Détection et suivi d'objets en mouvement dans des scenes complexes, application a la surveillance des conducteurs.

Aurélie Bugeau

N^o d'ordre : 3644

THÈSE

présentée

DEVANT L'UNIVERSITÉ DE RENNES I

pour obtenir

le grade de : ***DOCTEUR DE L'UNIVERSITÉ DE RENNES I***

Mention : Traitement du signal et télécommunications

par

Aurélie Bugeau

Équipe d'accueil : VISTA (IRISA, RENNES)

École doctorale : Mathématiques, Informatique, Signal, Électronique et Télécommunications

Composante Universitaire : SPM, Université de Rennes 1

Titre de la thèse :

DÉTECTION ET SUIVI D'OBJETS EN MOUVEMENT DANS DES SCÈNES COMPLEXES, APPLICATION À LA SURVEILLANCE DES CONDUCTEURS.

Composition du jury :

M.	Jean-Marc	ODOBEZ	Rapporteurs
M.	Nikos	PARAGIOS	
M.	Michel	BARLAUD	Examinateurs
Mme.	Jenny	BENOIS-PINEAU	
M.	Patrick	BOUTHEMY	
M.	Patrick	PÉREZ	Directeur

Table des matières

Notations **5**

Introduction générale **7**

I Détection d'objets en mouvement dans des scènes complexes **11**

Introduction **13**

1 État de l'art sur la détection et la segmentation du mouvement **15**
1.1 Détection basée sur la différence inter-images 16
1.1.1 Décision sur un pixel avec un seuil global 16
1.1.2 Décision basée région 17
1.2 Modélisation du fond 17
1.2.1 Méthodes non prédictives 18
1.2.2 Méthodes prédictives 20
1.3 Extraction de couches de mouvement 20
1.4 Détection de mouvement basée sur la cohérence 21
1.5 Conclusion et approche proposée 22
1.5.1 Particularité des séquences étudiées 22
1.5.2 Analyse des méthodes de l'état de l'art 23
1.5.3 Présentation de l'approche proposée 25

2 Approche locale : sélection et description de points **27**
2.1 Détection des pixels en mouvement 27
2.2 Sélection d'une grille de pixels en mouvement 30
2.3 Description des points sélectionnés 32
2.3.1 Caractéristiques de mouvement 32
2.3.2 Caractéristiques photométriques 37
2.4 Conclusion 41

3 Partitionnement des points **43**
3.1 Les méthodes de clustering 43
3.1.1 Approches basées sur le partitionnement des données 44
3.1.2 Approches hiérarchiques 44
3.1.3 Approches paramétriques d'estimation de la densité 44

3.1.4 Approches non paramétriques d'estimation de la densité 45
3.2 Estimateurs à noyau 45
3.2.1 Estimateur à noyau fixe 46
3.2.2 Estimateur sample point 46
3.2.3 Estimateur balloon 47
3.2.4 Qualité d'un estimateur à noyau fixe 47
3.2.5 Profile d'un estimateur à noyau 49
3.3 Filtrage et clustering mean shift 49
3.3.1 Filtrage mean shift à noyau fixe 50
3.3.2 Mean shift basé sur l'estimateur "sample point" : 51
3.3.3 Mean shift basé sur l'estimateur "balloon" 51
3.4 Conclusion 53

4 Sélection automatique de la taille du noyau pour le partitionnement mean shift 55
4.1 État de l'art sur la sélection de la taille d'un noyau 56
4.1.1 Méthodes statistiques pour la sélection du noyau 56
4.1.2 Méthodes basées sur l'évaluation des partitions 57
4.2 Sélection de la taille du noyau dans des espaces hétérogènes de grande dimension 58
4.2.1 Principe de l'algorithme de sélection 58
4.2.2 Sélection itérative 60
4.3 Application à la segmentation d'images couleur 62
4.3.1 Validation de l'approche itérative 64
4.3.2 Validation du clustering "pseudo balloon mean shift" 65
4.3.3 Ordre des espaces de caractéristiques 68
4.4 Application à la détection d'objets en mouvement 69
4.4.1 Détection des clusters en mouvement 69
4.4.2 Un premier résultat 71

5 Segmentation des objets en mouvement et résultats expérimentaux 73
5.1 Segmentation des objets en mouvement 73
5.1.1 Définition de la fonction d'énergie 74
5.1.2 Réglage des paramètres 75
5.1.3 Minimisation de l'énergie et validation des objets 76
5.2 Résultats de détection et segmentation des objets en mouvement 76
5.2.1 Résultats sur des séquences couleur 76
5.2.2 Résultats sur une séquence en niveau de gris 81
5.3 Comparaison avec d'autres méthodes de détection de mouvement 81

Conclusion 87

II Segmentation et suivi d'objets en mouvement 89

Introduction 91

6 Coupe minimale/flot maximal dans un graphe **93**
6.1 Définitions et propriétés sur les graphes . . . 93
6.1.1 Graphe . . . 93
6.1.2 Coupe . . . 94
6.1.3 Flot . . . 95
6.1.4 Relation entre coupe minimale et flot maximal . . . 95
6.1.5 Définitions : saturation, graphe résiduel, chemin augmentant . . . 97
6.1.6 Équivalence entre flot maximal et coupe minimale . . . 99
6.2 Méthodes pour trouver le flot maximal . . . 100
6.2.1 Flot maximal par saturation des chemins . . . 100
6.2.2 Flot maximal par poussage de flot . . . 102
6.3 Minimisation d'énergies . . . 103
6.3.1 Présentation des énergies considérées . . . 103
6.3.2 Algorithmes d'étiquetage et de minimisation . . . 103
6.3.3 Minimisation d'énergies par coupe minimale/flot maximal . . . 105
6.4 Applications à la segmentation d'images . . . 108
6.4.1 Segmentation d'images par coupe minimale/flot maximal . . . 108
6.4.2 Conlcusion : application à la détection et à la segmentation des objets en mouvement . . . 110

7 État de l'art sur le suivi d'objets **113**
7.1 Suivi par appariement de détections . . . 113
7.1.1 Méthodes déterministes . . . 114
7.1.2 Méthodes probabilistes . . . 114
7.1.3 Avantages et limitations . . . 115
7.2 Segmentation dynamique . . . 115
7.2.1 Suivi avec une représentation explicite du contour . . . 115
7.2.2 Minimisation d'une fonction d'énergie sur contour : représentation implicite 116
7.2.3 Avantages et limitations . . . 117
7.3 Suivi déterministe de fenêtre englobante par détection séquentielle . . . 117
7.3.1 Suivi différentiel d'imagettes . . . 118
7.3.2 Suivi de distributions . . . 118
7.3.3 Avantages et limitations . . . 118
7.4 Conclusion . . . 119

8 Suivi et segmentation par coupe minimale/flot maximal **121**
8.1 Principe . . . 122
8.1.1 Notations et caractérisation des observations . . . 122
8.1.2 Principe général . . . 122
8.1.3 Principe de la méthode de suivi . . . 123
8.1.4 Principe de la méthode de séparation des objets fusionnés . . . 124
8.2 Fonction d'énergie pour le suivi de chaque objet . . . 124
8.2.1 Forme du graphe . . . 125
8.2.2 Fonction d'énergie . . . 126
8.2.3 Création automatique de nouvelles pistes . . . 129
8.3 Séparation des objets ayant fusionné . . . 129

8.3.1 Forme du graphe 130
8.3.2 Énergie multi-étiquettes 130
8.4 Résultats Expérimentaux 131
8.4.1 Suivi d'objets détectés par soustraction de fond 132
8.4.2 Détection, segmentation et suivi d'objets dans des scènes complexes . . . 134

Conclusion **141**

Conclusion générale et perspectives **143**

Annexes **147**

A Preuve de convergence de l'algorithme pseudo balloon mean shift **149**

B Critère de stabilité pour la sélection de noyaux **151**

C Preuve du théorème 4 **155**

Liste des figures **156**

Liste des algorithmes **161**

Bibligraphie **163**

Liste des publications relatives aux travaux de thèse **175**

Notations

Nous présentons ici les notations utilisées tout le long du document.

Fonctions et opérateurs	
$\lfloor \cdot \rfloor$	Partie entière
$\langle \cdot \rangle$	Moyenne sur un petit voisinage autour de $\cdot$
$\lvert \cdot \rvert$	Cardinal d'un ensemble
$\cdot * \cdot$	Opérateur de convolution
$\nabla \cdot$	Gradient d'une fonction
$\Delta \cdot$	Laplacien d'une fonction

Notations	
$\mathcal{P}$	Ensemble des pixels de l'image
N	Nombre de pixels contenus dans une image ($N = \lvert\mathcal{P}\rvert$)
I_t	Image au temps t d'une séquence d'images
$s = (x, y)$	Un pixel de $\mathcal{P}$
$\mathbf{z}_t(s)$	Vecteur caractérisant le pixel s au temps t
$\mathbf{z}_t^{(G)}(s)$	Intensité du pixel s dans l'image I_t
$\mathbf{z}_t^{(C)}(s)$	Couleur du pixel s dans l'image I_t. Vecteur de dimension 3.
$\mathbf{z}_t^{(M)}(s)$	Mouvement au pixel s au temps t. Vecteur de dimension 2.
d_x	Mouvement horizontal
d_y	Mouvement vertical
G	Grille de points
$V(s)$	Voisinage de taille n autour du pixel s
$\mathcal{G}$	Graphe
$\mathcal{V}$	Ensemble des nœuds d'un graphe
$\mathcal{E}$	Ensemble de paires de pixels voisins
$\mathcal{E}_\mathcal{P}$	Ensemble de paires de pixels voisins de $\mathcal{P}$
$\mathcal{O}_t^{(i)}$	i^{eme} objet au temps t
$\mathcal{M}_t^{(i)}$	j^{eme} objet au temps t

Introduction générale

Avec la généralisation de l'utilisation d'images numériques, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi diverses que la vidéo surveillance, la compression vidéo, l'imagerie médicale, la robotique, l'interaction homme-machine, l'analyse de séquences sportives... En effet, les zones de mouvement d'une séquence d'images correspondent souvent à des événements sur lesquels un système de vision doit se focaliser.

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

- la détection du mouvement, qui consiste à étiqueter chaque pixel d'une image suivant si il correspond ou non à une région en mouvement dans la scène,

- la détection des objets en mouvement, c'est-à-dire la détection d'un ensemble de régions d'intérêt en mouvement dans la scène tridimensionnelle observée,

- la segmentation basée mouvement de la scène, pour laquelle chaque région de l'image ayant un mouvement distinct des autres est détectée et segmentée,

- l'estimation du mouvement, qui consiste à estimer, à partir d'une séquence d'images, le mouvement apparent des objets composants une scène tridimensionnelle,

- le suivi de primitives ou de régions, dont le but est de déterminer la position de chaque primitive ou région dans l'image à chaque instant,

- la reconnaissance et la modélisation d'activités ou de gestes.

Les trois premières problématiques (détection du mouvement, détection des objets en mouvement et segmentation basée mouvement), qui sont au cœur des travaux présentés, sont en général une première étape pour des outils automatiques de vision par ordinateur. Ces outils peuvent avoir pour vocation, soit uniquement de détecter, soit de détecter et reconnaître, soit de détecter et suivre des objets pour, par exemple, analyser le comportement ou la trajectoire de ces objets. Ainsi, des méthodes de détection fiables et automatiques sont indispensables pour de nombreuses applications de vision par ordinateur.

Les autres problématiques sont toutes aussi importantes et nécessitent la mise en place de méthodes simples et robustes. Tous ces sujets font l'objet d'un grand nombre de travaux, mais il n'existe pas, à l'heure actuelle, d' algorithmes aboutis s'adaptant à n'importe quelle situation.

Contexte

Cette thèse a débuté dans le contexte de l'Action Concertée Incitative (ACI) en sécurité et informatique, nommée "Behaviour" dont le sujet traite de la vision et l'apprentissage statistique pour la reconnaissance du comportement humain avec l'application à la surveillance du conducteur pour l'amélioration de la sécurité de la conduite automobile. Cette ACI a regroupé le laboratoire HEUDIASYC de l'université de technologie de Compiègne, le projet VISTA de l'IRISA et le centre de recherche PSA PEUGEOT CITROËN. Le travail entre ces trois équipes a été réparti de la manière suivante. Le centre de recherche de PSA a fourni un ensemble de vidéos annotées de conducteurs prises dans un simulateur de conduite. L'équipe HEUDIASYC avait pour rôle le suivi de visage et l'analyse du comportement [Hérault 06a, Hérault 06b]. Pour notre part, nous nous sommes concentrés sur la détection et le suivi simultané de parties du corps pour l'analyse des trajectoires.

Les vidéos fournies par PSA, bien que longues et annotées, avaient un certain nombre de défauts liés au fait qu'elles avaient été acquises à l'intérieur d'un simulateur en conditions de conduite nocturne. De ce fait, il s'agit de vidéos en niveau de gris (la pièce n'est pas éclairée et le conducteur n'est visible que grâce à l'utilisation d'une lumière infrarouge) avec un fond noir fixe et acquises avec des caméras immobiles (figure 0.1 a)). Dans le cadre de la détection et du suivi

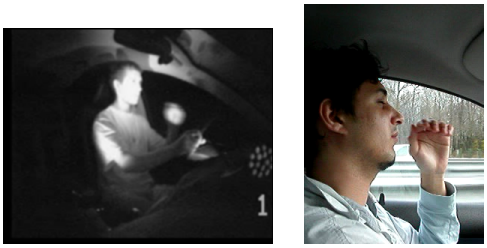
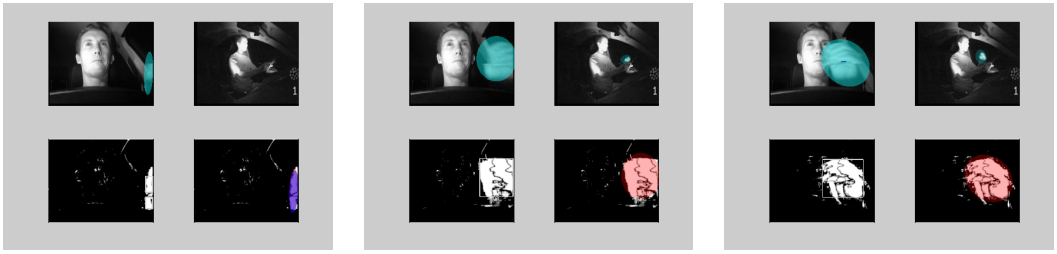


FIG. 0.1: Exemples d'images de séquences de conducteur. (a) Séquence fournie par PSA PEUGEOT CITROËN, acquise dans un simulateur. (b) Séquence en conditions réelles.

des objets en mouvement, ces vidéos sont relativement simples à traiter. En effet, la détection automatique de la main du conducteur peut être assez facilement obtenue par soustraction de fond et son suivi par une méthode telle que le filtrage de Kalman. Un exemple de résultat obtenu sur ces séquences est montré sur la figure 0.2. Nous avons alors décidé d'acquérir de nouvelles séquences, dans des conditions réelles, afin de tester la robustesse de ces algorithmes pour des vidéos plus difficiles à exploiter (figure 0.1 b)). Il s'est alors avérer que les méthodes étudiées n'étaient plus du tout adaptées. En effet, dans ces séquences prises de jour avec une caméra mobile, le conducteur, l'habitacle de la voiture et le paysage derrière les fenêtres sont visibles. Le fond, composé de l'habitacle mais aussi de toutes les régions à l'extérieur de la voiture, est dynamique : les changements dans le fond peuvent être brutaux, avec de forts changements d'illumination qui se répercutent sur le conducteur dans l'habitacle.

L'acquisition de ces vidéos dans des conditions réelles a totalement guidé cette thèse. En effet, comme nous le verrons dans les chapitres d'état de l'art de ce document, aucune méthode existante de détection et/ou de segmentation et/ou de suivi des objets en mouvement n'est adaptée à la complexité des vidéos considérées. Nous avons donc développé différents algorithmes permettant la détection et le suivi simultanés de parties du corps. Notons que nous ne nous sommes pas



t=3 t=5 t=7

FIG. 0.2: Exemples de résultats obtenus sur une séquence fournie par PSA PEUGEOT CITROËN. Première ligne : résultat du suivi par un filtre de Kalman [Kalman 60] des objets détectés. En haut à gauche, résultat sur le conducteur filmé de face. En haut à droite, résultat sur le conducteur filmé de profil. Seconde ligne : Résultats intermédiaires de la détection des objets en mouvement. En bas à gauche, résultat de la détection d'objets en mouvement extraits avec la méthode dite "gap/mountain" décrite dans [Wang 00] appliquée à une carte binaire caractérisant l'appartenance de chaque pixel au mouvement de la caméra (voir chapitre 2). En bas à droite, représentation des objets extraits par des ellipses obtenues avec la méthode décrite dans [Zivkovic 04].

uniquement concentrés sur ces séquences de conducteur, mais avons souhaité mettre en place des algorithmes génériques pouvant également s'appliquer à des séquences aux contenus dynamiques complexes de toute autre nature.

Contributions et description du document

Le but de cette thèse était finalement la détection et le suivi d'objets en mouvement dans des scènes relativement complexes. Nous avons donc, dans un premier temps, mis en place un nouvel algorithme de détection des objets en mouvement.

Nous définissons un objet en mouvement comme une région fermée de l'image ayant une couleur et un mouvement quasi constants. La détection se décompose en 3 étapes principales. Dans un premier temps, nous sélectionnons un nombre restreint de points afin de restreindre l'espace des données sur lequel les objets vont être recherchés. Nous décrivons ces points par leur position, leur mouvement et leur photométrie. Cet espace est ensuite divisé en plusieurs groupes de points, chacun correspondant à un objet en mouvement. L'utilisation des détections, pour des applications à la reconnaissance d'objets, à la surveillance, ou à de nombreux autres domaines, requière l'obtention des masques complets de chaque objet, c'est-à-dire de leur segmentation. La dernière étape de l'algorithme est donc une étape de segmentation des objets à partir des groupes de points obtenus.

Ces trois étapes font appel à un certain nombre de méthodes classiques en vision par ordinateur mais ont aussi nécessité la mise en place d'algorithmes mieux adaptés au problème. En particulier, un nouvel algorithme "mean shift" ainsi qu'un algorithme de sélection de la taille du noyau ont été mis en place pour l'étape de division des données en plusieurs groupes.

L'algorithme de détection des objets en mouvement dans des scènes complexes est décrit dans la partie I du document.

Le détecteur d'objets en mouvement proposé n'exploite pas la cohérence temporelle pouvant

exister entre les détections à des instants différents. Il n'y a donc aucune corrélation entre les détections à deux instants consécutifs. L'exploitation des détections pour un grand nombre d'applications (surveillance, analyse de séquences sportives, compression ...) nécessite l'ajout de cette cohérence temporelle. Nous avons donc, dans un deuxième temps, proposé une méthode de suivi ajoutant de la cohérence temporelle au système tout en tenant compte des nouvelles détections à chaque instant. Elle permet simultanément de segmenter et de suivre les objets. Elle se couple très bien avec l'algorithme de détection proposé précédemment, dans lequel la dernière étape (celle de segmentation) peut être omise. La méthode de suivi repose sur la théorie de coupe minimale/flot maximal dans un graphe. Elle est décrite dans la partie II de cette thèse.

Abus de langage et anglicisme

Tout au long du document, nous ferons l'abus de langage suivant. Nous parlerons de pixels en mouvement, même si un pixel dans une image est fixe. Cette terminologie fera référence à un pixel correspondant à un point physique en mouvement dans la scène tridimensionnelle observée. De même, nous parlerons de pixel pouvant bouger si le point observé dans la scène peut être en mouvement. Notons que les points dits en mouvement sont toujours en mouvement relativement aux mouvements éventuels de la caméra. De plus, nous emploierons simplement le terme d'intensité pour faire référence à l'intensité lumineuse.

Quelques anglicismes seront conservés tout le long du document. Ainsi, nous ferons référence à la division des données en plusieurs groupes par le terme anglais de "clustering". Les groupes de points obtenus seront alors appelés "cluster". Nous emploierons également parfois l'expression "Graph Cuts" pour parler des algorithmes de coupe minimale/flot maximal dans un graphe. Enfin, nous garderons les terminologies anglaises pour parler de l'algorithme "mean shift" et des estimateurs "sample point" et "balloon".

Première partie

Détection d'objets en mouvement dans des scènes complexes

Introduction

Cette partie s'intéresse à la détection instantanée des objets en mouvement dans une séquence d'images. On définit un objet comme une région fermée de couleur et mouvement quasi constants. Un objet au sens courant du terme peut, d'après cette définition, être composé de plusieurs objets définis de la manière précédente. Par exemple, une personne peut se retrouver décomposée en plusieurs objets : le buste (couleur potentiellement différente du reste du corps), ses jambes (couleurs et mouvements éventuellement différents du reste du corps), sa tête... Les séquences considérées dans cette partie sont complexes : fond dynamique avec des changements brutaux, caméra mobile, changement d'illumination... Nous décomposons cette première partie en cinq chapitres.

Chapitre 1 Le premier chapitre présente un état l'art des méthodes de détection et de segmentation de mouvement. Nous différencions les méthodes de détection basées sur la différence inter-images, celles reposant sur une modélisation du fond, celles extrayant des couches de mouvement et finalement les approches de détection de mouvement basées sur la cohérence. Toutes ces différentes techniques ont leurs propres avantages et leurs limites. Après une brève analyse des séquences complexes considérées, le principe de l'approche mise en place dans cette partie est détaillé. Cette méthode se décompose en 3 étapes principales : sélection et description d'un nombre restreint de points (chapitre 2), partitionnement de ces points (chapitre 3 et 4) et finalement segmentation de chaque objet à partir des partitions (chapitre 5).

Chapitre 2 Le deuxième chapitre présente ainsi la sélection d'un nombre restreint de points et leur description. Divers outils classiques en vision par ordinateur sont utilisés dans cette étape (estimation du mouvement dominant, estimation du flot optique, espaces de couleur ...) et sont donc détaillés successivement au cours du chapitre.

Chapitre 3 Le chapitre trois propose un état de l'art des méthodes de clustering. Nous détaillons la méthode retenue, c'est-à-dire le clustering mean shift. Il s'agit d'une méthode itérative de montée de gradient reposant sur l'utilisation d'un estimateur à noyau. Nous faisons donc quelques rappels sur l'estimation à noyau et présentons le clustering mean shift pour des noyaux fixes ou variables. Une nouvelle méthode de clustering appelée "pseudo balloon mean shift" est introduite.

Chapitre 4 Les méthodes de clustering mean shift ont une limite : le choix de la taille du noyau. Après un bref état de l'art sur le sujet, une technique de sélection de la taille du noyau est proposée. Elle est dédiée à des données composites de grandes dimensions. L'algorithme mis en place est appliqué et validé dans le contexte de la segmentation d'images couleur.

Chapitre 5 Ce dernier chapitre présente la méthode énergétique utilisée pour segmenter les objets à partir des clusters. Il montre également des résultats de l'algorithme complet de détection appliqué à plusieurs séquences ayant des particularités différentes. Une comparaison de cet algorithme avec d'autres méthodes (décrites dans le premier chapitre) de détection de mouvement termine ce chapitre.

Chapitre 1

État de l'art sur la détection et la segmentation du mouvement

Dans la première partie de ce document, nous nous intéressons à la détection d'objets en mouvement dans des scènes complexes. Les applications de la détection d'objets sont nombreuses. On peut notamment citer le suivi d'objets, l'analyse du comportement, la compression vidéo, la reconstruction 3D...

L'analyse du mouvement dans une séquence d'images repose soit sur la détection du mouvement soit sur la segmentation basée mouvement. La détection de mouvement a pour but de décider quelles parties des images (pixels ou régions) correspondent à des objets mobiles. Il s'agit d'une détection binaire. La segmentation basée mouvement est une détection multi-label dans laquelle chaque région de l'image ayant un mouvement distinct des autres est segmentée. Ces sujet ont été et sont toujours l'objet d'un nombre important de travaux. La diversité des recherches est liée à la difficulté de la tâche : les méthodes proposées doivent être le plus robuste possible aux fluctuations des intensités observées (en luminance, couleur ou température). Ces variations sont liées au capteur (légères vibrations, bruit électronique) ou au contenu dynamique de la scène (fond dynamique). De plus, même si certaines méthodes sont uniquement dédiées à des séquences acquises à l'aide d'une caméra fixe, les recherches tendent de plus en plus à considérer des vidéos acquises par une caméra mobile.

Dans la première partie de ce document, nous allons proposer un nouvel algorithme de détection des objets en mouvement dans des scènes complexes. Il consiste à détecter chaque objet en mouvement distinct du reste de la scène de part leur mouvement et leur couleur. Contrairement à la segmentation basée mouvement, la détection des objets en mouvement ne s'intéresse pas à l'ensemble de la scène mais uniquement aux régions correspondant à des objets. Ainsi le fond (éventuellement dynamique) n'est pas considéré.

Ce premier chapitre a pour but de donner un état de l'art des différentes techniques de détection et de segmentation de mouvement. La littérature sur le sujet étant extrêmement abondante, nous n'énoncerons pas toutes les méthodes existantes. L'idée est plus de donner un bref aperçu des méthodes les plus couramment utilisées et des différents types d'approches existantes pour la détection et/ou la segmentation de mouvement en vision par ordinateur. Pour cela, nous séparons les méthodes en quatre catégories. Cette classification n'est pas stricte car certaines méthodes pourraient être classées dans plusieurs de ces catégories. Nous distinguons ainsi les détections

basées sur la différence inter-images, celles basées sur la modélisation du fond, celles extrayant des couches de mouvement et enfin les détections utilisant la notion de cohérence.

Les particularités des séquences traitées dans ce document sont brièvement présentées à la fin du chapitre. Une analyse rapide des différents types de méthodes de détection ou segmentation de mouvement montre qu'aucune des méthodes existantes ne peut être directement appliquée aux vidéos complexes auxquelles nous nous intéressons. Ce chapitre se termine par une brève introduction et description de la nouvelle méthode de détection des objets en mouvement mise en place dans cette première partie du document.

1.1 Détection basée sur la différence inter-images

La méthode de détection de mouvement la plus simple consiste à faire la soustraction entre deux images successives [Jain 79]. Les pixels dont l'intensité résultante est proche de zéro sont assimilés comme étant les pixels du fond. De la même manière il est possible de soustraire à l'image courante une image de référence, cette image représentant uniquement la scène (où aucun objet en mouvement n'est présent). Ces approches basiques considèrent la caméra fixe durant l'acquisition de la séquence. Le choix du seuil de décision (proche de zéro) n'est pas évident et doit tenir compte du bruit et des changements de luminosité. De plus, ce seuil n'a aucune raison d'être global sur toute l'image. En effet, des objets et des mouvements différents entraînent des variations d'intensité différentes. De nombreuses méthodes, utilisant deux ou trois images successives, permettent de décider si un pixel a bougé. Elles s'appuient généralement sur un test d'hypothèse. Une analyse des différentes méthodes de détection de changement permettant de détecter le mouvement est donnée dans [Radke 05]. Nous reprenons ci-après [Veit 05a] et distinguons les décisions prises sur un pixel des décisions prises sur une région de l'image.

1.1.1 Décision sur un pixel avec un seuil global

Un ensemble de méthodes classiques de décision par test d'hypothèses est présenté dans [Konrad 00]. Le plus souvent, la différence d'intensité entre deux images successives est modélisée par une loi gaussienne sous l'hypothèse d'absence de mouvement et par une loi uniforme sous l'hypothèse de mouvement. Un rapport de vraisemblance sur la différence d'intensité donne alors le seuil de détection. Les paramètres des lois de probabilité influencent beaucoup les résultats et leur estimation n'est pas évidente. Le seuil de détection peut également être obtenu en utilisant des modèles statistiques de l'image plus complexes [Rosin 98]. Au lieu d'utiliser la différence d'intensité, des techniques similaires préfèrent s'appuyer sur le flot optique [Nelson 91, Thompson 93, Csurka 99, Irani 98]. Elles sont principalement dédiées au cas où la caméra est mobile. Plus récemment, l'utilisation du rapport d'intensité entre deux images a été proposée [Wu 05]. Un seuil unique de détection est automatiquement déterminé par une analyse de l'histogramme cumulé des rapports d'intensité. Le seuil global correspond à un point de changement dans l'histogramme des valeurs cumulées. L'utilisation d'un seuil global ne permet toutefois pas de bien caractériser les différents objets mobiles pouvant être présents dans la scène et la possible dynamique du fond.

La décision sur un pixel sans tenir compte de ses voisins donne des résultats sans aucune cohérence spatiale. Dans [Hsu 84] l'intensité de l'image est modélisée par des polynômes d'ordre 2 sur des petits supports. L'utilisation de ces voisinages locaux apporte ainsi de la cohérence spatiale. La comparaison entre les distributions d'intensité sur deux images successives permet de

décider si ces supports correspondent à des zones mobiles ou non, sans passer par la différence inter-images. Les seuils sont fixés à partir de niveaux de confiance sur le test de vraisemblance. Plus tard, des modèles markoviens, bien adaptés à la prise en compte d'un contexte spatial, ont été utilisés [Bouthemy 93, Aach 95]. Il est alors possible de ne prendre la décision qu'à partir de la vraisemblance de l'hypothèse d'absence de mouvement.

Les méthodes décrites dans cette section permettent de décider si un pixel est en mouvement ou non. En général, les résultats des algorithmes sont des cartes binaires de mouvement. Une étape d'isolation des objets doit alors être ajoutée pour obtenir une segmentation des différents objets en mouvement. Cela est le plus souvent fait par extraction des composantes connexes (en 4 ou 8 connexités) du masque binaire à l'aide d'algorithmes classiques peu coûteux.

1.1.2 Décision basée région

Décider si un pixel est en mouvement sur une région renforce la cohérence du masque de détection. Peu de méthodes de détection utilisent les régions, principalement parce qu'une décision erronée sur une région entière aura des conséquences plus importantes que la mauvaise classification d'un pixel. Parmi les approches qui travaillent sur les régions, on trouve essentiellement des techniques de segmentation de mouvement ou d'extraction de couches de mouvement (section 1.3).

Récemment, plusieurs méthodes se sont intéressées à des modèles *a contrario* pour détecter les zones de changement dans les images. Dans [Lisani 03] ou [Dibos 05] une information de changement est calculée à partir de différences d'orientation du gradient spatial d'intensité. Le modèle *a contrario* permet de détecter les fortes corrélations des différences d'orientation des gradients au niveau des zones de changements. Les régions qui sont soumises au test de détection sont des régions carrées de taille variable déplacées sur toute l'image. Une autre méthode, où la décision *a contrario* repose sur l'analyse des mesures locales de changement à partir des niveaux d'intensité de trois images successives, a été proposée dans [Veit 04]. Dans ces travaux, la décision est prise sur des régions fournies soit par un découpage en blocs, soit par une segmentation basée sur les niveaux d'intensité.

Détecter les pixels en mouvement peut également être vu comme la détection des pixels n'appartenant pas au mouvement de la caméra. Ceci est bien entendu valable que la caméra soit fixe ou mobile. On fait généralement l'hypothèse que le mouvement de la caméra correspond au mouvement dominant dans l'image. Ainsi, tous les pixels n'appartenant pas au mouvement dominant sont des pixels en mouvement. Des approches paramétriques visant à modéliser le mouvement de la caméra existent. Nous ne rentrons pas dans les détails ici puisque qu'une de ces méthodes [Odobez 97] fait l'objet d'une étude plus approfondie dans le chapitre suivant (chapitre 2).

1.2 Modélisation du fond

Une autre catégorie de méthodes, très populaire en détection de mouvement, repose sur la modélisation du fond. Plusieurs conditions sont nécessaires à l'utilisation de ces techniques. Tout d'abord, la caméra doit être maintenue fixe et les occultations de parties du fond par des objets en mouvement doivent rester temporellement minoritaires. De plus, il est préférable d'avoir, au début de la séquence, un certain nombre d'images sans objets en mouvement afin de pouvoir apprendre

correctement le fond. Ces techniques peuvent être prédictives ou non prédictives, ces dernières ayant fait l'objet d'un nombre plus important de recherche. La modélisation du fond, tout comme les méthodes basées sur la détection inter-images, permet de décider si un pixel est en mouvement ou non. Elle ne donne cependant pas directement le masque de chaque objet en mouvement et un module d'extraction de cible doit être ajouté.

1.2.1 Méthodes non prédictives

Les techniques non prédictives de modélisation du fond se décomposent en deux étapes principales : l'apprentissage du fond, le plus souvent en chaque pixel de l'image, et l'étiquetage binaire des pixels comme appartenant ou non au fond dans l'image courante. L'apprentissage du fond doit tenir compte de la fluctuation des intensités observées. L'étape d'étiquetage est similaire à la prise de décision décrite dans la section précédente.

Dans les premiers travaux de modélisation de fond, le fond était considéré statique. La distribution statistique en chaque pixel peut alors être représentée par une simple gaussienne [Wren 97, Kanade 98, Cavallaro 00, Huwer 00]. Quand le fond devient dynamique l'utilisation d'une gaussienne est insuffisante et la modélisation du fond par des mélanges de gaussiennes est préférable [Friedman 97, Grimson 98]. Nous détaillons brièvement ci-après la méthode de Grimson et Stauffer [Grimson 98], devenue une méthode de référence, qui nous servira à comparer nos résultats.

