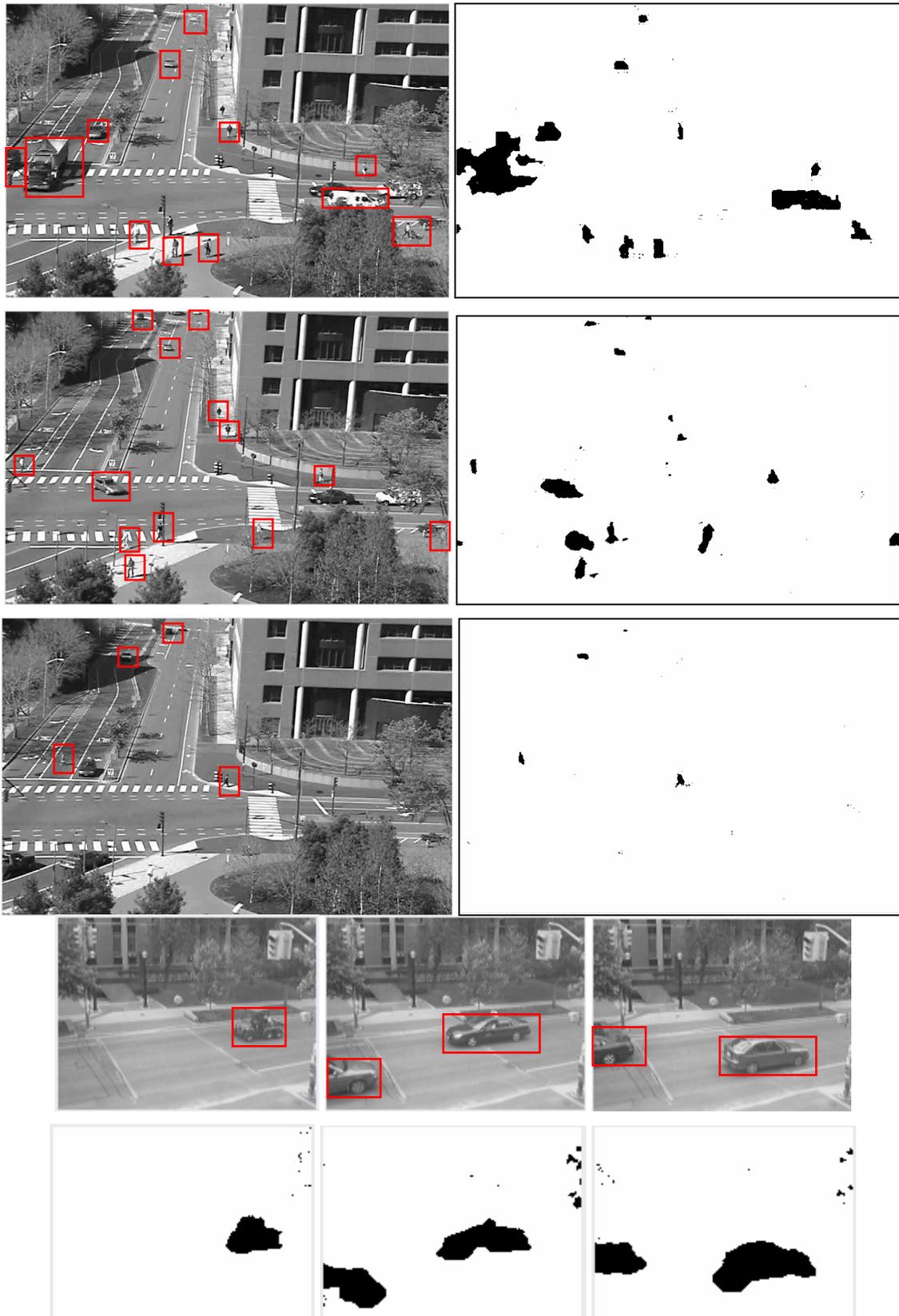


Figure 2.10 – Résultats obtenus sur deux séquences de trafic routier (*Browse et street_crossing*). Les rectangles correspondent aux régions associées à chaque objet. Cette estimation du mouvement des objets est satisfaisante puisque les objets détectés correspondent bien aux objets en mouvement.

2.5 Conclusion

Ce chapitre est principalement dédié à l'analyse et à l'étude de séquences de vidéo où la caméra est fixe dans un but de surveillance. Après une étude des méthodes de l'état de l'art, nous avons présenté comment la méthode des coupes de graphes peut être utilisée pour détecter les zones en mouvement. Cette méthode se décompose en trois étapes principales : le calcul de l'image de différence, la construction du graphe puis le calcul de la coupe minimale. Le principal avantage de notre méthode est qu'elle permet d'obtenir des cartes de détection du mouvement plus homogènes.

Les figures 2.8 et 2.9 montrent que l'utilisation de la différence de trois images successives, pour calculer les poids des arcs, permet de réduire l'erreur due aux occlusions partielles. Cependant, les résultats obtenus par cette technique peuvent contenir des trous à cause de l'information perdue au niveau des zones lisses ou encore au niveau des zones momentanément immobiles. L'utilisation de la différence par rapport à l'image de référence a permis de contourner de tel problème.

CHAPITRE 3

Etat de l'art des méthodes de suivi d'objets en mouvement

Le suivi d'objet en mouvement peut être défini comme étant la localisation spatio-temporelle d'un objet en mouvement au cours d'une séquence vidéo, où un objet est une zone de l'image qui peut être modélisée par des contours, silhouettes, primitives géométriques (rectangle englobant l'objet d'intérêt) ou encore par le point central. Cette localisation représente une tâche importante pour de nombreuses applications en vision par ordinateur à savoir :

- la surveillance automatique dont on cherche à détecter les activités suspectes ou inhabituelles par l'analyse de la trajectoire, de la vitesse ainsi que de la position d'objet en mouvement dans la scène ;
- la robotique (asservissement visuel) ;
- la reconnaissance de gestuelle ;
- la compression ;

Une approche robuste de suivi d'objets en mouvement doit résoudre trois grandes classes de problèmes :

- Problème de grands déplacements ;
- Problème de changement de topologie ;
- Problème de chevauchement des objets.

Pour résoudre ces problèmes, plusieurs approches ont été proposées dans la littérature : des méthodes basées sur un modèle d'objet (*model-based*) [Terzopoulos 92] [Isard 98] [MacCormick 00] [Wren 97] [Xu 02] [Rosenhahn 05] ou encore des méthodes basées sur le mouvement (*motion-based*) [Horn 81] [Barron 94] [Mansouri 03] [Adiv 85]. Cette diversité des approches est due aux contraintes du domaine d'application, les particularités des vidéos analysées, des caractéristiques (composantes) de l'image utilisées, de la nature du mouvement estimé, etc.

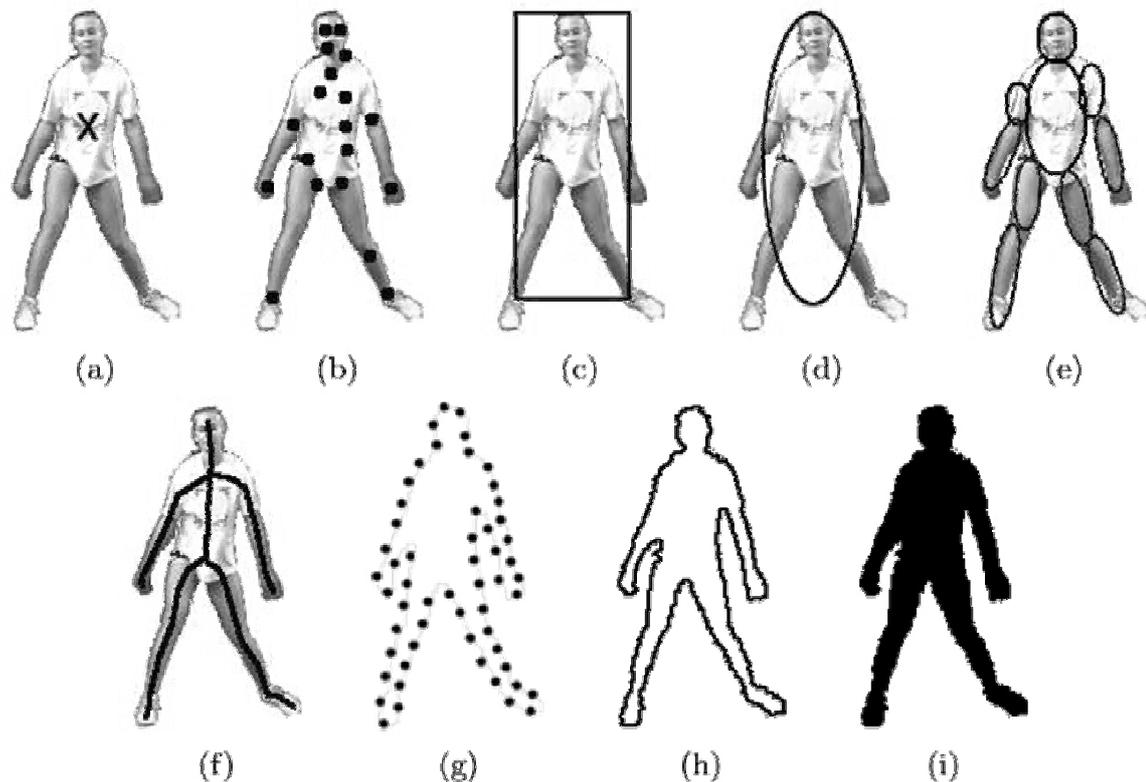


Figure 3.1 - Représentations d'objet. (a) centroïde, (b) plusieurs points, (c) fenêtre englobante rectangulaire, (d) fenêtre englobante elliptique, (e) modèle à base de blobs, (f) squelette d'objet, (g) les points de contrôle sur les contours de l'objet, (h) contour d'objet complet, (i) silhouette objet. Cette figure provient de [Yilmaz 06].

Dans ce chapitre nous présentons l'état de l'art des méthodes de suivi d'objets en mouvement. Nous ne détaillons pas ici les différents algorithmes mais nous donnons les avantages, les inconvénients et les domaines d'application des différentes familles de méthodes¹. Nous avons choisi de classer les différentes approches en quatre grandes catégories :

- Approches basées sur le suivi de points
- Approches basées sur le suivi de silhouettes
- Approches basées sur la correspondance de blocs
- Approches basées sur le suivi de noyaux

3.1 Suivi de points

Dans [Sethi 87] un objet est représenté par un ensemble de points. Le problème de suivi d'objets est formulé en termes de mise en correspondance de ces points en utilisant une

¹ Pour une analyse détaillée de l'état de l'art voir [Yilmaz 06]

approche gloutonne (*greedy*) basée sur des contraintes de proximité et de rigidité des objets. L'initialisation de l'algorithme est faite en associant les paires de points les plus proches. Ensuite un processus heuristique consistant à privilégier les associations qui favorise la cohérence des vitesses d'une image à l'autre est utilisé pour chercher des correspondances entre les paires de points. L'algorithme a l'avantage de simuler une prise en compte globale du problème de suivi. Cependant, il ne permet pas de prendre en compte les occultations.

En 2001, afin de résoudre le problème des occultations, Veenman et *al.* [Veenman 01] proposent l'utilisation de points fictifs créés par interpolation entre le dernier point de la trajectoire et le(s) point(s) candidat(s) de reprise de la trajectoire. Les occultations sont détectées en cherchant les trajectoires interrompues. Cependant, la contrainte de cohérence de mouvement n'est pas adaptée dans le cas où les points à suivre appartiennent à des objets ayant des mouvements différents.

Dans [Schmid 96], afin de réduire le taux d'erreurs dans le processus d'appariement de points, les auteurs introduisent une contrainte géométrique. Deux points voisins doivent être appariés à deux points également voisins dans l'image suivante. L'inconvénient de cette technique est le fait que les appariements corrects pourraient être aussi ignorés ce qui implique une augmentation de taux de vrais rejets.

Dans le même sens que les travaux de Schmid [Schmid 96], Trichet et *al.* [Trichet 07] utilisent une contrainte géométrique (triangulation de Delaunay) pour réduire le taux d'erreur. La contrainte utilisée est plus forte que celle utilisée par Schmid. Ceci implique un taux de vrais rejets plus élevé.

D'autres approches ont utilisé la technique de suivi multi-hypothèses MHT (*Multiple Hypothesis Tracking*). Cette technique consiste à utiliser, à chaque étape du suivi, les données collectées pendant les instants précédents pour examiner tous les cas possibles ; la poursuite d'une trajectoire existante, l'éventualité d'une occultation, l'apparition ou la disparition de nouveaux traits géométriques, ou bien la présence de fausses alarmes due à des données erronées. La technique des hypothèses multiples a été développée par Housewright et *al.* [Housewright 74] pour une unique cible et sans prise en compte du problème d'initialisation. Ces deux limitations ont été remédiées par Reid [Reid 79], plusieurs trajectoires sont initialisées et suivies dans un environnement encombré. Dans ce cas le problème de suivi peut être assimilé à un arbre d'hypothèse (décision) où chaque niveau représente une image et chaque branche représente une hypothèse. Les feuilles de bas niveau représentent les hypothèses actuelles. Les états des pistes sont estimés pour chaque hypothèse en utilisant un filtre de Kalman. L'avantage de la technique MHT est qu'une erreur de décision peut être rectifiée à la réception des nouvelles données. Cependant, elle est très coûteuse à la fois en mémoire et en temps d'exécution. Pour réduire le temps d'exécution, Cox [Cox 96] a utilisé l'algorithme de Murty [Murty 68] pour déterminer les k meilleures hypothèses en un temps polynomiale.

3.2 Suivi de silhouettes

« Une silhouette d'un point de vue optique est une vue d'un objet ou d'une scène comprenant le contour et un intérieur sans particularité » « **Wikipédia 22-09-2012** ».

Les méthodes de suivi basées silhouette reposent souvent sur une segmentation dynamique qui consiste à trouver, au moyen d'un modèle d'objet généré en utilisant les images précédentes, le contour ou la région relative à l'objet dans l'image courante. Ces méthodes sont souhaitables pour le suivi d'objet non rigide et sans informations *a priori* sur sa nouvelle forme à chaque instant.

On peut classer les méthodes basées silhouette en deux catégories : approches basées sur une représentation explicite du contour et d'autres utilisant la minimisation d'une fonction d'énergie sur contour (représentation implicite).

3.2.1 Approches explicites

Dans [Terzopoulos 92], l'objet à suivre est représenté par un ensemble de points de contrôle placés le long du contour (voir figure 3.1 (g)). Les auteurs ont utilisé un filtre de Kalman pour prédire ces points dans l'image courante, ceci permet d'obtenir le nouvel emplacement du contour. Le résultat de cette étape (étape de prédiction) est raffiné en utilisant le gradient de l'image. Une autre approche basée sur la représentation explicite de l'objet à suivre est présentée dans [Isard 98]. L'approche consiste à définir l'état d'un objet par un ensemble de splines et de paramètres de mouvement affine. La mise à jour de l'état de l'objet est faite par un filtrage particulière. Cette méthode a été étendue au cas multi-objets et à la gestion des occultations dans [MacCormick 00]. L'avantage des approches explicites est qu'elles permettent de suivre un objet avec un coût de calcul assez faible. Cependant, elles ne s'adaptent pas bien aux changements de topologie.

3.2.2 Approches implicites

Parmi les méthodes de suivi basées silhouettes, nous pouvons citer aussi celles utilisant un modèle de courbes déformables (connu sous le nom de contours actifs "*snake*" en anglais) [Paragios 00] [Peterfreund 00] [Isard 96] [Malik 97]. Ces méthodes tentent à faire évoluer une courbe placée autour de l'objet d'intérêt pour la rapprocher des zones de fort gradient (les frontières de l'objet) (figure 3.2). L'évolution de cette courbe est le résultat de la minimisation d'une fonctionnelle d'énergie.

Les algorithmes de contours actifs décrivent les objets à suivre plus simplement et plus efficacement et réduisent la complexité des calculs. Cependant, ils sont très sensibles à l'initialisation ce qui rend l'initialisation automatique difficile.

Dans [Xu 02], les auteurs ont utilisé les algorithmes des coupes de graphes pour le suivi d'un objet en mouvement. L'idée est, à partir de résultat obtenu à l'instant précédent, on détermine une zone d'intérêt en utilisant la dilatation (figure 3.3). Ensuite, le graphe est construit sur cette zone d'intérêt pour déterminer les nouvelles frontières. La fonction d'énergie associée à ce graphe ne comporte que des termes de régularité basés sur les gradients d'intensité de l'image.

3.3 Approches basées sur la correspondance de blocs

Les méthodes de mise en correspondance de blocs, connues sous le nom anglais "*block-*

matching" [Lee 97][Lin 98][Gyaourova 03][Hariharakrishnan 05], procèdent en deux étapes. La première étape consiste à diviser l'objet à suivre en blocs de taille égale (généralement de 8x8 ou 16x16 pixels) et chaque bloc est considéré comme étant un objet indépendant. La deuxième étape vise à mettre en correspondance les blocs d'une image à une autre à l'aide d'un critère de similarité (ou de ressemblance). Les critères les plus utilisés sont la différence moyenne absolue et l'erreur moyenne quadratique. Pour un état de l'art détaillé sur les algorithmes de mise en correspondance de blocs, consulter les travaux de Barjatya [Barjatya 04].

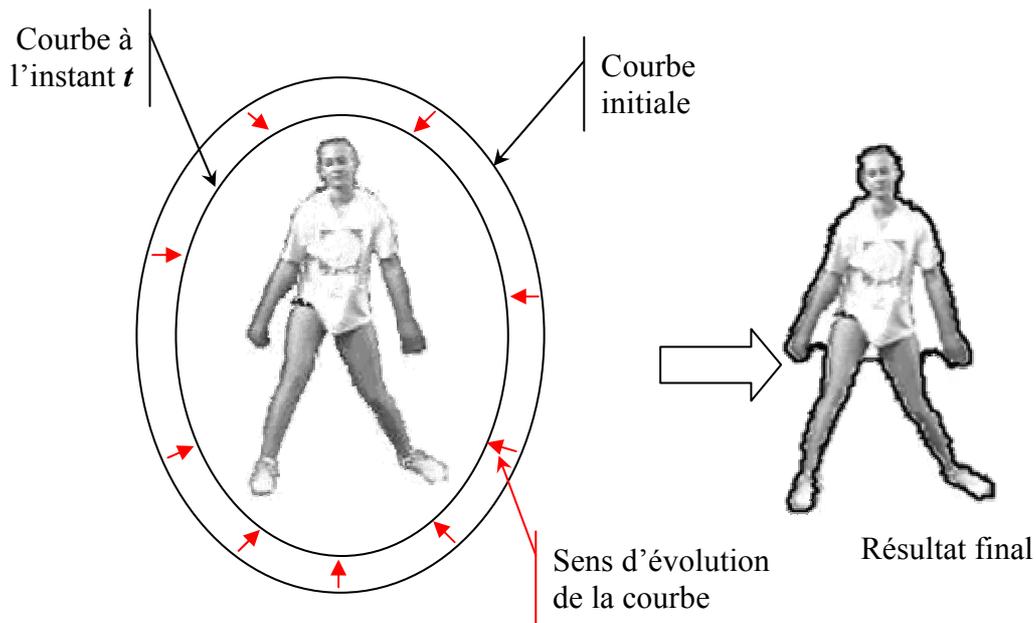


Figure 3.2- Evolution du contour actif vers l'objet d'intérêt. La courbe se déplace et épouse lentement les contours des objets en fonction de divers paramètres comme l'élasticité, la tolérance au bruit, etc.

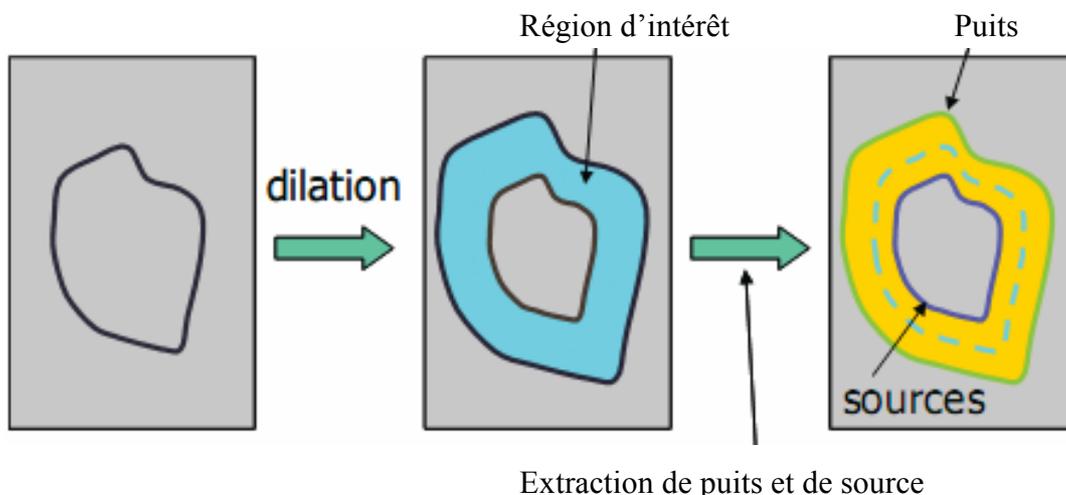


Figure 3.3 - Dilatation du contour pour obtenir l'emplacement de la source et du puits. Cette figure provient de [Xu 02].

3.4 Suivi de noyaux

Les méthodes de suivi de noyaux représentent l'objet à suivre par un rectangle, ellipse (figure 3.1 (c) et (d)) [Comanicu 03] [Mikram 06]. Dans [Comanicu 03], l'objet à suivre est représenté par un histogramme pondéré calculé sur une région elliptique contenant l'objet. La localisation de l'objet se fait à l'aide du procédé Mean Shift [Cheng 95] qui tente à maximiser itérativement la similarité d'apparence en comparant les histogrammes de l'objet et une fenêtre candidate. Le principal avantage de cette méthode est qu'elle permet d'estimer la position de l'objet en un nombre restreint d'itérations. Toutefois, dans le cas d'une scène contenant des objets ayant les mêmes distributions de couleurs et occupant des régions très proches, l'estimation sera ambiguë. Afin de contourner une telle ambiguïté, Mikram et *al.* [Mikram 06] utilisent une famille d'histogrammes élaborés à différentes échelles d'une image (figure 3.4).

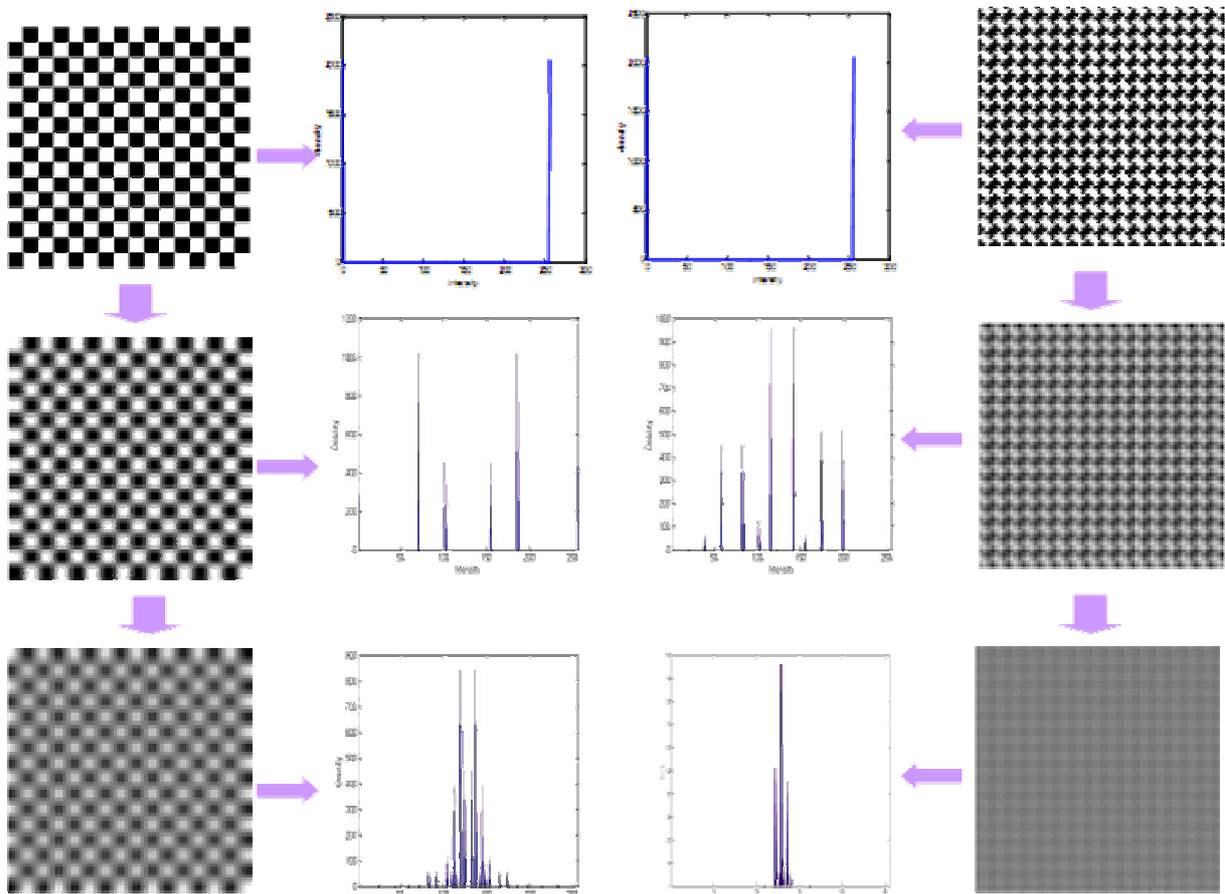


Figure 3.4 - Exemples de deux histogrammes multi-échelles. Les colonnes (a) et (d) illustrent la décomposition multi-échelles de deux images. Les colonnes (b) et (c) illustrent leurs histogrammes multi-échelles respectifs. Les histogrammes originaux sont identiques mais les histogrammes multi-échelles sont distincts. Cette figure provient de [Mikram 06].