Méthode de Grimson et Stauffer [Grimson 98]
En chaque pixel de l'image, un mélange de κ (généralement compris entre 3 et 5) gaussiennes (en dimension 3 pour des images couleur ou 1 pour des images monochromes) est défini puis progressivement mis à jour en fonction des nouvelles valeurs observées en ce pixel au cours du temps. Ce mélange modélise d'une part l'apparence du fond et d'autre part l'apparence des éventuels objets couvrant ce pixel. Il est initialement appris par K-moyennes puis EM (Expectation-Maximization) sur les premières images. Un avantage de cette méthode est qu'elle n'a pas besoin d'images sans objets pour la phase initiale d'apprentissage. Une fois l'apprentissage effectué, le mélange de gaussiennes est mis à jour de la manière suivante. Si la nouvelle valeur observée est bien décrite par l'une des composantes du mélange (distance de Mahalanobis entre la valeur du pixel et la moyenne de la composante inférieure à 2.5 fois l'écart-type de la composante), la moyenne et la variance de cette composante sont mises à jour. La mise à jour de la composante concernée dépend d'un taux de rafraîchissement prédéfini et de la vraisemblance de l'observation. Les autres composantes voient leur poids réduit du taux de rafraîchissement. Le mélange final du pixel à l'instant courant est obtenu en normalisant le poids de chaque composante. Dans le cas où aucune composante ne correspond à la nouvelle observation, la composante de poids le plus faible est remplacée par une composante de poids faible et de forte variance, centrée sur la nouvelle observation.
Afin de détecter si un pixel appartient au fond ou à un objet, il faut déterminer, à chaque instant, quelles composantes du mélange peuvent être attribuées au fond et à l'objet. Toutes les composantes dont le poids excède un certain seuil sont considérées comme décrivant le fond. Ainsi, si une nouvelle observation est associée à l'une de ces composantes, le pixel correspondant appartient au fond.

Lorsque des changements apparaissent trop rapidement dans le fond, les variances des

gaussiennes le caractérisant deviennent trop importantes et toutes les méthodes décrites précédemment échouent. Ainsi, quand la fonction de densité est plus complexe et ne peut être modélisée de manière paramétrique, une approche non-paramétrique capable de manipuler des densités arbitraires est plus adaptée. Dans [Elgammal 00], des noyaux gaussiens sont utilisés pour modéliser la densité en chaque pixel à tout instant, connaissant les instants précédents récents. Nous décrivons brièvement cette méthode qui nous servira également à comparer nos résultats.

Méthode non paramétrique d'Elgammal *et al.* **[Elgammal 00]**
Soient $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{t-1}$, les vecteurs de couleur (vecteur de dimension 3 : $\mathbf{x_i} = (\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \mathbf{x}_{i,3})$) d'un pixel aux instants précédents l'instant t. La densité de probabilité p à l'instant t est estimée non-paramétriquement en utilisant le noyau K[1] :

$$p(\mathbf{x}_t) = \frac{1}{t-1} \sum_{i=1}^{t-1} K(\mathbf{x}_t - \mathbf{x}_i) \ . \tag{1.1}$$

En choisissant un noyau normal, et en supposant indépendants les trois canaux de couleur, la densité s'écrit :

$$p(\mathbf{x}_t) = \frac{1}{t-1} \sum_{i=1}^{t-1} \prod_{j=1}^{3} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp - \frac{(\mathbf{x}_{t,j} - \mathbf{x}_{i,j})^2}{2\sigma_j^2} \ , \tag{1.2}$$

σ_j^2 définissant la taille du noyau pour le $j^{\text{ième}}$ canal de couleur. Cette taille doit tenir compte du bruit et des fortes variations possibles de l'intensité lors de la présence d'un mouvement. Pour chaque canal de couleur, elle est égale à :

$$\sigma_j = \frac{1}{0.68\sqrt{2}(t-1)} \sum_{i=1}^{t-1} |\mathbf{x}_{i,j} - \mathbf{x}_{i+1,j}| \ . \tag{1.3}$$

Une fois la densité de probabilité construite, un pixel est considéré comme appartenant au fond si sa densité est inférieure à un seuil. Une étape de suppression des mauvaises détections, principalement dues à de faibles mouvements dans le fond, peut être ajoutée à l'algorithme.

La méthode d'Elgammal *et al.* reste encore trop limitée à de faibles mouvements, sur de petites zones de l'image. Dans les cas réels, la plupart des scènes dynamiques affiche un mouvement assez répétitif. Ce comportement peut être correctement modélisé en utilisant des informations de mouvement données par le flot optique. Dans [Mittal 04], Mittal et Paragios présentent un algorithme non paramétrique permettant de coupler les informations photométriques standards au flot optique. Cela permet de détecter les objets se différenciant du fond soit par leur mouvement soit par leur couleur ou intensité. L'estimation du flot optique étant impossible dans les zones où le gradient spatial d'intensité est nul, les auteurs emploient des noyaux de taille variable pour le calcul de densités de probabilités.

Nous n'avons jusqu'ici discuté que des méthodes locales (la décision est prise en chaque pixel) qui n'ont pas de cohérence spatiale et peuvent être sensibles au bruit. La cohérence spatiale est introduite dans [Sheikh 05] en ajoutant un noyau spatial au noyau temporel de [Elgammal 00].

[1] Pour plus de détails sur l'estimation à noyau, se référer au chapitre 3

Dans [Sheikh 05], l'image est représentée comme un champ de Markov qui apporte une dépendance inter-pixels. Il s'agit alors de trouver le maximum *a posteriori* (MAP) en minimisant une fonction d'énergie contenant un terme unaire basé sur les distributions de fond et d'objets et un terme binaire traduisant la cohérence spatiale[2].

Dans le cas où la caméra est mobile, les méthodes précédentes de modélisation de fond ne sont pas directement applicables. Elles nécessitent en effet le recalage préalable de l'image courante dans un repère fixe commun (en général celui de la première image) à toute la séquence [Ren 03]. Cela revient au problème générique de construction de mosaïques à partir de séquences d'images. La fabrication et la mise à jour d'un modèle de fond mosaïque pour la détection par soustraction de fond semble en fait confiné aux situations plus contraignantes où le champ de vue total couvert par la caméra est borné et connu [Sullivan 06].

1.2.2 Méthodes prédictives

Les méthodes prédictives utilisent un modèle dynamique pour prédire la valeur d'un pixel à l'aide des observations précédentes. Elles emploient généralement un filtre de Kalman **??** pour modéliser la dynamique de l'état d'un pixel [Karmann 90, Koller 94]. L'algorithme le plus connu dans cette catégorie est probablement l'algorithme "wallflower" [Toyama 99] qui fait appel à une version simplifiée du filtre de Kalman, appelé filtre Weiner, pour prédire la valeur d'un pixel au temps courant à partir de ses κ valeurs précédentes. Les pixels dont l'erreur de prédiction est élevée sont étiquetés comme des pixels en mouvement. Des méthodes plus récentes utilisent des modèles plus complexes basés sur les textures dynamiques [Doretto 03, Zhong 03]. Les textures dynamiques permettent de représenter une image à l'aide d'une combinaison linéaire d'images de base. Chaque paramètre de l'image est projeté comme un point dans un espace de caractéristiques. Un modèle autorégressif prédit alors les paramètres et donc l'apparence des images suivantes. Les pixels éloignés de la prédiction sont considérés comme indépendants du modèle et sont suivis séparément par un filtre de Kalman.

1.3 Extraction de couches de mouvement

Dans [Ullman 79], l'interprétation du mouvement visuel est définie comme la tâche consistant à décrire l'environnement en terme d'objets (en considérant leur forme tridimensionnelle et leur mouvement dans l'espace) à partir de l'image produite par leur projection sur la rétine. L'objectif des méthodes décrites ci-après est de fournir une description d'une scène en terme d'objets et de leur mouvement dans le plan image ou, autrement dit, en terme de couches de mouvement.

La segmentation de mouvement peut donc être vue comme une décomposition de la scène en différentes couches, chacune étant associée à un modèle de mouvement. Dans [Dimitrova 95, Sahouria 97] un champ de mouvement (par exemple le flot optique) est segmenté pour mettre en évidence des mouvements indépendants. Ces travaux, basés sur l'estimation d'un champ dense de mouvement, restent pour l'instant minoritaires. En effet, la plupart des méthodes d'extraction de couches de mouvement [Ayer 95, Darrel 91, Jepson 93, Wang 94, Weiss 97] repose sur le calcul d'un certain nombre de modèles paramétriques et supposent que le mouvement 2D d'un objet

[2]Pour plus de détails de la minimisation d'énergie dans un champ de Markov, se référer au chapitre 6

3D projeté dans l'image suit un modèle paramétrique. Cette hypothèse est notamment valide si l'objet 3D est planaire. Le nombre de couches peut être estimé automatiquement en utilisant une approche MDL ("Minimum Descritption Length") [Ayer 95, Darrel 91]. L'extraction des couches est alors obtenue via un critère de maximisation de la vraisemblance et une optimisation de l'algorithme d'espérance-maximisation (EM).

Au lieu de construire un ensemble de modèles de mouvement correspondant à la vidéo, un autre type d'approche consiste à adapter un polynôme sur toute l'image. Une factorisation ou une dérivation de ce polynôme permet d'obtenir les paramètres de chaque modèle de mouvement 2D. Cette technique est appelée "multibody factorization" [Vidal 04]. Le polynôme est le résultat d'une résolution par une méthode de moindres carrés d'un système linéaire reliant les mesures aux modèles de mouvement. L'approche a été étendue au cas de scènes dynamiques dans [Vidal 05].

Récemment, d'autres méthodes utilisant des techniques de croissance de régions de même mouvement à partir de graines ont été mises en place. Par exemple, dans [Xiao 05], les graines sont déterminées en mettant en correspondance deux images. Les supports réels des couches sont alors obtenus en utilisant un champ de Markov (MRF) optimisé par un algorithme de coupe minimale/flot maximal dans un graphe[3]. Cette méthode est principalement dédiée à la détection d'occultations. La croissance de région est aussi utilisée dans [Pundlik 06], où des points caractéristiques sont détectés et suivis avec l'algorithme Kanade-Lucas-Tomasi (KLT) [Tomasi 91]. Une méthode de croissance de régions est appliquée pour grouper ces points : un point graine est d'abord sélectionné et forme un groupe, puis les points voisins lui sont successivement ajoutés si leur mouvement correspond au modèle affine du groupe. Les voisins sont obtenus par une triangularisation de Delaunay et le modèle affine de mouvement du groupe est mis à jour dès qu'un point est ajouté. Cette procédure est réitérée jusqu'à ce que tous les points soient classifiés.

Une combinaison entre la modélisation du fond et une approche par couche a été introduite dans [Kim 05]. La modélisation du fond permet de détecter successivement différents objets en mouvement au cours du temps. Chacun des objets détectés est associé à une couche, conservée même lors d'éventuels arrêts des objets. Le nouveau modèle de fond est un mélange du modèle de chaque objet déjà détecté ("short-term backgrounds") et du modèle du fond réel détecté dès le début de la séquence ("long-term backgrounds").

Pour toutes les méthodes décrites dans cette section, la segmentation de mouvement finale est obtenue en associant chaque pixel au modèle de mouvement le plus proche. L'avantage de ces approches est l'obtention directe des masques de chaque objet en mouvement (à condition bien sûr que tous les objets aient des mouvements distincts).

1.4 Détection de mouvement basée sur la cohérence

Une dernière approche consiste à définir un objet mobile comme une région ayant un mouvement cohérent. Dans ce contexte, un mouvement cohérent peut être défini comme un mouvement ayant de grande chance de provenir d'une "cible" classique (*e.g.* personne, véhicule) [Wildes 98]. Cette définition a été utilisée pour la segmentation de mouvement en ajoutant

[3] La théorie des coupes minimales dans un graphe est présentée dans le chapitre 6

l'hypothèse suivante : un objet avec un mouvement cohérent se déplace dans une direction approximativement constante pendant une courte période (en pratique quelques images). Ainsi, dans [Wixson 00], des mesures de flot optique cohérentes en direction sont accumulées pendant quelques pas de temps. Une estimation du déplacement de chaque pixel à travers une séquence d'images est ainsi obtenue. Il est alors possible de distinguer les objets mobiles, qui se déplacent avec une direction constante, des mouvements parasites. Cette méthode a été légèrement modifiée dans [Tian 05] pour fonctionner en temps réel. Dans ce dernier papier, l'accumulation des vecteurs de mouvement est faite sur 10 images consécutives.

Dans le cas d'applications particulières, on peut exiger une cohérence très forte sur les mesures de mouvement. Des motifs de mouvement correspondant à des comportement anormaux peuvent être détectés à partir de cartes de direction de mouvement [Gryn 05]. La procédure de prise de décision repose sur un seuillage de la distance au motif. Une autre manière de discriminer des mouvements pertinents consiste à s'intéresser à la classe des mouvements périodiques, en utilisant des points d'intérêt spatio-temporels [Laptev 05]. Cette méthode permet notamment d'extraire des piétons dans des situations où le fond est complexe. Citons enfin [Veit 05b] qui détecte des objets de mouvement localement rectiligne uniforme avec une méthode de clustering *a contrario*.

La combinaison d'une méthode de modélisation du fond et d'une méthode basée sur la cohérence a été proposée par [Zhu 05]. Une particularité de ce papier est de ne modéliser le fond que sur des points caractéristiques détectés avec le détecteur de coins Harris [Harris 88] et caractérisés par des descripteurs SIFT (Scale Invariant Feature Transform) [Lowe 99]. Un modèle de fond est défini ou mis à jour en chaque point détecté en utilisant une modélisation de la fréquence de détection du coin et de la moyenne temporelle du descripteur SIFT. Les coins n'appartenant pas au fond sont ensuite suivis en utilisant les KLT, et les objets mobiles sont finalement obtenus en groupant tous les coins suivis ayant des trajectoires similaires et cohérentes.

1.5 Conclusion et approche proposée

Cette dernière section a pour but de définir l'approche que nous proposons pour détecter des objets en mouvement. Une brève description de la difficulté des vidéos traitées dans la première partie de cette thèse justifie l'introduction d'une nouvelle approche. En effet, les méthodes présentées précédemment échouent la plupart du temps pour des vidéos aussi complexes.

1.5.1 Particularité des séquences étudiées

Cette thèse est principalement dédiée à l'analyse et à l'étude de séquences de conducteur dans un but de surveillance. Nous avons à notre disposition plusieurs séquences de conducteur (la plupart du temps filmé de profil), dans lesquelles sont visibles le conducteur, l'habitacle de la voiture et le paysage derrière les fenêtres. La difficulté principale de ces séquences vient du fond très dynamique visible derrière les différentes fenêtres de la voiture. On ne souhaite détecter et suivre que le conducteur ou tout du moins les zones en mouvement de son corps. Le fond est ainsi composé de l'habitacle mais aussi de toutes les régions à l'extérieur de la voiture. Les changements dans le fond peuvent être brutaux, avec de forts changements d'illumination qui se répercutent sur le conducteur dans l'habitacle. De plus, les fenêtres occupent parfois une grande partie des

images. Une autre difficulté vient du fait que le contraste (couleur ou luminance) entre les objets intéressants à détecter (visage, mains) et le fond peut être faible. Par ailleurs, le mouvement de ces objets peut être similaire à celui du fond pendant quelques instants successifs. Enfin, notons qu'une seule caméra, éventuellement mobile, est utilisée et qu'aucune information de profondeur n'est disponible.

Afin d'illustrer les difficultés décrites ci-dessus nous présentons quelques images d'une séquence de conducteur dans la figure 1.1. Le contraste entre la chemise et la barrière de sécurité est faible. Les trois premières images montrent que le mouvement de la main peut être proche de celui présent derrière la fenêtre. Enfin, les images suivantes présentent des changements brutaux tant au niveau de l'apparence que de l'illumination derrière la fenêtre.

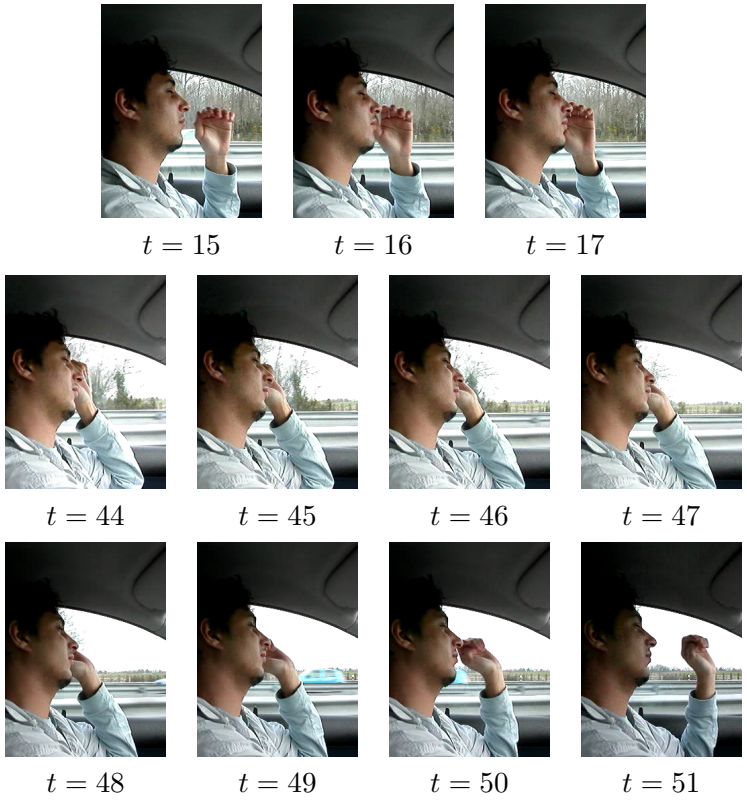


FIG. 1.1: Quelques images d'une séquence de conducteur.

Dans ce document, nous ne nous focalisons pas seulement sur des séquences de conducteur. Aussi aucune information sur le contexte ne sera utilisée.

1.5.2 Analyse des méthodes de l'état de l'art

Dans cette sous-section nous énonçons les aspects positifs et négatifs des quatre types de méthodes de détection et segmentation de mouvement décrits dans l'état de l'art. Nous montrons ainsi qu'aucune des méthodes précédemment citées ne peut être appliquée telle quelle aux séquences considérées.

Détection basée sur la différence inter-images

Les méthodes basées sur la différence entre deux ou trois images successives sont généralement rapides et faciles à mettre en place. Elles s'adaptent relativement bien aux changements brutaux dans le fond et moins bien aux changements de luminosité. La plupart des algorithmes ne se basent que sur l'intensité mais cela ne permet pas la détection d'objets ayant une intensité proche de celle des pixels du fond. D'autres méthodes s'appuient uniquement sur le flot optique ce qui ne permet pas de différencier les objets ayant un mouvement proche de celui du fond. Ainsi, déterminer la probabilité que chaque pixel appartienne au fond en n'utilisant que l'intensité ou le mouvement sur deux ou trois images successives ne permettra pas de détecter uniquement les objets mobiles intéressants. Pour finir, ces méthodes ne retournent qu'un masque binaire des pixels en mouvement. L'extraction des masques de chaque objet peut se faire en séparant les composantes connexes. Cependant, une seule composante connexe peut être associée à plusieurs objets, ce qui pose problème aux algorithmes d'identification ou de suivi reposant sur cet étiquetage. Ce problème est fréquemment rencontré lors de la détection d'ombres. La détection indésirable des ombres portées ne conduit pas nécessairement à la détection d'un objet inexistant mais déforme de façon préjudiciable les masques de certains objets.

Pour conclure, ces méthodes ne peuvent permettre à elles seules de détecter les objets mobiles dans les séquences que nous considérons. Néanmoins, ces techniques peuvent donner une bonne indication des pixels en mouvement et peuvent donc être utilisées comme pré-traitement pour améliorer et accélérer les autres techniques de détection ou de segmentation de mouvement.

Modélisation du fond

La modélisation du fond ou du mouvement en chaque pixel présente l'avantage de permettre une détection plus précise des objets en mouvement mais nécessite l'apprentissage de modèles sur plusieurs images successives. Les changements brutaux limitent donc l'utilisation d'une telle approche. La mise à jour continue des modèles de fond est indispensable pour prendre en compte des changements graduels mais elle conduit généralement à l'absorption d'objets restant immobiles un certain moment dans le modèle de fond. L'influence de l'immobilité d'un objet sur le modèle de fond dépend de la vitesse de rafraîchissement du modèle. Comme les méthodes de détection de changement basées sur la différence inter-images, ce type de méthodes ne retourne qu'une carte binaire des pixels en mouvement. L'extraction des composantes connexes doit être ajoutée, ce qui induit certains problèmes de différenciation d'objets comme décrit plus haut. Les méthodes de modélisation de fond sont en fait généralement dédiées à des séquences particulières (où la caméra est fixe). En effet, comme mentionné précédemment, la caméra doit avoir été maintenue fixe pendant toute la durée d'acquisition de la séquence. De plus il est généralement nécessaire d'avoir à sa disposition, au début de la séquence, un nombre assez important d'images sans objets en mouvement pour pouvoir correctement apprendre le fond.

Pour conclure, les méthodes visant à modéliser le fond se sont pas adaptées aux séquences considérées. Néanmoins nous retiendrons l'utilisation du flot optique et de la couleur introduite par Mittal et Paragios dans [Mittal 04]. En effet, étant donné l'éventuel faible contraste (tant au niveau de la couleur que du mouvement) entre le objets et le fond, il paraît indispensable d'utiliser ces deux espaces de caractéristiques pour pouvoir réaliser une bonne détection.

Extraction de couches de mouvement

L'extraction de couches de mouvement permet de détecter directement les masques de chaque objet. L'utilisation de ce type de méthodes suppose que la scène peut être décomposée en plusieurs régions bien distinctes toutes caractérisées par un mouvement bien défini. L'avantage de ces méthodes est qu'elles n'ont ni besoin d'une caméra fixe ni des résultats aux instants précédents. Elles se basent sur une estimation préalable du mouvement dans l'image. Appliquées à nos séquences, ces méthodes échoueront la plupart du temps car il est difficile de trouver un modèle de mouvement caractérisant les zones dynamiques du fond. En effet, les mouvements du fond peuvent n'avoir aucune cohérence et/ou une trop grande amplitude. Comme seul le mouvement est utilisé, ces techniques ne permettent pas de séparer les objets du fond lorsqu'ils ont des mouvements proches.

Nous retiendrons de ces méthodes l'idée de définir un objet comme une couche ayant des caractéristiques similaires (ou cohérentes) détectée par une méthode de clustering.

Détection de mouvement basée sur la cohérence

Une région cohérente peut être directement vue comme une couche. La différence et l'avantage ici est que seuls les objets cohérents sont détectés. Cela permet l'application directe d'autres traitements tels que l'identification ou le suivi. Cependant, la plupart des méthodes basées sur la cohérence n'utilisent comme information que la direction du mouvement. Celle-ci n'a aucun sens pour des mouvements de faibles amplitudes et l'utilisation du seul mouvement comme caractéristique n'est pas satisfaisant. De plus, dans nos séquences il arrive régulièrement qu'un objet ne soit en mouvement que pendant une durée trop courte pour permettre la construction de trajectoire.

La restriction de la détection à quelques points caractéristiques [Zhu 05] ou [Pundlik 06] permet une réduction non négligeable du temps de calcul mais ne donne pas la segmentation complète des objets mobiles. Cette dernière peut être assez simplement obtenue en ajoutant une étape supplémentaire de segmentation de type croissance de région.

1.5.3 Présentation de l'approche proposée

Il ressort de l'analyse précédente plusieurs idées intéressantes pour la détection des objets en mouvement. La première partie du document présente un algorithme s'appuyant sur les avantages de chaque catégorie de méthodes. Nous ordonnons les propriétés les plus intéressantes des méthodes décrites dans ce chapitre de la façon suivante. Tout d'abord nous avons vu qu'il est possible de ne réaliser la détection que sur un nombre restreint de pixels de l'image. Ces pixels peuvent être choisis parmi l'ensemble des pixels en mouvement, ces derniers étant le résultat d'une première détection basée sur la différence inter-images. Il est indispensable d'utiliser le mouvement et la couleur (ou l'intensité) pour décrire les points choisis. Une classification de ces points en couches ou régions cohérentes peut ensuite être obtenue par une méthode de clustering prenant en compte chaque caractéristique du point. Enfin, une étape de segmentation peut être ajoutée afin d'obtenir le masque complet de chaque objet mobile. L'algorithme de détection des objets en mouvement proposé dans cette partie respecte cette analyse. L'algorithme 1 résume les différentes étapes de l'algorithme et justifie le plan de cette première partie.

Algorithme 1 Principe de l'algorithme de détection des objets en mouvement et plan de la partie

1. Sélection et description de points en mouvement (chapitre 2)
   (a) Extraction d'une carte binaire de détection de changements
   (b) Sélection d'un nombre restreint de pixels en mouvement
   (c) Description des points sélectionnés
2. Partitionnement des points sélectionnés en régions cohérentes (chapitres 3 et 4)
3. Segmentation finale de tous les objets mobiles (chapitre 5). Cette étape utilise l'ensemble des pixels de l'image.

Chapitre 2

Approche locale : sélection et description de points

La première étape de l'algorithme de détection des objets mobiles (algorithme 1) présenté dans cette partie concerne la sélection et la description d'un ensemble restreint de pixels en mouvement dans l'image. L'utilisation d'un nombre restreint de points permet une réduction considérable du temps de calcul de notre algorithme. La sélection des points en mouvement est décomposée en deux étapes. Nous définissons d'abord ce qu'est un pixel en mouvement (section 2.1), puis nous présentons la sélection d'une grille de points uniformément répartie sur l'ensemble des pixels (section 2.2). La description des points sélectionnés doit être adaptée à l'application visée, c'est-à-dire à la détection d'objets mobiles. Nous utiliserons pour cela des informations sur la position, le mouvement et la photométrie. Le descripteur utilisé est présenté dans la section 2.3.

2.1 Détection des pixels en mouvement

La première étape de l'algorithme de détection des objets en mouvement consiste à extraire une carte binaire de détection de changements. Cette étape a deux buts principaux. Tout d'abord, l'objectif de cette partie est uniquement la détection des objets mobiles, c'est-à-dire des régions cohérentes composées de pixels en mouvement. Se concentrer uniquement sur les pixels en mouvement facilitera l'extraction de ces objets mobiles. De plus, l'extraction de la carte binaire permet de restreindre aux seuls pixels en mouvement l'étape suivante de l'algorithme, ce qui entraîne un gain de coût de calcul.

La caméra pouvant être mobile, tous les pixels peuvent bouger. On considère alors que les pixels fixes sont ceux suivant le mouvement de la caméra. Le mouvement de la caméra est souvent assimilé au mouvement dominant de l'image. Parmi les différents champs de mouvement généralement présents dans l'image, le mouvement dominant est celui qui concerne le nombre le plus important de pixels. Nous avons effectué un choix classique pour représenter le mouvement apparent induit par le déplacement de la caméra. Il consiste à utiliser un modèle de mouvement 2D paramétrique. Il s'agit en fait d'un modèle polynômial qui exprime le déplacement en chaque point de l'image comme une fonction polynômiale de la position du point. Généralement, des polynômes de degré 1 ou 2, correspondant respectivement au modèle affine ou au modèle quadra-

tique, sont employés. Nous avons utilisé le modèle affine complet à six paramètres défini par :

$$\mathbf{w}_\theta(s) = \begin{pmatrix} a_1 \\ a_4 \end{pmatrix} + \begin{pmatrix} a_2 & a_3 \\ a_5 & a_6 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} \quad , \tag{2.1}$$

où $\mathbf{w}_\theta(s)$ représente le vecteur de vitesse au pixel $s = (x, y)$, relativement au modèle paramétrique de paramètres $\theta = (a_1, a_2, a_3, a_4, a_5, a_6)$. Ce modèle de mouvement relativement simple, estimé sur l'ensemble du support de l'image, permet d'appréhender différents mouvements de la caméra tels que les panoramiques, les zooms, les travellings. Il s'adapte également aux plans statiques.

Des techniques robustes et multirésolutions associées à des algorithmes itératifs [Sawhney 96,Black 96,Odobez 95] permettent une estimation fiable de ce type de modèle. Ces approches présentent l'avantage d'un faible temps de calcul. En effet, l'algorithme que nous utilisons[1] [Odobez 95] estime les paramètres en temps réel (pour des images de 256×256 pixels).

Le vecteur de vitesse s'écrit :

$$\mathbf{w}_\theta(s) = \mathbf{B}(s)\theta \tag{2.2}$$

où $\mathbf{B}(s)$ est la matrice définie en chaque pixel $s = (x, y)$ par :

$$\mathbf{B}(s) = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \quad . \tag{2.3}$$

L'estimation du vecteur de paramètres θ est formulée comme un problème de minimisation :

$$\hat{\theta} = \arg\min_\theta \sum_{s \in \mathcal{R}} \varrho(DID(s, \theta)) \tag{2.4}$$

avec

$$DID(s, \theta) = \mathbf{z}_{t+1}^{(G)}(s + \mathbf{w}_{\theta,t}(s)) - \mathbf{z}_t^{(G)}(s) \tag{2.5}$$

la différence inter-images déplacée (DID) et $\mathbf{z}_t^{(G)}(s)$ la valeur au pixel s de l'intensité de l'image au temps t. Le vecteur de vitesse $\mathbf{w}_\theta(s)$ n'est pas forcément entier. Le calcul de la DID nécessite donc l'interpolation du point $s + \mathbf{w}_{\theta,t}(s)$ au pixel (entier) le plus proche. Dans l'équation (2.4), $\mathcal{R}$ représente le support sur lequel est réalisée l'estimation du modèle, c'est-à-dire le support de l'image. La fonction ϱ est une fonction de coût non quadratique permettant de ne pas perturber l'estimation du mouvement dominant par des points appartenant à des régions de mouvements secondaires (objets en mouvement dans la scène). Il s'agit en fait d'un M-estimateur car la minimisation de la fonction correspond à l'estimation du maximum de vraisemblance si la fonction ϱ est interprétée comme l'opposé de la log-vraisemblance associée au modèle. La minimisation (équation (2.4)) est réalisée de manière incrémentale par une méthode de Gauss-Newton dans un schéma multirésolution. Cela permet d'appréhender des mouvements de grande amplitude. L'estimé initial θ^0 est d'abord supposé nul. Puis, une succession d'incréments $\Delta\theta^k$ est évaluée et cumulée en exploitant une approximation au premier ordre $DID'(s, \theta)$ de l'erreur résiduelle relativement à l'estimé courant θ^k :

$$\Delta\theta^k = \arg\min_{\Delta\theta} \sum_{s \in \mathcal{R}} \varrho(DID'(s, \theta^k)) \quad . \tag{2.6}$$

[1] Le logiciel correspondant Motion-2D est disponible et téléchargeable sur la page web : *http ://www.irisa.fr/Vista/Motion2D*

L'approximation du premier ordre est donnée par :

$$DID'(s) = \mathbf{z}_{t+1}^{(G)}(s + \mathbf{B}(s)\theta^k) - \mathbf{z}_t^{(G)}(s) + \nabla\mathbf{z}_{t+1}^{(G)}(s + \mathbf{B}(s)\theta^k) \cdot \mathbf{B}(s)\Delta\theta \ , \qquad (2.7)$$

où $\nabla\mathbf{z}_{t+1}^{(G)}$ est le gradient spatial de la fonction intensité $\mathbf{z}_{t+1}^{(G)}$. La procédure de minimisation mise en place s'inscrit dans une stratégie pyramidale et est réalisée par une technique de moindres carrés pondérés itérés à chaque niveau de la pyramide. Il s'agit de résoudre un problème de minimisation en introduisant des poids $W_t(s)$, en chaque pixel s, spécifiant l'influence de chaque point dans l'énergie globale. La nouvelle formulation du critère (2.6) est alors :

$$\Delta\theta^k = \arg\min_{\Delta\theta} \sum_{s\in\mathcal{R}} \frac{1}{2} W_t(s) DID'(s, \theta^k)^2 \ , \qquad (2.8)$$

avec

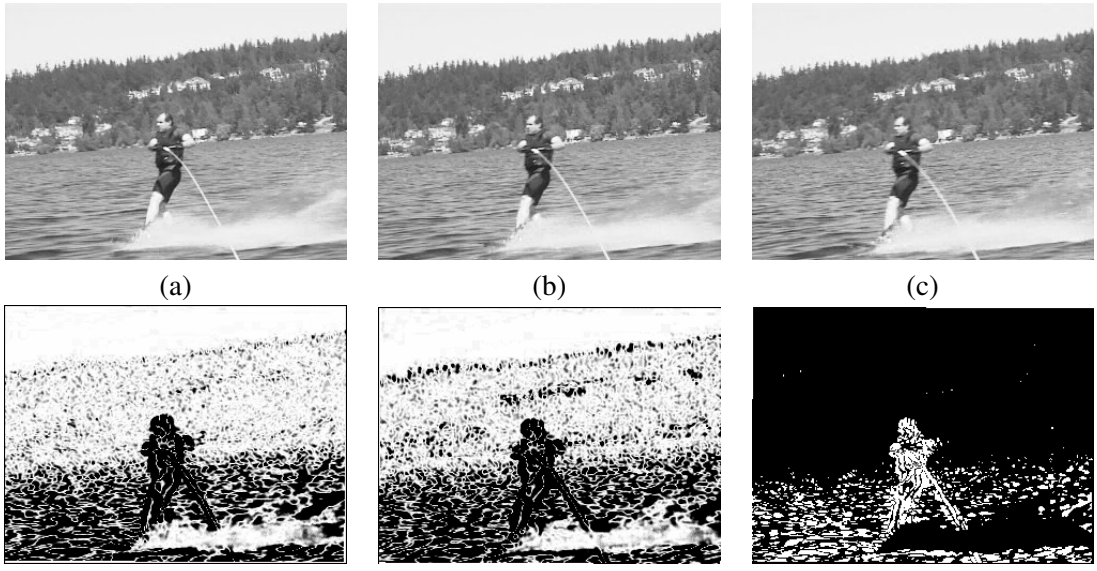
$$W_t(s) = \frac{\varrho'(DID'(s, \theta^k))}{DID'(s, \theta^k)} \ . \qquad (2.9)$$

La carte des poids W_t ($W_t(s) \in [0, 1]$) ainsi obtenue évalue la conformité des données au modèle estimé. Si un pixel s participe à l'estimation robuste du mouvement de la caméra alors $W_t(s)$ est proche de 1 et dans le cas contraire, $W_t(s)$ est proche de 0. Le fait d'utiliser la différence inter-images déplacée conduit à deux problèmes. Tout d'abord, la simple différence entre deux images successives (après compensation) aux instants t et $t + 1$ produit des valeurs élevées à la position d'un objet mobile à l'instant t (désoccultation du fond par l'objet mobile) et à la position de l'objet mobile à l'instant $t + 1$ (occultation du fond par l'objet mobile). Ceci empêche la localisation précise des objets en mouvement. Une façon de remédier à ce problème est de ne pas utiliser uniquement la DID entre t et $t + 1$ mais aussi celle entre t et $t - 1$. En notant $\mathbf{w}_{\theta,t}(s)$ le vecteur de vitesse au pixel $s = (x, y)$ de l'image à l'instant t, nous définissons la carte des pixels en mouvement M_t par :

$$M_t(s) = \begin{cases} 1 \ \text{si} \ W_t(s + \mathbf{w}_{\theta,t-1}(s))) + W_{t-1}(s) \ = \ 0 \\ 0 \ \text{sinon} \ . \end{cases} \qquad (2.10)$$

Le choix des pixels ayant un poids nul peut paraître drastique. Cependant il évite l'utilisation de fausses alarmes, c'est-à-dire de pixels détectés comme mobiles alors qu'ils sont en réalité immobiles. De plus, les différentes expérimentations réalisées sur nos séquences ont montré que ce choix n'entraîne que très rarement une non détection des objets mobiles *in fine*. Enfin, la segmentation des objets mobiles (dernière étape de l'algorithme), c'est-à-dire l'extraction des masques complets de chaque objet, utilisera l'ensemble des pixels de l'image et ne sera donc pas limitée par ce choix drastique. Les figures 2.1 et 2.2 montrent des résultats de la détection des pixels en mouvement. Le premier résultat illustre la difficulté de la tâche consistant à ne détecter que le skieur.

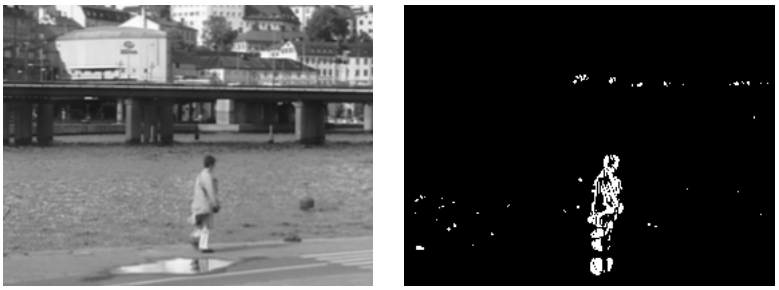
Le deuxième inconvénient de l'utilisation de la DID est lié à l'impossibilité de calculer une information de mouvement sur une zone d'intensité homogène. L'intérieur d'objets mobiles d'intensité uniforme se déplaçant relativement lentement se retrouve souvent étiqueté comme appartenant au mouvement de la caméra. En effet, les pixels concernés apparaissent localement comme des zones immobiles. Un exemple de ce problème est présenté à la figure 2.2 dans laquelle une partie du buste du piéton est apparenté au mouvement dominant (pixels noirs). Nous n'attachons pas beaucoup d'importance à ce problème pour l'instant. En effet, il sera assez bien corrigé par la dernière phase de l'algorithme de détection qui consiste à segmenter les objets mobiles.



(a) (b) (c)

(d) (e) (f)

FIG. 2.1: Carte de détection des pixels en mouvement sur la séquence de ski nautique. (a)-(c) Images 107, 108 et 109 en niveau de gris. (d) Carte des poids W_{t-1} calculée entre t et $t-1$. (e) Carte des poids W_t calculée entre t et $t+1$. (f) Carte M_t de détection des pixels en mouvement. Les pixels blancs sont les pixels en mouvement.



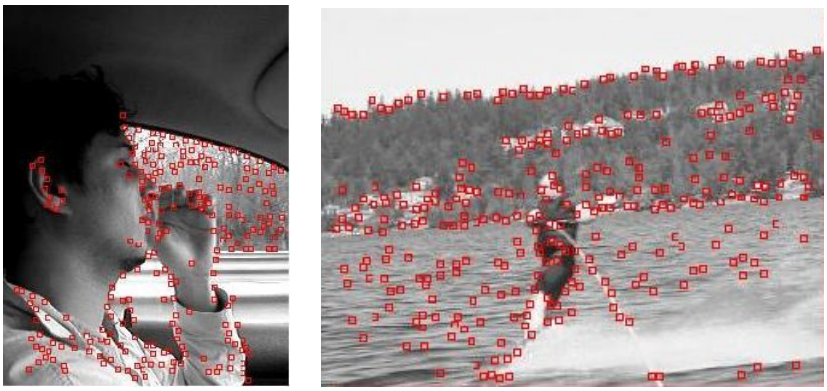
(a) (b)

FIG. 2.2: Carte de détection des pixels en mouvement sur la séquence montrant une personne marchant au bord d'un fleuve.(a) Images en niveau de gris. (b) Carte M_t de détection des pixels en mouvement. Les pixels blancs sont les pixels en mouvement.

2.2 Sélection d'une grille de pixels en mouvement

Dans le chapitre précédent nous avons énoncé plusieurs méthodes de détection et de segmentation de mouvement restreignant la détection à un nombre limité de points. La diminution de l'espace nous permet d'une part de réduire le temps de calcul et d'autre part de ne considérer que les pixels en mouvement. Les approches se limitant à un nombre restreint de points utilisent généralement des points caractéristiques, c'est-à-dire des coins. Le détecteur de coins le plus répandu dans la littérature est le détecteur de Harris [Harris 88]. Dans [Zhu 05], l'hypothèse justifiant l'uti-

lisation des coins est que le nombre de coins appartenant aux objets intéressants est plus important que le nombre de coins appartenant au fond. Nous avons voulu vérifier cette hypothèse sur nos séquences. La figure 2.3 montre le résultat obtenu sur une image de deux séquences différentes. Il est facile de constater que la justification basée sur le nombre de coins n'est pas suffisante. De plus, les coins ont le désavantage de ne pas être robustes aux changements de point de vue :



(a) (b)

FIG. 2.3: Résultat du détecteur de coin Harris sur a) l'image 16 de la séquence de conducteur b) l'image 108 de la séquence ski nautique.

si le fond varie et que les parallaxes changent, le nombre de coins et leurs voisins peuvent être considérablement différents d'une image à l'autre. Ainsi nous n'utiliserons pas les coins.

Comme aucun *a priori* sur la forme ou la texture des objets n'est supposé connu ici, il est préférable de sélectionner des points n'ayant aucune propriété particulière. La seule contrainte que nous nous fixons est que ces points doivent être répartis uniformément sur toute l'image, ou tout du moins sur tous les pixels en mouvement de l'image. La solution que nous proposons est alors d'utiliser une grille de points uniformément répartie sur l'image. Cela peut être vu comme un sous-échantillonnage. Nous avons arbitrairement choisi d'utiliser le même nombre de points pour les axes horizontaux et verticaux de l'image. la grille de points est alors définie par :

$$G = \left\{ s = \left(\frac{k.w}{N_G}, \frac{l.h}{N_G} \right), k = 0 \cdots N_G \ , \ l = 0 \cdots N_G \mid M_t(s) = 1 \right\} \ , \tag{2.11}$$

où w et h sont la largeur et la hauteur de l'image et N_G est un paramètre représentant la taille de la grille (avant suppression des pixels appartenant au mouvement de la caméra). Ce dernier paramètre est important. En effet, il permet de contrôler le compromis entre le coût de calcul et la précision. Les étapes de clustering et de segmentation utilisées ensuite dans notre algorithme de détection des objets mobiles peuvent devenir très coûteuses si un nombre trop important de points est sélectionné. Ainsi nous avons décidé de fixer approximativement le nombre de points en mouvement sélectionnés, noté n_G. Tous les résultats présentés dans ce document ont été obtenus avec $n_G = 500$. Il est intéressant de constater que le paramètre réglant la taille de la grille N_G peut être relié au nombre de pixels en mouvement dans l'image, noté $m = \sum_{s \in \mathcal{P}} M_t(s)$. Nous calculons finalement le paramètre N_G avec :

$$N_G = \lfloor \sqrt{whn_G/m} \rfloor \ , \tag{2.12}$$

où $\lfloor \bullet \rfloor$ représente la partie entière de $\bullet$.

Des exemples de grille après suppression des pixels en mouvement sont présentés dans les figures 2.4(c) et 2.5(c). Ils mettent bien en évidence la diminution du volume des données pour la recherche des objets en mouvement. L'utilisation de cet ensemble restreint ne permet pas de détecter directement les objets segmentés mais seulement des groupes de points correspondant aux objets. Pour plus de clarté, les "objets" détectés dans cet ensemble restreint seront appelés **clusters en mouvement**. Ces derniers sont obtenus après regroupement des points sélectionnés en couches ou régions cohérentes. Un critère de division doit alors être défini. Il passe par la définition d'un vecteur de caractéristiques, aussi appelé descripteur, en chaque point.

2.3 Description des points sélectionnés

Afin de détecter les clusters en mouvement, nous devons définir quelles caractéristiques seront utilisées par l'algorithme de clustering. Le clustering permet de partitionner un ensemble de données représentées par des vecteurs de caractéristiques en utilisant uniquement la proximité des vecteurs dans l'espace. Les résultats d'un algorithme de clustering dépendent du descripteur défini en chaque point. Il est donc important de bien choisir chaque composante du vecteur pour l'application visée. Nous cherchons dans cette partie à détecter des clusters cohérents pour la couleur et le mouvement. En effet, chaque cluster représente un objet en mouvement défini ainsi :

Définition 1 (Objet en mouvement)
Un objet en mouvement est une région fermée de l'image ayant une couleur et un mouvement quasi constants.

Un cluster en mouvement est défini similairement :

Définition 2 (Cluster en mouvement)
Un cluster en mouvement est un groupe de points de couleur et mouvement quasi constants. Il correspond à un objet en mouvement.

Les algorithmes de clustering peuvent devenir très coûteux si la dimension des données est trop grande. Il faut donc éviter de rajouter des caractéristiques peu discriminantes. La définition donnée d'un objet nous a amené à utiliser trois caractéristiques principales : la position, le mouvement et la photométrie. La couleur et le mouvement ont déjà été utilisés ensemble pour la détection de mouvement [Mittal 04]. La position est ajoutée afin d'éviter que des points éloignés dans l'image ne se retrouvent groupés dans le même cluster. Nous noterons le descripteur en chaque point d'index i dans la grille de la façon suivante :

$$\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)}), \tag{2.13}$$

où $\mathbf{x}_1^{(i)} = (x, y)$ représente la position, $\mathbf{x}_2^{(i)}$ le mouvement et $\mathbf{x}_3^{(i)}$ les caractéristiques photométriques.

2.3.1 Caractéristiques de mouvement

Avant de pouvoir être intégré dans le vecteur de caractéristiques, le mouvement doit être estimé. L'estimation du mouvement est un problème fondamental pour l'analyse de séquences d'images. Il consiste à mesurer la projection 2D dans le plan image d'un mouvement réel 3D, dû

à la fois au mouvement des objets dans la scène et aux déplacements de la caméra. Le mouvement 2D, appelé flot optique, est une variable cachée qui n'est accessible que par l'analyse des variations temporelles de la séquence d'images. Il peut être estimé globalement (champ dense) ou localement (généralement sur des éléments caractéristiques comme les contours ou les coins). Notre descripteur ne devant être établi que pour quelques points sélectionnés de l'image, il n'est pas nécessaire ici de calculer un champ dense de mouvement. Après un bref état de l'art sur les méthodes d'estimation de mouvement, nous détaillerons la méthode utilisée ici. Les vecteurs de mouvement obtenus n'étant pas toujours de bonne qualité, nous terminons cette sous-section en décrivant un test statistique permettant de valider les mesures de mouvement en chaque point de la grille.

2.3.1.1 Introduction aux mesures de mouvement

Afin d'utiliser les notations usuelles, dans toute cette sous-section I dénotera la fonction d'intensité et $I(x, y, t)$ la valeur de l'intensité au point de coordonnées (x, y) au temps t. Les méthodes d'estimation de mouvement reposent sur l'hypothèse de conservation de la luminance :

$$I(x + d_x, y + d_y, t + 1) \simeq I(x, y, t) \tag{2.14}$$

où d_x et d_y sont les vecteurs de déplacement horizontaux et verticaux entre les instants t et $t + 1$. Cette hypothèse n'est pas respectée dans le cas d'occultations, de transparences, de réflexions spéculaires et plus généralement de variations brutales de luminosité. L'invariance temporelle entre deux instants peut être caractérisée par une formulation différentielle linéaire, aussi appelée équation de contrainte du mouvement apparent (ECMA) :

$$\frac{dI}{dt}(x, y) = \frac{\partial I}{\partial t}(x, y) + \nabla I(x, y).(d_x, d_y) = 0 \tag{2.15}$$

Avec cette équation, seule la composante de la vitesse parallèle au gradient spatial d'intensité, appelée vitesse normale, est directement calculable. Ce problème classique et commun aux différentes approches d'estimation du mouvement est désigné sous le nom de "problème de l'ouverture". Par exemple, l'estimation du mouvement n'a pas de solution unique pour une sphère homogène tournant sur elle même. Différentes techniques ont été proposées pour s'affranchir de ce problème et disposer d'une caractérisation plus précise du mouvement. Il faut cependant noter qu'une caractérisation incomplète du mouvement n'est pas forcément une limitation pour certaines applications.

Nous détaillons maintenant brièvement les différents types d'approches existantes pour l'estimation du flot optique. Les techniques d'estimation peuvent être groupées en trois catégories, appelées respectivement méthodes de mise en correspondance, méthodes fréquentielles et méthodes différentielles.

Méthodes de mise en correspondance :
L'approche de mise en correspondance ou *block matching* consiste à trouver le déplacement (d_x, d_y) qui apparie au mieux des régions ou des éléments caractéristiques (contours, coins ...) de la scène entre deux instants consécutifs. L'appariement est en général calculé par une corrélation ou par distance entre les régions de l'image aux instants t et $t + 1$. Ces techniques sont simples à mettre en œuvre et permettent de mesurer des déplacements de grande amplitude.

Elles se retrouvent dans de nombreux standards de compression vidéo (MPEG1 et 2). Cependant, nécessitant une recherche exhaustive sur une région d'intérêt, elles peuvent être très coûteuses. De plus, les critères de similarité utilisés ne sont généralement pas invariants aux transformations géométriques (changement d'échelle, rotation, distorsion perspective) de l'image. Enfin, ces techniques sont légèrement imprécises du fait de la discrétisation du déplacement estimé.

Méthodes fréquentielles :
Les méthodes fréquentielles [Fleet 90, Heeger 88, Spinei 98] sont basées sur la transformée de Fourier en mouvement. Les fréquences temporelles sont déplacées par le produit de la vitesse et des fréquences spatiales. L'information de mouvement est en général extraite par des filtres orientés en espace et en temps (Gabor 3D ou filtre large-bande). La contrainte imposée pour éviter le problème d'ouverture est spatio-temporelle : le mouvement à déterminer est supposé constant à la fois sur le support spatial et temporel des filtres orientés. Le résultat obtenu est lissé en espace et en temps, ce qui peut poser des problèmes pour des séquences d'images où les changements sont rapides. Ces méthodes nécessitent généralement un coût de calcul assez important.

Méthodes différentielles :
Les méthodes différentielles reposent directement sur la résolution de l'équation de contrainte du mouvement apparent (équation 2.15). Cette dernière s'appuie sur deux hypothèses : la conservation de l'intensité et un déplacement faible entre deux images consécutives. Un approche multirésolution permet de considérer des déplacements de plus fortes amplitudes.

Le principal problème de l'ECMA, problème d'ouverture, est lié au fait que cette seule équation a deux inconnues. Elle ne peut donc être résolue sans ajouter des contraintes supplémentaires garantissant l'unicité de la solution. Les méthodes de régularisation du flot optique les plus utilisées sont la régularisation de Horn et Schunck [Horn 81] et celle de Lucas et Kanade [Lucas 81]. La régularité de Horn et Schunck, appelée contrainte de lissage de l'image, stipule que le champ de mouvement doit être régulier sur l'ensemble du support (pouvant être toute l'image). Celle de Lucas et Kanade suppose que le champ de mouvement est constant sur un support local. L'avantage de la méthode introduite par Lucas et Kanade est qu'elle n'est pas forcément utilisée pour calculer des champs denses. En effet, elle est majoritairement appliquée à l'estimation de mouvement en des points caractéristiques.

Les méthodes différentielles présentent de nombreux avantages face aux méthodes fréquentielles et de mise en correspondance. L'équation du flot optique permet une estimation subpixellique du mouvement (contrairement aux méthodes de mise en correspondance). De plus, même si l'estimation d'un champ dense par les méthodes de flot optique est coûteuse, elle l'est tout de même moins que les méthodes fréquentielles. Nous avons donc décider d'utiliser une de ces méthodes pour calculer les vecteurs de mouvement en chaque point en mouvement de la grille. Comme nous n'avons pas besoin d'un champ dense, nous utilisons la méthode introduite par Lucas et Kanade avec une approche multi-résolution. Le principe de cette méthode est détaillé dans la sous-section suivante.

2.3.1.2 Calcul des vecteurs de mouvement

La régularisation introduite par Lucas et Kanade [Lucas 81] repose sur l'hypothèse d'un mouvement constant sur le voisinage des points. Soit $V(x, y)$ le voisinage de taille n d'un point $s = (x, y)$. L'hypothèse d'homogénéité du mouvement conduit à un système sur-déterminé ré-

solu par une minimisation aux moindres carrés :

$$\text{argmin}_{(d_x,d_y)} \sum_{(x_i,y_i)\in V(x,y)} \Big[\frac{\partial I}{\partial x}(x_i,y_i,t)d_x + \frac{\partial I}{\partial y}(x_i,y_i,t)d_y + \frac{\partial I}{\partial t}(x_i,y_i,t)\Big]^2 \quad . \tag{2.16}$$

Le système (2.16) ne peut malheureusement être résolu que pour des voisinages texturés, c'est-à-dire des voisinages pour lesquels le gradient d'intensité n'est pas nul sur toute la région. Si cette condition n'est pas respectée pour un point, les vecteurs de mouvement ne seront pas calculés et le point sera enlevé de la grille. La grille se redéfinit alors ainsi :

$$G = \Big\{ s = \Big(\frac{k.w}{N_G}, \frac{l.h}{N_G}\Big) \mid M_t(s) = 1 \;\&\; \exists (x_i,y_i) \in V\big(\frac{k.w}{N_G}, \frac{l.h}{N_G}\big), |\nabla I(x_i,y_i)| \neq 0 \Big\} \quad . \tag{2.17}$$

Afin de prendre en compte de grands déplacements, nous avons utilisé une version multi-échelle de cette méthode. On considère pour cela une pyramide d'images construite par filtrages gaussiens et sous-échantillonages successifs de l'image originale. Les vecteurs de déplacements sont d'abord estimés aux échelles grossières et servent ensuite d'initialisation aux échelles plus fines (en descendant la pyramide).

Des champs denses peuvent être calculés avec cette méthode mais nous nous contentons d'estimer les vecteurs de mouvement pour les points de la grille. Des exemples de cartes de mouvement obtenues sont montrés sur les figures 2.4(f) et 2.5(f).

2.3.1.3 Validation des vecteurs par un test statistique

Le calcul du flot optique par la méthode de Lucas et Kanade se base sur deux hypothèses. La première, qui correspond à l'équation de contrainte du mouvement apparent, est la conservation de la luminance entre deux images consécutives. La deuxième suppose un mouvement quasi constant autour d'un point. Si ces hypothèses ne sont pas respectées, et si les gradients spatiaux et temporels ne sont pas assez significatifs, les vecteurs de déplacement peuvent être erronés. Nous avons donc rajouté une étape de validation des vecteurs de mouvement calculés en chaque point de la grille. Cette étape repose sur la comparaison de l'intensité du voisinage d'un point $s = (x, y)$ de l'image au temps t (échantillon de données X) avec le voisinage du point correspondant $s' = (x + d_x, y + d_y)$ dans l'image à l'instant $t + 1$ (échantillon de données Y).

La méthode la plus classique pour comparer deux échantillons est le calcul du coefficient de corrélation de Pearson. Si les deux échantillons X et Y ont pour variances respectives $V(X)$ et $V(Y)$, et pour covariance $cov(X, Y)$, le coefficient de corrélation s'écrit :

$$\gamma = \frac{cov(X,Y)}{\sqrt{V(X)V(Y)}} \quad . \tag{2.18}$$

Ses valeurs sont comprises entre -1 et 1. Une valeur proche de zéro indique que les deux ensembles ne sont pas corrélés. Cette relation ne prend pas en compte les distributions individuelles des deux ensembles et n'est donc pas la meilleure solution pour décider si les deux échantillons sont réellement corrélés. Des tests statistiques mieux adaptés existent.

Le but d'un test statistique est de décider entre deux hypothèses H_0 et H_1 laquelle doit être rejetée. Dans notre cas, l'hypothèse nulle H_0 affirme que les deux ensembles ne sont pas corrélés et l'hypothèse alternative H_1 qu'ils sont corrélés. Un risque, dit risque alpha, peut être associé à

l'hypothèse nulle. Il définit le risque que l'on prend en rejetant l'hypothèse nulle alors que cette hypothèse était vraie. Il traduit le niveau de confiance d'un test statistique.

Le test statistique relié au coefficient de corrélation que nous utilisons est l'indice de confiance appelé "p-value". Il s'agit de la probabilité d'avoir obtenu les résultats de la corrélation par chance. La p-value est basée sur la statistique de décision :

$$T = \gamma\sqrt{\frac{n-2}{1-\gamma^2}} \ , \tag{2.19}$$

où n est la taille des deux échantillons X et Y et γ le coefficient de corrélation. Une valeur de T proche de zéro assure que, sous l'hypothèse nulle, les deux ensembles de données ne sont pas corrélés. Sous l'hypothèse nulle, la loi de T est bien approchée par une distribution de Student définie par :

$$A(x) = \frac{1}{\sqrt{n-2}\, B(\frac{1}{2}, \frac{n-2}{2})} \int_{-x}^{x} \big(1 + \frac{y^2}{n-2}\big)^{\frac{3-n}{2}} dy \ , \tag{2.20}$$

où B est la fonction bêta donnée par :

$$B(a,b) = B(b,a) = \int_0^1 x^{a-1}(1-x)^{b-1} dx \ . \tag{2.21}$$

La probabilité, sous H_0, d'obtenir un T supérieur à la valeur observée définit la p-value. Elle est donnée par $A(|T|)$. Si l'on veut limiter à 5% (valeur la plus couramment utilisée par les statisticiens) le risque d'erreur ou risque alpha, c'est-à-dire le risque de rejeter l'hypothèse nulle alors qu'elle était vraie, on considère que les deux échantillons sont corrélés si la p-value est inférieure à 0.05. Pour plus d'informations sur la p-value et sur sa mise en œuvre, nous renvoyons à [Press 92].

Nous avons fait le choix de placer le risque à 5%. Si la p-value obtenue pour un point s est supérieure à 0.05, les vecteurs de déplacement ne seront pas considérés comme valides. Dans ce cas, nous ne garderons pas le point pour les étapes suivantes de l'algorithme. Une nouvelle grille,

$$G = \Big\{ s = \big(\frac{k.w}{N_G}, \frac{l.h}{N_G}\big) \mid M_t(s) = 1 \,\&\, \exists (x_i, y_i) \in \mathcal{V}\big(\frac{k.w}{N_G}, \frac{l.h}{N_G}\big), |\nabla I(x_i, y_i)| \neq 0 \,\&\, \text{p-value}(s, s') < 0.05 \Big\} \ , \tag{2.22}$$

est donc définie et un vecteur de déplacement

$$\mathbf{z}_t^{(M)}(s) = (d_x, d_y) \tag{2.23}$$

est associé à chacun de ses points. Contrairement aux informations de couleur, le vecteur de caractéristiques de mouvement n'est défini que pour les points de la grille. Dans la suite, nous noterons $|G|$ la taille de cette grille définitive.

Sur les figures 2.4 et 2.5 nous montrons l'influence de ce test statistique sur deux séquences différentes. La grille finale ainsi que les vecteurs de mouvement associés sont présentés. Comme on peut le voir, l'ajout d'une phase de validation des vecteurs de flot optique permet de supprimer les vecteurs aberrants. Sur ces figures sont également montrés les résultats obtenus en ne gardant que les points pour lesquels le coefficient de corrélation obtenu est supérieur à 0.5. Les grilles ont été obtenues avec les paramètres suivants. Pour la séquence de ski nautique, les images sont de taille $w = 320$ et $h = 240$, le nombre de pixels en mouvement est $m = 7794$, et finalement le

TAB. 2.1: Taille des grilles pour les images 108 de la séquence de ski nautique et 16 de la séquence de conducteur

	sans validation	correlation > 0.5	p-value < 0.05
ski nautique	302	140	177
conduteur	420	184	277

paramètre N_G définissant la taille de la grille vaut 5. Pour la séquence de conducteur on a $w = 240$, $h = 320$, $m = 10845$ et $N_G = 5$. Les tailles (nombre de points) des grilles sont montrées dans le tableau 2.1. Le test basé sur la p-value conserve plus de points que celui basé sur la corrélation, notamment des points importants sur le visage du skieur ou sur le bras du conducteur. Une comparaison approfondie devrait être menée. Néanmoins, un avantage non négligeable de la p-value est que le seuil à régler (risque alpha) n'influence pas autant les résultats (dès lors qu'il est choisi assez petit) que le seuil de corrélation.

2.3.2 Caractéristiques photométriques

Le mouvement n'est pas toujours suffisant pour différencier les objets du fond. Des caractéristiques photométriques doivent donc être ajoutées au descripteur. De nombreux descripteurs photométriques ont été étudiés, le plus connu étant probablement le descripteur SIFT (Scale Invariant Feature Transform) [Lowe 99]. Ce descripteur est invariant aux changements de luminosité et aux transformations géométriques. Cependant comme il s'agit d'un vecteur de dimension 128 nous ne pouvons l'utiliser dans une étape de clustering.

Le descripteur photométrique que nous utiliserons doit être assez discriminant tout en étant de dimension faible. Les séquences couleur contiennent plus d'informations photométriques que les séquences en niveau de gris. Nous n'utiliserons donc pas le même descripteur pour ces deux types de séquences.

2.3.2.1 Descripteurs dans les séquences monochromes

La première information utilisée dans le descripteur photométrique pour les séquences en niveau de gris est l'intensité $\mathbf{z}_t^{(G)}(s)$ du point $s = (x, y)$ pour lequel le descripteur est calculé. Afin d'être plus robuste au bruit, nous utilisons plutôt la moyenne, notée $\overline{\mathbf{z}}_t^{(G)}(s)$, de l'intensité sur un petit voisinage autour du point. Diverses expérimentations nous ont montré que cette information n'est pas suffisante. En effet, le contraste entre un objet et le fond peut être faible. Plutôt que de rajouter de nombreuses dimensions en utilisant le gradient dans plusieurs directions, nous avons décidé d'utiliser une information sur la texture. De nombreuses définitions et descriptions différentes de la texture existent. Une texture peut être définie comme une région d'intensité non constante, comme une organisation spatiale des pixels ou comme une répétition d'un motif ou de quelques éléments. Le point commun entre ces différentes définitions est le suivant : une région texturée est une région avec des variations d'intensité. En d'autres termes, toute région non homogène peut être vue comme une texture. Plusieurs méthodes pour analyser la texture existent également : statistiques du premier ordre (statistiques sur l'histogramme), matrices de co-occurence (recherche de motifs), analyse fréquentielle (filtres de Gabor)... Ces approches représentent toutes la texture avec des descripteurs de grandes dimensions.

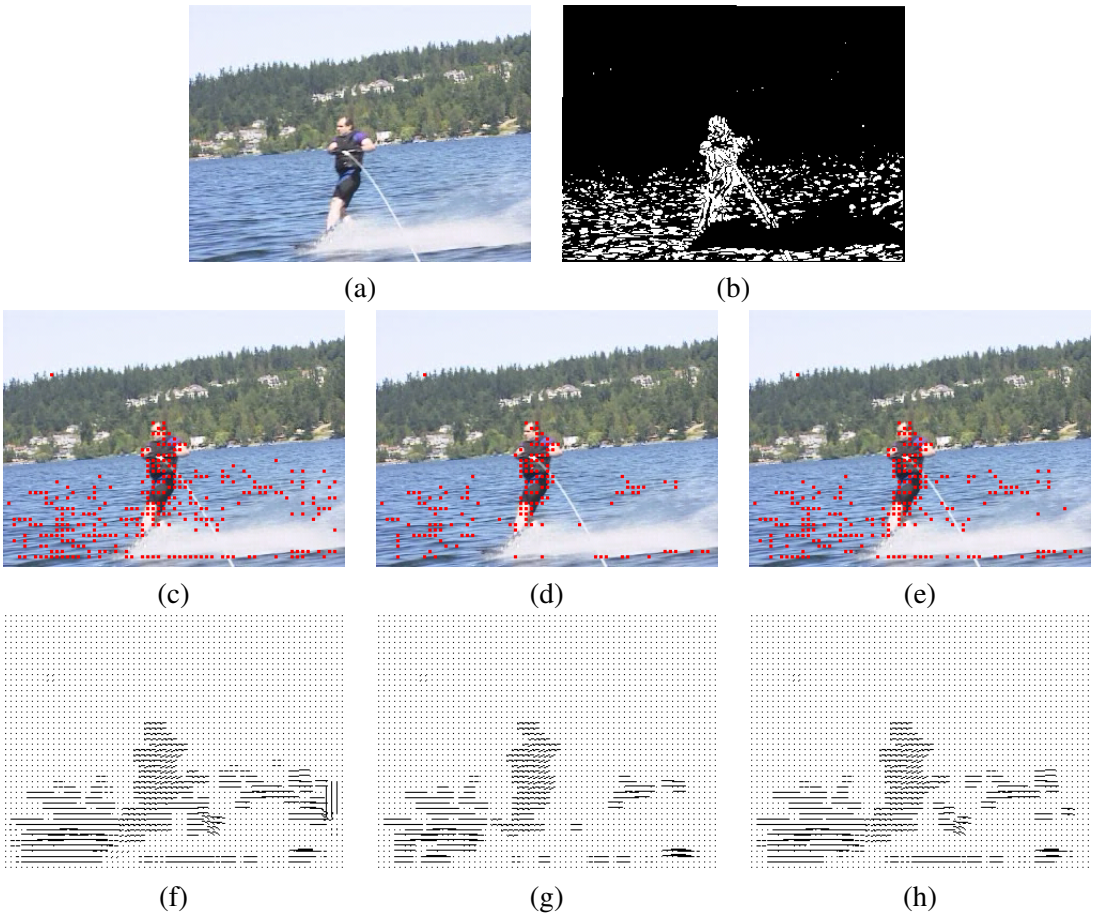


FIG. 2.4: Construction de la grille de points (deuxième ligne) et calcul du flot optique (troisième ligne) sur l'image 108 de la séquence de ski nautique. (a) Image initiale. (b) Carte des pixels en mouvement. (c) Grille restreinte uniquement aux pixels en mouvement (les vecteurs de mouvement sont montrés sur l'image (f)). (d) Grille finale obtenue après validation des vecteurs de mouvement par un test de corrélation (seuil = 0.5). (les vecteurs de mouvement sont montrés sur l'image (g)). (e) Grille finale obtenue après validation des vecteurs de mouvement par un test statistique (les vecteurs de mouvement sont montrés sur l'image (h)).

Dans ce document, nous caractérisons la texture simplement par la quantité ou la force des contours présents dans la région. Cette définition, très simple, ne pourrait être utilisée pour la mise en correspondance ou la classification de textures. Néanmoins, ajoutée aux autres dimensions du descripteur, elle est suffisante pour différencier les pixels d'un objet des pixels du fond. Ainsi, la texture en un point de la grille correspond ici à l'écart-type de la moyenne du Laplacien de l'intensité sur un petit voisinage autour du point : $\sigma_{\Delta \mathbf{z}_t^{(G)}(s)}$.