3.5 Conclusion

Dans ce chapitre, nous avons présenté l'état de l'art des méthodes de suivi d'objets en mouvements. Chaque méthode tente de résoudre un problème bien déterminé en utilisant une modélisation d'objet adéquate qui dépend des propriétés utilisées (mouvement, histogramme de couleurs, etc). Le but était de donner un bref aperçu des types de méthodes existants.

Il ressort de ce chapitre qu'il existe un réel besoin de méthodes générales robuste au changement de topologie et capable de gérer les occultations.

CHAPITRE 4

Une nouvelle approche de suivi d'objets en mouvement basée sur les coupes de graphes

Dans ce chapitre, nous mettons en place un algorithme de suivi d'objets qui intègre à la fois le suivi et la segmentation d'objets en mouvements [Housni 12a]. Cette approche est basée sur le calcul d'une coupe de coût minimale dans un graphe et les paramètres de mouvement affine estimés entre chaque deux images successives.

4.1 Principe

Nous avons choisi de mettre en place un système de suivi pour n'importe quel type de vidéo (caméra en mouvement ou non). Pour cela nous avons choisi de diviser le processus de suivi d'objets en quatre étapes:

- Etape d'initialisation : pour la sélection des objets à suivre nous avons utilisé la méthode de segmentation interactive à base des coupes de graphes suivi d'une identification des objets par l'algorithme des composantes connexes.
- Etape de prédiction : pour gérer le déplacement des objets, nous avons utilisé une méthode de prédiction basée sur le modèle de mouvement calculé entre deux images successives (image courante où on veut localiser les objets suivis et l'image précédente), ceci va nous permettre de gérer les grands déplacements.
- Etape de raffinement : pour résoudre le problème du changement topologique, nous avons utilisé une méthode de segmentation dynamique basée sur la détection des contours par Coupes de graphes. Cependant, le raffinement n'est fiable que dans le cas où les objets sont isolés. Le raffinement sera ambigu dans certains cas, notamment dans le cas des objets qui se chevauchent.

- Etape de gestion des occultations : Le raffinement par les coupes de graphes ne permet pas de classer les pixels des objets qui se chevauchent. L'introduction d'un critère d'apparence visuelle pour la classification permettra de contourner de telles ambiguïtés. A cette fin, nous avons choisi d'utiliser une mesure de similarité basée sur les intensités observées.

La figure 4.1 donne le schéma général de notre approche.

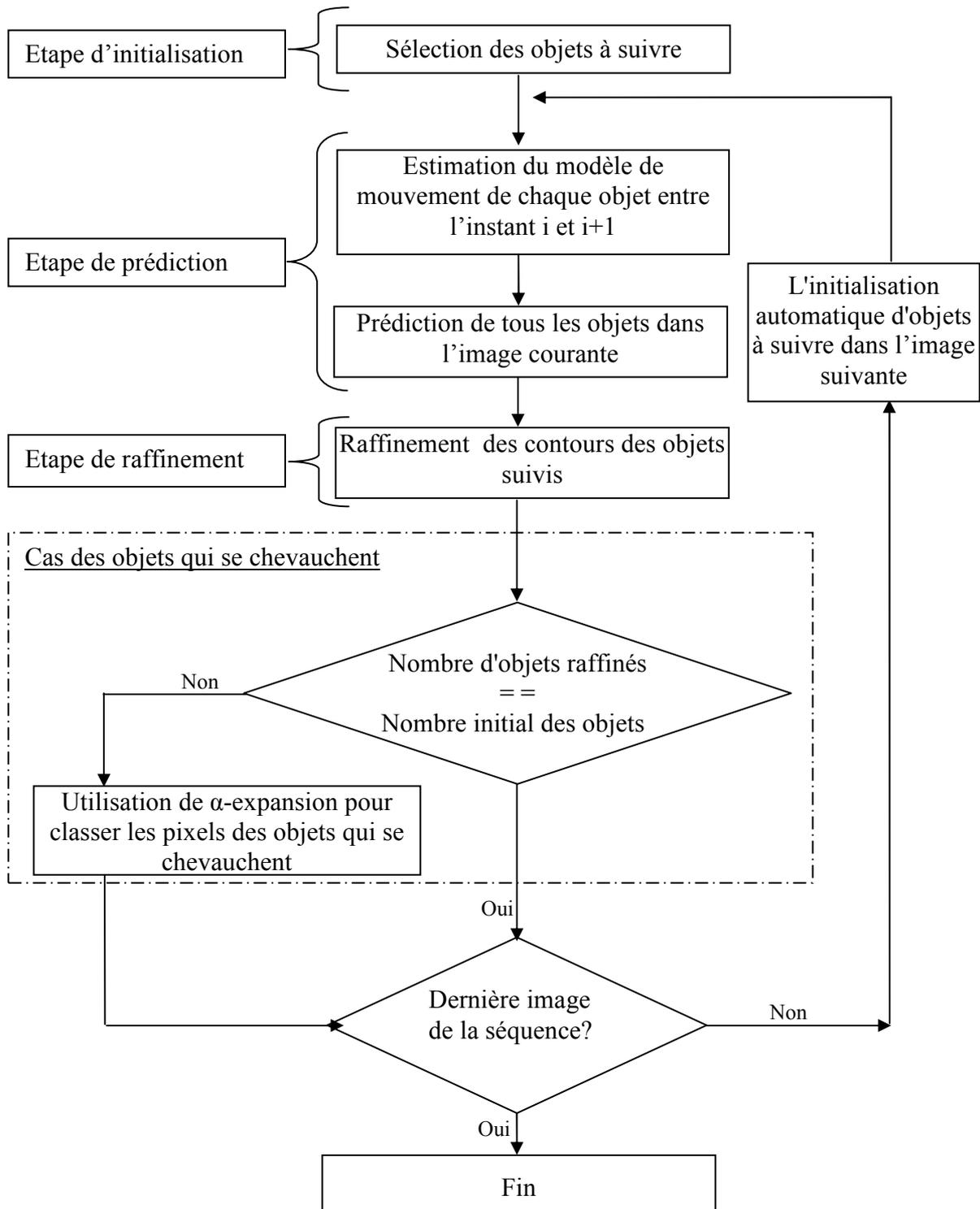
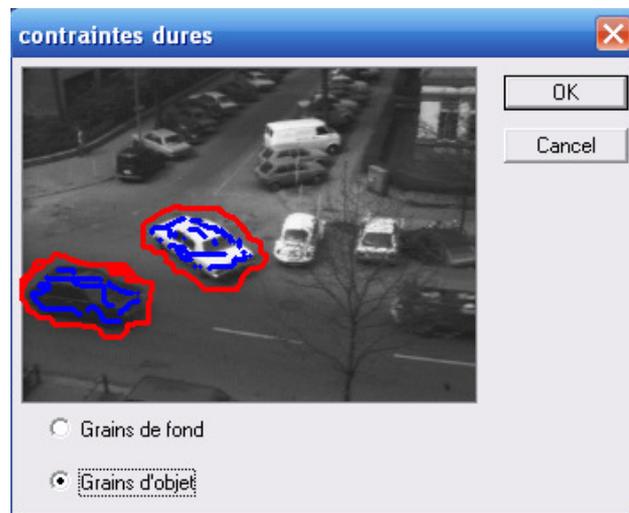


Figure 4.1 - Schéma général de notre approche.

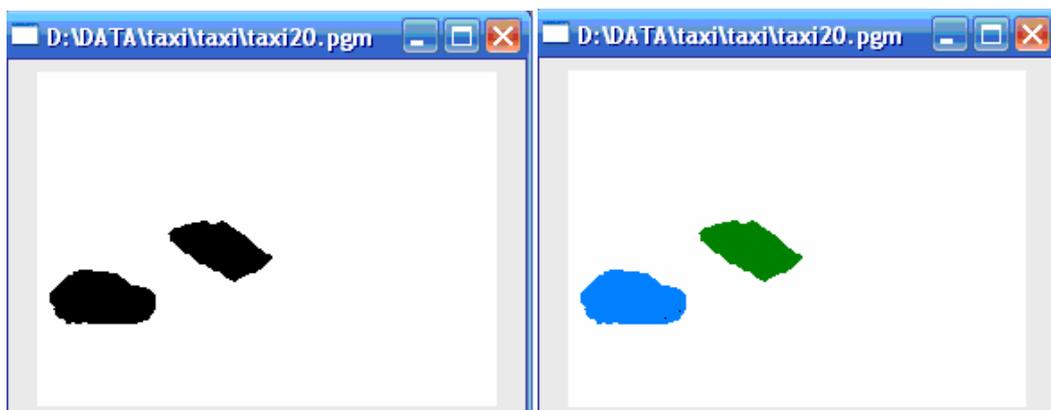
4.2 Sélection des objets à suivre

Dans toute technique semi-automatique, l'information initiale requise de la part de l'utilisateur doit être facile à saisir. La sélection des objets à suivre repose sur une approche utilisant une segmentation binaire et l'étiquetage des composantes connexes.

Pour le cas d'extraction d'objets d'intérêts par les coupes de graphes (connue sous le nom "méthode de segmentation interactive"), l'utilisateur doit marquer quelques pixels comme étant une partie des objets à extraire "*graines d'objets O*" ou du fond "*graines de fond F*". La segmentation doit se faire sans que les nœuds du graphe correspondant aux graines $\{O, F\}$ ne puissent changer d'étiquette (voir chapitre 1 section 1.6).



(a)



(b)

(c)

Figure 4.2 – L'interface utilisée pour la saisie de l'information initiale "contraintes fortes" (a) image avec des contraintes fortes, (b) les objets sélectionnés par les coupes de graphes, (c) l'identification des objets sélectionnés par l'algorithme de détection des composantes connexes.

Le résultat des coupes de graphes est une image binaire dont certains pixels sont étiquetés comme étant des pixels des objets segmentés et d'autres comme étant des pixels du fond (figure 4.2 (b)). L'étiquetage de l'image pour distinguer chaque objet se fait par l'algorithme des composantes connexes qui consiste à trouver les groupes de pixels connectés, appelés composantes connexes ou blobs, à partir d'une image binaire. Comme nous l'avons déjà introduit dans la section 1 du premier chapitre, la notion de connexité dérive de la notion de voisinage. Dans nos expérimentations, nous avons utilisé une connexité d'ordre 8 (figure 4.3). A noter que le choix de l'ordre de connexité influence sur le résultat.

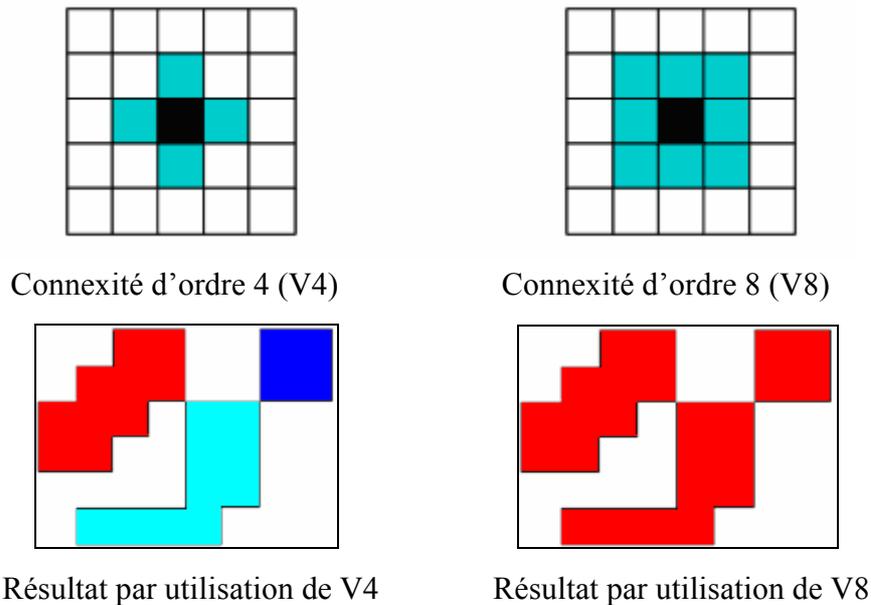


Figure 4.3 - Connexité et voisinage

Pour la détermination des composantes connexes, plusieurs algorithmes peuvent être utilisés à savoir : l'étiquetage séquentiel itératif, l'étiquetage séquentiel avec correspondance entre points, l'étiquetage par un parcours en profondeur, etc.. Dans nos expérimentations nous, avons utilisé l'algorithme d'étiquetage par un parcours en profondeur (algorithme 4.1).

Après la sélection des objets à suivre, l'étape suivante consiste à calculer pour chaque objet le modèle de mouvement caractérisant son déplacement. Ces modèles seront ensuite utilisés dans l'étape de prédiction.

4.3 Modèles de mouvement

La clé de tout système de suivi d'objets basé mouvement, réside dans le choix du modèle de mouvement pour chaque objet, car celui-ci doit représenter le mouvement de la majorité des pixels relatifs à un objet donné.

Notre objectif est de contrôler le déplacement des objets à suivre en exploitant une information *a priori* sur leur emplacement dans l'image précédente. Comme nous l'avons déjà énoncé, le modèle de mouvement sera utilisé pour prédire les objets dans l'image courante.

Algorithme 4.1 Etiquetage par un parcours en profondeur

```

Pour chaque x dans V faire // V est l'ensemble des pixels de l'image
| Etat[x] ← non Attend
Fin pour
L ← 1, A-traiter ← Vide
Pour chaque x dans V faire
| Si Etat[x] = non Attend alors
|   A-traiter ← A-traiter ∪ {x}
|   Etat [x] = Attend
|   Tant que A-traiter est non vide faire
|     Y ← un élément dans A-traiter
|     A-traiter ← A-traiter - {Y}
|     Pour chaque z ∈ V8(y) faire
|       Si z est un point objet et Etat[z] = non Attend alors
|         A-traiter ← A-traiter ∪ {z}
|         Etat [z] = Attend
|       Fin si
|     Fin Pour
|     Label [y] ← L
|     Etat [y] ← Examiné
|   Fin Tant que
|   Fin si
  L ← L+1
Fin Pour

```

Dans la littérature, il existe une large variété de façons de représenter, en terme mathématique, le déplacement des pixels ou des régions d'une image à l'autre. (Voir chapitre 5). Nous pouvons citer comme exemples le modèle homographie, le modèle rigide (projectif par exemple), le modèle quadratique, etc.. Le modèle de mouvement affine (4.1) est un compromis raisonnable entre une faible complexité et une bonne approximation. Ce modèle est une modélisation paramétrique à six paramètres ($c_1, c_2, a_1, a_2, a_3, a_4$) du mouvement définissant la transformation $T: I_t \rightarrow I_{t+1}$ qui permet de passer de pixel $I_t(P_i)$ au pixel $I_{t+1}(P_i')$.

$$\vec{\omega}_A(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} * \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (4.1)$$

Plusieurs méthodes ont été proposées pour l'estimation des modèles de mouvement paramétrique. La méthode la plus célèbre dans ce domaine est celle de Odobez et al. [Odobez 98] basée sur un estimateur multi-résolution [Odobez 95] qui permet de fournir en une seule itération des modèles paramétriques jugés suffisamment précis. Dans notre application, nous

avons utilisé cette méthode grâce à sa rapidité. Une implémentation de cet algorithme entre deux images successives peut être trouvée dans <http://www.irisa.fr/vista>.

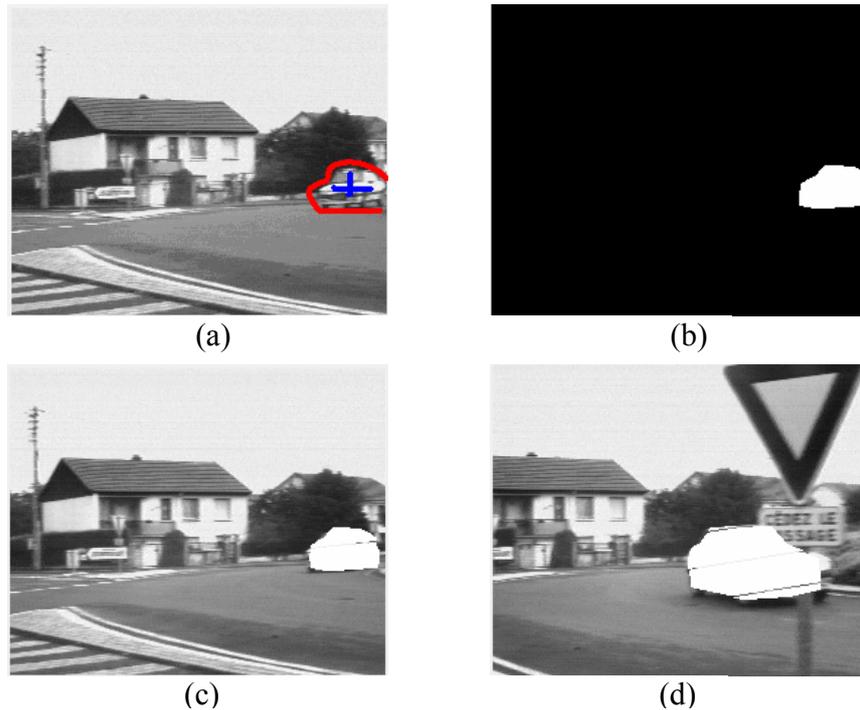


Figure 4.4 - (a) Première image avec des contraintes fortes, (b) l'objet à suivre sélectionné par la méthode des coupes de graphes, (c), (d) L'objet prédit dans les images 2, 23 de la séquence rond-point à l'aide du modèle de mouvement.

4.4 Raffinement des frontières des objets prédits

La prédiction par le modèle de mouvement ne permet pas dans tous les cas de suivre avec précision un objet en mouvement, surtout dans le cas de changement de topologie. Dans la figure 4.5, nous pouvons voir clairement qu'une partie de la route "fond" a été marquée comme une partie d'objet et inversement une partie de la voiture "objet" a été marquée comme partie de fond.

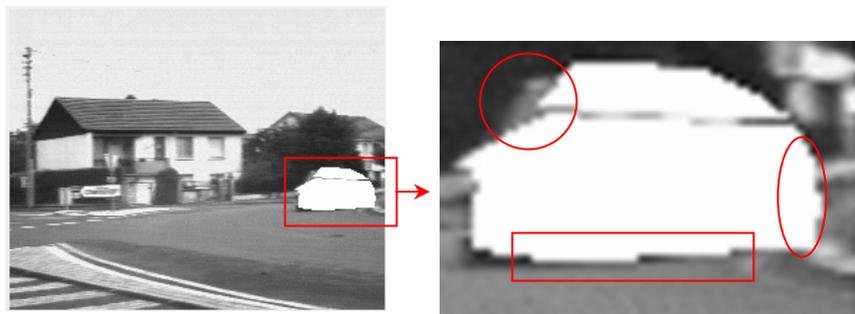


Figure 4.5 – Résultat d'une prédiction par le modèle de mouvement

Pour obtenir une description riche de l'objet, nous avons utilisé la méthode des coupes de graphes décrite dans la section 6 du premier chapitre. Cette méthode nécessite une

initialisation de la part de l'utilisateur (grains d'objet et de fond). Afin de générer automatiquement cette information nécessaire pour l'initialisation, nous avons utilisé la morphologie mathématique :

- Les graines de l'objet: quatre fois l'érosion¹ de l'objet prédit en utilisant le modèle de mouvement (figure 4.6 (b)).
- Graines de fond: résultat de la différence absolue d'images I_1 et I_2 . Où I_1 est le résultat de dilatation sept fois de l'objet prédit et I_2 est obtenu par la dilatation cinq fois l'objet détecté (figure 4.6 (c)).

Ceci permet d'optimiser les frontières de l'objet avec une précision de ± 9 pixels.

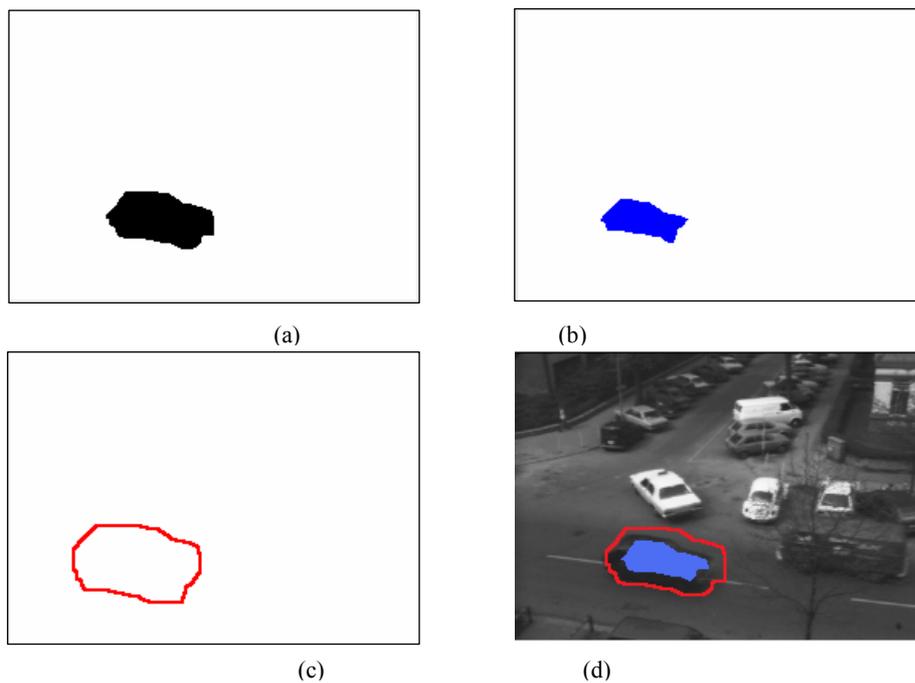


Figure 4.6 – Génération automatique des grains d'objet et de fond.

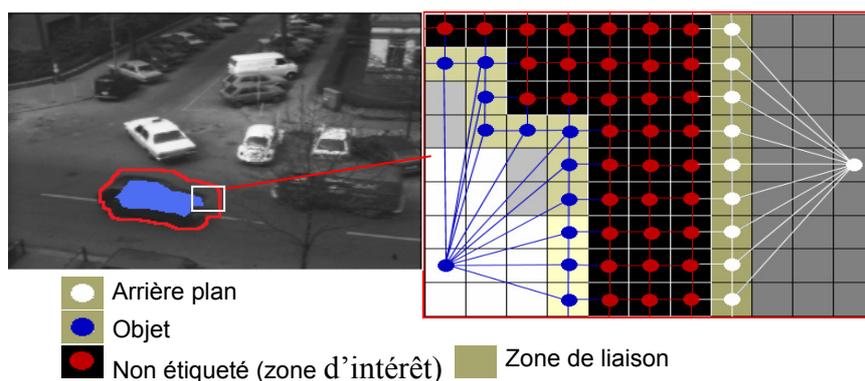


Figure 4.7 - Détection optimale des contours par les coupes de graphes. Le graphe est construit sur la zone d'intérêt située entre les grains d'objet et de fond pour déterminer les nouvelles frontières.

¹ Erosion morphologique par l'élément structurant V8.

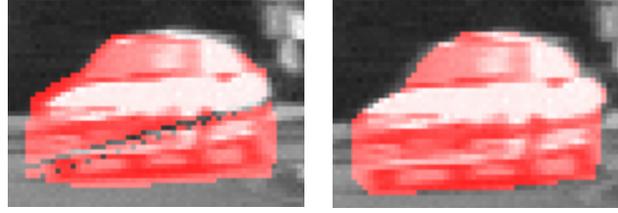


Figure 4.8 - Première image donne le résultat obtenu en utilisant le modèle de mouvement uniquement, seconde image donne le résultat obtenu après le raffinement par des coupes de graphes.

4.5 Gestion des problèmes d'occultation

A chaque instant, avant de calculer le modèle de mouvement pour chaque objet, nous utilisons l'algorithme de détection des composantes connexes afin d'étiqueter les objets. Cet algorithme permet de classer les pixels de l'image en fonction de leur appartenance à Objet_1, Objet_2 ... Objet_n ou au fond. Cependant, la classification n'est pas fiable que lorsque les objets prédits sont isolés et par conséquent, le suivi multi-objets peut être ambigu, dans certains cas, notamment dans le cas de chevauchement des objets.

Pour contourner de telles ambiguïtés, nous avons introduit une contrainte d'apparence visuelle, où une classification selon les intensités observées est prise en compte. A cette fin, nous avons choisi d'utiliser l'algorithme α -expansion [Boykov 01a] pour classer les pixels des objets qui se chevauchent.