Afin d'apporter de la cohérence temporelle à notre descripteur, nous ajoutons en chaque point s de l'image t l'information d'intensité et de texture du point $s' = (x + d_x, y + d_y)$ correspondant à $s = (x, y)$ dans l'image $t + 1$. Le descripteur au point $s = (x, y)$ correspondant à l'indice

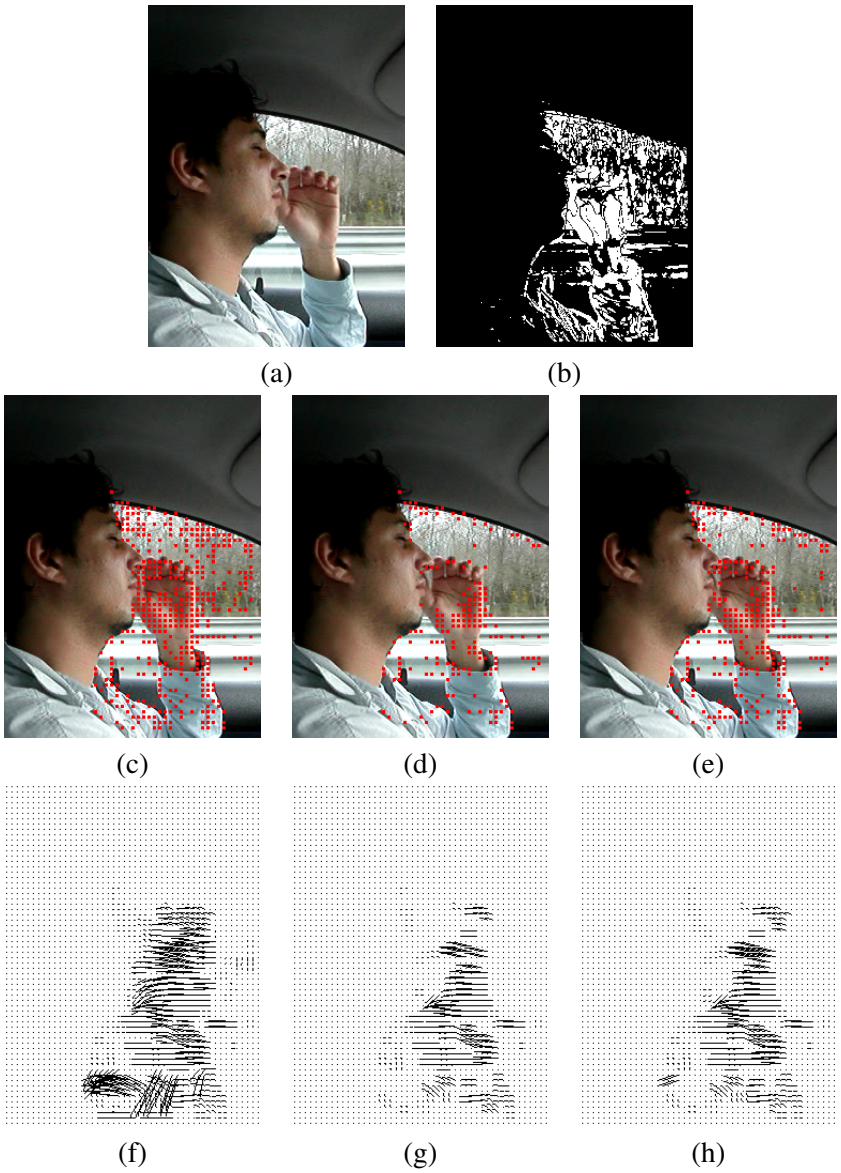


FIG. 2.5: Construction de la grille de points (deuxième ligne) et calcul du flot optique (troisième ligne) sur l'image 16 de la séquence de conducteur. (a) Image initiale. (b) Carte des pixels en mouvement. (c) Grille restreinte uniquement aux pixels en mouvement (les vecteurs de mouvement sont montrés sur l'image (f)). (d) Grille finale obtenue après validation des vecteurs de mouvement par un test de corrélation (seuil = 0.5). (les vecteurs de mouvement sont montrés sur l'image (g)). (e) Grille finale obtenue après validation des vecteurs de mouvement par un test statistique (les vecteurs de mouvement sont montrés sur l'image (h)).

$i \in \{1 \dots |G|\}$ de la grille est finalement défini par :

$$\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)}), \tag{2.24}$$

avec

$$\mathbf{x}_1^{(i)} = (x, y) \quad ,$$

$$\mathbf{x}_2^{(i)} = (d_x, d_y) \quad \text{et}$$

$$\mathbf{x}_3^{(i)} = (\overline{\mathbf{z}}_t^{(G)}(s), \sigma_{\Delta\mathbf{z}_t^{(G)}(s)}, \overline{\mathbf{z}}_{t+1}^{(G)}(s'), \sigma_{\Delta\mathbf{z}_{t+1}^{(G)}(s')}) \quad .$$

2.3.2.2 Descripteurs dans les séquences couleur

Différents tests expérimentaux nous ont montré que les trois canaux de couleur sont suffisants pour traiter des séquences couleur. Aucune information de texture ne sera alors rajoutée pour ce type de vidéos, ce qui permet également de réduire le nombre de dimensions du descripteur en chaque point. Les images couleur sont codées dans l'espace de couleur RGB. Cette représentation a un défaut : les trois canaux rouge, vert et bleu sont hautement corrélés ce qui rend cette représentation sensible aux changements de luminosité. Nous avons donc chercher un meilleur espace pour nos séquences. Les objets ou clusters intéressants à détecter contenant souvent de la peau, utiliser un espace de chrominance est approprié. En effet, la peau a une signature particulière dans l'espace des chrominances [Kjeldsen 96, Singh 03]. Il a été montré dans [Terrillon 00] que tous les espace de chrominance sont équivalents pour détecter de la peau. Nous avons choisi l'espace YUV. Le modèle YUV définit un espace colorimétrique en trois composantes. La première représente la luminance et les deux autres représentent la chrominance. Il s'agit du système utilisé dans les systèmes de diffusion télévisuelle PAL et NTSC. La conversion d'une image RGB en une image YUV se fait en appliquant la transformation linéaire :

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,436 \\ 0,615 & -0,515 & -0,100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.25}$$

La figure 2.6 montre un exemple de conversion sur la séquence de conducteur. On remarque bien que les régions avec de la peau se différencient nettement du reste de l'image pour les canaux de chrominance. Le vecteur de couleur en chaque pixel s de l'image sera défini dans toute la suite du document par :

$$\mathbf{z}_t^{(C)}(s) = (Y_t(s), U_t(s), V_t(s)) \ . \tag{2.26}$$

Afin de rendre le descripteur plus robuste au bruit, nous utilisons la moyenne de chaque canal de couleur calculée sur un voisinage autour du point. De plus, comme pour les séquences monochromes, nous incluons la correspondance dans l'image à l'instant suivant pour ajouter de la cohérence temporelle à notre détecteur de clusters en mouvement. Le descripteur au point $s = (x, y)$ correspondant à l'indice $i \in \{1 \dots |G|\}$ de la grille est finalement défini par :

$$\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)}), \tag{2.27}$$

avec

$$\mathbf{x}_1^{(i)} = (x, y) \quad ,$$

$$\mathbf{x}_2^{(i)} = (d_x, d_y) \quad \text{et}$$

$$\mathbf{x}_3^{(i)} = (\overline{\mathbf{z}}_t^{(C)}(s), \overline{\mathbf{z}}_{t+1}^{(C)}(s')) \ ,$$

où $s' = (x + d_x, y + d_y)$.

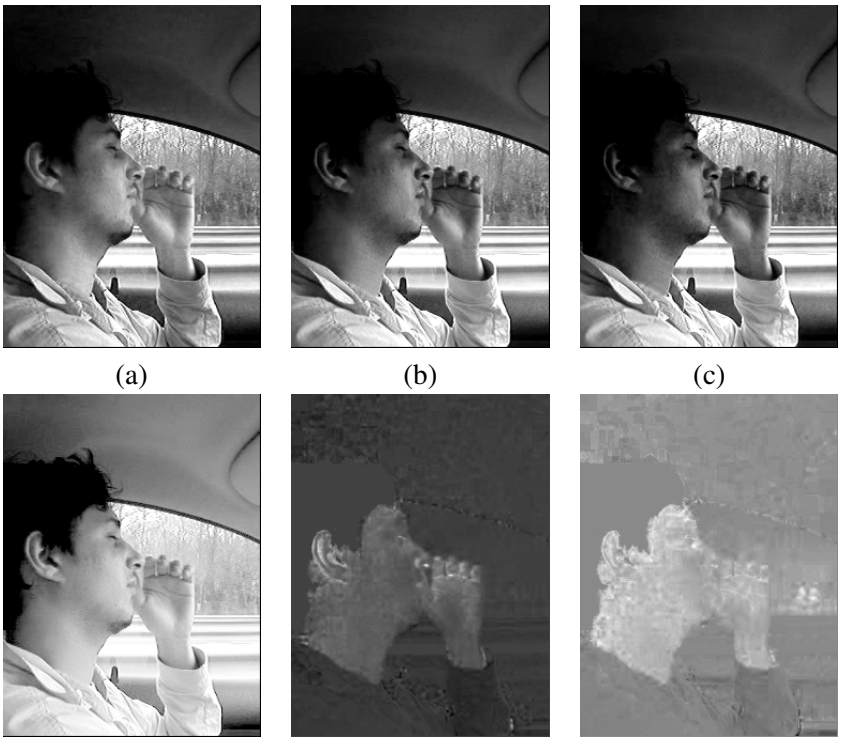


FIG. 2.6: Espace de chrominance YUV a)-c) Les trois canaux rouge, vert et bleu pour l'image 16 de la séquence de conducteur. d)-f) Les trois canaux Y,U,V correspondant

2.4 Conclusion

Dans ce chapitre nous avons présenté la sélection et la description des points qui serviront pour la détection des clusters en mouvement. Nous rappelons ici les résultats importants. Les points sélectionnés sont des pixels en mouvement uniformément répartis sur toute l'image et ayant des mouvements "valides". La validité des vecteurs de mouvement est évaluéee en appliquant un test statistique sur les vecteurs de déplacement calculés avec l'approche multi-résolution de Lucas et Kanade. La grille de points est finalement définie par l'équation (2.22). Un descripteur est ensuite associé à chaque point de cette grille. Les points sont décrits par leur position, leur mouvement et leurs caractéristiques photométriques. Pour des séquences en niveaux de gris, les caractéristiques photométriques retenues sont l'intensité et la texture (équation 2.24). Pour les séquences couleur, seules les informations de couleur dans un espace de chrominance sont prises en compte (équation 2.27). Le chapitre suivant présente la méthode de partitionnement qui permettra de détecter les clusters en mouvement.

Chapitre 3

Partitionnement des points

Dans la première partie de ce document nous cherchons à détecter les objets en mouvement. Un objet en mouvement est défini comme une région fermée ayant une couleur et un mouvement quasi uniforme (définition 1). La détection des objets se décompose en deux étapes : l'extraction de clusters en mouvement (définition 2) et la segmentation des différents objets, chacun correspondant à un cluster. Seuls les points de la grille définie dans le chapitre précédent servent à l'extraction des clusters. L'extraction peut immédiatement se faire à l'aide d'une méthode de partitionnement des points, aussi appelée clustering. Ce chapitre présente un état de l'art des méthodes de clustering et détaille la méthode dite "mean shift" qui sera utilisée. Le mean shift est une méthode non paramétrique basée sur une estimation à noyau. Après l'état de l'art, nous rappellerons quelques principes des estimateurs à noyau et terminerons en décrivant la méthode mean shift pour des noyaux de taille fixe sur tout l'espace des données et pour des noyaux de taille variable. Deux principaux estimateurs à noyau variable existent dans la littérature : l'estimateur "sample point" et l'estimateur "balloon". L'algorithme de clustering mean shift a déjà été introduit pour l'estimateur sample point [Comaniciu 01]. Nous introduisons dans ce chapitre le mean shift pour l'estimateur balloon, appelé "*pseudo balloon mean shift*", qui sera mieux adapté aux espaces multidimensionnels (comme nous le verrons au chapitre 4).

3.1 Les méthodes de clustering

Nous présentons dans cette section un état de l'art des algorithmes de clustering. Cet état de l'art n'est pas exhaustif et se limite aux algorithmes les plus cités. Pour une bibliographie plus complète, le lecteur peut se référer à [Jain 99,Grabmeier 02]. Les algorithmes de clustering sont ici divisés en quatre classes : approches basées sur le partitionnement des données, approches hiérarchiques, approches paramétriques d'estimation de la densité et approches non paramétriques d'estimation de la densité. Cette classification n'est pas stricte car certaines approches appartiennent à plusieurs catégories.

Les notations utilisées dans cette section sont les suivantes. Le nombre total de clusters, fixé préalablement par l'utilisateur ou automatiquement obtenu par l'algorithme, sera noté k_c. Nous introduisons la fonction c qui à chaque vecteur $\mathbf{x}^{(i)}$, correspondant au i^{ieme} point de l'espace des données, associe l'indice du cluster auquel appartient le point. Si le point i appartient au cluster d'indice u, alors $c(i) = u$. Nous utiliserons de manière abusive le terme point ou vecteur de caractéristiques pour faire appel au vecteur $\mathbf{x}^{(i)}$ représentant chaque point. De même, nous

appellerons parfois point l'indice i d'un point dans la grille. On définit par $\mathcal{C}_u$ l'ensemble des points appartenant au cluster u : $c(i) = u \Leftrightarrow \mathbf{x}^{(i)} \in \mathcal{C}_u$. Le nombre de points de l'espace correspond à la taille de la grille, noté $|G|$.

3.1.1 Approches basées sur le partitionnement des données

Les méthodes basées sur le partitionnement des données permettent de diviser l'ensemble des données en k_c clusters. Le nombre de clusters k_c est fixé *a priori* par l'utilisateur. Ces méthodes reposent généralement sur le choix d'une partition initiale qui est ensuite améliorée avec un algorithme itératif. Chaque cluster u est représenté par un vecteur $\mathbf{v}_u$, ce dernier étant le plus souvent le centre de gravité du cluster.

L'objectif de ces méthodes est de minimiser sur l'ensemble des clusters une mesure de similarité, appelée dist, entre le représentant d'un cluster et tous les vecteurs qui le composent :

$$sim = \sum_{u=1}^{k_c} \sum_{i|c(i)=u} \mathrm{dist}(\mathbf{x}^{(i)}, \mathbf{v}_u) \ . \tag{3.1}$$

Le plus connu des algorithmes de partitionnement des données est le K-means. Cet algorithme utilise la distance euclidienne comme mesure de similarité. La minimisation de l'équation (3.1). est un problème d'optimisation. Dans le K-means, l'optimisation appliquée est une descente de gradient. Cette dernière est très sensible aux conditions initiales et peut converger vers des minima locaux. D'autres méthodes de partitionnement des données existent mais elles restent toutes assez sensibles aux conditions initiales.

3.1.2 Approches hiérarchiques

Le principe des approches hiérarchiques est de construite un arbre binaire indicé appelé dendogramme. Le niveau le plus bas de l'arbre correspond à la partition la plus fine où chaque point de l'ensemble des données représente un cluster. Le niveau le plus haut correspond à la partition constituée d'un seul cluster englobant l'ensemble des points. Entre les deux, plusieurs partitions peuvent être construites. Le partionnement final des données est obtenu en coupant l'arbre horizontalement. Chaque nœud de l'arbre correspond à un cluster et englobe l'ensemble des points contenus dans le sous-arbre correspondant. La construction de l'arbre peut être ascendante (*bottom-up*) [Lance 67] ou descendante (*top-down*) [Hubert 74].

La hauteur d'un nœud dans l'arbre est importante. Elle représente l'indice de dispersion du nœud et correspond à une mesure de proximité entre les points qu'il contient. Elle est calculée après la fusion de deux groupes de points et est en fait basée sur une mesure d'écart (distance minimale ou maximale entre les points des deux groupes, variance des vecteurs [Lebart 84] ...) entre les deux groupes ou sur le nombre de clusters souhaités.

Une propriété intéressante des algorithmes hiérarchiques est la stabilité des résultats. En effet, aucun paramètre ou *a priori* ne doit être défini préalablement. Cependant, ils sont très coûteux et sont donc peu adaptés à de grands ensembles de données.

3.1.3 Approches paramétriques d'estimation de la densité

Les approches basées sur l'estimation de la densité partitionnent un ensemble de vecteurs en utilisant leur densité locale. Celle-ci est estimée à l'aide de méthodes statistiques d'estimation

de la densité. Chaque groupe de points localement dense correspond à un cluster. Il existe deux types de méthodes d'estimation de la densité : les méthodes paramétriques et les méthodes non paramétriques. Les premières supposent que les données sont distribuées suivant un modèle statistique connu. L'objectif est alors d'estimer les paramètres du modèle, généralement un mélange de distributions de probabilité de forme paramétrique :

$$f(\mathbf{x}) = \sum_{j=1}^{k_c} \pi_j \phi_j(\mathbf{x}, \alpha_j) \ , \tag{3.2}$$

où ϕ_j est une distribution, π_j un paramètre du modèle représentant le poids de la distribution j et α_j un paramètre traduisant la densité des clusters. Les distributions sont généralement supposées gaussiennes et les paramètres souvent estimés par l'algorithme EM. Les paramètres estimés sont alors les moyennes et les matrices de covariance des gaussiennes ainsi que le poids de chacune d'elles. Chaque composante du mélange correspond à un cluster. Une fois les paramètres estimés, chaque point est attribué à la composante la plus probable pour ce point.

Les approches paramétriques sont très simples à mettre en œuvre mais la qualité de l'estimation dépend fortement de l'adéquation du modèle choisi par rapport à la distribution réelle des points. De plus l'algorithme EM peut s'avérer assez coûteux.

3.1.4 Approches non paramétriques d'estimation de la densité

Avec les statistiques paramétriques, une hypothèse est formulée sur la loi suivie par la fonction de densité, ce qui correspond à une connaissance *a priori*. Une autre manière d'estimer la densité d'un ensemble de données est d'utiliser une méthode non paramétrique. La méthode non paramétrique la plus connue pour le clustering est le *mean shift*. Il s'agit d'un procédure itérative de montée de gradient qui estime les modes de la densité d'un nuage de points. Nous avons décider d'utiliser cette méthode dans notre algorithme de détection des objets mobiles. En effet, les algorithmes mean shift convergent rapidement vers un partitionnement unique et ne nécessitent pas l'apport d'un *a priori* sur les données. Ce chapitre décrit les algorithmes de clustering mean shift. Les algorithmes mean shift se basent sur l'utilisation d'un estimateur à noyau. Nous rappellerons dans la section suivante le principe de l'estimation à noyau et présentons trois algorithmes mean shift dans la section 3.3. La différence entre ces trois algorithmes repose sur le noyau utilisé. Il peut être fixe ou variable ce qui conduit à des algorithmes de partitionnement différents. Un défaut des algorithmes mean shift est que la taille du noyau doit être préalablement définie. Nous dédions tout le chapitre suivant à la sélection automatique de ce paramètre.

3.2 Estimateurs à noyau

Les estimateurs à noyau sont des outils classiques en statistique. Pour plus de clarté dans ce chapitre et le suivant, nous rappelons ici quelques résultats sur les estimateurs à noyau de taille fixe et de taille variable.

3.2.1 Estimateur à noyau fixe

Étant donnés $\{\mathbf{x}^{(i)}\}_{i=1..n}$, n points dans l'espace $\mathbb{R}^d$ de dimension d, l'estimation non paramétrique de la densité en un point $\mathbf{x}$ est donnée par :

$$\widehat{f}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}^{(i)}) \tag{3.3}$$

où $K_{\mathbf{H}}$ est un noyau de profil k. La matrice de lissage $\mathbf{H}$ est appelée fenêtre, taille ou bande passante (en anglais "bandwidth") du noyau. L'estimateur à noyau est aussi appelé fenêtre de Parzen. Son fondement et ses propriétés sont bien décrites dans [Tsybakov 03]. L'équation précédente peut se réécrire :

$$\widehat{f}(\mathbf{x}) = \frac{1}{n|\mathbf{H}|^{1/2}}\sum_{i=1}^{n} K(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})) \ . \tag{3.4}$$

Un noyau K, adapté à l'estimation non paramétrique, doit respecter certaines propriétés. En effet, le noyau doit être une fonction de $\mathbb{R}^d$ dans $\mathbb{R}$ positive, bornée, à support compact vérifiant :

$$\int_{\mathbb{R}^d} K(\mathbf{x})\mathrm{d}\mathbf{x} = 1 \text{ (normalisée)}, \tag{3.5}$$

$$\lim_{\|\mathbf{x}\|\to\infty} \|\mathbf{x}\|^d K(x) = 0 \text{ (décroissance rapide)}, \tag{3.6}$$

$$\int_{\mathbb{R}^d} \mathbf{x}K(\mathbf{x})\mathrm{d}\mathbf{x} = 0 \ , \tag{3.7}$$

$$\int_{\mathbb{R}^d} \mathbf{x}\mathbf{x}^T K(\mathbf{x})\mathrm{d}\mathbf{x} = c_K\mathbf{I} \ , \tag{3.8}$$

où c_K est une constante de normalisation et $\mathbf{I}$ est la matrice identité.

Les estimateurs à noyau fixe ne donnent pas toujours une bonne représentation des données. Ainsi, si la densité des données varie de manière significative d'une partie à l'autre de l'espace considéré, il est préférable d'utiliser un estimateur à noyau variable. Deux principaux estimateurs à noyau variable existent [Wand 95]. Pour le premier, appelé estimateur "sample point", le noyau s'adapte aux points de données observés, tandis que pour le second, appelé estimateur "balloon", le noyau s'adapte aux points d'estimation.

3.2.2 Estimateur sample point

L'estimateur sample point a été introduit par Breiman *et al.* dans [L.Breiman 77]. Il s'agit d'un mélange de noyaux de même forme mais de bande passante variable centrée en chaque observation (points de données) :

$$\begin{aligned}\widehat{f}(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}(\mathbf{x}^{(i)})}(\mathbf{x} - \mathbf{x}^{(i)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\mathbf{H}(\mathbf{x}^{(i)})|^{1/2}} K(\mathbf{H}(\mathbf{x}^{(i)})^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})) \ .\end{aligned} \tag{3.9}$$

Les avantages et inconvénients de cet estimateur ont été étudiés dans [Terrell 92]. L'estimateur sample point est une densité et son biais est considérablement réduit si les noyaux $\mathbf{H}(\mathbf{x}^{(i)})$ sont bien paramétrés. Cependant il est difficile de calculer les valeurs de ces paramètres, en particulier dans des espaces hétérogènes de dimensions importantes. Dans [Terrell 92], des expérimentations ont montré que cet estimateur donne de bien meilleurs résultats que l'estimateur à noyau fixe si la taille de l'espace de données est relativement faible mais détériore les résultats quand la taille de l'espace augmente. Un autre désavantage est que l'estimé en un point peut être influencé par des observations très éloignées de ce point.

3.2.3 Estimateur balloon

L'estimateur balloon a été introduit par Loftsgaarden et Quensberry [Loftsgaarden 65]. Il est défini par :

$$\begin{aligned}\widehat{f}(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}(\mathbf{x})}(\mathbf{x}-\mathbf{x}^{(i)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\mathbf{H}(\mathbf{x})|^{1/2}} K(\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{x}-\mathbf{x}^{(i)}))\ .\end{aligned} \tag{3.10}$$

Cet estimateur à de bonnes propriétés asymptotiques [Mack 79], mais il ne s'agit pas d'une densité (son intégration ne donne pas toujours 1). Dans [Terrell 92], il a tout de même été montré que cet estimateur devient très performant, comparé aux deux autres estimateurs évoqués, quand la dimension des données devient supérieure à 3.

Avant de décrire les trois méthodes de clustering mean shift, chacune étant associée à un des estimateurs précédents, nous faisons un bref rappel sur la qualité d'une estimation et sur le profil d'un estimateur.

3.2.4 Qualité d'un estimateur à noyau fixe

La qualité d'un estimateur est mesurée en calculant l'erreur d'estimation $\widehat{f} - f$. Une mesure couramment utilisée pour évaluer un estimateur est le risque quadratique local (MSE), qui est égal à la somme de la variance et du biais au carré :

$$\begin{aligned}\mathrm{MSE}(\mathbf{x}) &= E\big[(\widehat{f}(\mathbf{x}) - f(\mathbf{x}))^2\big] \\ &= \mathrm{var}(\widehat{f}(\mathbf{x})) + \big[\mathrm{Biais}(\widehat{f}(\mathbf{x}))\big]^2\ ,\end{aligned} \tag{3.11}$$

avec le biais :

$$\mathrm{Biais}(\widehat{f}(\mathbf{x})) = E\big[\widehat{f}(\mathbf{x})\big] - f(\mathbf{x})\ , \tag{3.12}$$

et la variance :

$$\mathrm{var}(\widehat{f}(\mathbf{x})) = E\big[(\widehat{f}(\mathbf{x}) - E\big[\widehat{f}(\mathbf{x})\big])^2\big]\ . \tag{3.13}$$

D'autres mesures, telles que le risque quadratique intégré (MISE) ou risque quadratique intégré asymptotique (AMISE), existent également. Les résultats détaillés des différentes mesures appliquées aux estimateurs à noyau peuvent être trouvés dans [Scott 92] et [Wand 95]. Nous ne montrons ici que le calcul du biais et de la variance pour l'estimateur à noyau fixe. Notons ∇f le

gradient de la fonction f, et $\mathcal{H}(f)$ la matrice Hessienne des dérivées secondes. Le développement de Taylor d'ordre 2 en $\mathbf{x}$ de $f(\bullet)$ [Wand 95, p.94] est :

$$f(\mathbf{x}+\delta\mathbf{x}) = f(\mathbf{x}) + \delta\mathbf{x}^T\nabla f(\mathbf{x}) + \frac{1}{2}\delta\mathbf{x}^T\mathcal{H}(f(\mathbf{x}))\delta\mathbf{x} + o(\delta\mathbf{x}^T\delta\mathbf{x}) \ . \tag{3.14}$$

L'espérance d'un estimateur à noyau peut être interprétée comme une convolution [Stoker 93]. Ainsi, l'espérance de l'estimateur à noyau fixe est donnée par :

$$\begin{aligned} E\big(\widehat{f}(\mathbf{x})\big) &= \int \frac{1}{|\mathbf{H}|^{1/2}} K(\mathbf{H}^{-1/2}(\mathbf{u}-\mathbf{x}))f(\mathbf{u})\mathrm{d}\mathbf{u} \\ &= \int K(\mathbf{s})f(\mathbf{x}+\mathbf{H}^{1/2}\mathbf{s})\mathrm{d}\mathbf{s} \\ &= \int K(\mathbf{s})\big(f(\mathbf{x}) + (\mathbf{H}^{1/2}\mathbf{s})^T\nabla f(\mathbf{x}) + \frac{1}{2}(\mathbf{H}^{1/2}\mathbf{s})^T\mathcal{H}(f(\mathbf{x}))(\mathbf{H}^{1/2}\mathbf{s}) + \\ &\quad o(\mathbf{s}^T\mathbf{H}\mathbf{s})\big)\mathrm{d}\mathbf{s} \\ &= \big[\int K(\mathbf{s})\mathrm{d}\mathbf{s} f(\mathbf{x}) + \int K(\mathbf{s})\mathbf{s}^T\mathrm{d}\mathbf{s}\ \mathbf{H}^{1/2^T}\nabla f(\mathbf{x}) + \\ &\quad \int \frac{1}{2}K(\mathbf{s})(\mathbf{H}^{1/2}\mathbf{s})^T\mathcal{H}(f(\mathbf{x}))(\mathbf{H}^{1/2}\mathbf{s})\mathrm{d}\mathbf{s} + \int K(\mathbf{s})o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s}\big] \ . \end{aligned} \tag{3.15}$$

En utilisant les propriétés du noyau introduites dans la sous-section précédente (équation 3.11), le fait que la trace (tr($\bullet$)) d'un scalaire est égale au scalaire lui-même et que tr($\mathbf{AB}$) = tr($\mathbf{BA}$), le biais de l'estimateur à noyau fixe devient :

$$\begin{aligned} \text{Biais}(\widehat{f}(\mathbf{x})) &= E\big(\widehat{f}(\mathbf{x})\big) - f(\mathbf{x}) \\ &= \int \text{tr}\big[K(\mathbf{s})(\mathbf{H}^{1/2}\mathbf{s})^T\mathcal{H}(f(\mathbf{x}))(\mathbf{H}^{1/2}\mathbf{s})\big]\mathrm{d}\mathbf{s} + \int K(\mathbf{s})o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s} \\ &= \int \text{tr}\big[\mathbf{H}^{1/2^T}\mathcal{H}(f(\mathbf{x}))\mathbf{H}^{1/2}\mathbf{s}\mathbf{s}^TK(\mathbf{s})\big]\mathrm{d}\mathbf{s} + \int K(\mathbf{s})o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s} \\ &= c_k\text{tr}\big[\mathbf{H}^{1/2^T}\mathcal{H}(f(\mathbf{x}))\mathbf{H}^{1/2}\big] + \int K(\mathbf{s})o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s} \ , \end{aligned} \tag{3.16}$$

et la variance est définie par :

$$\begin{aligned} \text{var}(\widehat{f}(\mathbf{x})) &= \text{var}\big[\frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}^{(i)})\big] \\ &= \frac{1}{n}\text{var}\big[\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}^{(i)})\big] \\ &= \frac{1}{n}\Big[E\Big(\big(\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}^{(i)})\big)^2\Big) - \Big(E\big(\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x}-\mathbf{x}^{(i)})\big)\Big)^2\Big] \\ &= \frac{1}{n}\big[\frac{1}{|\mathbf{H}|^{1/2}}\big(\int (K(\mathbf{s}))^2\mathrm{d}\mathbf{s} f(\mathbf{x}) + \int o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s}\big) - \\ &\quad \big(\int K(\mathbf{s})\mathrm{d}\mathbf{s} f(\mathbf{x}) + \int K(\mathbf{s})o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s}\big)^2\big] \\ &= \frac{1}{n|\mathbf{H}|^{1/2}}\big(\int (K(\mathbf{s}))^2\mathrm{d}\mathbf{s} f(\mathbf{x}) + \int K(\mathbf{s})o(\mathbf{s}^T\mathbf{H}\mathbf{s})\mathrm{d}\mathbf{s}\big) \ . \end{aligned} \tag{3.17}$$

Un bon estimateur est un estimateur non biaisé et de faible variance. La définition du biais et de la variance pour l'estimateur à noyau fixe conduit à l'analyse suivante : la valeur absolue du biais augmente et la variance diminue quand la taille du noyau $\mathbf{H}$ augmente. La qualité de l'estimation va donc fortement dépendre du choix du noyau, et un compromis entre le biais et la variance doit souvent être trouvé. Nous consacrons le chapitre suivant à l'étude du choix de la bande passante du noyau et ne donnons donc pas ici plus de détails sur les critères de sélection de la taille du noyau.

3.2.5 Profile d'un estimateur à noyau

Un noyau est caractérisé par son profil. Le profil est une fonction $k : [0, \infty) \rightarrow \mathbb{R}$. Nous nous intéressons ici aux noyaux symétriques. La relation reliant le noyau au profil est, dans ce cas, :

$$K(x) = c_k k(\|x\|^2) \ , \tag{3.18}$$

où c_k est une constante de normalisation strictement positive. Cette constante doit assurer la relation (3.5). L'estimateur à noyau de l'équation (3.3) se réécrit :

$$\begin{aligned} \widehat{f}(\mathbf{x}) &= \frac{1}{n|\mathbf{H}|^{1/2}} \sum_{i=1}^{n} K(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})) \\ &= \frac{c_k}{n|\mathbf{H}|^{1/2}} \sum_{i=1}^{n} k(\|\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2) \ . \end{aligned} \tag{3.19}$$

De nombreux noyaux, satisfaisant toutes les propriétés précédentes, existent. Les plus utilisés par les algorithmes mean shift sont le noyau d'Epanechnikov et le noyau gaussien. Le profil du noyau d'Epanechnikov est défini par :

$$k_E(x) = \begin{cases} 1 - x & 0 \leq x \leq 1 \\ 0 & x > 1 \ . \end{cases} \tag{3.20}$$

Il a de très bonnes propriétés statistiques. En effet, il peut être montré que la mesure AMISE est minimisée en utilisant un noyau d'Epanechnikov [Scott 92, p.139] [Wand 95, p.104]. Cependant sa dérivée n'est pas définie sur les frontières du domaine. L'autre noyau fréquement utilisé par les algorithmes mean shift est le noyau gaussien de profil :

$$k(x) = exp(-\frac{1}{2}x) \tag{3.21}$$

pour lequel :

$$g(x) = -k'(x) = \frac{1}{2}k(x) \ . \tag{3.22}$$

La constante de normalisation pour ce noyau est $c_k = (2\pi)^{-d/2}$.

3.3 Filtrage et clustering mean shift

Après avoir rappelé certaines propriétés des estimateurs à noyau, nous pouvons facilement introduire les algorithmes mean shift. Nous nous intéressons ici au filtrage et au clustering mean

shift. Le filtrage correspond à la recherche du mode de la densité à partir d'un point. Le clustering correspond au partitionnement final des données et découle directement du filtrage. Nous commençons par présenter le mean shift avec un estimateur à noyau fixe, suivi du mean shift avec un estimateur sample point, connu sous le nom de "variable bandwidth mean shift". Nous finissons ce chapitre en introduisant un algorithme mean shift basé sur l'estimateur balloon. Cet algorithme sera appelé "pseudo balloon mean shift".

3.3.1 Filtrage mean shift à noyau fixe

Le filtrage mean shift a d'abord été introduit par Fukunaga [Fukunaga 75] dans le but de proposer une estimation intuitive du gradient de la densité d'un nuage de points et de l'utiliser pour les problèmes de reconnaissance de formes. Il a ensuite été repris par Cheng [Cheng 95] et été très largement étudié par Comaniciu [Comaniciu 99, Comaniciu 01, Comaniciu 02]. Il s'agit d'une méthode itérative de montée de gradient permettant de trouver les modes de la densité d'un nuage de points. Le mode d'un nuage de point correspond à un maximum local de sa densité. Le principe de la procédure mean shift est de trouver ce mode en résolvant itérativement l'équation $\nabla f(\mathbf{x}) = \mathbf{0}$ sans estimer la densité. Le filtrage mean shift est bien décrit dans [Comaniciu 02] et nous ne rappelons ici que très brièvement le principe. Le gradient de l'estimateur à noyau fixe (équation 3.3) est donné par :

$$\nabla \widehat{f}(\mathbf{x}) = \mathbf{H}^{-1}\ \widehat{f}(\mathbf{x})\ \mathbf{m}(\mathbf{x}) \tag{3.23}$$

où $\mathbf{m}$ est le vecteur "mean shift",

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}^{(i)}\ g\big(\|\mathbf{H}^{-1/2}(\mathbf{x}-\mathbf{x}^{(i)})\|^2\big)}{\sum_{i=1}^{n} g\big(\|\mathbf{H}^{-1/2}(\mathbf{x}-\mathbf{x}^{(i)})\|^2\big)}\ -\ \mathbf{x}\quad , \tag{3.24}$$

avec$g = -k'$. Le vecteur mean shift est une estimation du gradient normalisé, à un coefficient multiplicatif près, et permet de définir un chemin menant à un maximum local de la densité, c'est-à-dire un mode du nuage de point. La procédure de filtrage mean shift est décrite dans l'algorithme 2. La convergence de cet algorithme est prouvée dans [Comaniciu 02]. Les points $\mathbf{y}^{(j)}$, calculés à

Algorithme 2 Filtrage mean shift

Étant donnés $\{\mathbf{x}^{(i)}\}_{i=1,\ldots,n}$, n points dans l'espace de dimension d et $\{\mathbf{z}^{(i)}\}_{i=1,\ldots,n}$ leurs modes associés.