Pour le calcul du poids des arcs, nous avons utilisé les fonctions suivantes:

$$si \begin{cases} I_p \notin objet_{i(i=1,2...n)} \text{ et} \\ I'_{\omega_{Ai}(p)} \in objet_{i(i=1,2...n)} \end{cases} \begin{cases} W(\omega_{Ai}(p), L_{i(i=1,2...n)}) = 0 \\ W(\omega_{Ai}(p), B) = 0 \end{cases} \quad (4.2)$$

$$si \begin{cases} I_p \in objet_{i(i=1,2...n)} \text{ et} \\ I'_{\omega_{Ai}(p)} \in objet_{i(i=1,2...n)} \end{cases} \begin{cases} W(\omega_{Ai}(p), L_{i(i=1,2...n)}) = \exp(-\|I_p - I'_{\omega_{Ai}(p)}\|) \\ W(\omega_{Ai}(p), B) = 0 \end{cases} \quad (4.3)$$

$$si I'_p \in Arrière \quad plan \begin{cases} W(p, L_{i(i=1,2...n)}) = 0 \\ W(p, B) = \infty \end{cases} \quad (4.4)$$

$$W(p, q) = \exp\left(-\frac{|I_p - I_q|^2}{2 * \sigma^2}\right) \frac{1}{dist(p, q)} \quad (4.5)$$

Où $objet_{(i=1,2...n)}$ est l'un des objets à suivre, I_p est l'intensité du pixel p dans l'image I , ω_{Ai} est le modèle de mouvement affine correspond à l'objet i , $I'_{\omega_{Ai}(p)}$ est l'intensité du pixel $\omega_{Ai}(p)$ dans la l'image I' , $\omega_{Ai}(p)$ est l'image de p par le modèle de mouvement ω_{Ai} , $L_{i(i=1...n)}$

est l'un des nœuds terminaux (terminal correspondant à l'objet i), B est le nœud terminal qui correspond à l'arrière-plan et σ peut être estimé comme "bruit de la caméra". (4.2), (4.3) et (4.4) sont utilisés pour calculer les capacités des arcs E_T reliant les nœuds P aux terminaux. Pour les arcs E_N reliant les nœuds voisins, les capacités sont calculées par (4.5).

4.6 Résultats expérimentaux

4.6.1 Données

La méthode a été testée sur cinq séquences (*taxi*, *rond-point*, *walk*, *croisement* et *Coastguarder*). La première séquence a 41 images de taille 191 x 256. La deuxième séquence a 33 images de taille 256 x 224. La troisième séquence a 127 images de taille 300 x 200. La quatrième séquence a 27 images de taille 300 x 200. La dernière séquence a 80 images de taille 300 x 240.

Dans les séquences *rond-point*, *croisement* et *coastguarder* la caméra est en mouvement, tandis qu'elle est fixe dans les deux autres séquences.

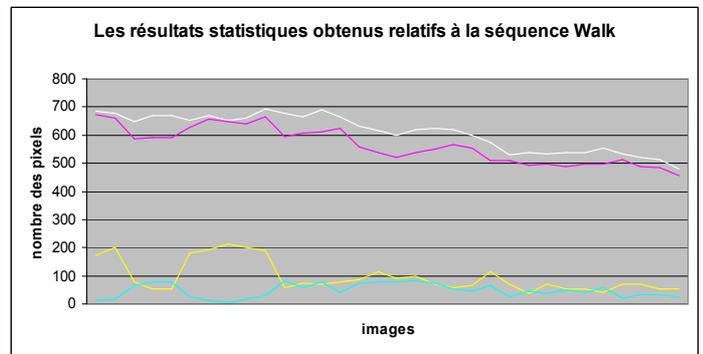
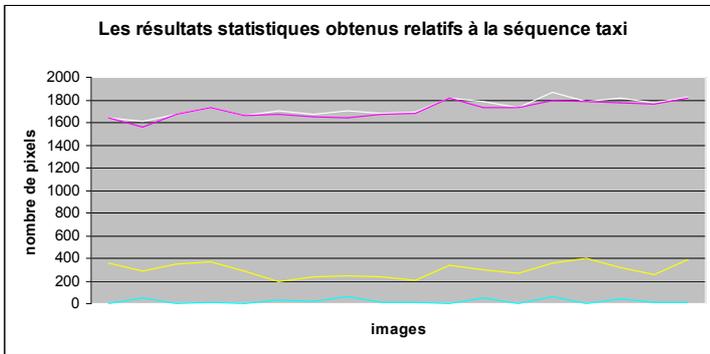
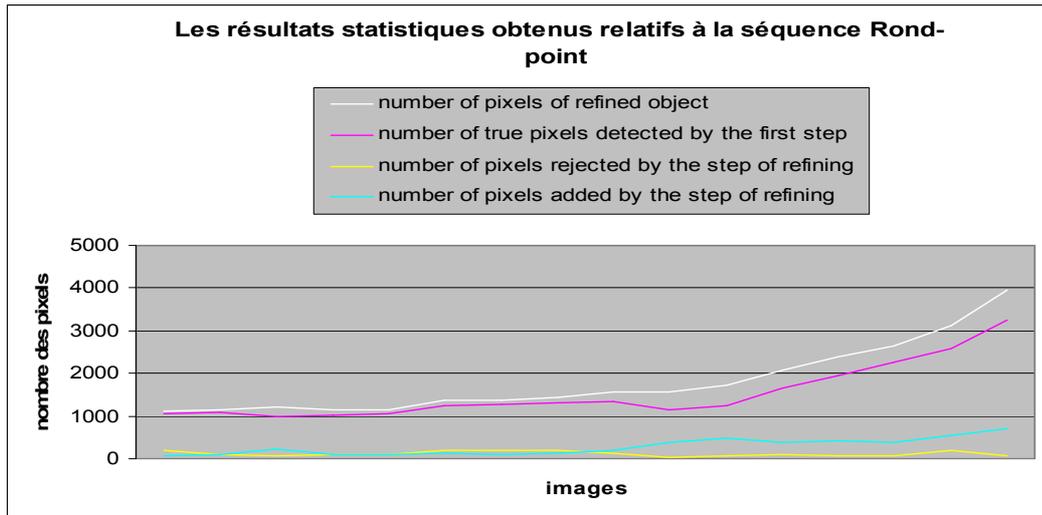
4.6.2 Résultats et complexité

L'approche des coupes de graphes est souvent utilisée pour résoudre un problème d'étiquetage. Sachant que, la complexité pour résoudre un tel problème dépend du nombre de pixels à étiqueter. Et afin de réduire la complexité d'étiquetage par les coupes de graphes, nous n'avons utilisé que le rectangle qui contient l'objet à traiter au lieu d'utiliser la totalité de l'image, cette méthode nous permet de réduire la taille du graphe. Par cette réduction de la taille des données, et en utilisant l'algorithme rapide décrit dans [Boykov 00], la coupe de graphe est faite en un temps réel.

Sur la figure 4.9, on peut d'une part voir que la courbe représentant le nombre de pixels détectés dans l'étape de prédiction est en dessous de la courbe représentant le nombre de pixels de l'objet raffiné. D'autre part, la courbe représentant le nombre de pixels rejetés par l'étape de raffinement ne touche jamais l'axe des x , ce qui montre que le modèle de mouvement (appelé mouvement dominant) ne couvre pas toujours tous les pixels de l'objet suivi. Compte tenu de ce qui précède, on conclut que le résultat de la prédiction par le modèle de mouvement ne représente pas précisément l'objet mais peut représenter une bonne approximation pour l'étape de raffinement.

La figure 4.10 montre la variation de la taille des objets suivis entre la première et la dernière image. Ce changement de taille peut être dû au rapprochement (séquence *rond-point*) ou l'éloignement (séquence *walk*) des objets de la caméra ou encore au changement de la topologie. Ce changement de la taille est géré par l'étape de raffinement.

Nous avons implémenté notre approche en C++. Toutes les expérimentations ont été effectuées sur Intel Core 2 Duo 2x2 GHz, 2 Go de mémoire, Windows XP. Le tableau 4.1 donne le temps d'exécution dans le cas de suivi d'objets sans chevauchement.



Calcul du modèle de mouvement	construction du graphe	Calcul de coupe minimale
0.013 s	0.005 s	0.021 s

Tableau 4.1 – Le temps d'exécution moyen par image (à noter que pour réduire le temps de calcul de coupe minimale, nous avons utilisé seulement le rectangle qui contient l'objet à raffiner au lieu d'utiliser la totalité de l'image).

Dans le cas du suivi des objets qui ne se chevauchent pas, le temps total d'exécution par image est 0.039 s. Dans ce cas, le suivi peut se faire en temps réel. Dans l'autre cas, à ce coût, il faut ajouter le temps de calcul nécessaire pour classer les pixels des objets qui se chevauchent (= nombre d'objets * le temps nécessaire pour la construction de graphe et le calcul de la coupe minimale), par exemple, pour la séquence *croisement* (deux objets se chevauchent) le temps exécution moyen par image est 0.094 s.

Figure 4.11 et 4.12 donne quelques exemples des résultats expérimentaux.



Figure 4.11 - Résultats obtenus dans des images 7, 13, 18 et 22 de la séquence *rond-point*.

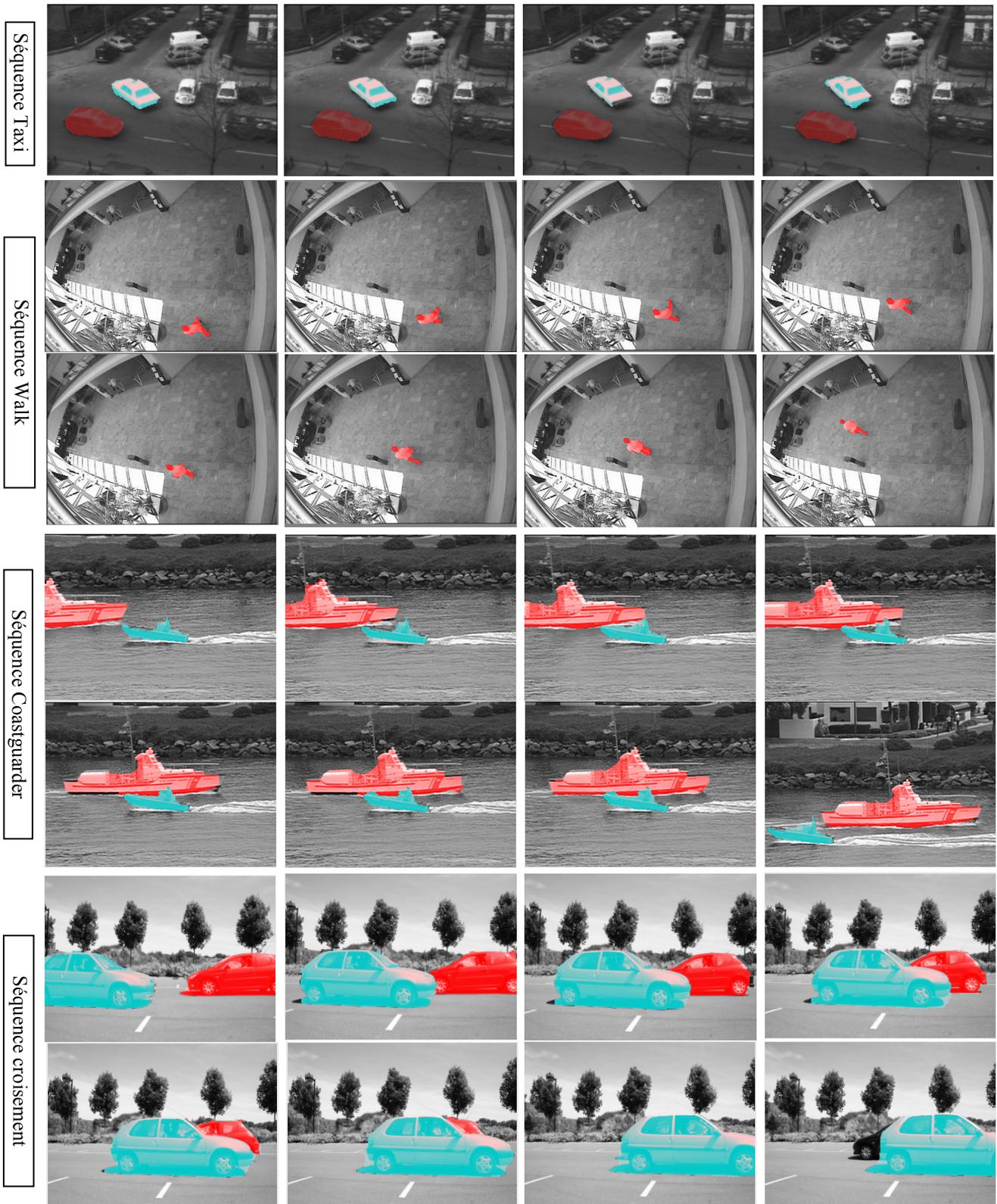


Figure 4.12 - Résultats expérimentaux. Nous pouvons voir dans la dernière image de la séquence *croisement* la réapparition de l'objet (voiture noire) alors que le programme n'a pas pu le détecter. Ceci est normal car l'objet a disparu dans l'image précédente et l'algorithme que nous avons proposé utilise le résultat de l'étape précédente pour localiser les objets dans l'image courante.

4.7 Conclusion et perspectives

Dans ce chapitre, nous avons présenté comment nous pouvons suivre des objets en mouvement dans une séquence vidéo en utilisant l'approche des coupes de graphes et le flot optique modélisé par le modèle affine à six paramètres. L'approche combine les avantages de différentes méthodes de suivi existantes (gestion des grands déplacements, de changement de topologie et des occultations). Globalement, les résultats expérimentaux (figures 4.11 et 4.12) montrent la bonne performance et la robustesse de notre approche.

Afin de suivre indépendamment chaque objet, la classification multi-étiquettes par l'algorithme α -expansion est ajoutée à la méthode de suivi. Elle permet de séparer les objets ayant fusionné après l'étape de raffinement des frontières.

D'autres recherches se concentreront sur l'amélioration de l'algorithme en définissant des critères permettant d'identifier les propriétés caractérisant chaque objet afin de pouvoir retrouver les objets disparus. Nous allons également essayer d'utiliser une initialisation plus sophistiquée (initialisation automatique).



Copier de [Mikram 06]. Résultat de suivi par la méthode de suivi de noyaux.



Nos résultats.

Figure 4.13 - Comparaison des résultats de suivi d'objet sur la séquence *walk*.

Deuxième partie :

*Détection de mouvements cohérents
et extraction de couches de même
mouvement*

Introduction de la deuxième partie

La détection de mouvements cohérents est une tâche qui consiste à grouper des mesures locales de mouvement en groupes de mouvement cohérent. Les mesures locales sont déterminées, soit au niveau de chaque pixel (flot optique) [Horn 80] [Horn 81], soit au niveau des points d'intérêt [Lowe 04] [Tomasi 91] [Ling 07] [Rabin 09].

Dans cette partie nous présentons un état de l'art et une nouvelle approche de détection de mouvements cohérents basée, en premier temps, sur des mesures locales calculées au niveau des points d'intérêt puis sur les modèles des mouvements paramétriques afin de fusionner les groupes cohérents en sens de mouvement paramétrique.

Un découpage en deux chapitres a été adopté pour la présentation de cette partie.

Dans le chapitre suivant (chapitre 5), nous présentons l'état de l'art des méthodes:

- de calcul des mesures locales basées sur la détection et la mise en correspondance des points d'intérêt ;
- d'estimation de mouvement ;
- de détection de mouvements cohérents.

Dans le dernier chapitre (chapitre 6) nous présentons une nouvelle approche de détection de mouvements cohérents et d'extraction des couches de même mouvement basée sur les points d'intérêt et le modèle de mouvement paramétrique

CHAPITRE 5

Etat de l'art des méthodes de détection des points d'intérêt, d'estimation de mouvement et de détection de mouvements cohérents



Figure 5.1 – La recherche d'image par la détection et la mise en correspondance des points d'intérêt SIFT (résultat obtenu par le logiciel SiftDimo¹).

Dans ce chapitre, nous présentons quelques méthodes utilisées dans la détection de mouvements cohérents. Le chapitre est organisé comme suit : la section 1 présente l'état de l'art des méthodes de détection et de mise en correspondance des points d'intérêt. La section 2 présente quelques méthodes d'estimation de mouvements existantes dans la littérature ainsi

¹ <http://www.cs.ubc.ca/~lowe/keypoints/>

qu'une version modifiée de celle utilisée par Fradet [Fradet 10] adaptée au cas des points d'intérêt et nous montrons que nous pouvons calculer le modèle de mouvement par une méthode directe basée sur la régression plane 3D. Ceci va nous permettre de réduire le temps de calcul à l'inverse des méthodes itératives classiques. La dernière section présente l'état de l'art des méthodes de détection de mouvements cohérents.

5.1 Détection et mise en correspondance des points d'intérêt

Les données vidéo sont sujettes à des perturbations qui peuvent venir de bruit lié au capteur, changement de luminosité, déplacement des objets et/ou au mouvement de la caméra etc.. La recherche des caractéristiques invariantes aux différentes perturbations est un axe de recherche très actif. Les points d'intérêt sont des caractéristiques de bas niveau simples, robustes et qui permettent une bonne caractérisation de contenu sémantique d'une scène. Ces caractéristiques sont très utilisées dans plusieurs applications en vision par ordinateur, on peut citer à titre d'exemple l'estimation du mouvement [Dupont 06] [Housni 12b], détection des objets en mouvement [Veit 07], détection de mouvement cohérent [Veit 07] [Poon 09] [Housni 12b], etc..

Les points d'intérêt sont définis comme des points où apparaît un changement (discontinuité) significatif dans les niveaux de gris. Par exemple, les coins, les intersections, les points isolés et les points spécifiques sur les textures, sont des points d'intérêt. En pratique, la détection de ces points consiste à détecter les points présentant un fort rayon de courbure de l'intensité ; c'est-à-dire des variations du second ordre de celle-ci.

5.1.1 Détection des points d'intérêt

Dans la littérature, il existe une grande variété de points d'intérêt tels que les points d'intérêt KLT [Shi 94] [Tomasi 91], Harris [Harris 88], les descripteurs SIFT [Lowe 04], etc. Les lecteurs peuvent se référer à [Mikolajczyk 04] pour plus de détails. Dans cette section, nous présentons celles basées sur la détection de coins et la méthode SIFT basée sur la détection d'extrema locaux.

5.1.1.1 Détection de coins

Les coins ont une grande probabilité de réapparaître d'une image à une autre. Ils représentent des caractéristiques invariantes aux transformations géométriques simples et aussi au changement de contraste, d'où leur très large utilisation dans la littérature. La détection des coins revient à détecter les endroits où le gradient est fort dans plus d'une direction.

5.1.1.1.1 Le détecteur de Harris [Harris 88]

La formule de Harris qui permet de localiser les points caractéristiques d'une image I est donnée par :

$$E(x, y) = (x, y) \cdot M \cdot (x, y)^T$$

$E(x,y)$ caractérise la variation de la luminance au voisinage du pixel (x,y) .

$$\text{Avec } M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial L}{\partial x}\right)^2 & \left(\frac{\partial L}{\partial x} \frac{\partial L}{\partial y}\right) \\ \left(\frac{\partial L}{\partial x} \frac{\partial L}{\partial y}\right) & \left(\frac{\partial L}{\partial y}\right)^2 \end{bmatrix} \quad (5.1)$$

Où $L = g_{\sigma_D} * I$ est l'image lissée avec une gaussienne d'écart type σ_D afin d'atténuer toutes les structures de tailles inférieures à la largeur de la gaussienne. Cette procédure est particulièrement efficace contre le bruit.

La détection des coins consiste à détecter les points où M représente une forte variation dans toutes les directions. Pour cela, on calcule les valeurs propres de M (λ_1 et λ_2) et en fonction des valeurs calculées, on a trois cas possible :

1. Si $\lambda_1 \approx 0$ et $\lambda_2 \approx 0$, le pixel (x,y) n'est pas un point d'intérêt car le gradient est faible dans toutes les directions ;
2. Si $\lambda_1 \approx 0$ et $\lambda_2 \gg 0$, le pixel (x,y) appartient au bord d'une région ;
3. Si $\lambda_1 \gg 0$ et $\lambda_2 \gg 0$, le pixel (x,y) est un point d'intérêt car le gradient est élevé dans toutes les directions.

Afin d'éviter le coût de calcul des valeurs propres, Harris et *al.* [Harris 88] utilisent la formule (5.2) qui permet de calculer en chaque pixel une valeur appelée valeur d'intérêt:

$$f(x, y) = \text{Det}(M) - k \text{Tr}(M)^2 \quad (5.2)$$

Où $\text{Det}(M)=AB-C^2$, $\text{Tr}(M)=A+B$ et le paramètre k permet de régler l'influence des bords par rapport à celle des coins. Dans [Harris 88], k est fixé à 0.04.

Les points d'intérêt sont ensuite déterminés via un seuil S fixé par l'utilisateur. Si $f(x,y) > S$ alors le pixel (x,y) correspond à un point d'intérêt, sinon il sera rejeté.

Dans [Montesinos 98], les auteurs proposent une généralisation de cet algorithme au cas des images couleurs (RVB). La matrice M est calculée par :

$$M = G(\sigma) * \begin{bmatrix} (R_x^2 + V_x^2 + B_x^2) & (R_x R_y + V_x V_y + B_x B_y) \\ (R_x R_y + V_x V_y + B_x B_y) & (R_y^2 + V_y^2 + B_y^2) \end{bmatrix} \quad (5.3)$$

Une autre amélioration de l'algorithme de Harris est développée par Dufournaud [Dufournaud 00]. La nouvelle version consiste à extraire les points d'intérêt dans différentes échelles en faisant varier la taille σ (figure 5.2).

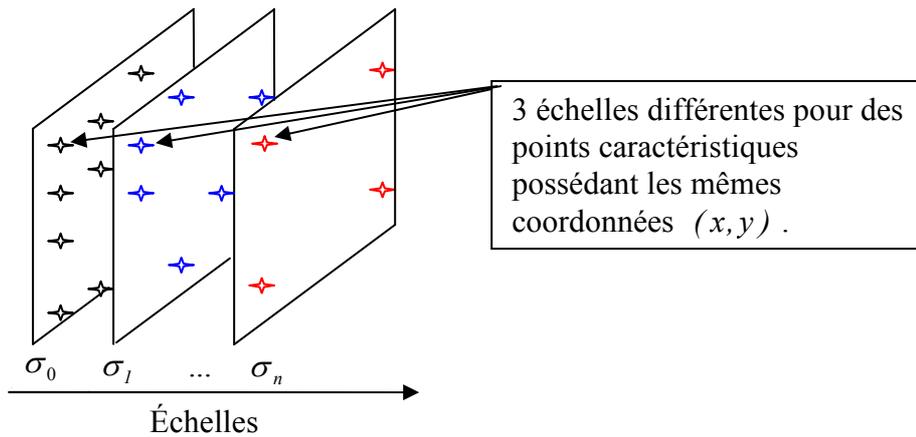


Figure 5.2 - Cartes de Harris pour différentes échelles.

5.1.1.1.2 Extraction et suivi de points caractéristiques KLT

De manière similaire aux travaux de Harris et *al.* [Harris 88], Tomasi et *al.* [Tomasi 91] proposent une méthode de détection et de suivi des points d'intérêt appelée KLT (Kanade-Lucas-Tomasi) basée sur l'analyse des valeurs propres de la matrice d'auto-corrélation M .

$$M = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (5.4)$$

Avec I_x et I_y les dérivées premières du signal d'intensité au point x .

Le déplacement $d=[dx, dy]^T$ d'un point x est déterminé en minimisant la fonction suivante :

$$\int_{\mathcal{W}} (I(x, t) - I(x + d, t + 1))^2 \cdot w(x) dx \quad (5.5)$$

Où \mathcal{W} est le voisinage du pixel x et $w(x)$ correspond à une fonction de pondération. Un développement linéaire du premier ordre permet de se ramener au système suivant :

$$Zd = a \quad (5.6)$$

où

$$Z = \int_{\mathcal{W}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} w(x) dx \quad (5.7)$$

et

$$a = \int_{\mathcal{W}} (I(x, t) - I(x, t + 1)) \cdot \nabla I dx \quad (5.8)$$

La résolution de ce système peut se faire par une la méthode itérative à savoir la méthode Newton-Raphson. Afin d'éviter des erreurs de suivi et d'assurer une résolution fiable du système, il faut que les deux valeurs propres λ_1 et λ_2 de la matrice d'auto corrélation M soient suffisamment grandes.

5.1.1.2 Détection d'extrema locaux : descripteurs SIFT

L'algorithme SIFT (*Scale Invariant Features Transform*) [Lowe 04] consiste à détecter les points d'intérêt qui sont stables dans l'espace échelle et de calculer pour chaque point un ensemble de descripteurs invariants par changement d'échelle et par rotation. Pour cela, on utilise des extrema locaux de l'opérateur DoG (*Difference of Gaussian*) défini par :

$$\begin{aligned} DoG(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, s\sigma) - L(x, y, \sigma) \end{aligned} \quad (5.9)$$

où $L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$; (5.10)

G est le filtre gaussien défini par :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (5.11)$$

σ est le paramètre d'échelle.

k est un facteur multiplicatif qui sépare deux échelles voisines. Lowe a montré que pour $k = 2^{\frac{1}{3}}$ les points obtenus sont plus répétables¹.