Pour $i = 1\ \ldots n$

1. Initialiser $j = 1$, $\mathbf{y}^{(1)} = \mathbf{x}^{(i)}$.
2. Répéter
   - $\mathbf{y}^{(j+1)} = \mathbf{y}^{(j)} + \mathbf{m}(\mathbf{y}^{(j)})$ (équation 3.24).
   - $j = j + 1$.

   Jusqu'à $\mathbf{y}^{(j-1)} = \mathbf{y}^{(j)}$.
3. $\mathbf{z}^{(i)} = \mathbf{y}^{(j)}$.

chaque itération, sont appelés points de trajectoire. Le premier point $\mathbf{y}^{(1)}$ est le point d'estimation $\mathbf{x}$ et le dernier $\mathbf{y}^{(t_m)}$ est le mode $\mathbf{z}$.

Un algorithme de clustering peut être directement déduit de cette procédure de filtrage en regroupant tous les points ayant convergé vers le même mode. Cette procédure est décrite dans l'algorithme 3.

Algorithme 3 Clustering mean shift

Étant donnés $\{\mathbf{x}^{(i)}\}_{i=1,\ldots,n}$, n points dans l'espace de dimension d et $\{\mathbf{z}^{(i)}\}_{i=1,\ldots,n}$ leurs modes associés.

1. Appliquer l'algorithme de filtrage mean shift.
2. Regrouper tous les $\mathbf{z}^{(i)}$ qui sont distants d'au plus $\mathbf{H}$, c'est-à-dire regrouper deux modes $\mathbf{z}^{(i)}$ et $\mathbf{z}^{(j)}$ si :
$$\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\| \leq \|\mathbf{H}\| \ .$$
3. Regrouper tous les points $\mathbf{x}^{(i)}$ dont les modes associés appartiennent au même groupe.

L'algorithme de filtrage mean shift converge généralement moins vite avec un noyau gaussien qu'avec un noyau d'Epanechnikov. Cependant la qualité de l'estimation est souvent meilleure avec un noyau gaussien. Ainsi, nous avons préféré utiliser ce noyau. En utilisant un noyau normal, l'équation (3.24) devient

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}^{(i)} \ \exp(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}^{(i)}, \mathbf{H}))}{\sum_{i=1}^{n} \exp(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}^{(i)}, \mathbf{H}))} \ - \ \mathbf{x} \tag{3.25}$$

où

$$D^2(\mathbf{x}, \mathbf{x}^{(i)}, \mathbf{H}) \equiv (\mathbf{x} - \mathbf{x}^{(i)})^T \mathbf{H}^{-1} (\mathbf{x} - \mathbf{x}^{(i)}) \tag{3.26}$$

est la distance de Mahalanobis (au carré) entre $\mathbf{x}$ et $\mathbf{x}^{(i)}$.

3.3.2 Mean shift basé sur l'estimateur "sample point" :

Le filtrage mean shift utilisant l'estimateur sample point a été introduit par Comaniciu *et al.* dans [Comaniciu 01]. Nous ne donnons ici que la définition du vecteur mean shift. En utilisant l'estimateur sample point, l'équation (3.25) devient :

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} |\mathbf{H}(\mathbf{x}^{(i)})|^{-1/2} \mathbf{x}^{(i)} \ \exp(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}^{(i)}, \mathbf{H}(\mathbf{x}^{(i)})))}{\sum_{i=1}^{n} |\mathbf{H}(\mathbf{x}^{(i)})|^{-1/2} \exp(-\frac{1}{2} D^2(\mathbf{x}, \mathbf{x}^{(i)}, \mathbf{H}(\mathbf{x}^{(i)})))} \ - \ \mathbf{x} \ . \tag{3.27}$$

De nouveau, des algorithmes de filtrage et de clustering peuvent être déduits de cette définition. La preuve de convergence du filtrage mean shift avec l'estimateur sample point peut être trouvée dans [Comaniciu 01].

3.3.3 Mean shift basé sur l'estimateur "balloon"

Aucun algorithme mean shift utilisant l'estimateur balloon n'existe dans la littérature. L'explication vient des défauts de cet estimateur : l'estimateur ballon $\widehat{f}(\mathbf{x})$ n'est pas toujours une densité et présente des problèmes de discontinuité. De plus sa dérivée $\nabla \widehat{f}(\mathbf{x})$ contient des termes en $(\mathbf{x} - \mathbf{x}^{(i)})^2$ et $\mathbf{H}'(\mathbf{x})$, ce qui ne permet pas d'en déduire une expression sous forme de vecteur.

La performance de l'estimateur balloon a pourtant été montrée pour des données de dimension supérieure à 3, ce qui est notre cas ici. Nous rappelons que les descripteurs qui seront utilisés par l'algorithme de clustering sont de dimension 8 pour des séquences en niveau de gris et de dimension 10 pour des séquences couleur. Il nous a donc paru intéressant de mettre en place un algorithme de clustering mean shift avec cet estimateur. Pour cela, il est nécessaire d'effectuer plusieurs hypothèses.

Dans le contexte des algorithmes mean shift, la fonction $\mathbf{H}$ représentant la taille du noyau est seulement définie de manière discrète en chaque point d'estimation. On peut donc supposer que $\forall i = 1 \dots n, \mathbf{H}'(\mathbf{x}^{(i)}) = 0$. En utilisant le profil k, l'estimateur balloon (équation (3.10)) peut se réécrire au point $\mathbf{x}^{(i)}$:

$$\widehat{f}(\mathbf{x}^{(i)}) = \frac{c_k}{n} \sum_{j=1}^{n} \frac{1}{|\mathbf{H}(\mathbf{x}^{(i)})|^{1/2}} k(\|\mathbf{H}(\mathbf{x}^{(i)})^{-1/2}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})\|^2) \ . \tag{3.28}$$

Avec l'hypothèse émise ($\mathbf{H}'(\mathbf{x}^{(\mathbf{i})}) = 0$), la dérivée de cet estimateur, pour $i = 1 \dots n$, vaut :

$$\begin{aligned}
\widehat{\nabla} f(\mathbf{x}^{(i)}) &= \nabla \widehat{f}(\mathbf{x}^{(i)}) \\
&= \frac{c_k}{n|\mathbf{H}(\mathbf{x}^{(i)})|^{1/2}} \sum_{j=1}^{n} \mathbf{H}(\mathbf{x}^{(i)})^{-1}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)}) k(\|\mathbf{H}(\mathbf{x}^{(\mathbf{i})})^{-1/2}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})\|^2) \\
&= \frac{c_k}{n|\mathbf{H}(\mathbf{x}^{(i)})|^{1/2}} \mathbf{H}(\mathbf{x}^{(i)})^{-1} \sum_{j=1}^{n} k(\|\mathbf{H}(\mathbf{x}^{(i)})^{-1/2}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})\|^2)(\mathbf{x}^{(j)} - \mathbf{x}^{(i)}) \\
&= \frac{1}{n}\Big[\sum_{i=1}^{n} \mathbf{H}(\mathbf{x}^{(i)})^{-1} K_{\mathbf{H}(\mathbf{x}^{(i)})}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})\Big]\Big[\frac{\sum_{j=1}^{n} \mathbf{x}^{(j)} K_{\mathbf{H}(\mathbf{x}^{(i)})}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})}{\sum_{j=1}^{n} K_{\mathbf{H}(\mathbf{x}^{(i)})}(\mathbf{x}^{(j)} - \mathbf{x}^{(i)})} - \mathbf{x}^{(i)}\Big] \ .
\end{aligned} \tag{3.29}$$

Le dernier terme de cette équation nous donne directement le vecteur mean shift :

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{j=1}^{n} \mathbf{x}^{(j)} K_{\mathbf{H}(\mathbf{x})}(\mathbf{x} - \mathbf{x}^{(j)})}{\sum_{j=1}^{n} K_{\mathbf{H}(\mathbf{x})}(\mathbf{x} - \mathbf{x}^{(j)})} - 1 \ , \tag{3.30}$$

et un algorithme de filtrage pourrait en être déduit. Cependant, il reste encore un problème à résoudre. En effet si $\mathbf{H}$ varie en chaque point de trajectoire, la convergence n'est pas assurée. La solution que nous proposons est de définir un algorithme mean shift "pseudo balloon" pour lequel le noyau varie en chaque point d'estimation (c'est-à-dire en chaque point de données) mais est fixe pour tous les points de trajectoire. Cela signifie que les points de données influençant le calcul d'un vecteur mean shift sont pris dans un voisinage de même taille tout le long de la trajectoire de montée de gradient. L'avantage est que l'estimée en un point a moins de risque d'être influencée par des observations trop éloignées. Nous supposons donc la taille $\mathbf{H}(\mathbf{x})$ du noyau constante pour tous les points de trajectoire $\mathbf{y}^{(j)}$ correspondant au point d'estimation $\mathbf{x}$. La convergence de ce nouvel algorithme de filtrage mean shift pseudo ballon est démontrée dans l'annexe A dans le cas où $\mathbf{H}(\mathbf{x})^T = \mathbf{H}(\mathbf{x})$. La procédure de clustering est définie dans l'algorithme 4. Nous utilisons le minimum des deux bandes passante dans l'étape 2 afin d'éviter le regroupement de deux modes trop éloignés.

Algorithme 4 Algorithme "pseudo balloon mean shift"

Étant donnés $\{\mathbf{x}^{(i)}\}_{i=1,\ldots,n}$, n points dans l'espace de dimension d et $\{\mathbf{z}^{(i)}\}_{i=1,\ldots,n}$ leurs modes associés.

1. Pour $i = 1, \ldots, n$, appliquer l'algorithme de filtrage mean shift en utilisant l'équation (3.30).
2. Regrouper deux modes $\mathbf{z}^{(i)}$ et $\mathbf{z}^{(j)}$ si :

$$\|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\| \leq \min(\|\mathbf{H}(\mathbf{x}^{(i)})\|, \|\mathbf{H}(\mathbf{x}^{(j)})\|) \ .$$

3. Regrouper tous les points $\mathbf{x}^{(i)}$ dont les modes associés appartiennent au même groupe.

La comparaison entre le clustering pseudo balloon mean shift et le clustering utilisant l'estimateur sample point est montrée dans le chapitre suivant.

3.4 Conclusion

Dans ce chapitre nous avons rappelé quelques principes de l'estimateur à noyau et présenté le clustering mean shift. Une nouvelle procédure mean shift, appelée pseudo balloon mean shift, reposant sur l'estimateur balloon a été introduite. Le défaut principal des estimateurs à noyau et des méthodes mean shift est que les résultats obtenus dépendent très fortement des tailles de noyau utilisées. La sélection automatique de la taille des noyaux est un problème complexe faisant toujours l'objet de nombreuses recherches. Elle devient une réelle difficulté si la dimension des données devient grande et l'espace hétérogène, ce qui sera notre cas. Nous dédions le chapitre suivant à la sélection automatique de la taille du noyau pour ce type de données. Les résultats de clustering appliqués à nos séquences dans le but de détecter les clusters en mouvement seront ensuite montrés dans le chapitre 5.

Chapitre 4

Sélection automatique de la taille du noyau pour le partitionnement mean shift

Dans le chapitre précédent, le clustering mean shift a été présenté. Nous avons choisi cette approche car elle ne requiert pas la connaissance *a priori* du nombre de clusters et de leur forme (ou leur distribution). Le mean shift est une technique non paramétrique qui repose sur l'estimation de la densité par un estimateur à noyau. La difficulté est alors de définir la taille du noyau car celle-ci influence beaucoup les résultats de partitionnement.

Nous considérons ici un espace de données hétérogènes de grande dimension. Les données, de dimension d, sont ainsi composées de P parties indépendantes (par exemple la position, le mouvement, la couleur), chacune étant de dimension $d_\rho, \rho = 1 \dots P$ ($\sum_{\rho=1}^{P} d_\rho = d$). Nous emploierons de manière identique les termes d'espace de caractéristiques, de domaine ou de type d'information pour faire référence à chacune des composantes des données. Les données peuvent être multimodales (données hétérogènes), ce qui encourage l'utilisation d'estimateurs à noyau de bande passante variable (estimateur "sample point" ou "balloon"). Cependant, cela complique le calcul de la meilleure taille de noyau, cette dernière devant maintenant être calculée pour chaque point de l'espace.

Plusieurs méthodes existent pour calculer la meilleure bande passante du noyau de l'estimateur. En particulier, un certain nombre de critères statistiques, faisant généralement un compromis entre le biais et la variance, ont été introduits dans la littérature. Ces critères sont néanmoins peu adaptés à des données de grande dimension. Quelques techniques, uniquement dédiées au clustering, préfèrent utiliser un critère de validation des partitions pour choisir la meilleure taille. En particulier, une de ces méthodes a été spécialement conçue pour le clustering mean shift [Comaniciu 01, Comaniciu 03a]. Cette méthode n'est toutefois pas très bien adaptée à nos données hétérogènes.

Nous consacrons ce chapitre à la mise en place d'un algorithme de sélection de la taille du noyau spécialement conçu pour le partitionnement mean shift de données hétérogènes de grande dimension. La méthode proposée a été très largement inspirée par l'algorithme mis en place dans

[Comaniciu 03a]. Elle se base sur un critère de validation des clusters en considérant qu'un bon cluster ne devrait pas changer si la taille du noyau varie légèrement. Afin de prendre en compte l'hétérogénéité des données, la technique proposée est itérative et sélectionne successivement la meilleure bande passante pour chaque espace de caractéristiques.

Le chapitre est organisé comme suit. Dans un premier temps (section 4.1) nous introduisons rapidement les méthodes existantes de calcul de la bande passante du noyau. Nous introduisons ensuite, dans la section 4.2, la méthode proposée, et la validons (section 4.3) dans le contexte de la segmentation d'images couleur. Nous terminons cette dernière section en montrant comment la technique proposée s'applique à la détection de cluster en mouvement et présentons un premier résultat de détection.

4.1 État de l'art sur la sélection de la taille d'un noyau

Cette première section présente rapidement les méthodes existantes pour la sélection automatique de la taille du noyau fixe ou variable d'un estimateur. Nous séparons les critères statistiques et les critères de validation d'une partition.

4.1.1 Méthodes statistiques pour la sélection du noyau

Un bon estimateur est un estimateur non biaisé et de faible variance. La définition du biais et de la variance, pour l'estimateur à noyau fixe (chapitre 3), conduit à l'analyse suivante : la valeur absolue du biais augmente et la variance diminue quand la taille du noyau $\mathbf{H}$ augmente. La qualité de l'estimation va donc fortement dépendre du choix du noyau et un compromis entre le biais et la variance doit souvent être trouvé. Minimiser le biais et la variance de l'estimateur revient à minimiser le risque quadratique local (MSE). Le MSE (équation (3.11)),

$$\begin{aligned} \mathrm{MSE}(\mathbf{x}) &= E\big[(\widehat{f}(\mathbf{x}) - f(\mathbf{x}))^2\big] \\ &= \mathrm{var}(\widehat{f}(\mathbf{x})) + \big[\mathrm{Biais}(\widehat{f}(\mathbf{x}))\big]^2 , \end{aligned}$$

mesure localement la distance entre l'estimateur à noyau $\widehat{f}$ et la densité f. Une mesure globale, appelée risque quadratique intégré (MISE), peut être calculée en intégrant la distance sur l'ensemble des données. En ignorant les termes d'ordre supérieur, on obtient finalement le risque quadratique intégré asymptotique (AMISE). Le MSE contient des termes fonction de la densité et de sa dérivée seconde, et les deux autres mesures sont fonction de la dérivée seconde uniquement. Ces mesures ne peuvent donc pas être utilisées directement puisqu'elles dépendent toutes de la densité que l'on cherche à estimer.

Un certain nombre de critères ont été mis en place pour minimiser ces mesures. On peut notamment citer la règle du pouce ("rule of thumb") [Silverman 86], les règles dites "plug-in" [Park 90, Sheather 91], ou encore les règles de validation croisée [Park 90] [Simonoff 96, p.46]. Le principe de la règle du pouce est de supposer que f est la densité d'une variable aléatoire, généralement gaussienne, ce qui permet de calculer la densité et sa dérivée seconde. Cette règle échoue dès lors que l'on s'écarte d'une distribution gaussienne ou si la vraie densité est multi-modale. Les règles "plug-in" s'inspirent de la règle du pouce. On commence par définir une bande passante initiale permettant d'avoir une première estimation, appelée "pilote", de l'estimateur à noyau. La densité est supposée égale à ce pilote et est injectée dans la MSE ou la MISE qui est ensuite minimisée. Les règles précédentes s'adaptent très bien aux estimateurs à

noyau variable. Un algorithme permettant la mise en place d'une loi plug-in pour des données mono-dimensionnelles peut être trouvée dans [Comaniciu 02]. Un algorithme existe aussi pour des données multi-variées [Wand 95, p.108] mais sa mise en œuvre est relativement complexe. À notre connaissance, il n'y a pas de techniques dans la littérature adaptant les règles du pouce ou plug-in à des données hétérogènes de grande dimension. Les dernières méthodes, souvent utilisées pour calculer la taille du noyau, sont celles de type validation croisée. Le principe est de sélectionner, parmi un ensemble de bandes passantes possibles, celle qui minimise la somme des distances entre chaque point et son estimé obtenue en utilisant tous les autres points. L'application de la validation croisée devient rapidement coûteuse dès lors que l'ensemble des données est grand.

Nous venons de présenter les critères les plus couramment utilisés pour calculer la taille du noyau d'un estimateur. Cependant, tous ces critères ne s'adaptent pas bien au type de données que nous considérons. En effet, bien que ces méthodes assurent que l'estimateur aura de bonnes propriétés statistiques, on ne peut être sûr qu'elles engendreront le meilleur partitionnement des données.

4.1.2 Méthodes basées sur l'évaluation des partitions

Des méthodes de calcul de la taille du noyau dédiées au clustering ont également été proposées, reposant sur une validation des clusters obtenus. Plusieurs techniques permettant de valider un cluster existent [Milligan 85]. Un bon cluster peut par exemple être un cluster compact (les données appartenant au cluster doivent être très proches les unes des autres), isolé (un cluster doit être différent des autres) ou stable. La stabilité est un critère basé sur la vision humaine : les clusters devraient pouvoir être perçus pour un nombre important d'échelles différentes. Ce critère a par exemple été utilisé dans [Leung 00] ou [Fukunaga 90]. Dans [Fukunaga 90], les données sont partitionnées pour plusieurs noyaux de différentes tailles. Le noyau finalement sélectionné est celui pour lequel le nombre de clusters obtenu est le plus proche des nombres obtenus pour les noyaux voisins. Un algorithme du même type a été proposé par Comaniciu dans [Comaniciu 03a].

La méthode de [Comaniciu 03a] est spécialement dédiée au clustering mean shift. L'idée de base est qu'une bonne partition ne devrait pas changer si une petite variation est appliquée au noyau de l'estimateur. En particulier, si un cluster peut être représenté par une loi normale, la moyenne et la variance de cette loi devraient rester constantes si le noyau ne varie que légèrement. Là encore, le clustering est effectué pour plusieurs tailles de noyau prédéfinies. Chaque partition obtenue est alors représentée par une loi normale et la stabilité est ensuite évaluée en chaque point : les lois normales obtenues pour des tailles voisines sont comparées à l'aide d'une mesure de divergence. Dans toute la suite, les tailles de noyau prédéfinies seront classées par ordre croissant et on utilisera le terme échelle pour se référer à l'indice d'une des tailles dans l'ensemble prédéfini. Le noyau final en un point est, dans [Comaniciu 03a], la covariance de la loi normale ayant donné la plus faible divergence.

4.2 Sélection de la taille du noyau dans des espaces hétérogènes de grande dimension

Cette section a pour but la mise en place d'un algorithme de sélection automatique de la taille du noyau pour des espaces de données hétérogènes de grande dimension. Plutôt que d'utiliser des critères statistiques, peu adaptés aux données considérées, nous avons préféré définir la meilleure taille de noyau en chaque point (nous utilisons des estimateurs à noyau variable) comme celle permettant d'obtenir la partition la plus stable. La meilleure taille de noyau $\Upsilon(\mathbf{x}^{(i)})$, en un point $\mathbf{x}^{(i)}$, est choisie parmi un ensemble de B matrices prédéfinies $\{\mathbf{H}^{(b)}, b = 1, \ldots, B\}$ (contrairement à [Comaniciu 03a] où la meilleure taille est la covariance de la partition la plus stable ; voir annexe B).

Nous organisons cette section de la façon suivante. Nous commençons par présenter le principe de la méthode proposée, puis montrons comment est prise en compte l'hétérogénéité des espaces de caractéristiques des données. Nous terminerons en énonçant l'algorithme final de sélection.

4.2.1 Principe de l'algorithme de sélection

Nous cherchons à trouver la meilleure bande passante en chaque point. La méthode mise en place dans ce chapitre est largement inspirée de l'algorithme de [Comaniciu 03a]. Elle se décompose en deux grandes étapes (figure 4.1).

4.2.1.1 Première étape : "évaluation de la taille pour les partitions"

La première est appelée "évaluation de la taille pour les partitions". Son principe est le suivant. Un algorithme de clustering mean shift est appliqué aux données pour chacune des matrices prédéfinies. Cela conduit à la création de plusieurs partitions, pour chaque échelle b de cet ensemble. Chaque cluster obtenu est alors représenté de manière paramétrique par une loi normale. La loi normale du cluster u à l'échelle b sera notée $\mathcal{N}(\mu_u^{(b)}, \mathbf{\Sigma}_u^{(b)})$. Notons $\mathcal{C}_u^{(b)}$ l'ensemble des points appartenant au cluster u à l'échelle b :

$$\mathcal{C}_u^{(b)} = \{i / c(i,b) = u\} \ , \tag{4.1}$$

en rappelant que la fonction c associe un point i au cluster auquel il appartient à l'échelle b. Si le point i appartient à la partition u après le clustering mean shift appliqué avec un noyau de taille $\mathbf{H}^{(b)}$, alors $c(i,b) = u$.

La moyenne $\mu_u^{(b)}$ est donnée par :

$$\mu_u^{(b)} = \frac{1}{|\mathcal{C}_u^{(b)}|} \sum_{i \in \mathcal{C}_u^{(b)}} \mathbf{x}^{(i)} \ , \tag{4.2}$$

et la covariance $\mathbf{\Sigma}_u^{(b)}$ par :

$$\mathbf{\Sigma}_u^{(b)} = \frac{1}{|\mathcal{C}_u^{(b)}|} \sum_{i \in \mathcal{C}_u^{(b)}} (\mathbf{x}^{(i)} - \mu_u^{(b)})(\mathbf{x}^{(i)} - \mu_u^{(b)})^T \ . \tag{4.3}$$

L'utilisation des formules traditionnelles précédentes (équations (4.2) et (4.3)) pour le calcul des covariances ne permet pas de prendre en compte la forme réelle d'un cluster, comme par exemple ses éventuelles queues (points éloignés du mode) non gaussiennes. Pour palier à ce problème, d'autres formules ont été mises en place dans [Comaniciu 03a]. Cependant, comme la covariance proposée peut être négative, nous n'avons pas souhaité les utiliser. Ainsi, dans la suite, nous considérons que toutes les moyennes et les covariances sont calculées par les formules traditionnelles (équations (4.2) et (4.3)). Nous pensons néanmoins qu'il sera utile dans des travaux futurs d'étudier d'autres mesures d'estimation robuste de la covariance en s'appuyant, par exemple, sur [Pena 01, Wang 02].

Après avoir calculé l'ensemble des lois normales, chaque point est associé à la distribution du cluster auquel il appartient. Le point i est ainsi associé à la distribution $p_i^{(b)} = \mathcal{N}(\mu_{c(i,b)}^{(b)}, \mathbf{\Sigma}_{c(i,b)}^{(b)})$ à l'échelle b.

4.2.1.2 Deuxième étape :"évaluation de la taille pour les données"

La deuxième étape de l'algorithme s'appelle "évaluation de la taille pour les données". Elle consiste à appliquer un critère de stabilité permettant de trouver le meilleur noyau en chaque point. Pour chaque point i, on compare les clusters $c(i, b)$, pour $b = 1 \dots B$ pour en extraire le plus stable. La sélection du meilleur noyau reposant sur un critère de stabilité, la comparaison entre clusters se fait en calculant une mesure de divergence entre clusters aux échelles voisines. Plus précisément, une mesure de divergence entre les lois normales des clusters voisins est calculée et comparée aux mesures obtenues pour les autres échelles. La meilleure échelle sera celle de divergence minimale.

De nombreuses mesures de divergence entre distributions existent. Une des plus connues est la distance de Kullback-Leibler $\mathcal{I}$ [Kullback 51]. Soit X un ensemble discret de variable aléatoire et p_1 et p_2 deux distributions de probabilité définies sur X. La divergence de Kullback-Leibler est définie par :

$$\mathcal{I}(p_1, p_2) = \sum_{x \in X} p_1(x) log_2 \frac{p_1(x)}{p_2(x)} \ . \tag{4.4}$$

Cette mesure n'est pas symétrique ($\mathcal{I}(p_1, p_2) \neq \mathcal{I}(p_2, p_1)$). Dans [Jeffreys 46] une distance symétrique J, issue de $\mathcal{I}$, a été proposée :

$$J(p_1, p_2) = \mathcal{I}(p_1, p_2) + \mathcal{I}(p_2, p_1) \ . \tag{4.5}$$

Cette mesure, bien que symétrique, ne respecte pas l'inégalité triangulaire. Il ne s'agit donc pas d'une distance au sens stricte mathématique du terme. Une mesure de divergence correspondant à une distance est la divergence de Jensen-Shannon, introduite dans [Lin 91]. Elle permet de comparer plusieurs distributions et est définie par :

$$JS_\pi(p_1, \dots, p_r) = E(\sum_i \pi_i p_i) - \sum_i \pi_i E(p_i) \ , \tag{4.6}$$

où r est le nombre de distributions, $\{\pi_i\}$ est une séquence de poids, chacun des poids $\{\pi_i\}$ étant associé à la distribution p_i, et $E(p_i)$ l'entropie de Shannon :

$$E(\{p_i\}) = -\sum_{x \in X} p_i(x) log_2(p_i(x)) \ . \tag{4.7}$$

Il a été montré dans [Comaniciu 03b] que la divergence de Jensen-Shannon entre deux distributions normales est égale à la distance de Bhattacharyya, couramment utilisée en vision par ordinateur pour comparer deux distributions. Nous utilisons cette mesure de divergence pour comparer les distributions normales représentant les clusters. Étant données r distributions normales $p_j \ , \ j = 1, \ldots, r$, caractérisées par leur moyenne μ_j et leur covariance $\mathbf{\Sigma}_j$, la divergence de Jensen-Shannon se réécrit ainsi :

$$JS(p_1 \ldots p_r) = \frac{1}{2}\log\frac{|\frac{1}{r}\sum_{j=1}^{r}\mathbf{\Sigma}_j|}{\sqrt[r]{\prod_{j=1}^{r}|\mathbf{\Sigma}_j|}} + \frac{1}{2}\sum_{j=1}^{r}(\mu_j - \frac{1}{r}\sum_{j=1}^{r}\mu_j)^T(\sum_{j=1}^{r}\mathbf{\Sigma}_j)^{-1}(\mu_j - \frac{1}{r}\sum_{j=1}^{r}\mu_j) \ . \quad (4.8)$$

Une mesure de divergence est calculée pour chaque point, à chaque échelle. La distribution du cluster $p_i^{(b)}$ contenant le point i est comparée avec les distributions des clusters aux deux échelles voisines $p_i^{(b-1)}$ et $p_i^{(b+1)}$ ($r = 3$). La meilleure échelle

$$b^* = \text{argmin}_b \ JS(p_i^{(b-1)}, p_i^{(b)}, p_i^{(b+1)}) \quad (4.9)$$

pour ce point sera celle qui minimise la divergence de Jensen-Shannon, définie par :

$$\begin{aligned} JS(p_i^{(b-1)}, p_i^{(b)}, p_i^{(b+1)}) = \frac{1}{2}\log\frac{|\frac{1}{3}\sum_{j=b-1}^{b+1}\mathbf{\Sigma}_{c(i,b)}^{(j)}|}{\sqrt[3]{\prod_{j=b-1}^{b+1}|\mathbf{\Sigma}_{c(i,b)}^{(j)}|}} + \\ \frac{1}{2}\sum_{j=b-1}^{b+1}(\mu_{c(i,b)}^{(j)} - \frac{1}{3}\sum_{j=b-1}^{b+1}\mu_{c(i,b)}^{(j)})^T(\sum_{j=b-1}^{b+1}\mathbf{\Sigma}_{c(i,b)}^{(j)})^{-1}(\mu_{c(i,b)}^{(j)} - \frac{1}{3}\sum_{j=b-1}^{b+1}\mu_{c(i,b)}^{(j)}) \ . \end{aligned} \quad (4.10)$$

La meilleure bande passante pour ce point sera finalement :

$$\mathbf{\Upsilon}(\mathbf{x}^{(i)}) = \mathbf{H}^{(b^*)} \ . \quad (4.11)$$

Dans [Comaniciu 03a], la meilleure bande passante pour le point i est la covariance du cluster le plus stable : $\mathbf{\Upsilon}(\mathbf{x}^{(i)}) = \mathbf{\Sigma}_{c(i,b^*)}^{(b^*)}$. Avec un tel choix, la taille finale peut ne pas être comprise entre les plus faible et plus forte bandes passantes prédéfinies. Nous discutons plus en détail du critère de sélection dans l'annexe B.

Comme nous allons le voir dans la prochaine sous-section, nous proposons un algorithme itératif permettant de trouver successivement la meilleure taille de noyau pour chaque espace de caractéristiques composant les données. À chaque itération, nous appliquerons les deux étapes précédentes résumées sur la figure 4.1.

4.2.2 Sélection itérative

En présence de données hétérogènes de grande dimension, l'ensemble des noyaux prédéfinis peut être de grande taille. En effet, si les données varient de manières indépendantes

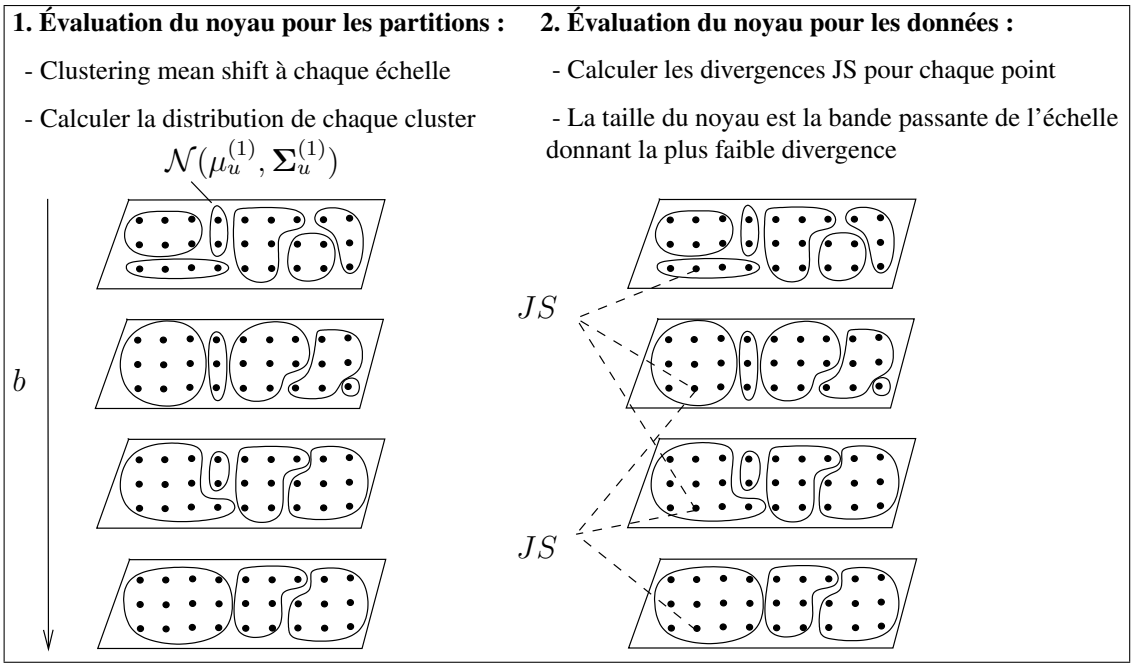


FIG. 4.1: Schéma représentant une itération de notre algorithme.

pour les différentes caractéristiques, le noyau associé à chacun de ces espaces doit être choisi indépendamment. Il est alors possible que le meilleur noyau ne soit pas obtenu pour la même échelle dans tous les espaces. La façon la plus triviale de tenir compte des différences de chaque espace est de définir l'ensemble des noyaux prédéfinis en combinant tous les noyaux possibles de chaque caractéristique. Cependant l'ensemble prédéfini découlant d'un tel choix peut vite devenir très grand : si B_ρ noyaux sont définis pour chaque espace ρ, l'ensemble prédéfini est de taille $\prod_{\rho=1}^{P} B_\rho$. Il faudra alors réaliser $\prod_{\rho=1}^{P} B_\rho$ clustering mean shift avant de pouvoir calculer la taille finale du noyau en chaque point. Ainsi, si le nombre de caractéristiques et le nombre de noyaux définis pour chacunes d'elles est important, utiliser directement l'algorithme défini précédemment (figure 4.1) devient trop coûteux. Nous proposons donc une méthode différente, basée sur une sélection itérative du noyau pour chaque domaine. Si B_ρ noyaux sont définis pour chaque espace ρ, cette technique permet de ne réaliser que $\sum_{\rho=1}^{P} B_\rho$ partitionnements mean shift, tout en permettant de sélectionner le meilleur noyau indépendamment pour chaque espace.

Supposons que nous cherchions à sélectionner la taille du noyau pour le premier domaine. Pour tous les autres espaces, nous fixons temporairement les bandes passantes notées $\tilde{\mathbf{H}}_\rho, \rho = 2, \ldots, P$. Ces matrices sont constantes pour toutes les échelles (pour tout $b = 1 \ldots B_\rho$) et égales à la moyenne :

$$\tilde{\mathbf{H}}_\rho = \frac{1}{B_\rho} \sum_{b=1}^{B_\rho} \mathbf{H}_\rho^{(b)}, \rho > 1 \ . \tag{4.12}$$

La méthode de sélection de la taille du noyau décrite dans la sous-section précédente (figure 4.1) est alors appliquée pour l'ensemble suivant :

$$\{\tilde{\mathbf{H}}^{(b)} = \mathrm{diag}[\mathbf{H}_1^{(b)}, \tilde{\mathbf{H}}_2, \ldots, \tilde{\mathbf{H}}_P], b = 1, \ldots, B_1\} \ .$$

Le noyau final $\boldsymbol{\Upsilon}_1(\mathbf{x}^{(i)})$ est ainsi obtenu en chaque point $\mathbf{x}^{(i)}$. La même procédure est alors réappliquée pour tous les autres domaines, en utilisant le noyau obtenu pour les espaces déjà étudiés. Ce dernier n'est cependant plus fixe puisqu'il est défini en chaque point :

$$\tilde{\mathbf{H}}^{(b)}(\mathbf{x}^{(i)}) = \mathrm{diag}[\boldsymbol{\Upsilon}_1(\mathbf{x}^{(i)}), \ldots, \boldsymbol{\Upsilon}_{\rho-1}(\mathbf{x}^{(i)}), \mathbf{H}_{\rho}^{(b)}, \tilde{\mathbf{H}}_{\rho+1} \ldots \tilde{\mathbf{H}}_P] \ . \tag{4.13}$$

Il est donc nécessaire d'utiliser un estimateur à noyau variable pour les procédures mean shift. Dans le chapitre précédent (chapitre 3), deux algorithmes de clustering mean shift utilisant des estimateurs à noyau variable ont été présentés. Comme nous considérons des données de dimension supérieure à 3, nous préférons utiliser l'estimateur "balloon", et appliquerons donc l'algorithme dit "pseudo balloon mean shift". Néanmoins, exactement la même méthode de sélection pourrait être obtenue avec le mean shift basé sur l'estimateur "sample point".

L'algorithme de sélection décrit dans la sous-section 4.2.1 est donc appliqué de manière itérative pour chaque espace de caractéristiques en utilisant le partitionnement "pseudo balloon mean shift". Cela permet d'obtenir, successivement pour chaque domaine, la meilleure taille de noyau (parmi l'ensemble des tailles prédéfinies) en chaque point de données.

L'algorithme itératif complet est présenté dans l'algorithme 5. Nous rappelons que pour chaque espace de caractéristiques, la taille du noyau en chaque point est choisie en deux étapes. La première correspond à l'évaluation de la taille pour les partitions et la seconde à l'évaluation de la taille pour les données.

4.3 Application à la segmentation d'images couleur

Dans cette section nous montrons l'application de la méthode de sélection des tailles de noyaux à la segmentation d'images couleur. Afin d'obtenir la segmentation finale, nous appliquons une nouvelle fois l'algorithme de clustering "pseudo balloon mean shift" avec les noyaux variables sélectionnés. Les données étant l'ensemble des pixels de l'image, un vecteur de caractéristiques de 5 dimensions (2 pour la position, 3 pour la couleur) est associé à chaque point. Nous considérons ici chaque dimension indépendante. Ainsi, 5 domaines ou espaces de caractéristiques, chacun de dimension 1, composent les données. Les domaines sont classés dans l'ordre suivant : abscisse, ordonnée, canal rouge, canal vert et canal bleu. L'ordre défini quel domaine est étudié à chaque itération de l'algorithme. Dans la sous-section 4.3.3 nous discuterons de l'influence que peut avoir cet ordre sur les résultats.

Pour toutes les expérimentations de cette section, le même ensemble de bandes passantes prédéfinies est utilisé. Pour chaque domaine, 9 bandes passantes sont définies, leur valeur étant comprise entre 10 et 30. Sur les images de segmentation présentées, la couleur de chaque pixel correspond à la couleur du mode vers lequel il a convergé.

Les particularités de notre algorithme sont successivement validées. Nous validons tout d'abord la sélection itérative en appliquant notre méthode pour 5 domaines (chacun de dimension 1) et pour 1 domaine de dimensions 5. Ce dernier cas revient à supposer que les données évoluent de la même façon pour les cinq dimensions. Nous comparons ensuite l'algorithme de clustering "pseudo balloon mean shift" mis en place dans le chapitre précédent avec l'algorithme de clustering mean shift basé sur l'estimateur sample point. Le choix final de la meilleure taille de

Algorithme 5 Algorithme itératif d'estimation de la taille du noyau

Étant données un ensemble de B_ρ bandes passantes prédéfinies $\{\mathbf{H}_\rho^{(b)}, b = 1 \dots B\}$ pour chaque espace de caractéristiques ρ.

Pour $\rho = 1, \dots, P$

- Évaluer la taille pour les partitions : pour tout $b = 1, \dots, B$
  1. Pour tout $\rho' = \rho + 1, \dots, P$, calculer $\tilde{\mathbf{H}}_{\rho'}$:

$$\tilde{\mathbf{H}}_{\rho'} = \frac{1}{B_{\rho'}} \sum_{b=1}^{B_{\rho'}} \mathbf{H}_{\rho'}^{(b)} \quad . \tag{4.14}$$

  2. Définir, pour $i = 1, \dots, n$,

$$\{\tilde{\mathbf{H}}^{(b)}(\mathbf{x}^{(i)}) = \mathrm{diag}[\mathbf{\Upsilon}_1(\mathbf{x}^{(i)}), \dots, \mathbf{\Upsilon}_{\rho-1}(\mathbf{x}^{(i)}), \mathbf{H}_\rho^{(b)}, \tilde{\mathbf{H}}_{\rho+1} \dots \tilde{\mathbf{H}}_P], b = 1, \dots, B_\rho\}.$$

  3. Partitionner les données en utilisant l'algorithme "pseudo balloon mean shift" (algorithme 4). Ainsi, $n^{(b)}$ clusters, notés $\mathcal{C}_u^{(b)}, u = 1 \dots n^{(b)}$, sont obtenus. On introduit la fonction c qui associe le point i à son cluster : $c(i, b) = u \quad \Leftrightarrow \mathbf{x}^{(i)} \in \mathcal{C}_u^{(b)}$.
  4. Calculer la loi normale $\mathcal{N}(\mu_u^{(b)}, \mathbf{\Sigma}_u^{(b)})$ caractérisant chaque cluster avec :

$$\mu_u^{(b)} = \frac{1}{|\mathcal{C}_u^{(b)}|} \sum_{i \in \mathcal{C}_u^{(b)}} \mathbf{x}^{(i)} = \begin{bmatrix} \mu_{u,1}^{(b)} \\ \vdots \\ \mu_{u,P}^{(b)} \end{bmatrix} \quad , \tag{4.15}$$

  et

$$\mathbf{\Sigma}_u^{(b)} = \frac{1}{|\mathcal{C}_u^{(b)}|} \sum_{i \in \mathcal{C}_u^{(b)}} (\mathbf{x}^{(i)} - \mu_u^{(b)})(\mathbf{x}^{(i)} - \mu_u^{(b)})^T = \mathrm{diag}[\mathbf{\Sigma}_{u,1}^{(b)}, \dots, \mathbf{\Sigma}_{u,P}^{(b)}] \quad . \tag{4.16}$$

  5. Pour toutes les échelles b, associer chaque point i à la moyenne $\mu_{c(i,b),\rho}^{(b)}$ et la covariance $\mathbf{\Sigma}_{c(i,b),\rho}^{(b)}$ du cluster auquel il appartient. La distribution correspondante est notée $p_{i,\rho}^{(b)}$.

- Évaluer la taille pour les données : pour chaque point $\mathbf{x}^{(i)}$
  1. Sélectionner l'échelle b^* donnant la partition la plus stable en résolvant :

$$b^* = \mathrm{argmin}_{r=2,\dots,B-1} \mathrm{JS}(p_{i,\rho}^{(r-1)}, p_{i,\rho}^{(r)}, p_{i,\rho}^{(r+1)}) \tag{4.17}$$

  où JS est la divergence de Jensen-Shannon définie par l'équation 4.10.
  2. La meilleure bande passante $\Upsilon_\rho(\mathbf{x}^{(i)})$ est $\mathbf{H}_\rho^{(b^*)}$.

noyau est analysé dans l'annexe B. En effet, dans notre méthode la taille finale est celle donnant le cluster le plus stable tandis que dans [Comaniciu 03a], la taille finale est la covariance de la loi normale calculée sur le cluster le plus stable.

4.3.1 Validation de l'approche itérative

Nous commençons par valider la sélection itérative sur plusieurs exemples. Nous appliquons d'une part l'algorithme de sélection (algorithme 5) en considérant que les données sont composées de cinq espaces de caractéristiques de dimension 1 (méthode itérative) et d'autre part en considérant qu'elles sont simplement composées d'un seul espace de dimension 5 (méthode non itérative).

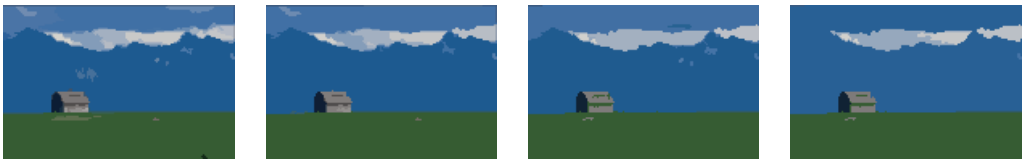
Les premiers résultats ont été obtenus en segmentant l'image d'un chalet. La figure 4.2 montre le résultat de la segmentation finale pour l'approche non itérative (figure (b)) et pour l'approche itérative (figure (c)). Avec la méthode non itérative 31 clusters ont été obtenus alors que la technique itérative en a donné 21. À la fin de la segmentation, le ciel et les montagnes se retrouvent fusionnés si l'on choisit une approche non itérative. Des différences sont aussi visibles dans les montagnes contenant moins de segments avec la méthode non itérative.

a) b) c)

FIG. 4.2: Validation de l'approche itérative sur l'image d'un chalet. a) Image originale ; b) Résultat de la segmentation avec une approche non itérative ; c) Résultat avec la méthode itérative.

Sur la figure 4.3, nous montrons l'évolution des partitionnements mean shift ("pseudo-balloon mean shift") calculés pour les différentes tailles de noyau prédéfinies au cours de l'algorithme non itératif. L'évolution pour l'algorithme itératif est visible sur la figure 4.4. Pour ce deuxième cas, les partitionnements sont calculés pour les différentes tailles de noyau et pour les différents espaces de caractéristiques. Avec l'approche non itérative, les tailles de noyaux augmentent simultanément pour les différents domaines, ce qui entraîne une perte rapide de nombreux détails. Au contraire la sélection itérative permet une meilleure stabilité entre les partitionnements obtenus à des échelles successives.

Sur la figure 4.5, un deuxième résultat de segmentation est présenté sur une image montrant une main. Les deux approches donnent des résultats assez similaires. Néanmoins, la méthode non itérative donne 13 clusters contre 37 pour l'approche itérative. La bague et les ongles ne sont effectivement pas détectés avec l'approche non itérative car, étant choisies simultanément avec la position, les tailles de noyaux finalement sélectionnées pour les espaces de couleur sont trop grandes.



$b = 2$ $b = 4$ $b = 6$ $b = 8$

FIG. 4.3: Évolution des segmentations avec les échelles (correspondant aux indices des bandes passantes) pour l'approche non itérative.

Un dernier résultat est présenté sur la figure 4.6. De nombreuses différences sont visibles sur le taureau. En particulier, l'algorithme itératif permet de conserver plus de détails sur la tête du taureau. L'algorithme itératif donne 136 clusters et l'algorithme non itératif 130. Le nombre de clusters est donc similaire avec les deux approches. En effet, on peut remarquer que même si l'algorithme itératif conserve plus de détails sur le taureau, la méthode non itérative produit quelques petits clusters supplémentaires (moins importants) dans l'herbe.

Les trois résultats précédents valident le choix d'une sélection itérative des noyaux. Une telle approche permet de garder des détails importants qui sont perdus si tous les espaces de caractéristiques sont étudiés simultanément.

4.3.2 Validation du clustering "pseudo balloon mean shift"

Dans le chapitre précédent, nous avons introduit un nouvelle méthode de clustering mean shift, appelée "pseudo balloon mean shift". Nous proposons ici de comparer cet algorithme avec l'algorithme de clustering mean shift appelé "variable bandwidth mean shift" [Comaniciu 01]. Nous avons choisi d'utiliser l'estimateur ballon en raison de la dimension des données. Dans [Terrell 92], il a été montré que l'estimateur "balloon" est performant dès lors que la dimension des données est supérieure à 3. Nous comparons les deux algorithmes mean shift de la façon suivante. Pour définir les tailles des noyaux en chaque point de l'espace des données, l'algorithme itératif de sélection de la taille (utilisant le "pseudo balloon mean shift") décrit dans ce chapitre est appliqué. À partir des noyaux obtenus, les deux différents algorithmes mean shift sont appliqués, donnant deux résultats de segmentation différents.

Nous analysons une nouvelle fois les résultats sur l'image du chalet. La figure 4.7 montre les deux résultats de clustering. Les partitions finales sont très proches avec les deux méthodes. Quelques légères différences sont visibles dans les nuages. Le mean shift basé sur l'estimateur balloon génére 31 clusters et le mean shift utilisant l'estimateur sample point en donne 29.

Le deuxième résultat est montré sur l'image de la main (figure 4.8). Les segmentations finales sont de nouveau très proches avec les deux estimateurs. L'algorithme "variable bandwidth mean shift" [Comaniciu 01] donne 35 partitions et l'algorithme "pseudo balloon mean shift" 37. La différence est principalement visible sur l'index de la main.

Un dernier résultat est présenté sur l'image du taureau à la figure 4.9. Contrairement aux deux

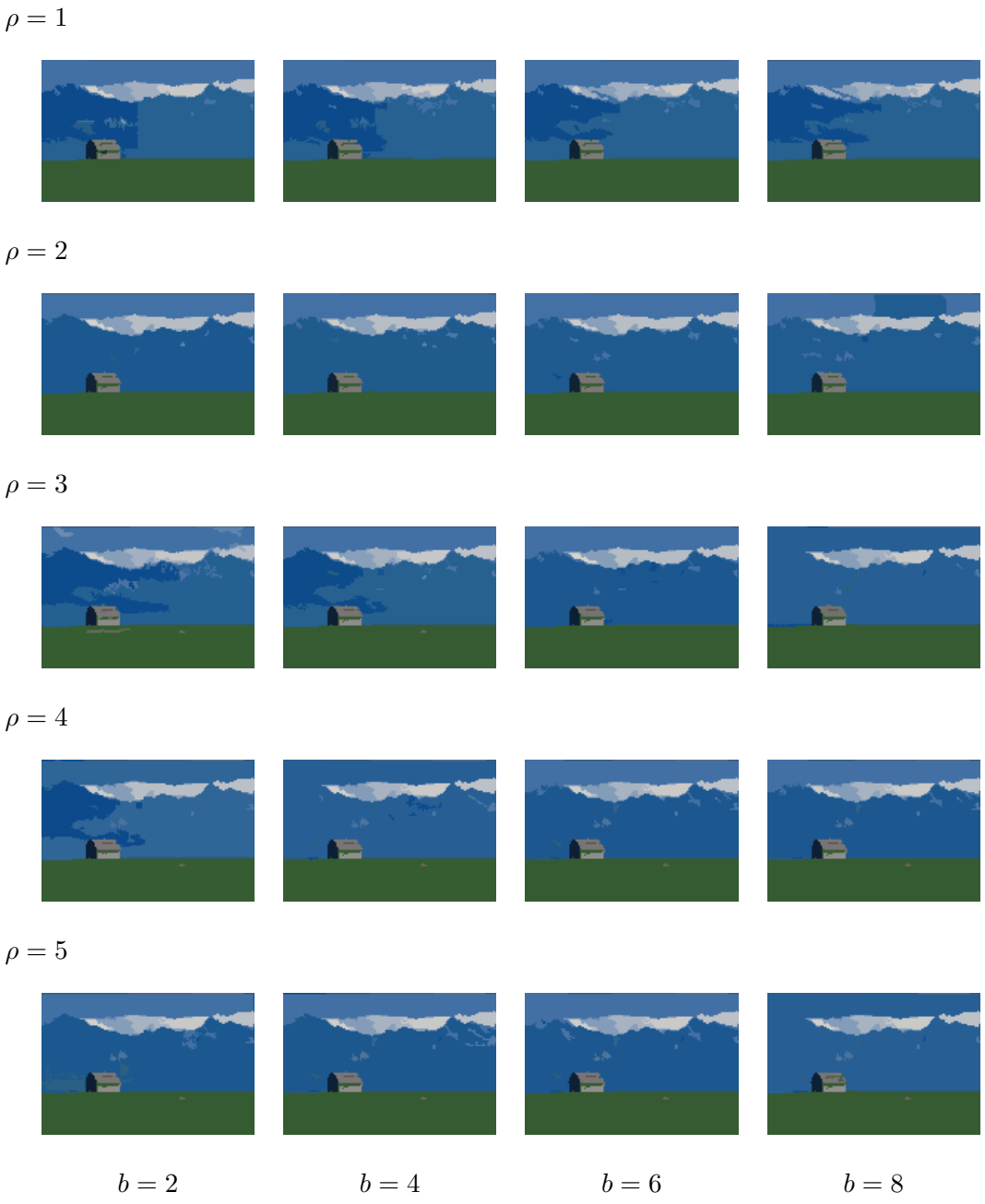


FIG. 4.4: Évolution des segmentations avec les échelles (correspondant aux indices des bandes passantes) et les espaces de caractéristiques pour l'approche itérative.

résultats précédents, de nombreuses différences apparaissent entre les deux estimateurs. L'estimateur sample point permet d'obtenir 128 clusters et l'estimateur balloon 136. Avec le nouveau mean shift introduit au chapitre 3, plus de détails sont conservés sur la tête du taureau. De plus, cette méthode permet d'obtenir ,sur le reste du corps du taureau, des clusters plus nets que le variable bandwidth mean shift basé sur l'estimateur sample point.

a) b) c)

FIG. 4.5: Validation de l'approche itérative sur l'image d'une main. a) Image originale ; b) Résultat de la segmentation avec une approche non itérative ; c) Résultat avec la méthode itérative.

a) b) c)

FIG. 4.6: Validation de l'approche itérative sur l'image d'un taureau. a) Image originale ; b) Résultat de la segmentation avec une approche non itérative ; c) Résultat avec la méthode itérative.

a) b) c)

FIG. 4.7: Validation du clustering "pseudo balloon mean shift" sur l'image d'un chalet. a) Image originale ; b) Résultat de la segmentation avec l'algorithme "variable bandwidth mean shift" ; c) Résultat avec l'algorithme "pseudo balloon mean shift".

En conclusion, les résultats précédents montrent, expérimentalement, la validité du nouvel algorithme mean shift que nous avons mis en place. En effet, les résultats sont très proches de ceux obtenus avec le clustering mean shift basé sur l'estimateur sample point introduit dans [Comaniciu 01], mais la nouvelle méthode s'adapte mieux à des données de dimension supérieure à trois et permet de conserver, après segmentation, plus de détails (qui peuvent potentiellement être importants).

a) b) c)

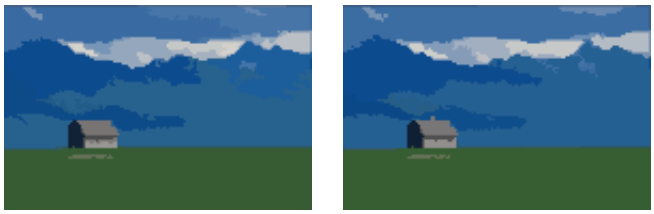
FIG. 4.8: Validation du clustering "pseudo balloon mean shift" sur l'image d'une main. a) Image originale ; b) Résultat de la segmentation avec l'algorithme "variable bandwidth mean shift" ; c) Résultat avec l'algorithme "pseudo balloon mean shift".

a) b) c)

FIG. 4.9: Validation du clustering "pseudo balloon mean shift" sur l'image d'un taureau. a) Image originale ; b) Résultat de la segmentation avec l'algorithme "variable bandwidth mean shift" ; c) Résultat avec l'algorithme "pseudo balloon mean shift".

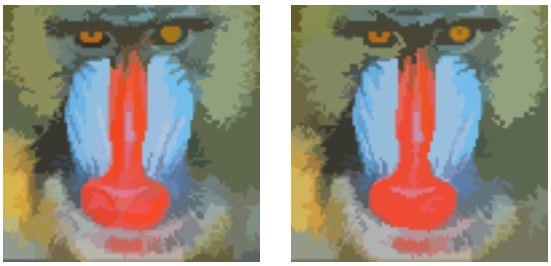
4.3.3 Ordre des espaces de caractéristiques

Pour terminer l'étude expérimentale de l'algorithme de sélection, nous étudions dans cette sous-section la question de l'ordre dans lequel les domaines sont étudiés. Nous montrons sur les figures 4.10 et 4.11 que l'ordre a effectivement une légère influence sur les résultats. Les résultats de ces deux figures ont été obtenus en utilisant 9 bandes passantes prédéfinies comprises entre 3 et 20. Nous ne pouvons pas, à partir de ces deux figures, conclure sur les meilleurs ordres à utiliser. En effet, les résultats de segmentation dépendent de l'application visée. Cependant, les différents tests réalisés pour la segmentation d'images, et pour la détection de clusters en mouvement, nous conduisent à penser qu'il est préférable de commencer par traiter les domaines les plus bruités, c'est-à-dire ceux pour lesquels une variation de la bande passante a une influence non négligeable sur les résultats. C'est ce critère que nous utiliserons dans notre algorithme de détection des objets en mouvement. Il est évalué en calculant, sur l'ensemble des données, l'écart-type des valeurs pour chaque domaine. L'espace de caractéristiques donnant le plus fort écart-type sera traité en premier. Bien entendu, ce critère n'a pas de sens pour la position. Pour tous les résultats présentés dans la suite, la position sera étudiée avant les autres caractéristiques, classées, elles, en fonction du bruit.



a) b)

FIG. 4.10: Ordre de traitement des espaces de caractéristiques. Résultats sur l'image d'un chalet du clustering "pseudo balloon mean shift" avec les bandes passantes obtenues en traitant les domaines dans les ordres suivants : a) abscisse, ordonnée, canal rouge, canal vert, canal bleu ; b) canal bleu, canal vert, canal rouge, ordonnée, abscisse.



a) b)

FIG. 4.11: Ordre de traitement des espaces de caractéristiques. Résultats sur l'image d'un singe du clustering "pseudo balloon mean shift" avec les bandes passantes obtenues en traitant les domaines dans les ordres suivants : a) abscisse, ordonnée, canal rouge, canal vert, canal bleu ; b) canal bleu, canal vert, canal rouge, ordonnée, abscisse.

4.4 Application à la détection d'objets en mouvement

Dans cette dernière section, l'application du clustering à la détection de clusters en mouvement est expliquée. Nous montrons simplement un premier résultat, un ensemble d'expérimentations étant présenté dans le chapitre suivant.

4.4.1 Détection des clusters en mouvement

Pour obtenir les clusters en mouvement correspondant aux objets, la méthode de clustering pseudo balloon mean shift est appliquée à l'ensemble des points de la grille G (chapitre 2). Nous rappelons qu'un descripteur $\mathbf{x}^{(i)}$ est associé à chaque point d'index i de la grille. Ce descripteur contient des informations de position (vecteur à deux dimensions $\mathbf{x}_1^{(i)}$), de mouvement (vecteur à deux dimensions $\mathbf{x}_2^{(i)}$) et de photométrie (vecteur $\mathbf{x}_3^{(i)}$ à quatre dimensions pour les séquences en niveaux de gris et à six dimensions pour les séquences couleur). Ainsi, le paramètre P vaut 3.

Avant d'appliquer l'algorithme de clustering, les noyaux en chaque point doivent être calculés. Ils sont obtenus en appliquant la méthode de sélection décrite dans ce chapitre. Cet algorithme

nécessite préalablement la définition d'un ensemble de tailles de noyau possibles. Le choix de cet ensemble est important car il peut fortement influencer les résultats. Néanmoins, nous ne souhaitons pas le sélectionner différemment (c'est-à-dire manuellement) pour chaque séquence traitée, mais au contraire rendre cette sélection automatique. On pourrait choisir un vaste et large ensemble permettant de couvrir un grand nombre de bandes passantes. Cependant, un clustering mean shift doit être calculé pour chaque bande passante de l'ensemble prédéfini, et ce pour chaque domaine. Il est donc préférable de se restreindre à un nombre plus petit de matrices prédéfinies, sélectionnées en fonction des caractéristiques des données. Comme pour les exemples de segmentation présentés précédemment, nous définirons neuf bandes passantes pour chaque domaine : $B_\rho = 9$, $\rho = 1 \dots P$. Nous donnons, pour chaque domaine, les formules permettant de calculer les tailles de noyau prédéfinies. Ces formules sont intuitives etn'ont pas de réelles justification théorique.

Les tailles de noyau pour la position dépendent de la taille de la grille. En effet, plus l'espacement entre deux points consécutifs de la grille est grand, plus les bandes passantes correspondantes doivent l'être. Ainsi, les fenêtres prédéfinies pour la position sont données par :

$$\mathbf{H}_1^{(b)} = \frac{w}{N_G}(1 + \frac{3b}{B_1 - 1}) \ , \quad \mathbf{H}_2^{(b)} = \frac{h}{N_G}(1 + \frac{3b}{B_1 - 1}) \ , b = 1 \dots B_1 \ , \tag{4.18}$$

où B_1 est le nombre de matrices prédéfinies pour le premier domaine, c'est-à-dire la position ($B_1 = 9$). Nous rappelons (voir chapitre 2) que N_G est le paramètre définissant la taille de la grille (avant suppression des points appartenant au mouvement dominant et des points ayant des vecteurs de flot optique non valides), et que w et h sont la largeur et la hauteur de l'image.

Pour la couleur et le mouvement, nous faisons varier les bandes passantes en fonction du bruit entre points voisins de la grille. Plus précisément, le bruit est défini comme l'écart type de la différence entre les valeurs (de couleur ou de mouvement) de deux points voisins de la grille. Ainsi, en notant $\mathcal{E}_G$ l'ensemble des paires de points voisins de la grille G, et $|\mathcal{E}_G|$ son cardinal, la moyenne α_ρ et le bruit β_ρ sont des vecteurs définis par :

$$\alpha_\rho = \frac{1}{|\mathcal{E}_G|} \sum_{(i,j) \in \mathcal{E}_G} |\mathbf{x}_\rho^{(i)} - \mathbf{x}_\rho^{(j)}| \tag{4.19}$$

et

$$\beta_\rho = \sqrt{\frac{1}{|\mathcal{E}_G|} \sum_{(i,j) \in \mathcal{E}_G} (|\mathbf{x}_\rho^{(i)} - \mathbf{x}_\rho^{(j)}| - \alpha_\rho)^2} \ , \tag{4.20}$$

avec $\rho = 2$ pour le mouvement et $\rho = 3$ pour la couleur. L'ensemble des tailles prédéfinies sera compris entre 0.5 fois et 2 fois cet écart-type. Finalement les matrices prédéfinies sont données par :

$$\mathbf{H}_\rho^{(b)} = \beta_\rho(0.5 + \frac{1.5b}{B_\rho - 1})\mathbf{I}_{d_\rho}, \ \ b = 1 \dots B_\rho \ \ , \tag{4.21}$$

où $\mathbf{I}_{d_\rho}$ est la matrice identité de dimension d_ρ.

L'application du clustering avec les meilleures tailles de noyau (c'est-à-dire avec les noyaux les plus stables) permet d'obtenir l'ensemble des clusters en mouvement. Nous montrons dans la sous-section suivante un premier résultat.

4.4.2 Un premier résultat

Nous finissons ce chapitre en appliquant la méthode de clustering introduite dans le chapitre précédent et la sélection de la taille des noyaux définie dans ce chapitre sur une séquence en niveau de gris montrant le trafic dans une rue. Il est difficile de détecter les objets en mouvement intéressants dans cette séquence car ils sont petits et les images sont très bruitées. La figure 4.12 présente les clusters en mouvement obtenus sur une image de la séquence. Une couleur différente est attribuée à chaque cluster. On remarque que la méthode de détection des clusters en mouvement permet de détecter tous les objets intéressants de l'image, c'est-à-dire les piétons et les voitures. Cette séquence sera étudiée plus en détail dans le chapitre suivant. Il est important

FIG. 4.12: Résultats de la détection de clusters en mouvement sur l'image 109 de la séquence trafic.

de noter que, pour ce résultat et pour tous ceux du chapitre suivant, seuls les clusters contenant plus de 15 points ont été conservés. Après suppression, k_t clusters en mouvement, notés $C_{u,t}, u = 1 \dots k_t$, sont détectés.

Dans le chapitre suivant, nous décrivons la méthode de segmentation utilisée pour obtenir les masques complets des objets en mouvement. Ce dernier chapitre de la partie présente également un ensemble de résultats sur des séquences diverses.

Chapitre 5

Segmentation des objets en mouvement et résultats expérimentaux

Nous avons présenté jusqu'ici une méthode de détection de clusters en mouvement. Le dernier chapitre de cette partie présente des résultats expérimentaux pour cette méthode. Afin de mieux visualiser les objets nous proposons de les segmenter à partir des clusters détectés. Ainsi au lieu de n'être qu'un ensemble de quelques pixels, la détection sera le véritable masque de pixels de l'objet. Nous rappelons qu'à la fin de la phase de partitonnement des données, seuls les clusters comprenant plus de 15 points de la grille sont conservés.

Ce chapitre est organisé comme suit. La première section présente la méthode de segmentation utilisée. La section 5.2 propose des résultats sur différentes séquences et la section 5.3 montre une comparaison de notre méthode avec des méthodes classiques de détections de mouvement basées principalement sur l'apprentissage et la soustraction du fond.

5.1 Segmentation des objets en mouvement

Obtenir le masque de pixels complet des objets en mouvement requiert l'ajout d'une étape supplémentaire de segmentation. De nombreuses méthodes de segmentation existent dans la littérature. On peut notamment citer les méthodes de type croissance de région, les méthodes basées sur un seuillage global de l'image (seuillage de l'intensité, d'histogrammes...), les méthodes de détection de contours ou des méthodes de type contours actifs. La littérature sur le sujet étant extrêmement vaste, nous ne proposerons pas d'état de l'art ici. Nous renvoyons le lecteur aux articles [Pal 93, Lucchese 01, Meziane 02] et surtout au récent livre [Zhang 06] pour une revue des différentes méthodes de segmentation d'images.

Nous avons ici choisi de segmenter les objets à l'aide d'une méthode de coupe dans un graphe ("Graph Cuts" en anglais ; voir chapitre 6). Segmenter un objet associé à un certain cluster revient alors à assigner une étiquette l_s ("fond" ou "objet") à chaque pixel s de l'image. Les algorithmes dits de coupe minimale/flot maximal dans un graphe ont, ces dernières années, connu un très grand succès dans le domaine de la segmentation d'image. La raison de cette popularité est qu'ils permettent de trouver le maximum *a posteriori* d'un champ de Markov, et cela avec un temps de calcul faible. Ils sont utilisés pour minimiser rapidement certaines fonctions d'énergie, tout en assurant la convergence vers le minimum global de la fonction. Le chapitre 6 de ce document est entièrement dédié à la présentation de la théorie de la coupe minimale dans un graphe.

5.1.1 Définition de la fonction d'énergie

Dans [Boykov 01b], un algorithme de segmentation intéractif, utilisant des contraintes apportées par l'utilisateur, a été introduit. L'étiquette de certains pixels est forcée à être soit "fond" soit "objet" dans l'étiquetage final, ce qui signifie qu'elle ne pourra pas changer au cours de la minimisation de la fonction d'énergie. Les étiquettes des autres pixels doivent respecter les contours de l'objet et dépendent de l'information de couleur ou d'intensité de l'objet et du fond. Ces informations sont des distributions apprises sur des régions à l'intérieur et à l'extérieur de l'objet, régions pré-sélectionnées par l'utilisateur. Cette méthode est détaillée dans la section 6.4.1.

La méthode de segmentation que nous proposons d'appliquer ici est similaire à la méthode de [Boykov 01b]. Il y a néanmoins deux différences majeures. D'une part, nous n'utilisons pas simplement des informations de couleur ou d'intensité mais ajoutons également le mouvement. D'autre part, la technique est automatique (l'utilisateur n'intervient pas). En effet, les clusters en mouvement définissent les pixels de l'objet sur lesquels peuvent être apprises les distributions, et donnent l'étiquetage forcé de quelques points de l'objet. Les points de la grille appartenant au cluster en mouvement sont ainsi forcés à avoir l'étiquette "objet". Les spécifications précédentes peuvent être traduites par la fonction d'énergie suivante :

$$E_t(L) = -\gamma_c \sum_{s\in\mathcal{P}} \ln(p(\mathbf{z}_t^{(C)}(s)|l_s)) - \gamma_m \sum_{s\in G} \ln(p(\mathbf{z}_t^{(M)}(s)|l_s)) + \lambda \sum_{(s,r)\in\mathcal{E}_\mathcal{P}} B_{\{s,r\}}(l_s,l_r)\delta(l_s,l_r) \tag{5.1}$$

où L regroupe l'ensemble des étiquettes l_s en chaque pixel $s \in \mathcal{P}$ de l'image, $\mathcal{E}_\mathcal{P}$ est l'ensemble des paires (s,r) de pixels voisins de $\mathcal{P}$ (nous utilisons ici un 8-voisinage), et δ est la fonction définie par :

$$\delta(l_s,l_r) = \begin{cases} 1 & \text{si } l_s \neq l_r \\ 0 & \text{sinon .} \end{cases} \tag{5.2}$$

Les paramètres γ_m, γ_c et λ sont des constantes discutées plus loin. Nous avons ici décidé de segmenter chaque objet indépendamment, une fonction d'énergie étant associé à chaque cluster en mouvement, mais nous aurions pu tout aussi bien utiliser une énergie dite multi-étiquettes permettant d'obtenir directement tous les masques de pixels avec une seule minimisation d'énergie. La raison pour laquelle nous segmentons chaque objet indépendamment vient de la méthode de suivi appliquée à la suite de l'algorithme de détection (voir chapitre 8).