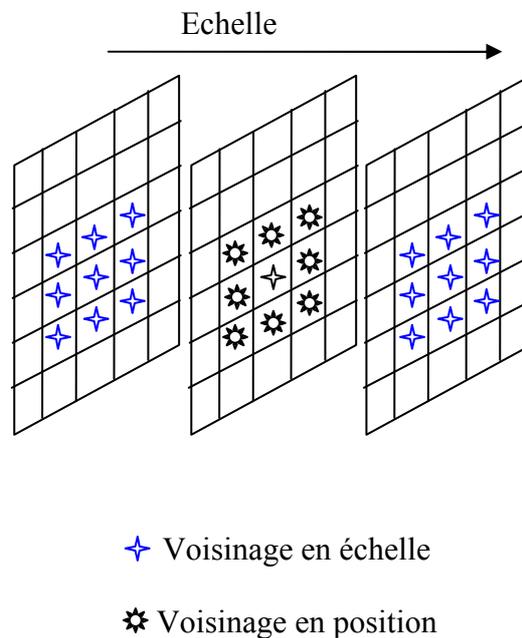


Figure 5.3 – Le voisinage (3x3x3) utilisé pour la recherche des extrema.

¹ Cette mesure (répétabilité) représente la stabilité d'un point et fut développée par Schmid [Schmid 97] dans sa thèse pour mettre en évidence la fiabilité du détecteur Harris.

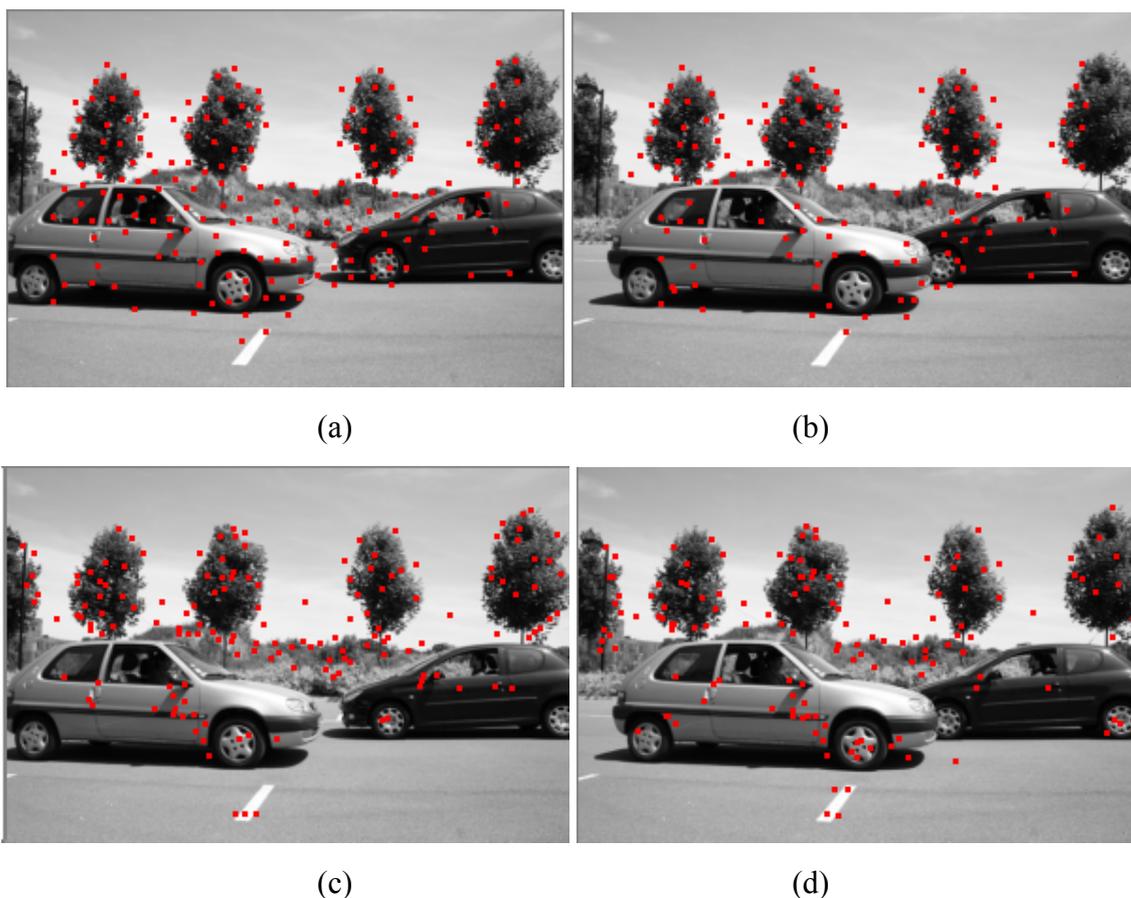


Figure 5.4 – Exemple des points d'intérêt. (a) et (b) détection et suivi des points d'intérêt par KLT. (c) et (d) points d'intérêt SIFT détectés dans deux images successives.

Les extrema sont recherchés dans de petits voisinages en position et en échelle (figure 5.3). Les points obtenus sont ensuite filtrés afin d'éliminer les points situés dans des zones insuffisamment contrastées ou sur des bords présentant une courbure trop faible. Ces derniers, points situés sur des bords présentant une courbure trop faible, sont éliminés par un opérateur proche de celui de Harris.

Pour chaque point d'intérêt sélectionné, on lui assigne une orientation qui correspond à l'orientation majoritaire des gradients spatiaux. Le descripteur de point d'intérêt sera ensuite représenté par rapport à cette orientation, ce que le rend invariant à la rotation d'image. Ce descripteur est constitué d'histogrammes d'orientation du gradient spatial d'intensité pondérés par la norme du gradient spatial.

La figure 5.4 donne quelques exemples des points d'intérêt calculés sur l'une de nos séquences.

5.1.2 Appariements

La mise en correspondance des points d'intérêt détectés dans des images successives I et I' consiste à déterminer pour chaque point de I , le point équivalent dans I' via un critère de similarité. C'est un processus intermédiaire entre l'étape de détection (*bas niveau*) et celle d'exploitation de l'information du mouvement en vue de la réalisation des tâches avancées (*haut niveau*) à savoir la détection et le suivi d'objet en mouvement.

Il existe un grand nombre de méthodes d'appariement de points d'intérêt. Ces méthodes peuvent être classées en fonction des descripteurs utilisés.

5.1.2.1 Descripteurs basés sur les intensités

Dans le cas de la mise en correspondance basée sur la conservation des valeurs d'intensité, l'appariement entre deux points est alors effectué par corrélation croisée qui permet de calculer une valeur mesurant la similarité entre deux points candidates à partir des intensités des pixels de leurs voisinages spatiaux. Le coefficient de corrélation croisée est donné par la formule :

$$Corr(x_1, x_2) = \frac{\sum_{i=-N}^N \sum_{j=-P}^P (I(x_1 + i, y_1 + j) - \bar{I}_{x_1}) \cdot (I(x_2 + i, y_2 + j) - \bar{I}_{x_2})}{\sqrt{\sum_{i=-N}^N \sum_{j=-P}^P (I(x_1 + i, y_1 + j) - \bar{I}_{x_1})^2 \cdot (I(x_2 + i, y_2 + j) - \bar{I}_{x_2})^2}} \quad (5.12)$$

Où N et P représentent la taille du voisinage (dans le cas d'un voisinage 3×3 , $N=1$ et $P=1$). \bar{I}_{x_k} est l'intensité moyenne des pixels appartenant au voisinage de point x_{k1} $k=1,2$.

Une autre méthode de mise correspondance des points d'intérêt basée sur les intensités des pixels du voisinage est proposée par Qin et *al.* [Qin 05]. La méthode consiste à comparer les histogrammes 2D calculés autour des points d'intérêt. Ceci rend ce descripteur robuste aux rotations. L'histogramme est ensuite normalisé afin de le rendre robuste aussi aux changements d'illuminations.

5.1.2.2 Descripteurs basés sur l'orientation

La méthode la plus simple pour appairer les points d'intérêt munis des descripteurs à base d'histogrammes est d'utiliser la distance euclidienne, ou celle de Manhattan. Néanmoins, pour ce type de descripteurs, d'autres distances s'avèrent plus adaptées. On peut citer la distance du χ^2 , la distance du terrassier EMD (*Earth Mover's Distance*) [Ling 07] [Rabin 09].

Dans [Ling 06], les auteurs ont montré que l'utilisation de la distance du χ^2 pour comparer les histogrammes améliore la mise en correspondance par rapport à la distance euclidienne.

$$\begin{aligned} dist(D(x_1), D(x_2)) &= \chi^2(D(x_1), D(x_2)) \\ &= \sum_{j=1}^N \frac{D(x_1)_j - D(x_2)_j}{D(x_1)_j + D(x_2)_j} \end{aligned} \quad (5.13)$$

Cependant, cette mesure nécessite des calculs supplémentaires par rapport la distance euclidienne.

Une autre approche présentée dans la littérature utilise la distance de terrassier EMD. Cette distance est particulièrement bien adaptée à la comparaison des descripteurs à base d'histogrammes.

La distance EMD correspond au coût minimum nécessaire pour qu'un terrassier construise l'histogramme x_2 à partir de l'histogramme x_1 .

La distance EMD est définie par :

$$dist_{EMD}(D(x_1), D(x_2)) = \min_{(\alpha_{i,j}) \in M} \sum_{i=1}^N \sum_{j=1}^N \alpha_{i,j} \cdot c(i, j) \quad (5.14)$$

$$\text{avec } M = \left\{ (\alpha_{i,j}) / \alpha_{i,j} \geq 0, \sum_j \alpha_{i,j} = (x_1)_i, \sum_i \alpha_{i,j} = (x_2)_j \right\} \quad (5.15)$$

et $c_{i,j}$ est le coût du transport entre les classes $(\mathbf{x}_1)_i$ et $(\mathbf{x}_2)_j$. dans [Rabin 09], ce coût est calculé par :

$$c(i, j) = \frac{1}{N} \min(|i - j|, N - |i - j|) \quad (5.16)$$

Nous renvoyons le lecteur pour plus de détails sur la distance EMD au [Rabin 09].

5.1.3 Conclusion

Dans cette section nous avons présenté quelques méthodes de détection et de mise en correspondance des points d'intérêt. Dans nos travaux, nous avons utilisé deux méthodes différentes pour tester notre approche de détection de mouvement cohérent :

- La détection et le suivi des points d'intérêt par la méthode KLT
- La détection des points d'intérêt SIFT et la mise en correspondance par la distance euclidienne.

5.2 Estimation du mouvement

L'estimation du mouvement est un point crucial pour plusieurs applications en analyse de vidéo, à savoir le suivi d'objets en mouvement, la détection de mouvements cohérents, la segmentation en couche de mouvement...etc. L'estimation du mouvement peut être définie comme étant la recherche de la corrélation entre deux images successives afin de prédire le changement de position du contenu. Dans cette section, nous allons dresser un état de l'art sur

les méthodes d'estimation du mouvement. Dans un premier temps, nous présentons les différents modèles représentant le mouvement puis nous dresserons ensuite un état de l'art des méthodes existantes pour l'estimation de mouvement. Nous nous intéresserons tout particulièrement aux méthodes utilisant les points d'intérêt pour l'estimation des modèles paramétriques.

5.2.1 Modèles de mouvement

Il existe une large variété de façons de modéliser le mouvement des objets, allant du plus simple (modélisation à deux paramètres) au plus complexe (modélisation à 12 paramètres). Le choix du modèle de mouvement se fait en fonction du type de mouvement étudié (translation, rotation, d'expansion/contraction...) et du degré de fiabilité voulue. Dans cette section, nous mettons le point sur les différents types de modélisation utilisés en analyse de vidéo, tout en précisant les avantages et les limitations de chaque type. Nous allons classer ces différents types de modélisation en deux grandes catégories :

- Modélisation du mouvement au niveau de chaque pixel (flot optique) : cette modélisation constitue le premier niveau de caractérisation du mouvement.
- Modélisation du mouvement par région : dans cette modélisation on s'attend à ce que tous les pixels de chaque région suivent un même modèle de mouvement commun.

5.2.1.1 Modélisation locale de mouvement (flot optique)

On peut définir le flot optique comme étant l'ensemble des vecteurs de vitesse (dx , dy) de chaque pixel dans l'image. L'estimation de ces vecteurs se fait sous l'hypothèse de *luminosité constante* qui suppose que l'intensité lumineuse réfléchie par un point de la scène reste constant lors d'un mouvement relatif scène/caméra.

En supposant que le mouvement entre deux instants t et $t+dt$ est très petit, le développement en série de Taylor d'ordre 1 nous donne:

$$I(x+dx, y+dy, t+dt) \approx I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + \text{OrdreSup} \quad (5.17)$$

Où **OrdreSup** les termes d'ordre plus élevés. Sous l'hypothèse de luminosité constante, nous arrivons donc à:

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (5.18)$$

$$(\nabla I)^T V + I_t = 0 \quad (5.19)$$

Où $\nabla I = \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix}$ est le gradient spatial (d'une image), $V = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}$ est le flot optique et $I_t = \frac{\partial I}{\partial t}$ est

la dérivée temporelle de l'intensité (entre deux instant t et $t+dt$).

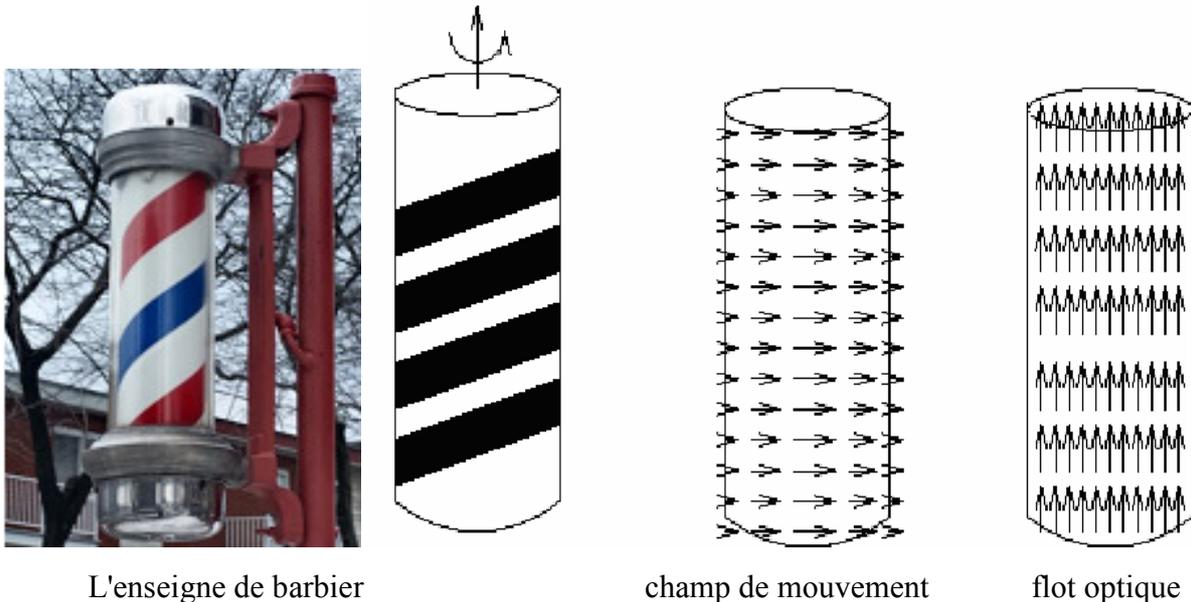


Figure 5.5 – Exemple d'ambiguïté inhérente à l'estimation du mouvement (aussi communément appelé problème de l'ouverture)

En chaque pixel, pour déterminer les deux composantes de vecteur de vitesse $\mathbf{v}=(dx, dy)$, on ne dispose que d'une seule équation (5.19), ce qui ne permet pas de déterminer de manière unique le flot optique. La figure 5.5 donne un exemple d'ambiguïté inhérente.

Des approches ont tenté de résoudre le problème de flot optique en ajoutant d'autres critères supplémentaires. Elles consistent généralement à lisser le mouvement localement, en autorisant ou non des discontinuités. Dans [Horn 80] [Horn 81], les auteurs ont donné une reformulation du problème mal posé en problème d'optimisation convexe quadratique. La fonctionnelle à minimiser comporte un terme d'adéquation qui est nul quand le champ de vecteurs \mathbf{v} vérifie l'équation du flot optique et un terme de régularisation (une norme de Sobolev sur \mathbf{v}).

Les auteurs de [Roy 00] [Boykov 01a] proposent une approche basée sur les champs de Markov aléatoire et les coupes de graphes. Cette approche consiste à discrétiser l'ensemble des valeurs que peuvent prendre les composantes des vecteurs de vitesses. Par conséquent, le nombre des cas possibles qui peuvent être générés en un point donné est limité par le pas de discrétisation.

La modélisation du mouvement par le flot optique souffre de plusieurs limitations qu'on peut résumer en :

- Le calcul des composantes du vecteur vitesse en un point se fait dans un voisinage borné de ce point, ce qui rend la méthode adaptée au petit déplacement.
- Le calcul des composantes du vecteur vitesse ne peut se faire que dans les directions du gradient (perpendiculaire au contour).

5.2.1.2 Modélisation de mouvement par région

Une autre façon de caractériser le mouvement d'une scène consiste à le modéliser par régions. Cette modélisation, connue sous le nom *modélisation paramétrique* repose sur l'hypothèse que le mouvement 2D d'un objet 3D projeté dans l'image suit généralement un modèle paramétrique de faible complexité. Les modèles de mouvement sont des fonctions qui utilisent de 2 à 12 paramètres. Plusieurs modèles paramétriques ont été proposés, nous présentons par la suite les modèles qui sont largement utilisés : le modèle affine et le modèle projectif.

5.2.1.2.1 Le modèle affine

Le modèle de mouvement affine est un modèle à six degrés de liberté (a_1, a_2, \dots, a_6) qui peut caractériser une classe importante de mouvement 2D: translation, rotation et expansion/contraction. Plusieurs auteurs [Bouthemy 93] [Dupont 06] [Fradet 08] justifient l'utilisation du modèle de mouvement affine par sa faible complexité et la précision de ses estimations.

La fonction mathématique définissant le modèle affine est déjà donnée dans le chapitre 4 (formule 4.1). Cette fonction peut être réécrite comme suit :

$$\omega(A, b; p_i) = Ax + b \quad , A \in \mathfrak{R}^{2 \times 2} \quad , b \in \mathfrak{R}^2 \quad (5.20)$$

Pour le cas où $A=0$, on retrouve le modèle translation [Tekalp 95].

5.2.1.2.2 Le modèle projectif

Le modèle projectif est une modélisation paramétrique à huit paramètres (h_1, h_2, \dots, h_8) du mouvement permettant de mettre en corrélation les objets d'une image à l'autre. La fonction mathématique définissant ce modèle est donnée par :

$$\omega(p_i) = \begin{bmatrix} u(p_i) \\ v(p_i) \end{bmatrix} = \begin{bmatrix} \frac{h_1 x_i + h_2 y_i + h_3}{h_7 x_i + h_8 y_i + 1} \quad , \quad \frac{h_4 x_i + h_5 y_i + h_6}{h_7 x_i + h_8 y_i + 1} \end{bmatrix}^T \quad (5.21)$$

Ce modèle peut caractériser le mouvement 2D d'un objet 3D projeté dans l'image. Un tel mouvement est aussi appelé une homographie [Olivier 93]. Dans le cas où $h_7=0$ et $h_8=0$, on retrouve le modèle affine.

5.2.1.2.3 Modèle retenu

Pour toutes les raisons déjà citées (la faible complexité et la précision de ses estimations), nous avons retenu le modèle affine complet à six paramètres. La qualité d'estimation et des résultats obtenus par l'utilisation de ce modèle sont plus pertinentes dans les cas où le mouvement entre deux instants donnés est faiblement éloigné. Dans nos travaux, nous avons modélisé les mouvements entre deux images consécutives, ce qui est raisonnable de supposer qu'on peut les approcher par des modèles affines.

5.2.2 Estimation du mouvement paramétrique

L'estimation du mouvement est un problème difficile, sa résolution se fait par la mise en œuvre des hypothèses locales. L'hypothèse la plus utilisée est la conservation de la luminance des pixels le long de leurs mouvements dans la scène. Cette hypothèse est à la base de plusieurs approches dont on peut citer les méthodes d'estimation par la mise en correspondance des blocs (*block matching*), les méthodes différentielles, les méthodes fréquentielles, etc. Nous présentons par la suite deux méthodes d'estimation du mouvement paramétrique basées sur les points d'intérêt, celle de Wills et *al.* [Wills 03] et celle de Dupont et *al.* [Dupont 06].

5.2.2.1 Méthode utilisant une paire d'images distantes

Wills et *al.* [Wills 03] ont développé une méthode d'estimation de mouvement paramétrique entre deux images distantes pour lesquelles l'estimation du flot optique n'est pas efficace. Leur méthode est basée sur :

1. l'extraction des points d'intérêt et leurs descriptions sur les deux images,
2. la mise en correspondance de points d'intérêt d'une image à l'autre,
3. le partitionnement des points d'intérêt en groupes de mouvement cohérent via une variation de l'algorithme de RANSAC,
4. le calcul direct du modèle homographique du mouvement de chaque groupe à partir des couples de correspondances des points relatifs à cette dernière.

Pour réduire l'influence des correspondances incorrectes, les auteurs dilatent les points d'intérêt. Les pixels appartenant aux zones dilatées sont ensuite ajoutés à la liste des points d'intérêt. Cette technique ne peut pas être appliquée dans le cas des petits déplacements car le risque d'ajouter des points appartenant aux zones occultées est très grand, ce qui explique pourquoi la segmentation des images intermédiaires n'est pas mentionnée dans leur article.

5.2.2.2 Détails sur la méthode de Dupont et *al.* [Dupont 06]

Dans [Dupont 06], les auteurs proposent une méthode d'estimation de mouvement basée sur un processus de raffinement itératif. L'estimation initiale est calculée pour une région

donnée \mathcal{S}_i par la minimisation de la fonction (5.22) par la méthode linéaire standard développée par Paragios et *al.* [Paragios 05].

$$\mathcal{E}(\mathcal{A}) = \int_{\mathcal{S}_i} \|\mathcal{A}(\mathbf{x}) \cdot \nabla I^t(\mathbf{x}) + I^{t+1}(\mathbf{x}) - I^t(\mathbf{x})\|^2 d\mathbf{x} \quad (5.22)$$

où \mathcal{A} est le modèle de mouvement à estimer et ∇I^t le gradient spatial de l'image à l'instant t .

Le processus de raffinement s'appuie sur la méthode de Odobez et *al.* [Odobez 95] qui consiste, à chaque itération, à estimer $\Delta\mathcal{A}$ de telle sorte à minimiser l'erreur résiduelle suivante :

$$\mathcal{E}(\Delta\mathcal{A}) = \int_{\mathcal{S}_i} \|I^t(\mathbf{x}) - I^{t+1}(\mathbf{x} + \mathcal{A}(\mathbf{x})) - \Delta\mathcal{A}\nabla I^{t+1}(\mathbf{x} + \mathcal{A}(\mathbf{x}))\|^2 d\mathbf{x} \quad (5.23)$$

La précision de l'estimation dépend de l'initialisation qui ne doit pas être trop éloignée de la solution optimale.

Pour la détermination des régions \mathcal{S}_i , les auteurs utilisent un procédé basé sur les points d'intérêt. La méthode s'exécute en quatre étapes :

1. extraction des points d'intérêt et leurs descriptions sur toutes les images de la séquence.
2. la détermination de correspondance d'une image à l'autre.
3. à partir des couples de correspondances, l'algorithme K-Means permet de déterminer, pour chaque point d'intérêt extrait, sa couche (région) correspondante. Le nombre de couche n est fixé au début de telle sorte à avoir une sur-segmentation.
4. le calcul direct du mouvement relatif à chaque région.

Le regroupement des points d'intérêt par l'algorithme de K-Means est très sensible à l'initialisation, les auteurs proposent de l'exécuter plusieurs dizaines de fois avec des initialisations différentes. La meilleure convergence sera ensuite conservée. Cependant, ce procédé est très coûteux en temps d'exécution.

5.2.2.3 Estimation de mouvement paramétrique à partir des mesures de flot optique.

L'estimation de mouvement paramétrique à partir des mesures de flot optique est très utilisée dans la littérature. Citant à titre d'exemple [Fradet 10] où l'auteur a proposé d'utiliser la régression linéaire par la méthode des moindres carrés pondérés pour calculer les paramètres du modèle affine à partir des mesures locales de flot optique.

L'inconvénient de ces méthodes est qu'elles héritent des problèmes de la méthode du flot optique cités précédemment (sous section 5.2.1.1) et qu'on peut résumer dans la non fiabilité des mesures de déplacement locales (dx, dy) calculées en chaque point.

5.2.2.4 Estimation de mouvement via les points d'intérêt et la régression plane 3D

Dans cette sous-section, nous présentons la méthode que nous avons utilisée pour l'estimation de mouvement paramétrique et qui consiste à utiliser la méthode de régression plane 3D récemment développée par Jacquelin [Jacquelin 11] pour calculer les paramètres du modèle affine à partir des mesures locales déterminées après la détection et la mise en correspondance des points d'intérêt. L'objectif est de contourner le problème de non fiabilité des mesures locales calculées par la méthode du flot optique et d'éliminer ainsi l'influence des pixels aberrants (occlusions, luminosité localement non constante d'une image à l'autre, etc.) sur le calcul des paramètres du modèle affine.

La méthode présentée ici s'appuie sur l'hypothèse que les points d'intérêt sont partitionnés en groupes de mouvements cohérents et que l'on souhaite seulement estimer le mouvement paramétrique (affine) relatif à chaque groupe.