Les deux premiers termes de la fonction d'énergie sont appelés termes d'attache aux données et le troisième est appelé terme de régularisation. Le rôle des termes d'attaches aux données est d'inciter la segmentation finale, c'est-à-dire l'étiquetage, à respecter les informations de couleur et de mouvement de l'objet. Ces informations sont apportées par des distributions calculées sur l'ensemble des points du cluster en mouvement pour la distribution de l'objet, et sur l'ensemble des pixels de l'image pour la distribution du fond. Nous avons choisi d'utiliser des mélanges de gaussiennes, obtenus par un algorithme EM (Expectation-Maximization), pour calculer toutes les distributions. Quatre distributions sont calculées pour la segmentation de chaque objet associé au cluster u : deux distributions pour la couleur, construites respectivement à partir de l'ensemble des vecteurs $\{\mathbf{z}_{t-1}^{(C)}(s)\}_{s\in\mathcal{C}_{u,t}}$ pour l'objet et $\{\mathbf{z}_{t-1}^{(C)}(s)\}_{s\in\mathcal{P}}$ pour le fond, et deux pour le mouvement,

construites respectivement sur les ensembles $\{\mathbf{z}_{t-1}^{(M)}(s)\}_{s\in\mathcal{C}_{u,t}}$ pour l'objet et $\{\mathbf{z}_{t-1}^{(M)}(s)\}_{s\in\mathcal{P}}$ pour le fond. Nous rappelons que $\mathcal{C}_{u,t}$ est le cluster u à l'instant t.

Le but du terme de régularisation est d'encourager la segmentation à être suffisamment "régulière" (masques compacts) tout en respectant les contours forts de l'image. Nous utilisons le même terme que dans [Boykov 01b] :

$$B_{\{s,r\}}(l_s,l_r) = \exp\big(-\frac{\|\mathbf{z}_t^{(G)}(s)-\mathbf{z}_t^{(G)}(r)\|^2}{\sigma^2}\big).\frac{1}{\mathrm{dist}(s,r)} \ , \tag{5.3}$$

où $\mathrm{dist}(.)$ est une distance et le paramètre σ est lié au bruit [Rother 04] :

$$\sigma = 2 * \langle(\mathbf{z}_t^{(G)}(s)-\mathbf{z}_t^{(G)}(r))^2\rangle \ . \tag{5.4}$$

La fonction $\langle.\rangle$ représente la moyenne sur toute l'image.

Comme nous l'avons déjà évoqué, il a été montré dans [Boykov 01b] qu'il est possible de fixer l'étiquette finale de certains pixels. L'étiquette de ces pixels ne changera pas au cours de la minimisation. Pour plus de détails, nous renvoyons à la section 6.4.1. Si l'on force les points du cluster en mouvement à avoir l'étiquette "objet", la fonction d'énergie devient :

$$\begin{aligned} E_t(L) = &-\gamma_c \sum_{s\in\mathcal{P}\setminus\mathcal{C}_{u,t}} \ln(p(\mathbf{z}_t^{(C)}(s)|l_s)) - \gamma_m \sum_{s\in G\setminus\mathcal{C}_{u,t}} \ln(p(\mathbf{z}_t^{(M)}(s)|l_s)) - \\ &\sum_{s\in\mathcal{C}_{u,t}} \Big(1+\max_{s'\in\mathcal{P}}\lambda \sum_{r|(s',r)\in\mathcal{E}_\mathcal{P}} B_{\{s',r\}}\Big)\delta(l_s,\text{"objet"}) + \lambda \sum_{(s,r)\in\mathcal{E}_\mathcal{P}} B_{\{s,r\}}(l_s,l_r)\delta(l_s,l_r) \ . \end{aligned} \tag{5.5}$$

5.1.2 Réglage des paramètres

Le mouvement étant un champ épars, le terme d'attache aux données basé sur le mouvement a moins d'influence dans la fonction d'énergie que le terme basé sur la couleur. Pour essayer de lui donner autant d'importance nous posons $\gamma_c = 1$ et lions le paramètre γ_m à la taille de la grille :

$$\gamma_m = N_G^2. \tag{5.6}$$

Il reste à fixer le paramètre λ. Il a une influence très importante sur les résultats mais n'a pourtant pas beaucoup été étudié dans la littérature. Certains conseillent juste de lui donner la valeur 50 [Rother 04] et d'autres la valeur 20 [Blake 04]. Afin d'éviter l'éventuelle saturation de tous les arcs reliant deux pixels voisins, nous avons choisi de prendre pour λ la moyenne sur tous les pixels de l'image des termes d'attache aux données :

$$\lambda = \frac{1}{N}\Big(-\gamma_c \sum_{s\in\mathcal{P}} \sum_{l_s="fond","objet"} \ln(p(\mathbf{z}_t^{(C)}(s)|l_s)) - \gamma_m \sum_{s\in G} \sum_{l_s="fond","objet"} \ln(p(\mathbf{z}_t^{(M)}(s)|l_s))\Big) \ , \tag{5.7}$$

en rappelant que N est le nombre de pixels de l'image.

5.1.3 Minimisation de l'énergie et validation des objets

La minimisation de la fonction d'énergie par une méthode dite de "coupe minimale / flot maximal" (chapitre 6) permet d'obtenir l'étiquetage final de chaque pixel qui correspond au minimum global de cette fonction. L'étiquetage final donne directement la segmentation en terme de masque de pixels de l'objet en mouvement. Le même principe est appliqué successivement à chaque cluster en mouvement.

Il est possible qu'à la fin de la minimisation d'énergie la segmentation d'un objet ne contienne pas plus de pixels que le cluster en mouvement servant d'initialisation à la segmentation. Les pixels appartenant à la segmentation ne sont alors que ceux ayant été forcés à lui appartenir. Un tel résultat n'est pas surprenant si le cluster fait partie d'une grande région dynamique du fond, ou si ses contours ne sont pas assez forts. En effet, si le cluster fait partie d'une grande région du fond, les probabilités d'appartenance au fond ou à l'objet seront très proches tandis qu'aucun contour ne permettra de stopper la segmentation. Nous montrerons un exemple dans la section suivante (voir figure 5.1, image 124). Nous désignerons un cluster conduisant à une telle segmentation comme un cluster non valide. Ainsi si une étape de suivi (comme celle de la partie II) ou de reconnaissance est appliquée à la suite du détecteur, ces clusters non valides ne seront pas traités. En conclusion, la phase de segmentation permet non seulement d'obtenir les masques de pixels complets de chaque objet en mouvement mais aussi de supprimer les objets ou clusters non valides.

5.2 Résultats de détection et segmentation des objets en mouvement

Nous présentons maintenant des résultats de la méthode de détection des objets en mouvement sur un ensemble de séquences. Afin de bien visualiser les résultats, une couleur arbitraire est associée à chaque objet, à chaque instant. Cette couleur ne dépend que de l'ordre arbitraire dans lequel les objets ont été détectés. Comme nous ne faisons que de la détection (pas de cohérence temporelle ou de suivi), un même objet peut avoir des couleurs différentes au cours de la séquence. Nous avons décomposé la section en deux sous-sections, la première traitant des séquences couleur et la seconde montrant les résultats obtenus sur une séquence en niveau de gris.

5.2.1 Résultats sur des séquences couleur

Nous commençons par présenter des résultats de notre algorithme de détection d'objets en mouvement sur des séquences couleur acquises par une caméra mobile.

La première séquence analysée est la séquence de ski nautique déjà présentée dans le chapitre 2. Il est difficile de ne détecter que le skieur d'une part parce que le nombreux pixels en mouvement sont détectés dans l'eau, et d'autre part parce que la caméra peut bouger très brutalement. Sur la figure 5.1, la première colonne présente les images originales et la seconde la carte de détection des pixels en mouvement. Les deux dernières colonnes montrent les résultats de notre détecteur d'objet en mouvement. Ainsi, la troisième colonne montre les clusters en mouvement tandis que la quatrième expose les objets en mouvement, c'est-à-dire les segmentations finales des objets. Le skieur est généralement bien détecté et segmenté. Cependant il arrive régulièrement au cours de la séquence que le skieur ne soit pas du tout détecté. Il peut y avoir plusieurs raisons à cela. Pour l'image 124 (montrée sur la figure 5.1) par exemple, une grande partie des pixels du skieur est

assimilée au mouvement dominant de l'image. Il n'y a donc plus assez de pixels en mouvement sur le skieur pour pouvoir réaliser une bonne détection. Les différentes expérimentations menées nous ont également montré que, parfois, l'ensemble des noyaux prédéfinis n'est pas bien paramétré (noyaux de taille trop grande ou trop petite), et ne permettent pas la détection du skieur. Cela nous montre qu'un nombre plus important de noyaux devraient être prédéfinis mais cela impliquerait un coût de calcul supplémentaire non négligeable. Enfin le détecteur échoue aussi lorsque le skieur bouge rapidement relativement à la caméra, ce qui entraîne un grand flou de bougé et des couleurs pas assez cohérentes pour former un cluster. Néanmoins, la détection du skieur reste très satisfaisante, étant donné la dynamique du fond et de la caméra, et les échecs du détecteur n'apparaissent que pour un nombre minoritaire d'images de la séquence.

Un autre phénomène important à remarquer dans cette séquence est qu'il n'y a que très peu de clusters ou d'objets en mouvement détectés dans le fond et en particulier dans l'eau. De plus la plupart du temps, les clusters en mouvement détectés dans le fond sont, après segmentation, désignés comme non valides. Les résultats sur l'image 124 de la figure 5.1 montrent un cluster en mouvement détecté dans l'eau. Cependant il apparaît que l'objet segmenté correspondant n'est pas valide (le nombre de pixels appartenant à l'objet segmenté est égal au nombre de pixels du cluster en mouvement). Ce résultat n'est pas surprenant étant donné que le cluster est détecté dans un région présente sur un grande partie de l'image. Les probabilités d'appartenance au fond ou à l'objet sont alors très proches pour tous les pixels de l'eau.

Remarquons enfin que sur l'image 214, le cluster en mouvement contient des pixels appartenant d'un part à des régions de peau (visage, bras, jambes) et d'autre part à la combinaison du skieur. Cela signifie que des points de couleurs assez distinctes ont été regroupés dans le même cluster par l'algorithme de partitionnement des données. Dans le domaine YUV, les valeurs de chrominances sont très proches pour la combinaison et pour la peau mais les valeurs de luminance Y sont très différentes. Ce partitionnement, pouvant paraître surprenant, est de nouveau lié au choix automatique de la taille du noyau, ou, plus précisément, au calcul des noyaux prédéfinis. En effet, la variance sur toute l'image de l'intensité entre pixels voisins est grande, ce qui entraîne des noyaux de grande taille pour la dimension correspondant à la luminance.

Les deux résultats suivants montrent l'application de la méthode à la surveillance des conducteurs. Tout d'abord, nous présentons la détection des objets en mouvement sur la séquence de conducteur déjà évoquée au cours de cette partie. Les difficultés de cette vidéo sont les suivantes. Tout d'abord, le fond est composé de l'habitacle et du paysage (dynamique) visible derrière les différentes fenêtres de la voiture. Les changements dans le fond peuvent être brutaux avec de fortes variations d'illumination qui se répercutent sur le conducteur dans l'habitacle. De plus, le contraste entre la chemise et la barrière de sécurité est faible et le bras du conducteur peut avoir un mouvement en apparence similaire à celui des arbres derrière la fenêtre. Les clusters et segmentations correspondants aux objets en mouvement sont visibles sur la figure 5.2. Malgré le nombre de pixels en mouvement présents dans le paysage, aucun objet n'est détecté dans cette région. La main et le bras sont bien détectés et segmentés sur la majorité des images de la séquence. Il arrive cependant parfois que l'un de ces deux objets ne soit pas détectés (par exemple, sur l'image 17, le bras n'est pas détecté). En effet, les pixels en mouvement peuvent ne pas avoir des mouvements assez similaires ou alors le nombre de pixels en mouvement sur ces objets être trop faible. Un défaut (visible sur les images 15 et 17) est que sur certaines images une partie de la barrière de sécurité est détectée comme un objet en mouvement. Cependant, comme cette région n'est pas détectée à tous les instants, l'ajout de cohérence temporelle ou de suivi (voir partie II) permettra

FIG. 5.1: Résultats de la détection sur la séquence de ski nautique pour les images 74, 124, 144, 214, 232, 236 et 242

de la supprimer. Les résultats obtenus, même s'ils ne sont pas parfaits, sont vraiment très encourageants compte tenu de la complexité de la séquence.

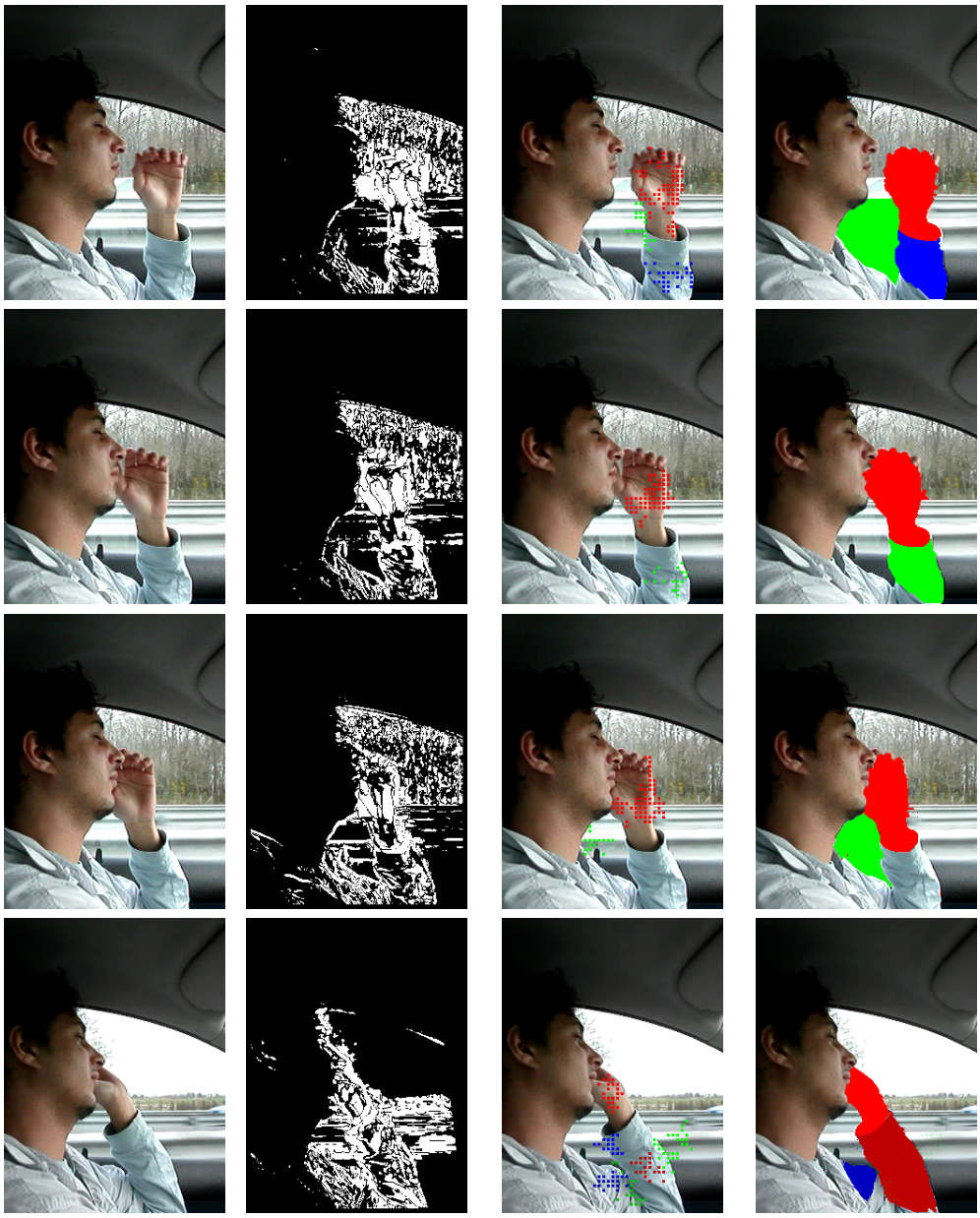
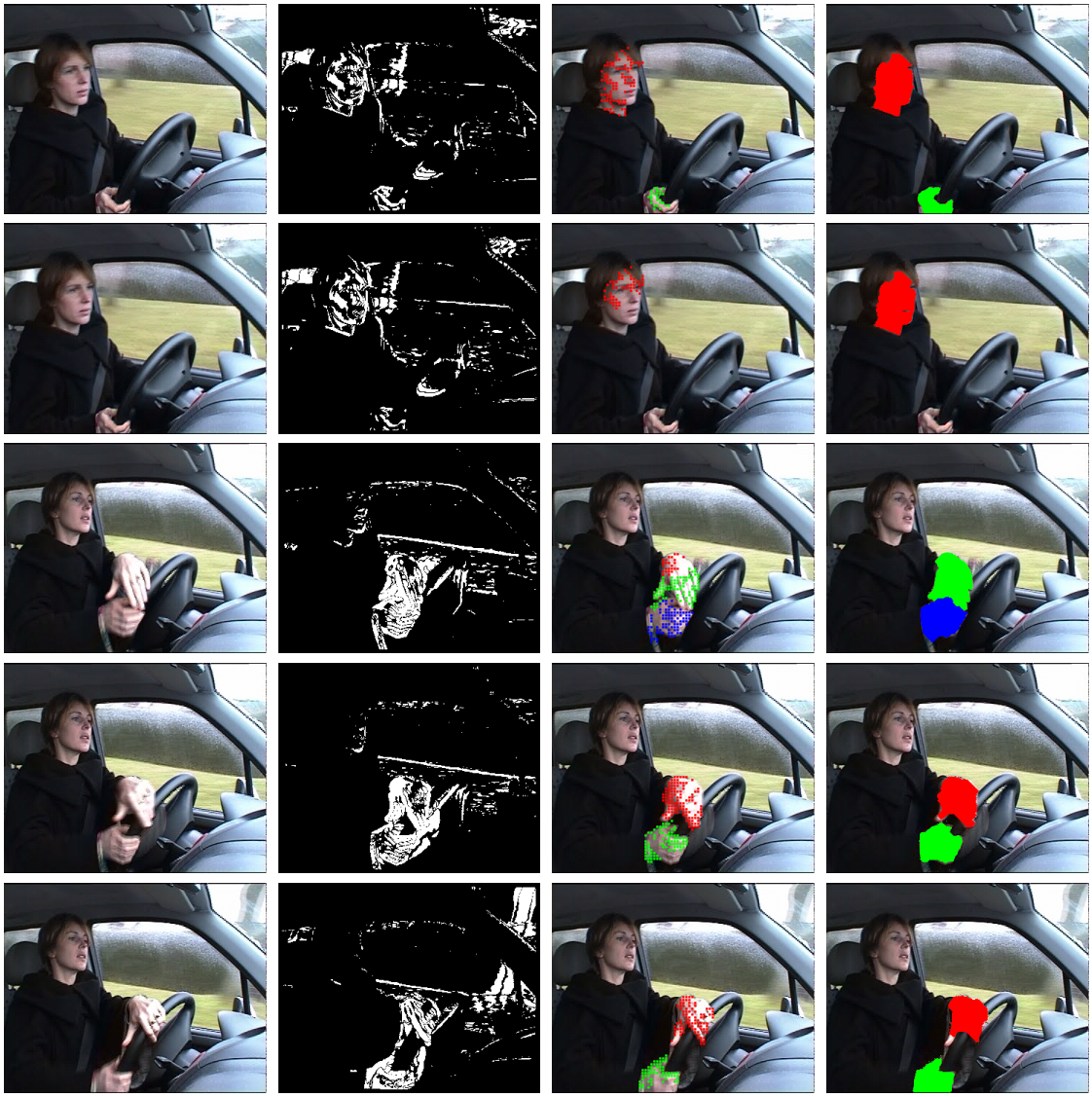


FIG. 5.2: Résultats de la détection sur la séquence de conducteur pour les images 15, 16, 17 et 48

Une deuxième séquence de conducteur est présentée sur la figure 5.3. Les difficultés sont relativement similaires à celles de la séquence précédente. Le contraste entre les objets dans l'habitacle et le paysage derrière la fenêtre est plus fort, mais les mains sont plus petites ce qui les rend plus difficiles à détecter. La tête est bien détectée et segmentée sur les premières images (image 16 et

17). Comme elle reste fixe aux instants suivants, elle n'est plus considérée comme un objet en mouvement. On remarque également que les deux mains sont bien détectées. En particulier, on peut voir sur l'image 40 que l'utilisation du mouvement dans le descripteur permet de bien détecter les deux mains séparément. Sur cette image, deux clusters en mouvement ont été obtenus sur la main gauche. Cependant, la phase de segmentation, qui encourage le masque des objets à respecter les contours, permet de bien regrouper ces deux clusters en un seul objet.



Images originales | Pixels en mouvement | Clusters en mouvement | Objets segmentés

FIG. 5.3: Résultats de la détection sur la séquence d'une conductrice pour les images 16, 17, 40, 42 et 47

5.2.2 Résultats sur une séquence en niveau de gris

Nous terminons cette section en montrant des résultats de détection sur une séquence en niveaux de gris très bruitée, avec un faible contraste. La caméra utilisée pour cette séquence est fixe. De nombreux petits objets en mouvement sont présents et doivent être détectés. En raison de la petite taille des objets et de la faiblesse du contraste, nous n'avons pas été capables d'obtenir de bonnes segmentations sur cette séquence malgré les détections très encourageantes des clusters en mouvement. Pour cette raison, sur la figure 5.4, nous présentons simplement des boîtes englobant chacun de ces clusters. Comme on peut le voir, la plupart des piétons les plus grands (en apparence) sont détectés. Celui au milieu à droite des images ne l'est cependant presque jamais. Cela n'est pas surprenant étant donné qu'il marche dans le sens de l'axe optique de la caméra et appartient donc en grande partie au mouvement dominant. En effet, cette région de l'image ne contient que très peu de pixels en mouvement avec des vecteurs de flot optique valides. Il arrive parfois que des clusters correspondant à du bruit soient détectés. Néanmoins, comme leur détection n'arrive que très rarement, l'ajout de cohérence temporelle ou d'une phase de suivi permettrait probablement de s'affranchir de ces mauvaises détections.

Nous avons présenté dans cette section plusieurs résultats très encourageant de notre méthode sur quatre séquences différentes.

5.3 Comparaison avec d'autres méthodes de détection de mouvement

Afin de valider complètement notre algorithme, nous le comparons à d'autres méthodes de détection de mouvement. Les méthodes servant de comparaison sont :

- la méthode de modélisation et de soustraction de fond de Grimson et Stauffer [Grimson 98] (voir chapitre 1)
- la méthode non paramétrique de Elgammal *et al.* [Elgammal 00] (voir chapitre 1)
- la méthode de la section 2.1 de détection des pixels en mouvement basée sur l'extraction du mouvement dominant [Odobez 95].

Les résultats de ces trois techniques sont des cartes binaires de détection de mouvement ou de changement. Notre méthode, au contraire, permet directement de détecter les masques de chacun des objets. Pour pouvoir la comparer, tous les pixels de l'image appartenant aux objets en mouvement segmentés sont étiquetés comme des zones de changement dans une carte binaire. Notons que les deux séquences traitées ici ont été acquises par une caméra fixe. En effet, les méthodes basées sur l'apprentissage de modèle de fond [Grimson 98, Elgammal 00] requièrent cette particularité, chaque pixel de l'image étant comparé avec le même pixel dans les images précédentes.

La première séquence est une séquence en niveaux de gris montrant un piéton et un cycliste se déplaçant devant un fleuve (figure 5.5). Comme on peut le voir sur la deuxième ligne, peu de pixels en mouvement sont détectés, ce qui rend cette séquence relativement facile à traiter par notre algorithme. Tandis que notre méthode détecte et segmente correctement le piéton, le cycliste et une partie des voitures roulant sur le pont, les autres techniques détectent de nombreux changements dans l'eau. De plus avec ces autres techniques, le masque pixels du piéton n'est

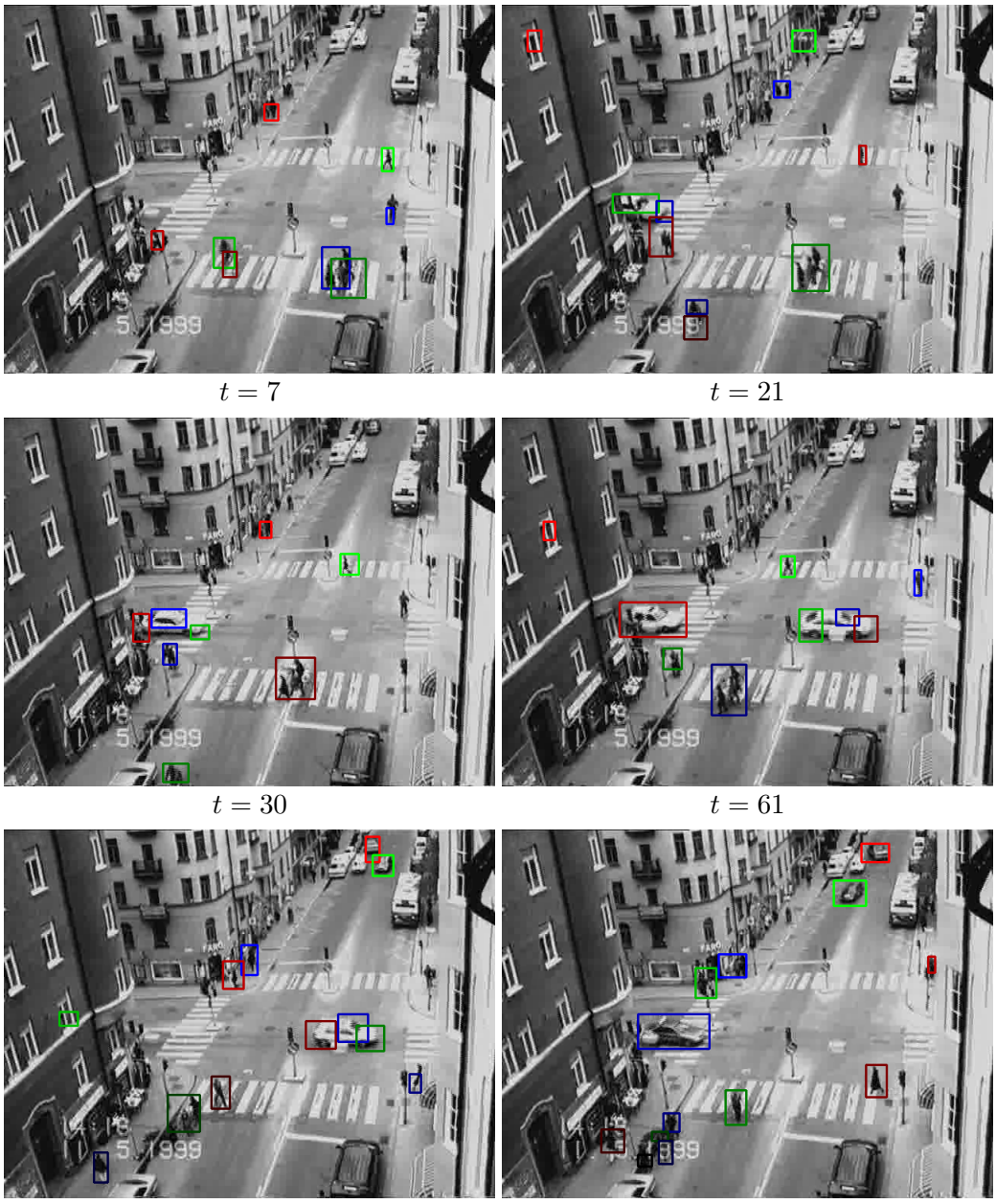


FIG. 5.4: Résultats de la détection sur la séquence trafic. Seules les boîtes englobant les clusters en mouvement sont montrées.

pas parfait et comporte des trous. Nous avions déjà évoqué le problème de trou à l'intérieur des détections dans le chapitre 2 (figure 2.2). Les trous dans les masques de détection du piéton sont dus au fait que la personne marche lentement et que son manteau est peu texturé. L'algorithme de segmentation utilisé par notre méthode permet de s'affranchir de ce problème sans aucune difficulté.

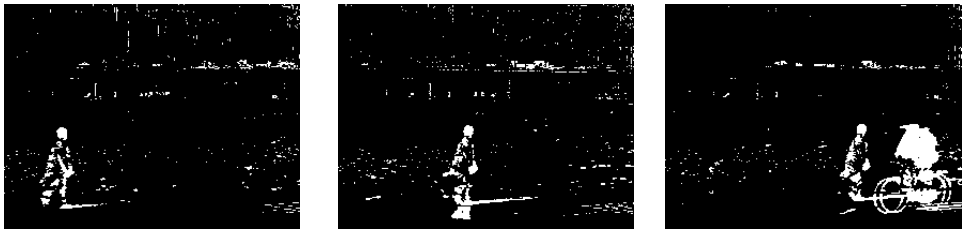
La deuxième séquence sur laquelle nous comparons nos résultats est une séquence couleur montrant deux personnes marchant au milieu d'arbres oscillants (figure 5.6). Le mouvement dans les arbres, qui couvrent une grande partie de l'image, et les nombreuses occultations partielles ou totales des personnes rendent cette séquence nettement plus difficile à traiter que la précédente. Le nombre de pixels en mouvement dans les arbres peut être important (deuxième ligne de la figure). Sur une telle séquence, la méthode de Grimson et Stauffer échoue totalement (quatrième ligne), et n'arrive ni à détecter les personnes, ni à séparer les arbres des bâtiments en arrière plan. La méthode non paramétrique (troisième ligne) donne par contre de bien meilleurs résultats, qui sont assez proches des cartes de détection des pixels en mouvement (deuxième ligne). Ces deux méthodes ne détectent aucun bâtiment mais détectent tout de même quelques pixels dans les arbres. De plus, comme pour la séquence précédente, les masques de détection des personnes contiennent des trous. Avec notre méthode, il arrive que des clusters en mouvement soient détectés au milieu des arbres, mais la phase de segmentation permet de les déclarer non valides (la segmentation associée ne contient que les pixels du cluster) et de les éliminer. Bien que prometteurs, les masques de segmentation ne sont pas parfaits. Cela vient des nombreuses occultations de certaines parties des piétons, qui ont pour conséquence de forts gradients d'intensité stoppant le flot au moment de la segmentation. De plus, les clusters représentant les objets sont généralement petits ce qui ne permet pas de bien apprendre les distributions de couleur et de mouvement nécessaires à la segmentation. Malgré toutes ces difficultés, nos résultats sont très prometteurs, comparés aux autres méthodes.

Images originales

Pixels en mouvement (section 2.1)

Méthode non paramétrique [Elgammal 00]

Méthode de Grimson et Stauffer [Grimson 98]



Notre méthode

$t = 34$ $t = 59$ $t = 84$

FIG. 5.5: Masques de détection obtenus par différentes méthodes pour la séquence montrant un piéton et un cycliste se déplaçant devant un fleuve.

Images originales

Pixels en mouvement (section 2.1)

Méthode non paramétrique [Elgammal 00]

Méthode Grimson et Stauffer [Grimson 98]

Notre méthode

$t = 108$ $t = 168$ $t = 235$

FIG. 5.6: Masques de détection obtenus par différentes méthodes pour la séquence montrant des piétons marchant au milieu d'arbres.

Conclusion

Cette première partie a traité de la détection d'objets en mouvement dans des scènes complexes. L'état de l'art du premier chapitre a montré qu'aucune des méthodes existantes ne peut être utilisée seule pour le type de vidéos que nous considérons ici (fond hautement dynamique avec des changements brutaux, caméra mobile, changements d'illumination...). Une analyse des différents types de méthodes a permis de mettre en avant leurs avantages et leurs limites. La méthode proposée combine les points positifs, des différentes méthodes pour tenter de s'attaquer aux défis posés par les vidéos que nous traitons.

L'algorithme de détection proposé se décompose en trois principales étapes. La première consiste à définir une grille de pixels en mouvement. Les pixels en mouvement sont ceux n'appartenant pas au mouvement de la caméra. Ces points sont alors décrits par leur position, leur couleur dans un espace de chrominance et leur mouvement. Les mouvements sont des vecteurs de flot optique valides. La validité est le résultat d'un test statistique (basé sur la p-value). Les points dont les vecteurs de flot optique ne sont pas valides sont retirés de la grille. Une étude plus approfondie sur le test statistique utilisé devrait être menée. En particulier, il serait intéressant de comparer plus en détail la p-value à un test de corrélation, afin de mieux mettre en avant ses atouts et l'amélioration des résultats finaux de détection.