Notons G_j un ensemble de points d'intérêt ayant un mouvement cohérent défini comme étant le vecteur de paramètres $\mathbf{a}=(\mathbf{a}_x, \mathbf{a}_y)^T$ avec $\mathbf{a}_x=(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ et $\mathbf{a}_y=(\mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6)$. Pour un point $(\mathbf{x}, \mathbf{y}, \mathbf{dx}, \mathbf{dy})^T \in G$ pour lequel le vecteur de déplacement $(\mathbf{dx}, \mathbf{dy})$ est connu, on a les relations suivantes :

$$dx = a_1 \cdot x + a_2 \cdot y + a_3 \quad (5.24)$$

$$dy = a_4 \cdot x + a_5 \cdot y + a_6 \quad (5.25)$$

On a \mathbf{a}_x (respectivement \mathbf{a}_y) ne dépend que de \mathbf{dx} (respectivement \mathbf{dy}), la régression peut être appliquée séparément sur chacune des composantes horizontale et verticale. Pour cela, chaque groupe de points d'intérêt $G_j = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{dx}_i, \mathbf{dy}_i)_{i=1..N}\}$ est scindé en deux sous groupes $G_{jx} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{dx}_i)_{i=1..N}\}$ et $G_{jy} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{dy}_i)_{i=1..N}\}$. Par la suite, on cherche à estimer pour chaque groupe de points \mathbf{a} (autrement dit \mathbf{a}_x et \mathbf{a}_y) de telle sorte à minimiser par la méthode des moindres carrés des distances vraies les erreurs quadratiques suivantes :

$$\sum_{(\mathbf{x}, \mathbf{y}, \mathbf{dx})^T \in G_{jx}} \left\| (a_1 \cdot x + a_2 \cdot y + a_3) - dx \right\|^2 \quad (5.26)$$

$$\sum_{(\mathbf{x}, \mathbf{y}, \mathbf{dy})^T \in G_{jy}} \left\| (a_4 \cdot x + a_5 \cdot y + a_6) - dy \right\|^2 \quad (5.27)$$

Pour résoudre ce problème, nous avons choisi d'utiliser la méthode de régression plane 3D décrite dans [Jacquelin 11] (cf. annexe pour plus de détails).

A noter que la fiabilité d'estimation est liée à celle des vecteurs de déplacement $(\mathbf{dx}, \mathbf{dy})^T$. Nous verrons dans le chapitre suivant comment partitionner l'ensemble des points d'intérêt d'une image en groupes de mouvement cohérent.

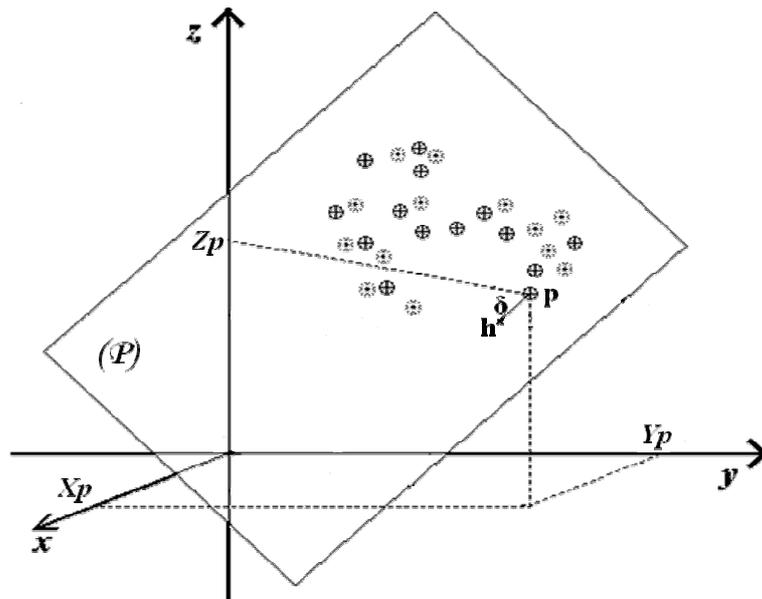


Figure 5.6 - Les points * dans la figure représentent l'ensemble des points d'intérêt $\{(x_i, y_i, d_i)\}_{i=1..N}$ appartenant à un même groupe. h est la projection orthogonale de point p sur le plan (\mathcal{P}) . δ égale à la distance entre p et le plan (\mathcal{P}) . Le but est de déterminer (\mathcal{P}) tel que $\sum_{i=1}^N \delta_i^2$ soit minimum. Pour déterminer l'équation affine (5.24) d_i est réglé sur dx_i , et il est mis à dy_i pour déterminer (5.25).

Nous venons de présenter la méthode que nous avons utilisée pour l'estimation du mouvement afin de fusionner les groupes de points d'intérêt cohérents en sens de mouvement paramétrique. Ces informations de mouvement vont pouvoir aussi être exploitées pour extraire les couches de même mouvement.

5.3 Détection de mouvements cohérents

5.3.1 Etat de l'art

La détection de mouvements cohérents est un problème qui tente de déterminer le nombre de mouvements différents existant dans une scène, leurs modèles de mouvement, leurs répartition spatiale, etc.. Ces informations sont souvent déterminées en exploitant les mesures locales obtenues après la détection et la mise en correspondance des points d'intérêt (section 5.1). La détection de mouvements cohérents peut être vue comme une description de la scène en termes d'objets en mouvements [Ullman 82].

Dans [Costeira 98], les auteurs proposent une méthode de groupement des trajectoires des points d'intérêt relatifs à un objet en mouvement par l'analyse d'une matrice formée des coordonnées des points d'intérêt à travers plusieurs images successives (au moins une dizaine). La méthode est connue sous le nom de la méthode de factorisation multi-objets.

Dans le cas de N points d'intérêt extraits dans F images successives, la matrice est donnée par :

$$W = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & & \vdots \\ x_{F1} & \cdots & x_{FN} \\ y_{11} & \cdots & y_{1N} \\ \vdots & & \vdots \\ y_{F1} & \cdots & y_{FN} \end{bmatrix}$$

où x_{ij} et y_{ij} sont les coordonnées du point i dans l'image j .

Après la décomposition en valeurs singulières $W = U \Sigma V$, les auteurs de [Costeira 98] ont montré que la matrice $Q = V^* V^T$, appelée matrice d'interaction de forme (*shape interaction matrix*), possède plusieurs propriétés intéressantes et utiles dont la principale est que chaque élément de Q^* fournit des informations importantes indiquant si un pair d'éléments (deux points d'intérêt) appartiennent à un même objet ou non. Q^* est Q mise sous forme diagonale par blocs. La stabilité de la mise sous forme diagonale par blocs de Q a été l'objet de plusieurs améliorations [Inoue 01] [Wu 01] [Park 04].

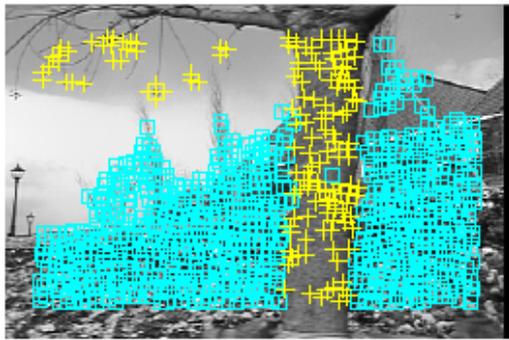
L'avantage principal de la méthode de factorisation par blocs est qu'elle est invariante au mouvement des objets. Cependant, elle est très sensible aux occultations.

Dans [Yuille 98], les auteurs ont proposé un cadre théorique pour l'analyse du mouvement à partir des mesures locales. Ce cadre théorique a donné lieu à plusieurs approches dont on peut citer [veit 07] [Poon 09].

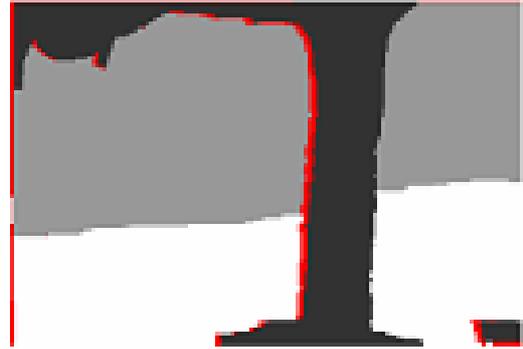
Dans [Veit 07], les auteurs proposent une approche basée sur l'algorithme de classification hiérarchique ascendante (*single linkage*) appliquée sur les mesures de déplacement locales calculées au niveau des points d'intérêt pour créer des groupes cohérents. Ces groupes sont ensuite validés et fusionnés par approche *a contrario* [Desolneux 03].

Poon et al. [Poon 09] utilisent l'algorithme RANSAC [Fischler 81] pour déterminer la correspondance entre les points d'intérêt des images consécutives. Les mouvements cohérents sont ensuite déterminés par la méthode *a contrario* dans un espace 3D. Ce travail présente les résultats expérimentaux sur des images réelles acquises par des caméras fixes, alors que les questions essentielles de la navigation autonome ne sont pas considérées.

Ces deux dernières méthodes, qui peuvent être vues comme des applications du cadre théorique proposé par Yuille et Grzywacz [Yuille 98], sont incapables de regrouper des zones distantes ayant le même mouvement au sens de mouvement paramétrique ou de séparer les points voisins ayant des mouvements très proches (voir figure 5.7).



(a) [Veit 07] [Poon 09]



(b) [Xiao 05]

Figure 5.7 - Exemple de limitation à la détection de mouvements cohérents dans les séquences acquises avec une caméra en mouvement. (a) le résultat de détection de mouvements cohérents par les méthodes de Veit [Veit 07] et Poon [Poon 09]. (b) le résultat de segmentation en sens de mouvement par la méthode de Xiao [Xiao 05]. Le nombre de mouvements différents, comme c'est détecté dans (b), est 3 alors que les méthodes qui se basent seulement sur les mesures locales, obtenues après la détection et la mise en correspondance des points d'intérêt, n'ont pas pu séparer correctement ces mouvements.

5.3.2 Conclusion

En conclusion, il existe un réel besoin de méthodes générales capable de détecter les mouvements cohérents en sens de mouvement paramétrique dans le cas de la caméra fixe ou non.

CHAPITRE 6

Nouvelle approche d'estimation de mouvements cohérents et d'extraction de couches de même mouvement

La détection de mouvements cohérents et l'extraction de couches de même mouvement sont deux problèmes liés. La plupart des approches d'extraction de couches de même mouvement nécessitent des initialisations : à savoir le nombre de couche, les modèles des mouvements relatifs aux objets, etc. Ces initialisations peuvent être déterminées par des techniques dites de détection de mouvements cohérents.

Dans ce chapitre, nous présentons une nouvelle approche de détection de mouvements cohérents et d'extraction de couches de même mouvement [Housni 12b].

6.1 Détection de mouvements cohérents

Nous présentons dans cette section une nouvelle approche de détection de mouvements cohérents qui est basée sur le regroupement des mesures locales calculées par la détection et la mise en correspondance des points d'intérêt. L'avantage principal de notre approche est qu'elle permet de détecter les mouvements cohérents en sens de mouvements paramétriques.

La détection se fait en deux étapes. La première étape, appelée étape de regroupement local, permet de réduire la taille du problème en regroupant chaque deux points voisins (**définition 1 & 2**) ayant le même mouvement local (**définition 2**). La deuxième étape, appelée étape de fusion, permet de regrouper chaque deux groupes de points cohérents en sens de mouvements paramétriques.

Les étapes de notre approche sont :

I. Etape 1 : Regroupement local

- a) L'extraction des points d'intérêt dans la première image
- b) Le suivi de ces points dans la seconde image (par l'algorithme KLT tracking [Tomasi 91]). Seuls les points suivis sont sélectionnés pour être regroupés par la méthode de clustering (algorithme 6.1).
- c) Le regroupement des points d'intérêt par l'algorithme 6.1 en utilisant un critère local '**définition 2**'. Cette étape génère une liste de groupes initiaux. Chaque groupe est défini par un ensemble de points.

II. Etape 2 : Fusion des groupes cohérents en sens de mouvements paramétriques.

- d) Le fusionnement des groupes dont le modèle du mouvement calculé à partir de l'ensemble des points de ces groupes supporte plus que 85% des points de chaque groupe via un processus itératif '**algorithme 6.2**'
- e) Les groupes dont le nombre de points est supérieur à 3 sont retenus.

Remarque :

Dans le cas des points d'intérêt munis des descripteurs SIFT, les étapes I-a et I-b sont remplacées par une étape de détection et de mise en correspondance. Pour tester notre approche sur ce type de points, nous avons utilisé la distance euclidienne pour mesurer la similarité entre deux points d'intérêt.

6.1.1 Regroupement local

Dans cette étape, nous nous intéressons au regroupement local des points d'intérêt spatio-temporellement voisins. Pour cela, on définit la notion de voisinage spatio-temporel et de connexité entre deux points d'intérêt.

Soit $P(x, y)$ et $Q(x', y')$ deux points d'intérêt dont les mouvements locaux sont respectivement $T_P(dx, dy)$ et $T_Q(dx', dy')$.

Définition 1 (Voisinage Spatio-Temporel VST)

On dit que P appartient au voisinage spatio-temporel de Q $P \in VST_Q$ si et seulement si :

$$|x - x'| < w * \rho(T_Q) \quad \text{et} \quad |y - y'| < w * \rho(T_Q) \quad (6.1)$$

Avec w est la distance minimale entre les points sélectionnés.

ρ est le module de vecteur de déplacement $\rho(T_Q) = \sqrt{dx'^2 + dy'^2}$.

Définition 2 (connexité de deux points d'intérêt)

P et Q sont dits connexes si et seulement si :

$$\begin{cases} P \in VST_Q \\ |\rho(T_P) - \rho(T_Q)| < \varepsilon_1 \\ |\theta(T_P) - \theta(T_Q)| < \varepsilon_2 \end{cases} \quad \text{ou} \quad \begin{cases} Q \in VST_P \\ |\rho(T_P) - \rho(T_Q)| < \varepsilon_1 \\ |\theta(T_P) - \theta(T_Q)| < \varepsilon_2 \end{cases} \quad (6.2)$$

Avec θ est l'orientation du vecteur de déplacement.

$\varepsilon_1, \varepsilon_2$ sont deux constantes à calculer en fonction du module et de l'orientation du vecteur de déplacement (6.3).

Cas où $P \in VST_Q$

Soit $\vec{v}(|dx'| + 1, |dy'|)$, $\vec{u}(|dx'|, |dy'| + 1)$ et $\vec{T}(|dx'|, |dy'|)$ trois vecteurs (voir figure 6.1)

$$\begin{aligned} \varepsilon_1 &= \max \left(\sqrt{|\vec{v}| + |\vec{T}|}, \sqrt{|\vec{u}| + |\vec{T}|} \right) \\ \varepsilon_2 &= \max \left(\left| \theta(\vec{v}) - \theta(\vec{T}) \right|, \left| \theta(\vec{u}) - \theta(\vec{T}) \right| \right) \end{aligned} \quad (6.3)$$

Pour le cas de $Q \in VST_P$, (dx', dy') sera remplacé par (dx, dy) .

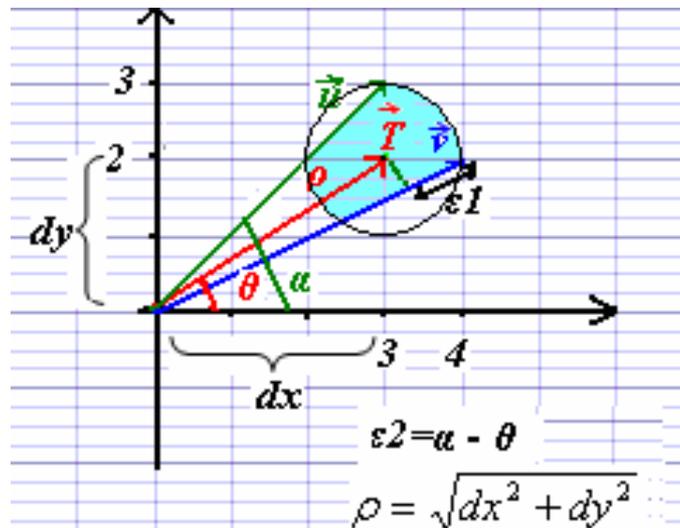


Figure 6.1- Cette figure illustre les différents paramètres utilisés pour la détermination du voisinage spatio-temporel et pour le calcul de la connectivité dans le cas de $dx = 3$ et $dy = 2$.

L'algorithme de regroupement local est donné comme suit:

Algorithme 6.1 : regroupement local par l'approche des composantes connexes.

Pour tout x dans V faire // V est l'ensemble des points d'intérêt

 Etat[x] ← Non atteint

Fin Pour

Ett ← 1

Pour tout x dans V faire

 Si Etat[x] = Non atteint alors

 A_traiter ← A_traiter \cup { x }

 Etat[x] = Atteint

 Tant que A_traiter Non Vide faire

Y ← dépiler (A_traiter)

 A_traiter ← A_traiter - { y }

 Pour tout $z \in N_y$ faire

 Si z et y sont connexes et Etat[z] = Non atteint alors

 A_traiter ← A_traiter \cup { z }

 Etat[z] = Atteint

 Fin si

 Fin pour

 Fin tant que

Fin si

Ett ← Ett + 1

Fin Pour.

Quelques exemples obtenus par étape de regroupement local sont donnés dans la figure 2, colonne 2.

6.1.2 Fusion des groupes cohérents en sens de mouvements paramétriques

La première étape « Regroupement local » permet de regrouper les points d'intérêt en fonction de leurs vecteurs de déplacement. Cependant, la détection de mouvements cohérents n'est fiable que pour les zones continues (cas de la séquence taxi de la figure 2). Par conséquent, la détection de mouvements cohérents sera sur-groupement dans certains cas, notamment dans les régions où les points d'intérêts sont spatio-temporellement discontinus (cas de la séquence *flowers* voir figure 2).

L'introduction de la classification par modèle de mouvement permettra de contourner de tels problèmes. A cette fin, nous avons opté pour l'utilisation de modèle de mouvement affine calculé à partir des points groupés.

6.1.2.1 Calcul du modèle de mouvement

Soit $G = \{(x_i, y_i, dx_i, dy_i)\}_{i=1..N}$ l'ensemble des points d'intérêt appartenant au même groupe. Le calcul du modèle de mouvement affine consiste à déterminer les deux équations affines $dx = a_1 \cdot x + a_2 \cdot y + a_3$ et $dy = a_4 \cdot x + a_5 \cdot y + a_6$. La première permet de calculer en chaque point le déplacement suivant X et la deuxième suivant Y.

Pour un ensemble de points d'intérêt G, nous avons calculé les paramètres du modèle de mouvement par la méthode décrite dans la sous-section 5.2.2.4. L'avantage de cette méthode est que, en plus de la qualité des résultats, le calcul se fait directement à l'inverse des méthodes itératives tel que [Dupont 06].

6.1.2.2 Algorithme de fusionnement de paires de groupes

Afin de résoudre le problème de sur-groupement, nous avons utilisé un algorithme permettant de fusionner les groupes dont le modèle du mouvement calculé à partir de l'ensemble des points de ces groupes supporte plus que 85% des points de chaque groupe via un processus itératif '**algorithme 2**'.

A chaque itération, la sélection des groupes candidats se fait en fonction de la distance inter-groupes définie par :

$$D(G_1, G_2) = \min_{p \in G_1, q \in G_2} d(p, q) \quad (6.4)$$

où $d(p, q)$ est la distance euclidienne entre les deux points p et q.

L'algorithme de fusionnement est donné comme suit:

Algorithme 2 : Fusion des groupes cohérents en sens de modèle de mouvement.

Calculer la matrice **D** des distances euclidiennes inter-groupe par (6.4)

Seuil ← 90

Répéter

 Sélectionner (Gi,Gj) qui ont la distance inter-groupe la plus petite

 Tant que **D**(Gi,Gj) ≠ infini faire

 Calculer le modèle de mouvement pour l'ensemble des points de $G_i \cup G_j$

 Si le modèle de mouvement support **Seuil** % des points de chaque groupe alors

 Fusionner Gi et Gj

 Actualiser la matrice des distances inter-groupe

 Sinon

D(Gi,Gj) ← infini

 Fin si

 Sélectionner (Gi,Gj) qui ont la distance inter-groupe la plus petite

 Fin Tant que

Réinitialisation de **D**

Seul ← **Seuil**-2

Tant que (Seuil > 85)

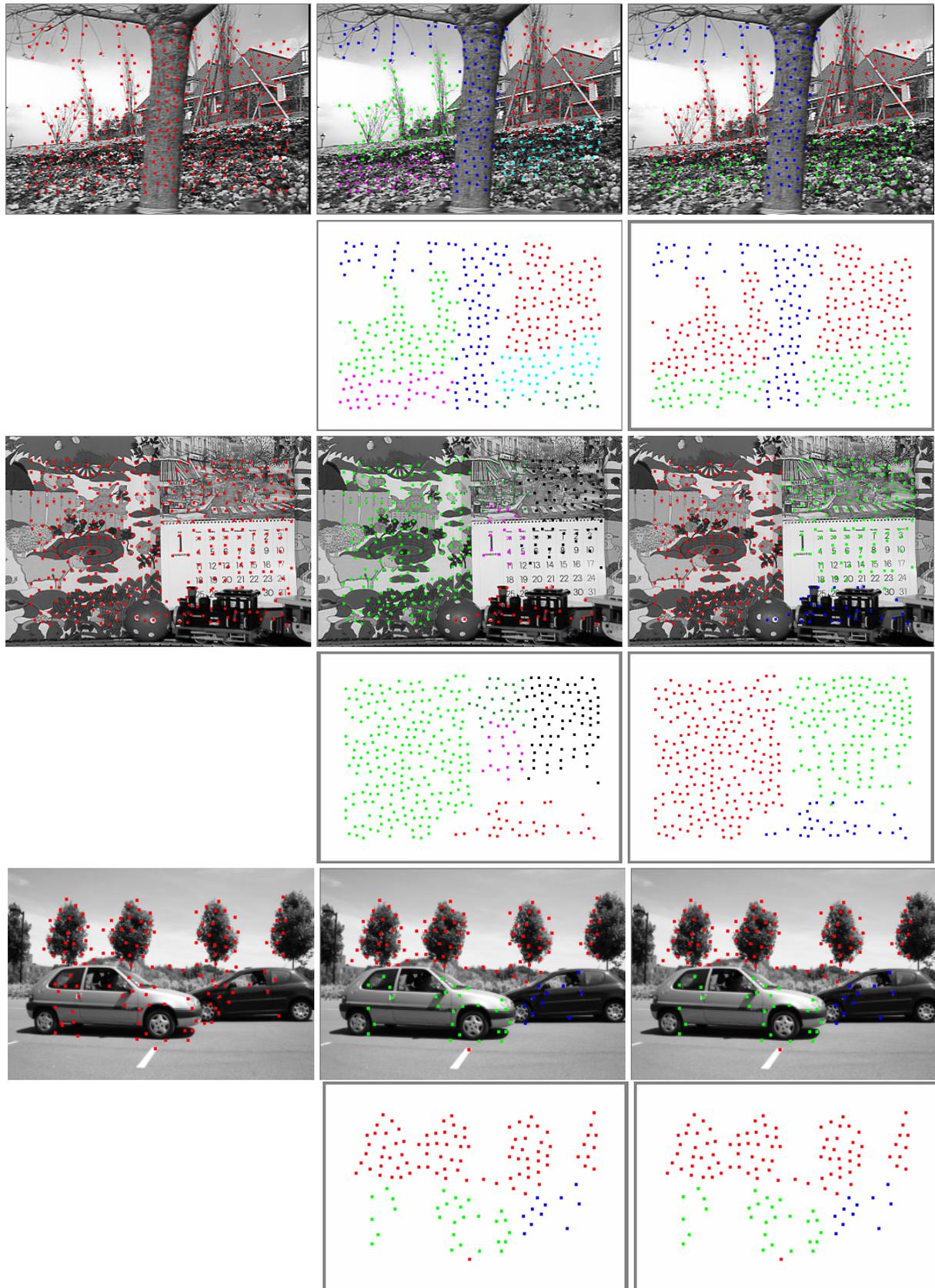


Figure 6.2 - La première colonne donne les points d'intérêt sélectionnés sur la première image. La deuxième colonne donne le résultat obtenu après le groupement local. La troisième colonne donne le résultat final de la détection de mouvements cohérents.

Quelques résultats de la détection de mouvements cohérents sont donnés dans les figures 6.2, 6.6, 6.7 et 6.8.

6.2 Extraction de couches de même mouvement

Dans la section précédente, nous avons montré comment déterminer les mouvements cohérents (nombre de couche n) ainsi que leurs modèles du mouvement affine A_i . Dans cette section, nous exploitons ces informations afin d'extraire les supports de couches de même mouvement. Nous présentons, dans un premier temps, la notion de couches de même mouvement. Un bref aperçu de quelques méthodes de l'état de l'art est ensuite donné. A la fin de la section, nous détaillons les différentes étapes de la méthode que nous avons utilisée. Cette dernière partie peut être considérée comme une application simple des résultats de la détection de mouvements cohérents.

6.2.1 Définition

La segmentation en couche de même mouvement consiste à déterminer le nombre de couche n , leurs mouvements paramétriques ainsi que leurs supports de couche. Autrement dit, la segmentation en mouvement est l'opération qui consiste à partitionner l'image I en régions S_i ayant chacune un mouvement différent. Formellement, on écrit:

$$\bigcup_{i=1}^n S_i = I \quad \text{et} \quad S_i \cap S_j = \emptyset, i \neq j \quad (6.5)$$

La figure 6.3 donne un exemple de résultat de segmentation en couches de même mouvement. Chaque couche est représentée par une couleur différente.

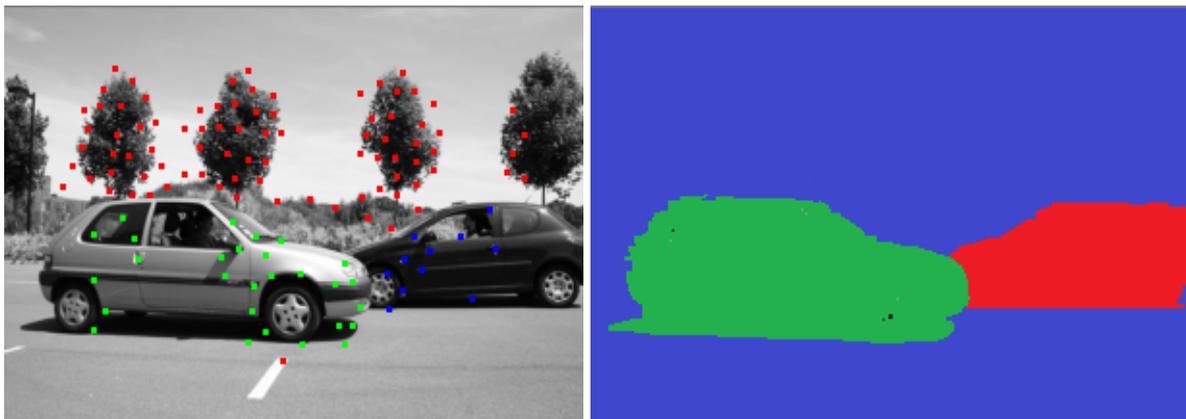


Figure 6.3 Exemple de décomposition en couches. Dans cet exemple, il y a 3 couches : arrière-plan, voiture noire et la voiture grise. Chaque couche à son propre mouvement.

6.2.2 Etat de l'art

L'extraction de couches de même mouvement est une tâche cruciale pour de nombreuses applications telles que l'analyse sémantique d'une scène, la compression de la vidéo, la détection d'objets en mouvement, etc.. Wang et *al.* [Wang 93] sont les premiers à avoir proposé en 1993 la décomposition d'une scène en couches de même mouvement. Et depuis, ce problème est devenu le sujet de nombreuses recherches en vision par ordinateur.

La plupart des méthodes de segmentation en couches de même mouvement alterne entre deux grandes étapes : une étape d'estimation du mouvement et une étape de segmentation de la scène (estimation de régions) en fonction des modèles de mouvement estimé dans la première étape. Après chaque estimation de régions, on teste si deux ou plusieurs couches peuvent fusionner.

Plusieurs techniques de segmentation d'une séquence vidéo en couches de même mouvement ont été développées à savoir :

- Les techniques basées sur le calcul du flot optique en chaque pixel à partir des contraintes visuelles [Horn 81], [Barron 92], [Brox 04], [Black 96a], [Ju 96]. Ces méthodes souffrent du nombre des inconnus à estimer qui est supérieur au nombre des contraintes ce qui impose l'utilisation de contraintes supplémentaires.
- Les techniques basées sur la croissance des régions, à savoir [Ke 01] qu'à partir d'une sur-segmentation basée sur la couleur et les modèles de mouvements affines les mieux estimés sur ces régions, le problème de segmentation peut se formuler comme un problème de clustering dans ce sous-espace formé par ces régions où les clusters sont plus facilement distinguables. Cette technique comporte le risque de regrouper dans la phase initiale les points dont le mouvement est différent mais dont la couleur est proche. Dans [Xiao 05], les auteurs proposent pour assurer une cohérence spatiale et éviter le problème des zones occultées, une méthode basée sur un regroupement d'un ensemble d'images ($n=5$), ce qui s'avère extrêmement coûteux en temps de calcul.
- Les techniques basées sur le mouvement paramétrique [Odobez 95], [Black 96b], [Dupont 06], en s'appuyant sur l'hypothèse standard que le mouvement 2D d'un objet 3D projeté dans l'image suit un modèle paramétrique, l'estimation des paramètres du modèle se fait via un processus itératif qui minimise l'erreur résiduelle. Ces méthodes souffrent de l'influence des zones d'occultations sur le calcul du modèle de mouvement, ce qui impose l'utilisation de contraintes supplémentaires, à savoir les contraintes visuelles basées sur la distribution de couleurs.

6.2.3 Extraction des supports des couches

6.2.3.1 Formulation du problème

Le problème d'extraction des supports de couches est un problème d'étiquetage où l'on cherche à classer les pixels en fonction de mouvement et d'apparence visuelle. Ce problème peut être représenté en termes de minimisation d'énergie. Cette énergie est de la forme (6.6):

$$E(l) = \sum_{x \in P} D_x(l_x) + \lambda \cdot \sum_{(x,y) \in N} V_{x,y}(l_x, l_y) \quad (6.6)$$

D représente un terme d'attachement aux données qui incite le résultat à être proche des données initiales, V est un terme de régularité portant sur des pixels voisins pour imposer une cohérence spatiale, l est l'ensemble des étiquettes, P est l'ensemble des pixels de l'image et λ est une constante réelle positive qui règle la pondération relative au terme de lissage par rapport au terme d'attache aux données.

Sous l'hypothèse que les mouvements cohérents déterminés dans la section précédente sont correctes, on peut considérer que les erreurs résiduelles sont dues au bruit gaussien $G(\mu, \sigma)$ où μ et σ sont la moyenne et l'écart type des erreurs résiduelles de la couche l_i calculés à partir de ses points d'intérêt. Par conséquent, le terme d'attachement aux données s'écrit :

$$D(x, L_i) = k - \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\delta_i(x)^2}{2\sigma_i^2}\right) \quad (6.7)$$

où $\delta_i(x) = \max(\|I(x,t) - I(x + A_i(x), t + 1)\|, \|I(x,t) - I(x - A_i(x), t - 1)\|)$,

K est une constante suffisamment grande pour que $D(x, l_i)$ soit positif. On peut prendre par

$$\text{exemple } k = \max_{\substack{l_i \in L \\ x \in P}} \left(\frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{\delta_i(x)^2}{2\sigma_i^2}\right) \right) \quad (6.8)$$

A_i est le modèle du mouvement affine de la couche i calculé dans l'étape précédente (étape de la détection de mouvement cohérents).

Remarque : L'avantage de l'utilisation des points d'intérêt pour le calcul de σ est qu'il permet d'éviter l'influence, d'une part, des zones occultées, ainsi que celui des points mal classés ou non supportés par le modèle du mouvement d'autre part.

Pour assurer un lissage temporel, nous avons utilisé un triplet d'images ($t - 1, t, t+1$) avec le modèle du mouvement calculé entre les instants $t-1$ et t .

Pour le terme de lissage, au lieu d'utiliser un simple modèle de pott [Dupont 06] [Xiao 05], nous avons utilisé une fonction ad-hoc définie par :

$$V_{x,y}(l_x, l_y) = T(l_x \neq l_y) \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|I_x - I_y|^2}{2\sigma^2}\right) \quad (6.9)$$

$$\text{où } \sigma = \frac{1}{n} \sum_{i=1}^n \sigma_i \text{ et } T(l_x \neq l_y) = \begin{cases} 1 & \text{si } (l_x \neq l_y) \\ 0 & \text{sinon} \end{cases}.$$

La fonction (6.9) encourage la segmentation qui passe par des régions où le gradient de l'image est assez fort $|I_x - I_y| > \sigma$.

6.2.3.2 Minimisation d'énergie

Pour minimiser l'énergie précédente (6.6), le calcul direct d'une solution optimale n'est pas possible. En effet, c'est un problème NP-complet. Nous avons vu dans le chapitre 1 comment est minimisée une telle énergie. Dans nos expérimentations, nous avons utilisé la méthode d'alpha-expansion [Boykov 01a] (voir chapitre 1, sous section 1.4.2) qui permet d'obtenir rapidement une solution proche de la solution optimale.

6.3 Résultats et expérimentations

6.3.1 Données

Pour valider notre approche et comparer nos résultats avec d'autres techniques, nous avons utilisé trois séquences classiques *flowers*, *calendar* and *croisement*. La première séquence contient 51 images de 352 x 240 pixels, la deuxième séquence contient 101 images de 352 x 240 pixels et la troisième séquence contient 27 images de 300 x 200 pixels.

6.3.2 Résultats et complexité

La complexité des deux algorithmes 1 et 2 est linéaire $O(n)$ où n est le nombre des points d'intérêt. Le tableau 6.1 donne le temps d'exécution moyen pour chaque séquence. Sachant que nous avons implémenté notre approche en langage C++ (avec un code non optimisé) et que pour tous les tests que nous avons fait, nous avons utilisé un Intel Core 2 Duo à 3x3 GHz et 2Go de RAM.

Séquence	Nombre de points d'intérêt	Détection des points d'intérêt	Le suivi des points d'intérêt par KLT	Groupement local (local clustering)	Détection de mouvements cohérents	Calcul de modèle de mouvement	Extraction des couches de même mouvement
<i>Flowers</i>	379	0,062 s	0,109 s	0,000 s	0,027s	0,000 s	1,716 s
<i>Calendar</i>	389	0,062 s	0,109 s	0,000 s	0,031s	0,000 s	1,590 s
<i>Croisement</i>	147	0,046 s	0,094 s	0,000 s	0,023s	0,000 s	0,940 s

Tableau 6.1 - Le temps d'exécution moyen par image (cas de détection et suivi des points d'intérêt par la méthode klt)

A noter que l'utilisation des points d'intérêt pour le calcul des modèles de mouvement nous a permis d'éviter l'influence des zones d'occultation. La figure 6.4 donne les pourcentages de recouvrement de mouvement des pixels de chaque région par les modèles de mouvement calculés à partir des points d'intérêt. On peut remarquer que la modélisation de mouvement par le modèle affine est plus adaptée au cas des petits déplacements (cas des séquences *flower* et *calender*).

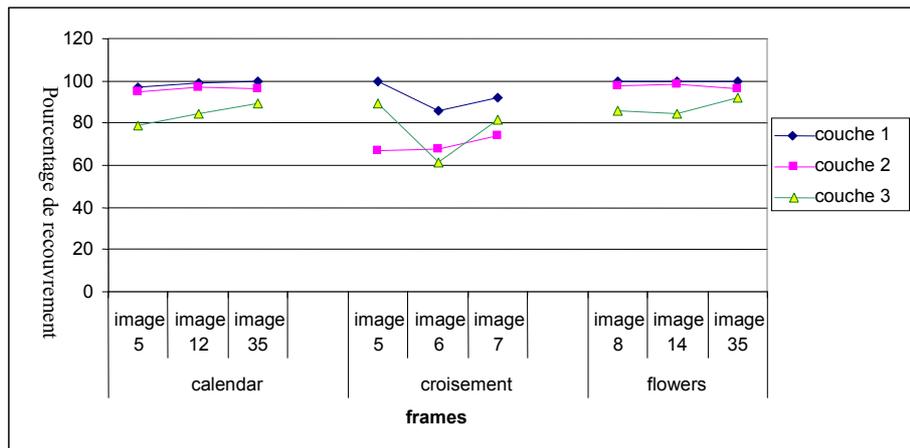


Figure 6.4 - Cette figure donne les pourcentages de recouvrement de mouvement des pixels de chaque couche par les modèles de mouvement calculés. Dans les figures 6.6, 6.7 et 6.8 la couche 1 correspond aux zones où les points d'intérêt sont en rouge, la couche 2 correspond aux zones où les points d'intérêt sont en vert et la couche 3 correspond à la zone où les points d'intérêt sont en bleu.

Approche	Configuration	Temps d'exécution moyen
[Dupond 06]	« la complexité en terme de calculs reste toujours l'une des limitations de l'approche proposée » [Dupond 06]	
[Xiao 05]	Pentium IV 2.0G	<ul style="list-style-type: none"> ➤ Dans la première étape de l'extraction des descriptions de couche, l'approche nécessite un temps relativement long, ce qui est environ 8-10 minutes. ➤ A partir de la deuxième image, le temps d'exécution moyen est 30 secondes.
[Fradet 08]	Pentium IV à 3,6GHz et 3GB de RAM	➤ temps d'exécution moyen est 6 s.
Notre Approche	Core 2 Duo à 2Ghz et 2GB de RAM	➤ temps d'exécution moyen est 1,931s

TABLEAU 6.2 - Comparaison des résultats d'extraction de couches de même mouvement sur la séquence *flowers* en terme de temps d'exécution des approches [Dupond 06] [Xiao 05] [Fradet 08] et notre méthode.

Le tableau 6.2 présente une comparaison en terme de temps d'exécution. Il est difficile de comparer le temps d'exécution des différentes approches pour une même configuration à cause de non disponibilité des codes sources. Les résultats obtenus par notre approche en temps d'exécution sont encourageants.

Dans la séquence *flowers* (figure 6.6), il est intéressant de noter que malgré la différence entre les mouvements des couches 1 et 2 qui est très petite (approximativement le mouvement translatif de la couche 1 est (1,0) celui de la couche 2 est (2,0)), notre approche de détection de mouvements cohérents a pu les distinguer, d'une part. D'autre part, notre approche a aussi

pu regrouper des parties correspondant à la même couche même si ces parties ne sont pas contiguës dans l'image.

Contrairement à [Veit 07][Poon 09] où la détection de mouvement commence par la soustraction de fond afin de réduire le nombre de point et de faciliter la répartition des points restant en groupes cohérents par simple représentation de ces points en 4D (x, y, θ, ρ) où θ est l'orientation du vecteur de vitesse et ρ est l'amplitude du vecteur de vitesse, notre approche est générale et peut être appliquée dans le cas où la caméra est en mouvement ou non.

Les résultats de l'extraction de couches de même mouvement sont très encourageants, sachant que nous avons utilisé seulement la classification par le mouvement. Une amélioration peut être faite par l'ajout d'autres contraintes à savoir les contraintes visuelles basées sur la distribution de couleurs comme dans [Dupont 06].

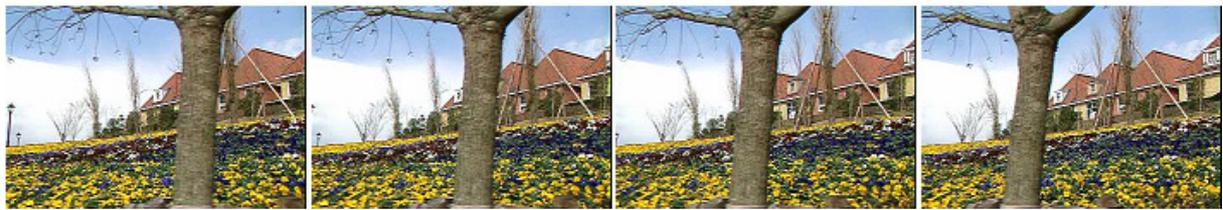
La figure 6.5a montre les résultats que nous avons obtenus sur la séquence *flowers*. Le nombre de couches ainsi les supports de chaque couche est les mêmes que celles obtenus par la méthode de Xiao [Xiao 05] figure 6.5b.

Les résultats de Dupont et al. (figure 6.5d dupliquée depuis [Dupont 05]) montrent que l'initialisation utilisée mène à un nombre de couches inattendu.

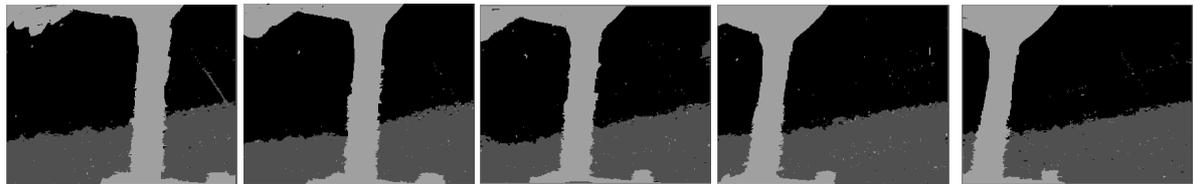
6.4 Conclusion

Dans ce chapitre nous avons présenté notre approche automatique de détection de mouvements cohérents et d'extraction de couche de même mouvement. Les résultats expérimentaux (Figures 6.2, 6.5, 6.6, 6.7 et 6.8) sont prometteurs et démontrent le potentiel de la méthode proposée.

L'approche proposée peut être appliquée en analyse de vidéo surveillance pour la détection d'objets en mouvement. Ceci peut être fait, dans le cas où la caméra est fixe, en détectant les zones dont le modèle de mouvement est différent de zéro. Pour le cas de la caméra en mouvement et sous l'hypothèse que le mouvement dominant est relatif à l'arrière-plan et avec la condition que le mouvement de l'arrière-plan soit représenté par un seul modèle de mouvement (comme dans le cas de la séquence *croisement*), la détection d'objets en mouvement peut se faire alors par la classification des modèles du mouvement en fonction de leur domination.



Quelques images de la séquence *flower*.



(a) Nos résultats. Seules les images 1- 8-14 -29 et 35 sont montrées.



(b) Résultats de Xiao et Shah copiés de [Xiao 05]. Les régions d'occultation sont en rouge. Trois couches ont été détectées.



(c) Résultats de Fradet et al copiés de [Fradet 08] (segmentation semi-automatique). La première carte de segmentation (soulignée) est fournie au système par l'utilisateur.



(d) Résultats de Dupont et al. copiés de [Dupont 06]. L'initialisation automatique produit une sur segmentation (5 couches) avant que le nombre de couches ne finisse par se stabiliser à 3. Seules les images 1, 4, 7, 10, 13, 16 sont montrées.

Figure 6.5 – Comparaison des résultats de segmentation sur la séquence *flower*. Les images d'une même colonne ne correspondent pas aux mêmes instants.

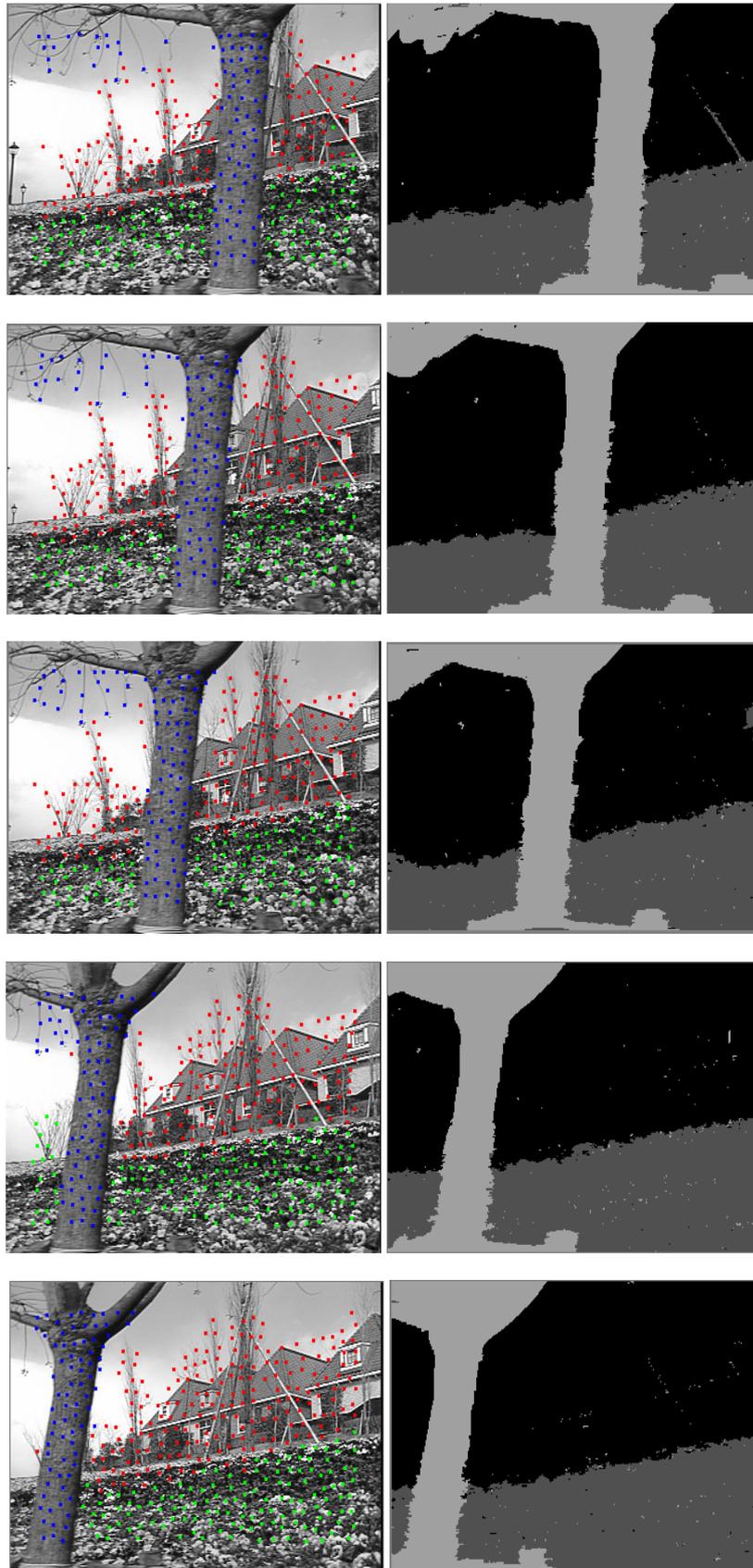


Figure 6.6 – Résultats de la détection de mouvements cohérents et d'extraction de couches de même mouvement. Cas de la séquence *flowers* images 1- 8-14 -29 et 35.

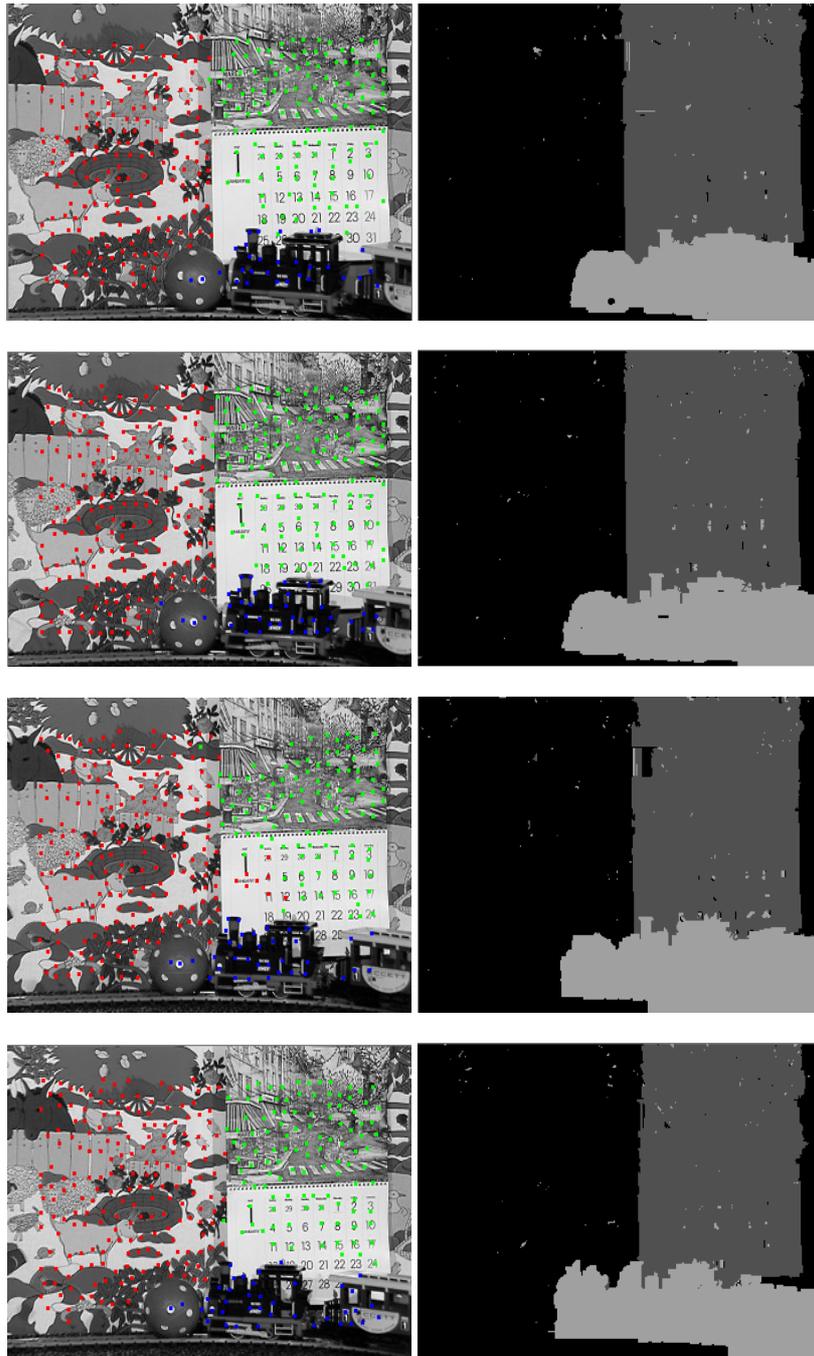


Figure 6.7 – Résultats de la détection de mouvements cohérents et d'extraction de couches de même mouvement. Cas de la séquence *calendar* images 5, 12, 24 et 35.

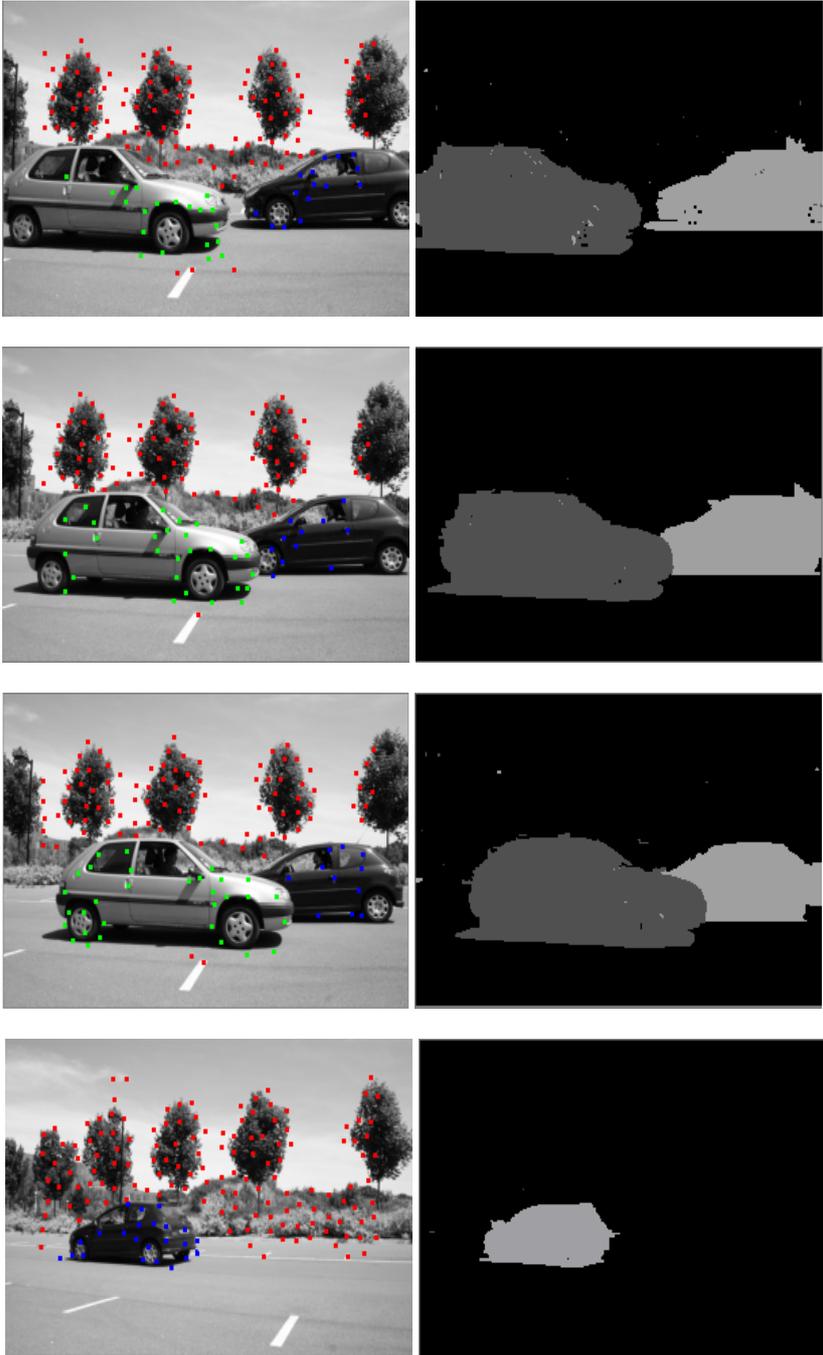


Figure 6.8 – Résultats de la détection de mouvements cohérents et d'extraction de couches de même mouvement. Cas de la séquence *taxi* images 5, 6, 7 et 22.

Conclusion générale et perspectives

Cette étude a exploré le thème de l'analyse du mouvement dans les séquences d'images. Nous avons présenté dans deux parties différentes deux types d'approches bien distincts : la détection et le suivi d'objets en mouvement et le clustering d'un ensemble de trajectoires de points (détection de mouvements cohérents). Dans cette conclusion générale nous proposons d'abord une synthèse des travaux effectués puis évoquons un certain nombre de perspectives.

a) Synthèse des travaux effectués

Dans ce travail de thèse, nous nous sommes intéressés à trois problématiques:

- Détection et segmentation d'objets en mouvement
- Suivi d'objets en mouvement
- Détection de mouvements cohérents et extraction de couches de même mouvement.

Dans la première partie, deux nouvelles approches ont été proposées. Nous avons proposé une nouvelle méthode automatique pour la détection d'objets en mouvement basée sur une modélisation des interactions spatio-temporelles entre étiquettes (arrière-plan et objets mobiles) par un champ de Markov. La minimisation de la fonction d'énergie résultant de cette modélisation est faite par la méthode des coupes de graphes. Puis, une nouvelle approche de suivi d'objets en mouvement a été proposée. Cette méthode se décompose en trois étapes principales. La première étape consiste à déterminer pour chaque objet le modèle de mouvement affine caractérisant son déplacement. La deuxième étape consiste à prédire les objets à suivre dans l'image suivante en utilisant les modèles de mouvement calculé dans l'étape précédente. L'utilisation des modèles de mouvement pour la prédiction des objets permet de résoudre le problème de large déplacement. La dernière étape consiste à raffiner les frontières des objets suivis par une méthode de segmentation dynamique afin de prendre en compte des éventuels changements de topologie. Une quatrième étape est ajoutée dans le cas de chevauchement des objets suivis. Cette dernière étape est basée sur la classification des pixels des objets qui se chevauchent par la méthode des coupes de graphes (algorithme alpha-expansion).

La deuxième partie du document a traité le problème de détection de mouvements cohérents et d'extraction de couche de même mouvement. Dans un premier temps, nous avons présenté une nouvelle approche de détection de mouvements cohérents. Cette méthode se décompose en trois étapes principales. La première étape consiste à détecter et à mettre en correspondance les points d'intérêt.

La deuxième étape consiste à grouper les points d'intérêt spatio-temporellement connexes. Cette dernière étape permet de regrouper les points d'intérêt en fonction de leurs vecteurs de déplacement locaux, donc incapable de regrouper les points distants (non connexes) ayant le même mouvement, ainsi incapable de regrouper les points dont les vecteurs de déplacement

locaux sont différents et qui peuvent être modélisés par un seul modèle de mouvement paramétrique.

Dans la troisième étape, afin de résoudre le problème de sur-groupement, nous avons utilisé un algorithme permettant de fusionner les groupes dont le modèle du mouvement calculé à partir de l'ensemble des points de ces groupes supporte plus que 85% des points de chaque groupe via un processus itératif.

L'approche précédente de détection de mouvements cohérents permet d'obtenir le nombre de mouvements cohérents ainsi que leurs modèles de mouvement. Ces informations sont ensuite utilisées afin de segmenter la scène en couches de même mouvement.

b) Perspectives

Les perspectives de ce travail sont les suivantes : pour la détection de mouvement, il est intéressant, dans le cas d'utilisation de la différence d'images successives comme l'observation, d'exploiter les résultats obtenus à des instants précédents afin de classifier les zones momentanément immobiles. La détection de mouvement dans le cas de la caméra en mouvement se base sur la différence d'images pixels par pixels (avec un modèle de mouvement nul), la généralisation de la méthode en cas de séquences acquises avec des caméras en mouvement peut être faite en détectant le mouvement relatif à l'arrière plan (généralement le mouvement dominant) puis remplacer la formule qui permet de calculer la différence d'images (2.1) par :

$$\zeta(s) = \left| \frac{dI(s)}{dt} \right| = \left| \Delta_{I_{t,t-1}}(s) \right| = \left| I_t(s) - I_{t-1}(s + A) \right|$$

Où A est le modèle du mouvement relatif à l'arrière plan.

Le résultat présenté sur la figure 4.12 a mis en avant une limitation de la méthode de suivi d'objets en mouvement. En effet, sur cette figure nous pouvons voir dans la dernière image de la séquence *croisement* la réapparition de l'objet (voiture noire) alors que le programme n'a pas pu le détecter. Ceci est dû au fait que l'objet a disparu dans l'image précédente (occultation totale) et l'algorithme que nous avons proposé utilise le résultat de l'étape précédente pour localiser les objets dans l'image courante. L'amélioration de l'algorithme proposé peut se faire par l'ajout d'autres critères caractérisant les objets afin de les retrouver dans le cas des occultations totales.

Le couplage de la méthode de détection d'objets en mouvement et celle de suivi d'objets peut aboutir à un système automatique de détection de suivi d'objets en mouvement.

Pour la détection de mouvements cohérents, Le modèle de mouvement rectiligne calculé par la méthode de régression plane en 3D permet déjà une bonne approximation d'un grand nombre de mouvements. Les mouvements de rotation d'un objet sur lui-même mériteraient aussi d'être étudiés en utilisant d'autres approches d'estimation des paramètres de modèle de mouvement où aussi par une modélisation de mouvement par morceaux.

Les résultats présentés sur les figures 6.4, 6.5, 6.6, 6.7 et 6.8 reflètent la qualité des modèles de mouvement calculés à partir des points d'intérêt en utilisant la méthode de régression plane 3D. Les résultats sont très encourageants, néanmoins, l'ajout des contraintes visuelles basées sur la distribution de couleurs comme dans [Dupont 06] peut améliorer davantage les résultats de la segmentation en couches de même mouvement.

La perspective la plus ambitieuse est certainement la proposition d'une nouvelle description des objets vidéo, s'appuyant sur la segmentation, l'estimation précise de mouvement sur de courts, moyens et longs intervalles temporels et les interactions d'un opérateur. L'extraction d'attributs supplémentaires (texture, forme, orientation, illumination ...) permettrait une modélisation plus complète des séquences traitées.

ANNEXE

Régression plane 3D

Dans cette annexe, nous présentons la méthode que nous avons utilisée pour l'estimation de mouvement paramétrique basée sur la régression plane 3D récemment développée par Jacquelin [Jacquelin 11]. L'objectif est de contourner le problème de non fiabilité des mesures locales calculées par la méthode du flot optique et d'éliminer ainsi l'influence des pixels aberrants (occlusions, luminosité localement non constante d'une image à l'autre, etc.) sur le calcul des paramètres du modèle affine.

La méthode présentée ici s'appuie sur l'hypothèse que les points d'intérêt sont partitionnés en groupes de mouvements cohérents et que l'on souhaite seulement estimer le mouvement paramétrique (affine) relatif à chaque groupe.

Notons G_j un ensemble de points d'intérêt ayant un mouvement cohérent défini comme étant le vecteur de paramètres $\mathbf{a}=(\mathbf{a}_x, \mathbf{a}_y)^T$ avec $\mathbf{a}_x=(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$ et $\mathbf{a}_y=(\mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6)$. Pour un point $(\mathbf{x}, \mathbf{y}, \mathbf{dx}, \mathbf{dy})^T \in G$ pour lequel le vecteur de déplacement $(\mathbf{dx}, \mathbf{dy})$ est connu, on a les relations suivantes :

$$dx = a_1 \cdot x + a_2 \cdot y + a_3$$

$$dy = a_4 \cdot x + a_5 \cdot y + a_6$$

On a \mathbf{a}_x (respectivement \mathbf{a}_y) ne dépend que de \mathbf{dx} (respectivement \mathbf{dy}), la régression peut être appliquée séparément sur chacune des composantes horizontale et verticale. Pour cela, chaque groupe de points d'intérêt $G_j = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{dx}_i, \mathbf{dy}_i)_{i=1..N}\}$ est scindé en deux sous groupes $G_{jx} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{dx}_i)_{i=1..N}\}$ et $G_{jy} = \{(\mathbf{x}_i, \mathbf{y}_i, \mathbf{dy}_i)_{i=1..N}\}$. Par la suite, on cherche à estimer pour chaque groupe de points \mathbf{a} (autrement dit \mathbf{a}_x et \mathbf{a}_y) de telle sorte à minimiser les erreurs quadratiques suivantes :

$$\sum_{(\mathbf{x}, \mathbf{y}, \mathbf{dx})^T \in G_{jx}} \|(a_1 \cdot x + a_2 \cdot y + a_3) - dx\|^2$$

$$\sum_{(\mathbf{x}, \mathbf{y}, \mathbf{dy})^T \in G_{jy}} \|(a_4 \cdot x + a_5 \cdot y + a_6) - dy\|^2$$

Pour résoudre ce problème, nous avons choisi d'utiliser la méthode de régression plane 3D décrite dans [Jacquelin 11]. La suite de cette annexe est un résumé de [Jacquelin 11] auquel nous renvoyons le lecteur pour plus de détails. Nous utilisons ici les mêmes notations que dans [Jacquelin 11].

A.1 Introduction

Etant données les coordonnées de n points :

$$P_1(X_1, Y_1, Z_1), P_2(X_2, Y_2, Z_2), \dots, P_k(X_k, Y_k, Z_k), \dots, P_n(X_n, Y_n, Z_n).$$

dont on sait qu'ils sont voisins d'un plan (Π) inconnu au départ, il s'agit de déterminer les paramètres caractérisant ce plan tel que soit minimum la somme des carrés des distances entre le plan et les points.

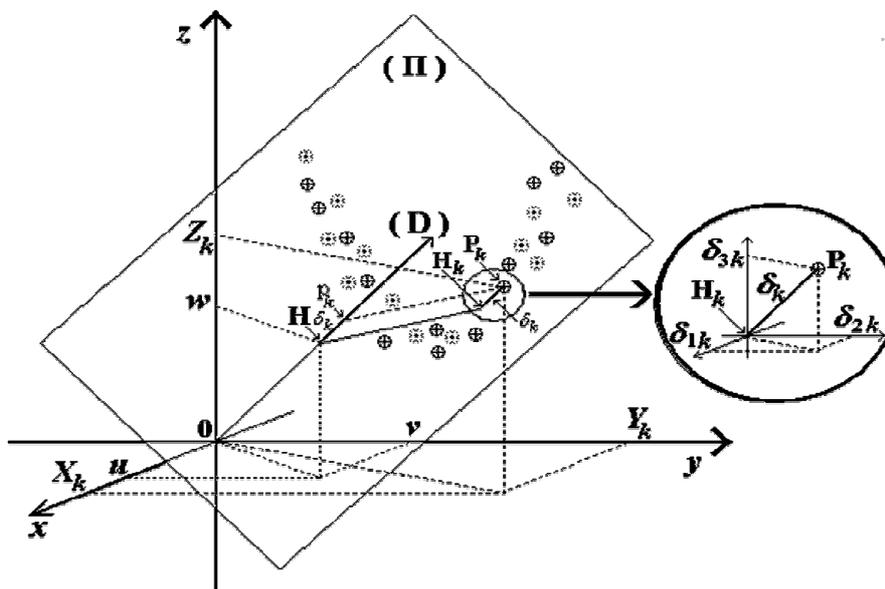


Figure A.1 - Les points * dans la figure représentent l'ensemble des points $\{P_1, P_2 \dots P_n\}$. H_k est la projection orthogonale de point P_k sur le plan (Π) . δ_k égale à la distance entre P_k et le plan (Π) . Le but est de déterminer (Π) tel que $\sum_{i=1}^N \delta_i^2$ soit minimum. Cette figure provient de [Jacquelin 11].

L'équation du plan (Π) sera : $A x + B y + C z + I = 0$.

La droite (D) , passant par l'origine du système d'axes et orthogonale à (Π) intersecte ce plan au point $H(w, v, w)$. La droite (D) sera définie par les paramètres a et b figurant dans les équations :

$$(D) \begin{cases} x = a.z \\ y = b.z \end{cases} \quad \text{En conséquence :} \quad (H) \begin{cases} u = a.w \\ v = b.w \\ w \end{cases}$$

Il est facile d'établir les relations qui relient (u, v et w) à (A, B et C) ainsi qu'à (a et b) :

$$\begin{cases} u = \frac{-A}{A^2 + B^2 + C^2} \\ v = \frac{-B}{A^2 + B^2 + C^2} \\ w = \frac{-C}{A^2 + B^2 + C^2} \end{cases} \quad \begin{cases} A = \frac{-u}{u^2 + v^2 + w^2} = \frac{-a}{(a^2 + b^2 + 1)w} \\ B = \frac{-v}{u^2 + v^2 + w^2} = \frac{-b}{(a^2 + b^2 + 1)w} \\ C = \frac{-w}{u^2 + v^2 + w^2} = \frac{-1}{(a^2 + b^2 + 1)w} \end{cases} \quad \begin{cases} a = \frac{A}{C} \\ b = \frac{B}{C} \end{cases}$$

La projection orthogonale d'un point $P_k (X_k, Y_k, Z_k)$ sur (Π) est le point H_k .

La projection orthogonale de P_k sur (D) est le point p_k dont les coordonnées sont (x_k, y_k, z_k) .

La distance entre P_k et (Π) est δ_k , égale à la distance entre p_k et H_k $\delta_k = \overline{H_k P_k} = \overline{H p_k}$.

$$\delta_k^2 = \delta_{1,k}^2 + \delta_{2,k}^2 + \delta_{3,k}^2 = (x_k - u)^2 + (y_k - v)^2 + (z_k - w)^2$$

Le but est de calculer les paramètres a, b et w qui sont suffisants pour que la droite (D) , le point (H) et le plan (Π) perpendiculaire à (D) en (H) soient tous déterminés. Il en serait de même si on cherchait à calculer les paramètres A, B et C figurant dans l'équation de (Π) , puisqu'ils sont liés par les relations précédentes. On cherche à ce que la somme des $(\delta_k)^2$ soit minimum.

Note : Pour réduire le volume des écritures, on utilisera le symbole Σ à la place de $\sum_{k=1}^{k=n}$.

A.2 Moindres carrés des distances vraies, méthode directe (non itérative)

Soit D_k la distance entre le point $P_k (X_k, Y_k, Z_k)$ et un point quelconque $P(x, y, z)$ du plan (Π) :

$$D_k^2 = (x - X_k)^2 + (y - Y_k)^2 + (z - Z_k)^2$$

$$D_k^2 = (x - X_k)^2 + (y - Y_k)^2 + \left(\frac{-(Ax + By + 1)}{C} - Z_k \right)^2$$

La distance minimum est atteinte lorsque P est en H_k (projection orthogonale de P_k sur (Π))

$$\begin{cases} \frac{\partial(D_k^2)}{\partial x} = 2(x - X_k) - 2 \frac{A}{C} \left(\frac{-(Ax + By + 1)}{C} - Z_k \right) = 0 \\ \frac{\partial(D_k^2)}{\partial y} = 2(y - Y_k) - 2 \frac{B}{C} \left(\frac{-(Ax + By + 1)}{C} - Z_k \right) = 0 \end{cases}$$

La résolution de ce système de deux équations linéaires donne d'abord les deux inconnues x et y puis z est obtenu avec $z = -(Ax+By+1)/C$.

$$\begin{cases} x = \frac{(B^2 + C^2)X_k - ABY_k - ACZ_k - A}{A^2 + B^2 + C^2} \\ y = \frac{-ABX_k + (A^2 + C^2)Y_k - BCZ_k - B}{A^2 + B^2 + C^2} \\ z = \frac{-ACX_k - BCY_k + (A^2 + B^2)Z_k - C}{A^2 + B^2 + C^2} \end{cases}$$

En reportant ces expressions dans $\delta_k^2 = (x - X_k)^2 + (y - Y_k)^2 + (z - Z_k)^2$ et après simplification, on obtient :

$$\delta_k^2 = \frac{(AX_k + BY_k + CZ_k + 1)^2}{A^2 + B^2 + C^2}$$

D'où la somme des carrés :

$$\sum \delta_k^2 = \frac{1}{A^2 + B^2 + C^2} \sum (AX_k + BY_k + CZ_k + 1)^2$$

Les dérivations partielles relativement à A , B et C conduirait à un système compliqué de trois équations non linéaires. Un changement de variables permet de simplifier les écritures :

$$a = \frac{A}{C} \quad ; \quad b = \frac{B}{C} \quad ; \quad \lambda = -\frac{1}{C} \quad \rightarrow \quad w = \frac{\lambda}{a^2 + b^2 + 1}$$

On reconnaît les notations a , b et w définies en introduction et correspondant à celles de la figure jointe. Avec ces notations, on a :

$$\sum \delta_k^2 = \frac{1}{a^2 + b^2 + 1} \sum (aX_k + bY_k + Z_k - \lambda)^2$$

Le minimum de cette somme est obtenu lorsque les dérivées partielles relativement à a , b et λ sont nulles. Celle relative à λ est particulièrement simple :

$$\frac{\partial}{\partial \lambda} \sum \delta_k^2 = \frac{2}{a^2 + b^2 + 1} \sum (aX_k + bY_k + Z_k - \lambda) = 0$$

D'où l'on tire :

$$\lambda = \frac{1}{n} (a \sum X_k + b \sum Y_k + \sum Z_k)$$

$$\begin{aligned} \sum \delta_k^2 &= \frac{1}{a^2 + b^2 + 1} \sum \left(aX_k + bY_k + Z_k - \frac{1}{n} (a \sum X_k + b \sum Y_k + \sum Z_k) \right)^2 \\ &= \frac{1}{a^2 + b^2 + 1} \sum \left(a \left(X_k - \frac{1}{n} \sum X_k \right) + b \left(Y_k - \frac{1}{n} \sum Y_k \right) + \left(Z_k - \frac{1}{n} \sum Z_k \right) \right)^2 \end{aligned}$$

Pour simplifier les écritures, on pose :

$$x_k = X_k - \frac{1}{n} \sum X_k \quad ; \quad y_k = Y_k - \frac{1}{n} \sum Y_k \quad ; \quad z_k = Z_k - \frac{1}{n} \sum Z_k$$

$$\sum \delta_k^2 = \frac{1}{a^2 + b^2 + 1} \sum (ax_k + by_k + z_k)^2$$

Le système de deux équations à deux inconnues (a, b) est obtenu par les dérivations partielles :

$$\begin{cases} \frac{2}{a^2 + b^2 + 1} \sum (ax_k + by_k + z_k)x_k - \frac{2a}{(a^2 + b^2 + 1)^2} \sum (ax_k + by_k + z_k)^2 = 0 \\ \frac{2}{a^2 + b^2 + 1} \sum (ax_k + by_k + z_k)y_k - \frac{2b}{(a^2 + b^2 + 1)^2} \sum (ax_k + by_k + z_k)^2 = 0 \end{cases}$$

Il se simplifie :

$$\begin{cases} \sum (ax_k + by_k + z_k)x_k - \frac{2a}{(a^2 + b^2 + 1)} \sum (ax_k + by_k + z_k)^2 = 0 \\ \sum (ax_k + by_k + z_k)y_k - \frac{2b}{(a^2 + b^2 + 1)} \sum (ax_k + by_k + z_k)^2 = 0 \end{cases}$$

$$(1)*b - (2)*a \rightarrow \sum (ax_k + by_k + z_k)bx_k - (ax_k + by_k + z_k)ay_k = 0$$

Pour développer les équations, on définit les coefficients suivants :

$$\begin{aligned} S_{xx} &= \sum x_k^2 \quad ; \quad S_{yy} = \sum y_k^2 \quad ; \quad S_{zz} = \sum z_k^2 \\ S_{xy} &= \sum x_k y_k \quad ; \quad S_{xz} = \sum x_k z_k \quad ; \quad S_{yz} = \sum y_k z_k \end{aligned}$$

Ce qui donne :

$$\begin{cases} (aS_{xx} + bS_{xy} + S_{xz}) - \frac{a(a^2S_{xx} + b^2S_{yy} + S_{zz} + 2abS_{xy} + 2aS_{xz} + 2bS_{yz})}{a^2 + b^2 + 1} = 0 \\ (aS_{xy} + bS_{yy} + S_{yz}) - \frac{b(a^2S_{xx} + b^2S_{yy} + S_{zz} + 2abS_{xy} + 2aS_{xz} + 2bS_{yz})}{a^2 + b^2 + 1} = 0 \end{cases}$$

$$b^2 S_{xy} + bS_{xz} + ab(S_{xx} - S_{yy}) - a^2 S_{xy} - aS_{yz} = 0$$

$$b^2 = \frac{1}{S_{xy}} (a^2 S_{xy} + aS_{yz} - (a(S_{xx} - S_{yy}) + S_{xz})b)$$

En reportant plusieurs fois cette expression b^2 dans le système précédent, on le réduit à deux équations en b au premier degré. Ce qui conduit à deux formulations de b exprimé en fonction de a . L'égalisation de ces deux expressions de b donne une équation où ne figure plus

$$\begin{cases} c_0 = S_{yz}(S_{xy}^2 - S_{xz}^2) + S_{xy}S_{xz}(S_{zz} - S_{yy}) \\ c_1 = S_{xy}^3 + S_{xy}(S_{xz}^2 - 2S_{yz}^2 - S_{zz}^2) + S_{xy}(S_{xx}S_{zz} + S_{yy}S_{zz} - S_{xx}S_{yy}) + \\ \quad + S_{xz}S_{yz}(S_{yy} + S_{zz} - 2S_{xx}) \\ c_2 = S_{yz}^3 + S_{yz}(S_{xz}^2 - 2S_{xy}^2 - S_{xx}^2) + S_{yz}(S_{xx}S_{zz} + S_{xx}S_{yy} - S_{yy}S_{zz}) + \\ \quad + S_{xy}S_{xz}(S_{xx} + S_{yy} - 2S_{zz}) \\ c_3 = S_{xy}(S_{yz}^2 - S_{xz}^2) + S_{xz}S_{yz}(S_{xx} - S_{yy}) \end{cases}$$

Ensuite, la résolution de l'équation du troisième degré :

$$r = \frac{c_2}{c_3}; s = \frac{c_1}{c_3}; t = \frac{c_0}{c_3}; p = s - \frac{r^2}{3}; q = \frac{2r^3}{27} - \frac{rs}{3} + t; R = \frac{q^2}{4} + \frac{p^3}{27}$$

$$\text{Si } R > 0 \rightarrow a = -\frac{r}{3} + \sqrt[3]{-\frac{q}{2} + \sqrt{R}} + \sqrt[3]{-\frac{q}{2} - \sqrt{R}}$$

$$\text{Si } R < 0 \rightarrow \begin{cases} \rho = \sqrt{-\frac{p^3}{27}} \\ \varphi = \arccos\left(\frac{-q}{2\rho}\right) \end{cases} \begin{cases} a_1 = -\frac{r}{3} + 2\sqrt[3]{\rho} \cos\left(\frac{\varphi}{3}\right) \\ a_2 = -\frac{r}{3} + 2\sqrt[3]{\rho} \cos\left(\frac{\varphi + 2\pi}{3}\right) \\ a_3 = -\frac{r}{3} + 2\sqrt[3]{\rho} \cos\left(\frac{\varphi + 4\pi}{3}\right) \end{cases}$$

$$b = \frac{S_{xy}S_{yz}a^2 + (S_{yz}^2 - S_{xy}^2)a - S_{xy}S_{yz}}{(S_{yz}(S_{xx} - S_{yy}) - S_{xy}S_{xz})a + S_{xy}(S_{yy} - S_{zz}) + S_{xz}S_{yz}}$$

S'il y a plusieurs valeurs réelles de a, calculer chaque b correspondant, ainsi que les sommes

$$\sum \delta_k^2 = \frac{1}{a^2 + b^2 + 1} \sum (ax_k + by_k + z_k)^2 \text{ la plus petite détermine le choix optimal.}$$

L'équation du plan (Π) : $Ax + By + Cz + 1 = 0$ est ainsi obtenue avec :

$$\lambda = \frac{1}{n} \left(a \sum X_k + b \sum Y_k + \sum Z_k \right) \rightarrow C = \frac{1}{\lambda}; A = -\frac{a}{\lambda}; B = -\frac{b}{\lambda}$$

Une comparaison de cette méthode avec d'autre méthode de l'état de l'art, à savoir la méthode basée sur l'analyse en composantes principales, est donnée dans [Jacquelin 11].

A.3 Discussion sur le choix de l'équation $Ax+By+Cz+1=0$

Un plan est défini par trois paramètres indépendants, non pas quatre. En conséquence, la régression ne peut porter que sur trois paramètres indépendants.

Si l'équation du plan est $Ax+By+Cz+D=0$, l'un des paramètres doit être fixé arbitrairement. Le choix s'est porté sur $D=1$. Mais on aurait aussi bien pu choisir de fixer $A=1$ et d'optimiser B, C, D . Ou de choisir $B=1$, ou $C=1$. Tout cela est possible avec des calculs similaires.

Il est alors possible, dans certains cas, que ce plan calculé passe très près de $(0,0,0)$ ce qui correspondra à de grandes valeurs de A, B, C . Il n'y a donc pas de problème, hormis éventuellement des difficultés de calcul liés aux grands nombres.

Même dans ce cas exceptionnel, il n'y a pas de difficulté car une simple translation du système d'axes permet de se retrouver dans la situation où le plan ne passe pas par la nouvelle origine, ni à son voisinage. Et le calcul se fait alors sans difficulté.

Bibliographie

- [Adiv 85] G. Adiv. Determining 3D motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 7(4), 384–401, 1985.
- [Barjatya 04] A. Barjatya, “Block Matching Algorithm for motion estimation”, in Technical Report, Utah State, 2004
- [Barron 94] J. Barron, D. Fleet, S. Beauchemin. Performance of optical flow techniques. *Int. J. of Comput. Vis.* 12(1), 43–77, 1994.
- [Bekkari 10] A. Bekkari, K. Housni, S. Idbraim, D. Mammass and Y. Chahir, Classification of high resolution urban satellite Images combining SVM and Graph Cuts 5th International Symposium on Image/Video Communications and Mobile Networks September 30, October 1-2, 2010, Rabat Maroc.
- [Besag 86] J. Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society - Series B*, vol. 48, no. 3, pages 259–302, 1986.
- [Bouthemy 93] Patrick Bouthemy & Edouard François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. J. Computer Vision (IJCV)*, vol. 10, no. 2, pages 157–182, 1993.
- [Boykov 00] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. 3rd. International Workshop on EMMCVPR. Springer-Verla, September 2000.
- [Boykov 01a] Y. Boykov, O. Veksler & R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Machine Intell. (PAMI)*, vol. 23, November 2001.
- [Boykov 01b] Yuri Y. Boykov & Marie-Pierre Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *Proc. Int. Conf. Computer Vision (ICCV)*, volume 1, pages 105–112, July 2001.
- [Boykov 04] Y. Boykov, and V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 9, 1124-1137. 2004.
- [Cheng 95] Y. Cheng, “Mean shift mode seeking and clustering,” *IEEE Trans, PAMI*, vol. 17, pp. 790-799, 1995.
- [Cheung 03] S-C.S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice, France, 2003.
- [Cook 98] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1998.

- [Costeira 98] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *Int. J. Comput. Vision*, 29(3) :159–179, 1998.
- [Cox 1996] I. Cox et S. Hingorani. An E-cient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking. *IEEE PAMI*, vol. 18, no.
- [Desolneux 03] A Desolneux, L Moisan, J Morel, A Grouping Principle and Four Applications, *IEEE Trans. Pattern Anal. Machine Intell*, Vol. 25, n. 4, pp.508-513, 2003.
- [Dibos 05] F. Dibos, G. Pelletier, S. andKoepfler. – Real-time segmentation of moving objects in a video sequence by a contrario detection. *Proc. Int. Conf. Image Processing*, 2005.
- [Dinic 70] E.A.Dinic. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Doklady Akademii Nauk SSSR*, 1277–1280, 1970.
- [Dufournaud 00] Y. Dufournaud, Cordelia Schmid, Radu Horaud, Matching Images with Different Resolutions, *International Conference on Computer Vision & Pattern Recognition*, June 2000.
- [Dupont 06] Romain Dupont, Nikos Paragios, Renaud Keriven et Philippe Fuchs. Extraction de Couches de Même Mouvement via des Techniques Combinatoires, au congré français Reconnaissance des Formes et Intelligence Artificielle (RFIA) en Janvier 2006.
- [Edmonds 72] Edmonds J., and Karp, R.M. (1972). Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *J. ACM* 19, 248–264.
- [Felzenszwalb 06] P.F. Felzenszwalb & D.P. Huttenlocher. Efficient Belief Propagation for Early Vision. *Int. J. Computer Vision (IJCV)*, vol. 70, no. 1, pages 41–54, October 2006.
- [Fischler 81] M. A. Fischler, R. C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Comm. of the ACM*, Vol 24, pp 381-395, 1981.
- [Ford 62] L. Ford and D. Fulkerson, *Flows in Networks*. Princeton University Press. 1962.
- [Fradet 08] M. Fradet, P. Pérez & P. Robert. Semi-automatic Motion Segmentation with Motion Layer Mosaics. In *Proc. Euro. Conf. Computer Vision (ECCV)*, pages 210–223, december 2008.
- [Goldberg 88] A. V. Goldberg and R. Tarjan, A new approach to the maximum flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940. October 1988.

- [Goldberg 88] A. V. Goldberg and R. Tarjan, A new approach to the maximum flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940. October 1988.
- [Greig 86] D. M. Greig, B. T. Porteous, and A. H. Seheult. Discussion of : “On the statistical analysis of dirty pictures” (by j. e. besag.). *Journal of the Royal Statistical Society B* 48, 282~284. 1986.
- [Greig 89] D. M. Greig, B. T. Porteous, and A. H. Seheult. “Exact maximum a posteriori estimation for binary images.” *J. R. Statist. Soc. B*, 51, 271-279, 1989.
- [Gyaourova 03] A. Gyaourova, C. Kamath, and S.-C. Cheung, Block matching for object tracking, LLNL Technical report. UCRL-TR-200271, October 2003.
- [Hariharakrishnan 05] K. Hariharakrishnan, D. Schonfeld, P. Raffy, F. Yassa, Video tracking using block matching, *ICIP* (3), pp.945-948, 2003.
- [Harris 88] Harris, C., and Stephens, M. A Combined Corner and Edge Detector. In 4th ALVEY Vision Conference, pp. 147-151. 1988.
- [Harris 88] C. Harris, M. Stephens. – A combined corner and edge detector. *Alvey Vision Conference*, 1988.
- [Horn 80] B.K.P Horn et B.G. Schunck, Determining Optical Flow,' *A.I. Memo No. 572, Massachusetts Institute of Technology*, 1980.
- [Horn 81] B.K.P Horn et B.G. Schunck, Determining Optical Flow, *Artificial Intelligence*, Vol. 17, pp. 185-204, 1981.
- [Housewright 74] R.B. Housewright, R.A. Singer, R.G. Sea, Derivation and evaluation of improved tracking filters for use in dense multitarget environments, In *IEEE Transactions on Information Theory*, Vol. 20, July 1974.
- [Housni 09a] K. Housni, D. Mammass, Y. Chahir, Interactive ROI Segmentation using Graph Cuts, *International Journal on Graphics, Vision and Image Processing*, Volume 9, Issue 6, Pages 1-6, December 2009.
- [Housni 09b] K. Housni, Y. Chahir, D. Mammass, Automatic Moving Object Segmentation based on Graph Cut and Optical Flow. 2nd International Conference on Applied Informatics (ICAI 2009) 15-17 November 2009 Algérie.
- [Housni 10] K. Housni, A. Bekkari, D. Mammass, Y. Chahir, Détection de tumeurs cérébrale dans les images IRM par SVM et GraphCuts , deuxième Journées Doctorales en Technologies de l'Information et de la Communication (JDTIC'10) le 15-17 Juillet 2010 Fes, Maroc.
- [Housni 12a] K. Housni, D. Mammass, Y. Chahir, Moving Objects Tracking in Video by Graph Cuts and Parameter Motion Model , *International Journal of Computer Applications (IJCA)*, February 2012 Edition. Volume 40, No. 10, pages 20-27, February 2012.

- [Housni 12b] K. Housni, D. Mammass, Y. Chahir, Coherent Motion Detection and Motion Layers Extraction of Similar Motion based on Image Feature Points and Graph Cuts Algorithm, *International Review on Computers and Software (I.RE.CO.S.)*, Vol. 7, n. 4, pages 1478- 1485, 2012.
- [Hsu 84] 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 :73–106, 1984.
- [Inoue 01] K. Inoue and K. Urahama. Separation of multiple objects in motion images by clustering. In *International Conference on Computer Vision (ICCV'01)*, pages 219–224, 2001
- [Isard 96] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density,” in *Proc. European Conf. Computer Vision*, 1996, pp. 343–356.
- [Isard 98] M. Isard, A. Blake. Condensation – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1) :5–28, 1998.
- [Ishikawa 03] H. Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *IEEE Trans. Pattern Anal. Machine Intell. (PAMI)*, vol. 25, no. 10, pages 1333–1336, October 2003.
- [Ishikawa 09] Hiroshi Ishikawa. Higher-order clique reduction in binary graph cut. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2993 – 3000. IEEE, 2009.
- [Jain 79] R. Jain, H.H. Nagel. – On the analysis of accumulative difference pictures from image sequence of real world scenes. *IEEE Trans. Pattern Anal. Machine Intell.*, 1(2) :206–214, 1979.
- [Jehan-Besson 01a] S. Jehan-Besson, M. Barlaud, and G. Aubert. Region-based active contours for video object segmentation with camera compensation. In *International Conference on Image Processing*, pages 61–64, 2001.
- [Jehan-Besson 01b] S. Jehan-Besson, M. Barlaud, and G. Aubert. Video objects segmentation using eulerian region-based active contours. In *ICCV*, pages 353–361, 2001.
- [Juan 06] O. Juan & Y. Boykov. Active Graph Cuts. In *Proc. Conf. Computer Vision Pattern Rec. (CVPR)*, volume I, pages 1023–1029, 2006.
- [Karmann 90] K.P. Karmann and A. Von Brandt. *Time-varying image processing and moving object recognition*. Elsevier, 1990.
- [Kirkpatrick 83] S. Kirkpatrick, C. D. Gelatt & M. P. Vecchi. Optimization by Simulated Annealing. *Science*, vol. 220, no. 4598, pages 671–680, May 1983.
- [Kohli 05] P. Kohli & P.H.S. Torr. Efficiently Solving Dynamic Markov Random Fields Using Graph Cuts. In *Proc. Int. Conf. Computer Vision (ICCV)*, volume II, pages 922–929, 2005

- [Kohli 07] P. Kohli & P.H.S. Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *IEEE Trans. Pattern Anal. Machine Intell. (PAMI)*, vol. 29, no. 12, pages 2079–2088, December 2007
- [Kolmogorov 04] V. Kolmogorov, FEBRUARY, What Energy Functions Can Be Minimized via Graph Cuts? *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 26, NO. 2. 2004.
- [Kolmogorov 06b] Vladimir Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Pattern Anal. Machine Intell. (PAMI)*, vol. 28, no. 10, pages 1568–1583, 2006.
- [Kolmogorov 07] V. Kolmogorov, C. Rother. – Minimizing nonsubmodular functions with graph cuts-a review. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(7) :1274–1279, 2007.
- [Kon 00] J. Konrad. *Handbook of Image and Video Processing*, chapter Motion detection and estimation, pages 207–225. Academic Press, 2000.
- [Lee 97] C.H. Lee and L.H. Chen. A fast estimation algorithm based on the block sum pyramid. In *Int. Trans. on Image Processing*, volume 6, November 1997.
- [Lin 98] C-W. Lin, Y-J. Chang, and Y-C. Chen. Hierarchical motion estimation algorithm based on pyramidal successive elimination. In *Proc. of the Int. Computer Symp.*, pages 41–44, Tainan, Taiwan, October 1998.
- [Ling 06] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 246–253, 2006.
- [Ling 07] H. Ling and K. Okada. An efficient earth mover’s distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5) :840–853, 2007.
- [Lisani 03] J. Lisani, J. Morel. – Detection of major changes in satellite images. *Proc. Int. Conf. Image Processing*, 2003.
- [Lowe 04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2) :91–110, 2004.
- [MacCormick 00] J. MacCormick, A. Blake. – A probabilistic exclusion principle for tracking multiple objects. *Int. J. Computer Vision*, 39(1) :57–71, 2000.
- [Malik 97] J. Malik and S. Russell, “Traffic Surveillance and Detection Technology Development: New Traffic Sensor Technology,” Univ. of California, Berkeley, California PATH Research Final Rep., UCB-ITS-PRR-97-6, 1997.
- [Mansouri 03] A.R. Mansouri, J. Konrad. 2003. Multiple motion segmentation with level sets. *IEEE Trans. Image Process.* 12(2), 201–220.

- [Mikolajczyk 04] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, *International Journal of Computer Vision*, Vol. 60, n. 1, pp.63-86, 2004.
- [Mikram 06] M. Mikram, R. Megret, Y. Berthoumieu, and M. Donias, Multi-Scale Histograms for Kernel-Based Object Tracking, ISCCSP, 2006.
- [Mitiche 96] A. Mitiche and P. Bouthemy. Computation and analysis of image motion : A synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1) :29–55, 1996.
- [Montesinos 98] P. Montesinos, V. Gouet, and R. Deriche, Differential invariants for color images, *International conference on pattern recognition*, 1998.
- [Murty 68] K. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Resear.* 16, 682–686, 1968.
- [Odobez 95] J.-M. Odobez, P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348-365, 1995.
- [Odobez 98] J.-M. Odobez & P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, vol. 66, no. 2, pages 143–155, 1998.
- [Olivier 93] F. Olivier, *Three-Dimensional Computer Vision (Artificial Intelligence)*. The MIT Press, November 1993.
- [Paragios 00] N. Paragios and R. Deriche. Geodesic active contour and level sets for the detection and tracking of moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(3) :266–280, March 2000.
- [Paragios 05] N. Paragios, Y. Chen, and O. Faugeras, Chapitre "Optical Flow Estimation" du livre "Mathematical Models in Computer Vision: The Handbook", pages 239-257. Springer, 2005.
- [Park 04] J. H.Park, H. Zha, and R. Kasturi. Spectral clustering for robust motion segmentation. In *European Conference on Computer Vision*, pages 390–401, 2004.
- [Pearl 82] J . Pearl. "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach, " *Proceedings of the second annual conference on Artificial Intelligence*, Pittsburgh , Pennsylvania , August 1982.
- [Peterfreund 00] N. Peterfreund, "Robust tracking of position and velocity with Kalman snakes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 564–569, June 2000.
- [Piccardi 04] M. Piccardi. Background subtraction techniques: a review. In *Proc. IEEE Conference on Computer*, February 2004.

- [Poon 09] H. S. Poon, F. Mai, Y. S. H. G. Chesi, *Robust Detection and Tracking of Multiple Moving Objects with 3D Features by an Uncalibrated Monocular Camera*, 4th international conference, MIRAGE, pp. 140-149, 2009.
- [Qin 05] L. Qin, W. Gao, Image Matching Based on A Local Invariant Descriptor, IEEE International Conference on Image Processing, Volume 3, 11-14, Page(s):377 – 380, Sept. 2005.
- [Rabin 09] J. Rabin, J. Delon, and Y. Gousseau. A statistical approach to the matching of local features. *SIAM Journal of Imaging Sciences*, 2(3) :931–958, 2009.
- [Radke 05] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms : a systematic survey. *IEEE Transactions on Image Processing*, 14(3) :294–307, March 2005.
- [Ranchin 04] F. Ranchin and F. Dibos. Moving objects segmentation using optical flow estimation. In *Mathematics, Image and Analysis, Les Cahiers du CEREMADE 2004-41*, 2004.
- [Reid 1979] D. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, vol. 24, no. 6, pages 843-854, 1979.
- [Rosenhahn 05] B. Rosenhahn, U. Kersting, S. Andrew, T. Brox, R. Klette and H-P. Seidel. A silhouette based human motion tracking system Technical Report (The University of Auckland, New Zealand: CITR) ISSN: 1178-3581, 2005.
- [Rother 05] C. Rother, S. Kumar, V. Kolmogorov, and A. Blake. Digital tapestry. In *CVPR*, 2005.
- [Roy 98] S. Roy & I. J. Cox. A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem. In *Proc. Int. Conf. Computer Vision (ICCV)*, 1998.
- [Roy 00] S. Roy, and V. Govindu, Mrf solutions for probabilistic optical flow formulations. *icpr 03* (2000).
- [Schmid 96] C. Schmid. Appariement d'images par invariants locaux de niveaux de gris. Doctoral thesis, Institut National Polytechnique de Grenoble, GRAVIR, July 1996.
- [Sethi 87] I. K. Sethi and R. C. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions Pattern Analysis Machine Intelligence*, 9(1) :56–73, 1987.
- [Shi 94] J. Shi, C. Tomasi, Good features to track, *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Seattle, pp. 593-600. 1994.
- [Smith 05] K. Smith, D.G. Perez, et J.M. Odobez (2005), 'Using particles to track varying numbers of interacting people', 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'.

- [Stauffer 00] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real time tracking. *IEEE trans. on PAMI*, 8(22) :747–757, August 2000.
- [Szeliski 06] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, and al. A comparative study of energy minimization methods for markov random fields. In *European Conference on Computer Vision (ECCV'06) (2006)*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3952 of LNCS, Springer, pp. 16-29
- [Tappen 03] M. F. Tappen, and W. T. Freeman, Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. *iccv 02 (2003)*.
- [Tekalp 95] Tekalp, A. (1995), *Digital Video Processing*, Prentice Hall.
- [Terzopoulos 92] D. Terzopoulos, R. Szeliski. Tracking with kalman snakes. *Active vision*, pp. 3–20, 1992.
- [Tomasi 91] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.
- [Toyoma 99] K. Toyoma, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : principles and practice of background maintenance. In *Proc. IEEE Conference on Computer*, pages 255–261, Kerkyra, Greece, 1999.
- [Trichet 07] R. Trichet and B. Mérialdo. Probabilistic matching algorithm for keypoint based object tracking using a delaunay triangulation. In *International Workshop on Image Analysis for Multimedia Interactive Services*, Santorini, Greece, June 2007.
- [Vapnik 95] V. Vapnik, “The nature of statistical learning theory”, Springer, 1995.
- [Veenman 01] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions Pattern Analysis Machine Intelligence*, 23(1) :54–72, 2001.
- [Veit 04a] T. Veit, F. Cao, and P. Bouthemy. Probabilistic parameter-free motion detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.
- [Veit 04b] T. Veit, F. Cao, P. Bouthemy. – An a contrario framework for motion detection. *Rapport de Recherche n5313*, INRIA, 2004.
- [Veit 05] T. Veit. – Détection et analyse du mouvement dans des séquences d’images selon une approche probabiliste a contrario. – PhD. Thesis, Université de Rennes 1, Mention Traitement du Signal et des Télécommunications, December 2005.
- [Veit 07] T. Veit, F. Cao, P. Bouthemy, *Space-time A Contrario Clustering for Detecting Coherent Motions*, IEEE International Conference on Robotics and Automation Roma, Italy, pp. 10-14, 2007.

- [Wainwright 05] M.J. Wainwright, T.S. Jaakkola & A.S. Willsky. MAP estimation via agreement on (hyper) trees : message-passing and linear programming approaches. *IEEE Trans. on Information Theory*, vol. 51, no. 11, pages 3697–3717, November 2005.
- [Wren 97] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland.. Pfinder: real-time tracking of the human body. *IEEE Trans. Pattern Anal.Mach. Intell.* 19(7), pages 780–785, 1997.
- [Wills 03] J. Wills, S. Agarwal & S. Belongie. What Went Where. In *Proc. Conf. Computer Vision Pattern Rec. (CVPR)*, volume 1, pages 37–454, Madison, WI, June 2003.
- [Wu 01] Y.Wu, Z. Zhang, T. S. Huang, and J. Y. Lin. Multibody grouping via orthogonal subspace decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 2, page 252, 2001.
- [Wu 05] Q. Wu, H. Cheng, B. Jeng. – Motion detection via change-point detection for cumulative histograms of ratio images. *Pattern Recognition Letters*, 26(5) :555–563, 2005.
- [Xiao 05] J. Xiao & M. Shah. Motion Layer Extraction in the Presence of Occlusion Using Graph Cuts. *IEEE Trans. Pattern Anal. Machine Intell. (PAMI)*, vol. 27, no. 10, pages 1644–1659, October 2005.
- [Xu 02] N. Xu, N. Ahuja. Object contour tracking using graph cuts based active contours. *Proc. Int. Conf. Image Processing*, 2002
- [Yilmaz 06] A. Yilmaz, O. Javed, and M. Shah. Object tracking : A survey. *ACM Computing Surveys*, 38(4) :13, 2006.
- [Zhu 05] S. Zhu, Q. Avidan, K.-T. Cheng. Learning a sparse, corner-based representation for time-varying background modeling. *Proc. Int. Conf. Computer Vision*, 2005.

Liste des publications relatives aux travaux de thèse

Journaux internationaux

- ⊕ **Coherent Motion Detection and Motion Layers Extraction of Similar Motion based on Image Feature Points and Graph Cuts Algorithm.** Khalid Housni, Driss Mammass, Youssef Chahir, International Review on Computers and Software (I.RE.CO.S.), Vol. 7, n. 4. P 1478-1485, July 2012.
- ⊕ **Moving Objects Tracking in Video by Graph Cuts and Parameter Motion Model.** Khalid Housni, Driss Mammass, Youssef Chahir, International Journal of Computer Applications IJCA, Volume 40, N°10, P: 20—27, February 2012.
- ⊕ **Interactive ROI Segmentation using Graph Cuts.** Khalid Housni, Driss Mammass, Youssef Chahir, ICGST International Journal on Graphics, Vision and Image Processing GVIP, Volume 9, Issue 6, P: 1—6, December 2009.

Conférences internationales

- ⊕ **Classification of high resolution urban satellite Images combining SVM and Graph Cuts** A. Bekkari, K. Housni, S. Idbraim, D. Mammass and Y. Chahir, 5th International Symposium on Image/Video Communications and Mobile Networks September 30, October 1-2, 2010, Rabat Maroc.
- ⊕ **Automatic Moving Object Segmentation based on Graph Cut and Optical Flow.** Khalid Housni, Youssef Chahir, Driss Mammass, 2nd International Conference on Applied Informatics (ICAI 2009) 15-17 November Algérie.
- ⊕ **Extraction de zones d'intérêt par graph cuts.** Khalid Housni, Driss Mammass, Youssef Chahir, Fifth International Conference SETIT 2009, Hammamet 22-26, March 2009, Tunisia.

Conférences nationales

- ⊕ **Détection de tumeurs cérébrale dans les images IRM par SVM et GraphCuts** Khalid Housni, Aissam Bekkari, Driss Mammass, Youssef Chahir, deuxième Journées Doctorales en Technologies de l'Information et de la Communication (JDTIC'10) le 15-17 Juillet 2010 Fes, Maroc.
- ⊕ **Coupes de graphe – théorie et applications** Khalid Housni, Driss Mammass, Youssef Chahir, Journée de mathématiques et applications, 10 juillet 2010, Casablanca Maroc.
- ⊕ **Restauration d'images par l'algorithme alpha-expansion.** Khalid Housni, Driss Mammass, Youssef Chahir, Premières Journées Doctorales en Technologies de l'Information et de la Communication (JDTIC'09) le 16-18 Juillet 2009 Rabat, Maroc.