La deuxième étape consiste à regrouper les points en plusieurs groupes, appelés cluster en mouvement, de couleur et mouvement quasi constants. Ce clustering est réalisé avec une nouvelle méthode de clustering mean shift appelée "pseudo balloon mean shift". Afin de sélectionner automatiquement en chaque point la taille du noyau de l'estimateur utilisé, un algorithme de sélection automatique de la bande passante a été introduit. Il consiste à sélectionner, pour chaque point des données, le noyau donnant le cluster le plus stable, parmi un ensemble de noyaux prédéfinis. Cette technique a été appliquée à la segmentation d'image couleur puis à la détection de clusters en mouvement. Nous avons tenu à mettre en place une règle de calcul des matrices prédéfinies afin de rendre la technique automatique, c'est-à-dire sans paramètre. Certains des résultats expérimentaux obtenus ont montré que le choix de l'ensemble des matrices n'est pas optimal et devrait encore être étudié. De plus, la méthode de sélection de bandes passantes proposée nécessite la représentation paramétrique par une loi normale de chaque cluster intermédiaire construit au cours de l'algorithme. Nous avons proposé d'utiliser les définitions traditionnelles de la moyenne et de la covariance mais un calcul de covariance robuste caractérisant mieux la forme du cluster serait probablement plus approprié. Pour cela, on pourrait par exemple s'appuyer sur les travaux [Pena 01] ou [Wang 02].

La dernière étape a pour but l'obtention des masques complets de chaque objet. Il s'agit alors de segmenter l'image pour chaque objet à partir du cluster en mouvement correspondant. Chaque masque est obtenu en minimisant une fonction d'énergie grâce au calcul de la coupe minimale,

c'est-à-dire du flot maximal, dans un graphe. Nous avons proposé ici de segmenter chaque objet indépendamment : une fonction d'énergie est minimisée pour chaque objet. Les algorithmes de coupe minimale/flot maximal donnent la possibilité de minimiser des énergies multi-étiquettes. Nous utiliserons d'ailleurs cette particularité dans le chapitre 8. Une extension de la méthode pourrait alors consister à modifier notre algorithme afin que tous les objets soient segmentés simultanément. Cela permettrait, d'une part de réduire le coût de calcul de l'algorithme, et d'autre part de pas considérer chaque objet objet indépendamment et de prendre en compte leurs interactions mutuelles.

Plusieurs résultats expérimentaux et comparaisons avec d'autres méthodes ont démontré les performances de l'algorithme sur des séquences issues de contextes différents. Le résultat présenté sur la figure 5.4 met en avant une limitation de la méthode. En effet, sur cette figure le piéton au milieu à droite des images n'est presque jamais détecté comme un objet en mouvement. La majorité des pixels de ce piéton appartenant au mouvement dominant de l'image, trop peu de pixels en mouvement sont disponibles pour le détecter correctement. Des travaux futurs devraient se concentrer sur l'étude de la détection de petits objets appartenant majoritairement au mouvement dominant.

Enfin, la méthode de détection des objets en mouvement mise en place dans cette partie n'exploite aucune cohérence temporelle. En effet, les objets sont détectés indépendamment à chaque instant. Dans la partie II, un algorithme de suivi d'objets est proposé. Il permettra d'utiliser, sous la forme d'observations extérieures, les clusters en mouvement afin de suivre chacun des objets en mouvement détectés.

Deuxième partie

Segmentation et suivi d'objets en mouvement

Introduction

La deuxième partie de ce document traite du suivi d'objets initialisés par un module externe de détection. Les méthodes de suivi permettent de construire les trajectoires d'objets mobiles en s'accommodant du bruit, des déformations subies par les objets et d'éventuelles occultations. Certaines méthodes de suivi nécessitent une initialisation manuelle tandis que d'autres utilisent directement des initialisation obtenues préalablement par une méthode de détection d'objets. Nous proposons ici une méthode de suivi s'appuyant sur la prédiction des objets suivis et sur des observations extérieures. Ces deux principaux ingrédients sont insérés dans une fonction d'énergie minimisée par un algorithme de coupe minimale/flot maximal permettant de segmenter et de suivre simultanément des objets. Cette seconde partie est organisée en trois chapitres.

Chapitre 6 Le premier chapitre présente la théorie des coupes dans un graphe. Trouver la coupe minimale dans un graphe peut servir à minimiser des énergies. Le principe est de ramener le problème de minimisation d'énergie à un problème de "coupe minimale" dans un graphe ; cette coupe minimale étant elle-même équivalente à un problème de "maximisation de flot". Nous détaillons donc les algorithmes permettant de trouver le flot maximal dans un graphe et démontrons les équivalences entre coupe minimale, flot maximal et minimisation d'énergie. L'application à la segmentation d'image est ensuite étudiée.

Chapitre 7 Dans le chapitre 7, un état de l'art des méthodes de suivi est donné. Nous séparons ces méthodes en trois grandes catégories : suivi par appariement de détection ("detect-before-track"), suivi par segmentation dynamique et suivi déterministe par détection séquentielle. Chacune de ces catégories possède ses propres avantages et limites que nous analyserons dans ce chapitre.

Chapitre 8 L'algorithme de suivi d'objets proposé est détaillé dans le chapitre 8. Il combine une partie des avantages des différentes catégories de méthodes décrites dans l'état de l'art du chapitre 7 et utilise des observations extérieures correspondant à des objets en mouvement détectés par un module extérieur. Cette détection peut être par exemple obtenue par simple soustraction de fond ou être le résultat de l'algorithme de détection décrit dans la partie I. Des résultats de la méthode de suivi sont montrés sur des séquences relativement simples ou le fond est fixe et sur les séquences complexes déjà utilisées dans la première partie. Ainsi, à la fin de ce chapitre nous montrons les résultats finaux de cette thèse traitant de la détection, de la segmentation et du suivi d'objets en mouvement dans des scènes complexes.

Chapitre 6

Coupe minimale/flot maximal dans un graphe

Ce chapitre est entièrement dédié à la présentation du principe de la coupe minimale dans un graphe (en anglais "Graph Cuts"). Il a pour objectif d'expliquer le principe de la minimisation d'énergie et de présenter ses applications les plus courantes en vision par ordinateur. L'ensemble de ce chapitre est largement inspiré du cours "Introduction aux GraphCuts" de Mickaël Péchaud[1].

Les algorithmes de coupe minimale/flot maximal dans un graphe ont été utilisés pour la première fois en vision par ordinateur par Greig et Porteous [Greig 89]. Ils ont connu depuis quelques années un très grand succès pour des applications diverses. Ce succès est dû au fait qu'ils permettent de trouver le minimum global, en temps polynômial, de nombreuses fonctions d'énergie couramment employées en vision.

Ce chapitre est organisé comme suit. Dans la première section, nous énonçons quelques définitions sur les graphes et introduisons les notions de coupe minimale et flot maximal. Nous présentons ensuite le problème de minimisation d'énergie et les algorithmes de coupe minimale/flot maximal permettant cette minimisation. Nous terminons le chapitre en montrant l'application à la segmentation d'images.

6.1 Définitions et propriétés sur les graphes

Le but des algorithmes de coupe minimale/flot maximal dans un graphe est de ramener le problème de minimisation d'énergie à un problème de "coupe minimale" dans un graphe ; cette coupe minimale étant elle-même équivalente à un problème de "maximisation de flot". Nous définissons dans cette section le type de graphe utilisé en vision par ordinateur, la notion de coupe minimale puis de flot maximal. Nous montrons ensuite l'équivalence entre coupe minimale et flot maximal.

6.1.1 Graphe

Un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est composé d'un ensemble d'arcs ou arêtes $\mathcal{E}$ et d'un ensemble de nœuds ou sommets $\mathcal{V}$. Un poids $\omega(p, q)$ est associé à chaque arc $e = (p, q) \in \mathcal{E}$, connectant le nœud $p \in \mathcal{V}$ au nœud $q \in \mathcal{V}$. Il est positif et appelé **capacité de l'arc**. Un graphe peut être orienté ou non orienté. Nous ne nous intéresserons ici qu'au cas des graphes orientés. Un graphe orienté

[1] http ://mickaelpechaud.free.fr/graphcuts.pdf

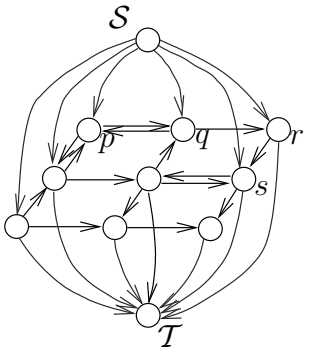


FIG. 6.1: Exemple de graphe.

est un graphe dont les arêtes sont orientées : $(p,q) \neq (q,p)$ et $\omega(p,q) \neq \omega(q,p)$, si $(p,q) \in \mathcal{E}$ et $(q,p) \in \mathcal{E}$ (ce qui n'est pas nécessairement le cas). On distingue sur le graphe deux sommets particuliers : une source $\mathcal{S}$ et un puits $\mathcal{T}$, et on considère qu'il n'y a pas d'arête entrante dans $\mathcal{S}$ ni d'arête sortante de $\mathcal{T}$. Les autres sommets sont les nœuds intermédiaires du graphe. Plusieurs sources peuvent être ajoutées mais nous traiterons principalement dans ce chapitre les graphes contenant uniquement deux terminaux. Afin d'illustrer ces premières définitions, un exemple de graphe est donné sur la figure 6.1.1.

6.1.2 Coupe

Une coupe dans un graphe est un ensemble d'arcs déconnectant la source du puits, c'est-à-dire qu'il n'existe plus de chemin orienté de $\mathcal{S}$ à $\mathcal{T}$. Une coupe peut également être vue comme une partition $S \cup T$ des sommets où $\mathcal{S}$ appartient à S et $\mathcal{T}$ appartient à T. Plus généralement une coupe entre deux nœuds quelconques p et q est définie de la manière suivante :

Définition 3 (Coupe)
Si p et q sont 2 sommets d'un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, une ($p$,$q$)-coupe est un ensemble C d'arcs déconnectant p de q : dans le graphe partiel $(\mathcal{V}, \mathcal{E} \backslash C)$ il n'existe pas de chemin orienté de p à q.

Pour la recherche de la coupe minimale dans un graphe, seules les ($\mathcal{S}$,$\mathcal{T}$)-coupes séparant la source du puits sont étudiées. Nous les nommerons directement coupes, et les noterons C, dans la suite de ce chapitre. Comme les arêtes du graphe, une coupe possède une capacité :

Définition 4 (Capacité d'une coupe)
La capacité $|C|$ d'une coupe C est la somme des capacités des arcs de la coupe :

$$|C| = \sum_{\substack{(p,q) \in \mathcal{E} \\ p \in S, q \in T}} \omega(p,q) \ . \tag{6.1}$$

La définition de la capacité d'une coupe permet directement d'introduire la notion de coupe minimale :

Définition 5 (Coupe minimale)
On appelle coupe minimale dans un graphe une coupe C_{min} de capacité minimale ($\forall$ coupe C, $|C_{min} \leq |C|$).

6.1.3 Flot

Comme nous allons le démontrer, le problème de calcul d'une coupe minimale dans un graphe est dual au problème du calcul du flot maximal dans un graphe. Un flot représente en quelque sorte l'acheminement d'un liquide depuis la source $\mathcal{S}$ vers la destination $\mathcal{T}$ et est alors décrit par la quantité de liquide, appelée flux, transitant sur chacun des arcs du graphe. Le flux une fonction $f : \mathcal{E} \rightarrow \mathbb{R}$, qui, sur un arc, doit être inférieur à la capacité de l'arc.

Un flot doit vérifier, en chaque sommet, une loi de conservation analogue aux lois de Kirchoff en électricité : le flux entrant en chaque sommet est égal au flux sortant. Dans toute la suite nous utiliserons le mot flot pour parler du flot ou du flux. Les deux propriétés d'un flot sont énoncées ci-dessous :

Propriété 1
Un flot f vérifie les propriétés suivantes :

1. $\forall\ p \in \mathcal{V}\backslash\{\mathcal{S},\mathcal{T}\},\ \sum_{q|(p,q)\in\mathcal{E}} f(p,q) = \sum_{q|(q,p)\in\mathcal{E}} f(q,p)$ *(loi de Kirchoff)*
2. $\forall\ (p,q) \in \mathcal{E},\ 0 \leq f(p,q) \leq \omega(p,q)$

Nous pouvons maintenant introduire la notion de "valeur d'un flot" :

Définition 6 (Valeur d'un flot)
On dit que f est un flot de $\mathcal{S}$ à $\mathcal{T}$ dans le graphe $\mathcal{G} = (\mathcal{V},\mathcal{E})$ si et seulement si les lois de conservation (lois de Kirchoff) sont vérifiées pour tous les nœuds exceptés les terminaux où l'on a :

$$\sum_{p\in\mathcal{V}\backslash\{\mathcal{S},\mathcal{T}\}} f(\mathcal{S},p) = \sum_{p\in\mathcal{V}\backslash\{\mathcal{S},\mathcal{T}\}} f(p,\mathcal{T}) = |f|\ . \tag{6.2}$$

La quantité $|f|$ est appelée valeur d'un flot.

Un flot maximal est alors tout naturellement un flot de valeur maximale.

Définition 7 (Flot maximal)
Le problème du flot maximal consiste à trouver un flot F_{max} de valeur maximale sur le graphe ($\forall$ flot f de $\mathcal{S}$ à $\mathcal{T}$, $|f| < |F_{max}|$).

Dans la suite de cette section nous montrons qu'il y a équivalence entre flot maximal et coupe minimale.

6.1.4 Relation entre coupe minimale et flot maximal

Une coupe $C = S \cup T$ est un passage obligé pour un flot, et potentiellement un goulot d'étranglement. En effet, un flot transitant de S à T doit nécessairement emprunter les arêtes de la coupe (par définition, tous chemins de S à T comportent au moins une arête). Comme nous allons le montrer, la valeur du flot peut être définie comme le flot sortant de la coupe moins le flot entrant. Pour cela, dénotons par $\tilde{\mathcal{G}} = (\mathcal{V},\tilde{\mathcal{E}})$ le graphe déduit de $\mathcal{G}$ en ajoutant l'arc $(\mathcal{T},\mathcal{S})$, appelé arc de

retour de flot, de capacité infinie. Dans $\tilde{\mathcal{G}}$, la loi de Kirchoff s'étend aux terminaux. Le point 2 de la propriété 1 devient alors : $\forall\ p \in \mathcal{V},\ \sum_{q|(p,q)\in\tilde{\mathcal{E}}} f(p,q) = \sum_{q|(p,q)\in\tilde{\mathcal{E}}} f(q,p)$.

Le problème du flot maximal de $\mathcal{S}$ à $\mathcal{T}$ dans $\mathcal{G}$ revient à trouver un flot f dans $\tilde{\mathcal{G}}$ vérifiant les contraintes de capacité ($\forall (p,q) \in \tilde{\mathcal{E}}\ 0 \leq f(p,q) \leq \omega(p,q)$) et tel que la valeur du flot f sur l'arc de retour ($f(\mathcal{T},\mathcal{S}) = |f|$) soit maximale. On peut maintenant énoncer la propriété de la valeur d'un flot dans une coupe.

Propriété 2 (Valeur d'un flot dans une coupe)
Si $C = S \cup T$ est une coupe et f un flot de $\mathcal{S}$ à $\mathcal{T}$ dans le graphe $\mathcal{G}$, alors la valeur du flot est égale au flot sortant de S moins le flot entrant dans S :

$$|f| = f^+(S) - f^-(S)$$

avec

$$f^+(S) = \sum_{p\in S, q\in T \mid (p,q)\in\mathcal{E}} f(p,q)$$

et

$$f^-(S) = \sum_{p\in S, q\in T \mid (q,p)\in\mathcal{E}} f(q,p)$$

Preuve
Soit f un flot quelconque dans $\tilde{\mathcal{G}}$, auquel correspond le flot f entre la source et le puits dans $\mathcal{G}$. Dans $\tilde{\mathcal{G}}$, la propriété sue les lois de Kirchoff s'écrit :

$$\forall\ p \in \mathcal{V},\ \sum_{q|(p,q)\in\tilde{\mathcal{E}}} f(p,q) = \sum_{q|(q,p)\in\tilde{\mathcal{E}}} f(q,p)\ .$$

On a alors :

$$\sum_{p\in S} \sum_{q|(p,q)\in\tilde{\mathcal{E}}} f(p,q) = \sum_{p\in S} \sum_{q|(q,p)\in\tilde{\mathcal{E}}} f(q,p)\ ,$$

d'où, en distinguant les nœuds de S et les nœuds de T :

$$\sum_{p\in S, q\in S|(p,q)\in\tilde{\mathcal{E}}} f(p,q) + \sum_{p\in S, q\in T|(p,q)\in\tilde{\mathcal{E}}} f(p,q) = \sum_{p\in S, q\in S|(q,p)\in\tilde{\mathcal{E}}} f(q,p) + \sum_{p\in S, q\in T|(q,p)\in\tilde{\mathcal{E}}} f(q,p)\ ,$$

ce qui est équivalent à,

$$\sum_{p\in S, q\in S|(p,q)\in\mathcal{E}} f(p,q) + \sum_{p\in S, q\in T|(p,q)\in\mathcal{E}} f(p,q) = \sum_{p\in S, q\in S|(q,p)\in\mathcal{E}} f(q,p) + \sum_{p\in S, q\in T\setminus\mathcal{T}|(q,p)\in\tilde{\mathcal{E}}} f(q,p) + f(\mathcal{T},\mathcal{S})\ .$$

D'après les lois de Kirchoff, $\sum_{p\in S\setminus\mathcal{S}, q\in S\setminus\mathcal{S}|(p,q)\in\mathcal{E}} f(p,q) = \sum_{p\in S\setminus\mathcal{S}, q\in S\setminus\mathcal{S}|(q,p)\in\mathcal{E}} f(q,p)$, ce qui

implique :

$$\begin{aligned}\sum_{p\in S,q\in S|(p,q)\in\mathcal{E}} f(p,q) &= \sum_{p\in S\setminus\mathcal{S},q\in S\setminus\mathcal{S}|(p,q)\in\mathcal{E}} f(p,q) + \sum_{p\in\mathcal{V}\setminus\mathcal{S}|(\mathcal{S},p)\in\mathcal{E}} f(\mathcal{S},p)\\ &= \sum_{p\in S\setminus\mathcal{S},q\in S\setminus\mathcal{S}|(q,p)\in\mathcal{E}} f(q,p) + \sum_{p\in\mathcal{V}\setminus\mathcal{S}|(\mathcal{S},p)\in\mathcal{E}} f(\mathcal{S},p)\\ &= \sum_{p\in S,q\in S|(q,p)\in\mathcal{E}} f(q,p)\ .\end{aligned}$$

On a donc :

$$\sum_{p\in S,q\in T|(p,q)\in\mathcal{E}} f(p,q) = \sum_{p\in S,q\in T(q,p)\in\mathcal{E}} f(q,p) + |f|\ ,$$

et finalement

$$|f| = \sum_{p\in S,q\in T|(p,q)\in\mathcal{E}} f(p,q) - \sum_{p\in S,q\in T\setminus\mathcal{T}|(q,p)\in\mathcal{E}} f(q,p)\ .$$

On a ainsi montré que la valeur du flot est égale au flot sortant de S moins le flot entrant dans S. ■
Les propriétés 1 et 2 montrent que la valeur de tout flot entre $\mathcal{S}$ et $\mathcal{T}$ est toujours inférieure à la capacité de n'importe quelle coupe. En effet, comme $\forall\ (p,q)\in\mathcal{E},\ \ 0\leq f(p,q)\leq\omega(p,q)$, on a

$$|f| \leq \sum_{p\in S,q\in T|(p,q)\in\mathcal{E}} \omega(p,q),$$

c'est-à-dire $|f|\leq|C|$. Cela conduit donc à la propriété suivante :

Propriété 3 (Équivalence flot maximal / coupe minimale (forme faible))
Si f est un flot entre S et T et $C = S\cup T$ est une coupe, alors : $|f|\leq|C|$. Il est possible de réécrire cette propriété en fonction du flot maximum et de la coupe minimum :

$$|F_{max}| \leq |C_{min}| \tag{6.3}$$

Nous venons de voir que la capacité d'une coupe constitue une borne supérieure pour la valeur maximale d'un flot. Nous allons maintenant voir que cette borne supérieure est toujours atteinte : la valeur maximale d'un flot est toujours égale à la capacité minimale d'une coupe. Pour pouvoir démontrer cette égalité, nous avons préalablement besoin d'introduire les notions de graphe résiduel, de chemin augmentant et de saturation.

6.1.5 Définitions : saturation, graphe résiduel, chemin augmentant

Nous énonçons dans cette sous-section une liste de définitions qui permettront de démontrer le théorème d'égalité entre coupe minimale et flot maximal.

Définition 8 (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 : $f(p,q)=\omega(p,q)$. Un chemin est saturé si l'un de ses arcs est saturé.

Tant que l'arc n'est pas saturé, il est possible de lui faire transiter plus de flot. Bien entendu les contraintes sur les capacités doivent être respectées. La quantité de flot que l'on peut faire passer en plus dans un arc s'appelle la capacité résiduelle.

Définition 9 (Capacité résiduelle)
La capacité résiduelle d'un arc (p, q) est la quantité $\omega(p, q) - f(p, q)$ de flot qu'il peut encore faire transiter. La capacité résiduelle d'un chemin est la plus petite capacité résiduelle de ses arcs.

Saturer un chemin entre le nœud $\mathcal{S}$ et le nœud $\mathcal{T}$ consiste à augmenter la valeur du flot des arcs du chemin de la capacité résiduelle du chemin. Lorsque l'on sature un chemin entre $\mathcal{S}$ et $\mathcal{T}$, la propriété de flot est conservée sur le graphe : le flux sur un arc reste inférieur à la capacité de l'arc et la loi de Kirchoff est encore vérifiée pour tous les sommets. En effet, le chemin repart de tout nœud intermédiaire autant de fois qu'il y arrive. Si il existe encore des chemins non saturés entre $\mathcal{S}$ et $\mathcal{T}$, le flot f n'est pas maximum. Il suffit alors de saturer l'un de ces chemins pour obtenir un flot de valeur supérieure. La définition précédente permet d'introduire la notion de graphe résiduel.

Définition 10 (Graphe résiduel)
Un graphe résiduel $\mathcal{G}_f$ associé à un flot f sur un graphe $\mathcal{G}$ est un graphe comprenant les mêmes sommets que $\mathcal{G}$. Chaque arc (p, q) de $\mathcal{G}$ est associé dans $\mathcal{G}_f$ à :

- *l'arc "forward" $(p, q)_F$ de capacité $\omega(p, q) - f(p, q)$.*
- *l'arc "backward" $(p, q)_B$ de capacité $f(p, q)$.*

La capacité de l'arc forward indique qu'il est possible d'augmenter le flot sur (p, q) d'au plus $\omega(p, q) - f(p, q)$ ce qui correspond à la saturation de l'arc. Un arc forward "existe" (il est de capacité non nulle) dans $\mathcal{G}_f$ si et seulement si il n'est pas saturé dans $\mathcal{G}$. La capacité de l'arc backward indique qu'il est possible de diminuer le flot entre p et q d'au plus $f(p, q)$. Cela correspond à annuler le flot. Un arc backward "existe" (il est de capacité non nulle) dans $\mathcal{G}_f$ si et seulement si le flot n'est pas nul sur l'arc (p, q) dans $\mathcal{G}$. Si la valeur du flot de $\mathcal{S}$ à $\mathcal{T}$ est nulle alors le graphe résiduel est le même que le graphe initial.

Si tous les chemins ne sont pas saturés, il est possible d'ajouter du flot dans le graphe résiduel. Ce flot est alors appelé flot résiduel. Il a les propriétés suivantes.

Propriété 4 (Ajout d'un flot résiduel)
Si f_r est un flot sur le graphe résiduel $\mathcal{G}_f$, nous définissons la valuation des arcs $f' = f \oplus f_r$ sur le graphe $\mathcal{G}$ par :

$$f'(p, q) = f(p, q) + f_r(p, q)_F - f_r(q, p)_B$$

On a alors les propriétés suivantes :

- *f' est un flot sur le graphe $\mathcal{G}$*
- *La valeur du flot f' est la somme des 2 flots : $|f'| = |f| + |f_r|$*

Nous terminons cette sous-section en définissant la notion de chemin augmentant.

Définition 11 (Chemin augmentant)
Un chemin augmentant pour un flot f sur un graphe $\mathcal{G}$ est un chemin (orienté) de capacité non nulle reliant $\mathcal{S}$ à $\mathcal{T}$ dans le graphe résiduel $\mathcal{G}_f$.

Cela revient à dire que le chemin augmentant est un chemin le long des arcs non saturés du graphe résiduel. Le flot correspondant à la saturation du chemin augmentant R sera noté f_R dans la suite.

6.1.6 Équivalence entre flot maximal et coupe minimale

Les définitions données précédemment vont maintenant nous permettre d'établir l'équivalence entre une flot maximal et une coupe minimale. Cette égalité découle directement du théorème suivant.

Théorème 1 (Ford & Fulkerson)
Si f est un flot dans un graphe $\mathcal{G}$, il y a équivalence entre les propriétés :

1. *La valeur du flot f est maximale*
2. *Il n'existe pas de chemin augmentant pour f*
3. *Il existe une coupe C de capacité égale à la valeur du flot f*

Preuve
Pour montrer l'équivalence entre ces 3 propriétés, nous allons établir la suite des implications.

(1) $\Rightarrow$ (2) Démontrons la contraposée. S'il existait un chemin augmentant R, alors $f \oplus f_R$ serait un flot de valeur $|f|+|f_R|$, ce qui contredit le fait que f est un flot de valeur maximale.

(2) $\Rightarrow$ (3) Supposons qu'il n'existe pas de chemin augmentant. Soient S l'ensemble des sommets p tels qu'il existe un chemin orienté de $\mathcal{S}$ à p dans le graphe résiduel, et T son complémentaire. La partition $C = S \cup T$ définit une coupe sur le graphe. Montrons que la capacité C de cette coupe est égale à la valeur du flot f. Comme $|f| = f^+(S) - f^-(S)$, ceci revient à démontrer les deux points suivants :

a) Les arcs sortants de S sont saturés par le flot f , c'est-à-dire $f^+(S)$ est maximal :
Soit un arc (p, q), $p \in S$ et $q \in T$. Supposons par l'absurde que cet arc n'est pas saturé. Alors dans le graphe résiduel l'arc forward $(p, q)_F$ a une capacité non nulle. En prolongeant le chemin de $\mathcal{S}$ à p avec l'arc $(p, q)_F$, on obtient alors un chemin de $\mathcal{S}$ au sommet q, ce qui contredit $q \in T$. Ainsi tous les arcs sortants de S sont saturés.

b) Les arcs entrants de S n'ont aucun flot transitant sur eux, c'est-à-dire $f^-(S) = 0$:
Soit un arc (q, p), $q \in T$ et $p \in S$. Supposons également par l'absurde que le flot f est non nul sur cet arc. Alors, dans le graphe résiduel, l'arc backward $(q, p)_B$ a une capacité non nulle, ce qui contredit que le sommet q n'appartienne pas à S.

Nous avons donc :

$$|f| = \sum_{p \in S, q \in T | (p,p) \in \mathcal{E}} \omega(p, q) - 0 = |C|.$$

(3) $\Rightarrow$ (1) Cette implication découle directement de la propriété 3 (équation 6.3).

■

Ainsi, si f est un flot maximal dans un graphe alors il existe une coupe de capacité égale à la valeur du flot. Or la capacité de la coupe minimale est toujours supérieure au flot maximal (propriété 3). On peut donc directement énoncer le théorème "Coupe minimale / Flot maximal" :

Théorème 2 (Coupe minimale / Flot maximal)
La valeur maximale d'un flot entre deux sommets dans un graphe est égale à la capacité minimale d'une coupe :

$$|F_{max}| = |C_{min}| \ .$$

6.2 Méthodes pour trouver le flot maximal

Il existe de nombreux algorithmes pour trouver un flot maximal dans un graphe. Pour une bonne revue de toutes ces méthodes, on peut se référer à [Schrijver 03]. Les algorithmes peuvent être divisés en deux classes : les algorithmes par saturation de chemins et les algorithmes par poussage de flot.

6.2.1 Flot maximal par saturation des chemins

Les algorithmes par saturation de chemins s'appuient directement sur la définition de la saturation. Saturer un chemin entre le nœud $\mathcal{S}$ et le nœud $\mathcal{T}$ consiste à augmenter le flot sur les arcs du chemin de la capacité résiduelle du chemin. Si il existe encore des chemins non saturés entre $\mathcal{S}$ et $\mathcal{T}$, le flot f n'est pas maximum. Il suffit alors de saturer l'un de ces chemins pour obtenir un flot de valeur supérieure. Ainsi, un algorithme de saturation peut être construit en saturant itérativement tous les chemins entre $\mathcal{S}$ et $\mathcal{T}$. Le principe est de partir d'un flot nul et de trouver itérativement un chemin de la source au puits sur lequel il n'y a pas d'arête saturée. Du flot est alors ajouté en saturant l'arête qui a la capacité résiduelle la plus faible. Ce principe est illustré par l'algorithme 6.

Algorithme 6 Algorithme par saturation de chemins

- Initialisation :

  $\forall (p,q) \in \mathcal{E} \;\; f(p,q) = 0$ (On part d'un flot nul)

- Tant Que il existe un chemin R de $\mathcal{S}$ à $\mathcal{T}$ non saturé par le flot f
  Augmenter f en saturant R :
  1. Calculer $a = \text{argmin}_{\omega(p,q),(p,q)\in R} w(p,q) - f(p,q)$
  2. $\forall (p,q) \in R$, $f(p,q) \leftarrow f(p,q) + a$

  Fin TantQue

- Retourner f

Cet algorithme ne conduit pas toujours au flot maximal [Ford 62, Schrijver 03]. En effet, les chemins non saturés de $\mathcal{S}$ à $\mathcal{T}$ ne sont pas recherchés dans le graphe résiduel mais uniquement dans le graphe initial. Il ne s'agit donc pas de chemins augmentant et le théorème 1 ne peut alors pas être appliqué. Afin de trouver le flot maximal, Ford et Fulkerson [Ford 62] ont proposé de ne pas saturer les chemins dans le graphe initial $\mathcal{G}$ mais dans un graphe résiduel $\mathcal{G}_f$ (algorithme 7), ce qui rejoint le théorème 1.

Les deux algorithmes précédents peuvent sembler similaires, mais, comme nous allons le montrer sur un exemple, ils ne conduisent pas au même résultat. Considérons ainsi le graphe, présenté sur la figure 6.2, composé d'une source, d'un puits et de deux nœuds intermédiaires p et q. Les capacités des arcs sont indiquées au-dessus de chaque arc. Considérons tout d'abord un flot nul sur ce graphe. Un chemin possible entre $\mathcal{S}$ et $\mathcal{T}$ (en rouge sur la figure 6.2a)) est $(\mathcal{S}, q, p, \mathcal{T})$ de capacité résiduelle 2. La saturation du chemin donne un flot f égal à 2. Le flot circulant dans chaque arc est indiqué en vert. L'algorithme de saturation s'arrête à cette étape : il n'y a plus de chemin

Algorithme 7 Algorithme de Ford & Fulkerson

- Initialisation :

  $\forall (p,q) \in \mathcal{E} \;\; f(p,q) = 0$ (On part d'un flot nul)

- Tant Que il existe un chemin augmentant R de $\mathcal{S}$ à $\mathcal{T}$

  Saturer le chemin R dans le graphe résiduel $\mathcal{G}_f$

  $f \leftarrow f \oplus f_R$

  Fin TantQue

- Retourner f

non saturé de $\mathcal{S}$ à $\mathcal{T}$ dans le graphe. Cependant, il existe un chemin augmentant $(\mathcal{S}, p, q, \mathcal{T})$ dans le graphe résiduel (figure 6.2b)). La saturation du chemin augmentant fait passer la valeur du flot à 3. Après cette itération, il n'existe plus de chemin augmentant dans le graphe résiduel (figure 6.2 c)). L'algorithme de Ford & Fulkerson s'arrête alors et délivre le flot f, de valeur 3, qui est maximum.

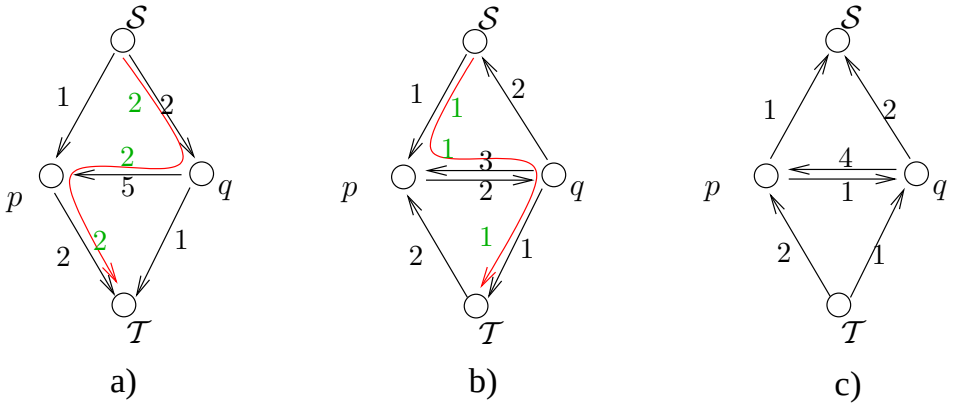


FIG. 6.2: Comparaison de l'algorithme de saturation (algorithme 6) et de l'algorithme de Ford & Fulkerson (algorithme 7). En rouge sont montrés des chemins augmentant possibles et en vert est indiqué le flot dans chaque arc. a) Algorithme de saturation correspondant aussi à la première étape de l'algorithme de Ford & Fulkerson. b) Saturation des chemins dans le graphe résiduel (algorithme de Ford & Fulkerson). La figure montre le graphe résiduel avec ses capacités. Les arcs de capacité nulle ne sont pas représentés. c) Graphe résiduel final, dans lequel il n'existe plus de chemin augmentant.

Plusieurs algorithmes ont été mis en place depuis. Par exemple, Dinic [Dinic 70] reprend le principe de saturation des chemins mais cherche à chaque étape un chemin de longueur minimale dans le graphe résiduel. Une forme "dynamique" ce cette méthode a été introduite dans [Boykov 04]. Les arbres de recherche de chemins dans le graphe résiduel sont conservés et modifiés à chaque itération (plutôt que complètement recalculés). Le coût calculatoire de cet algorithme est faible (temps de calcul quasiment linéaire par rapport à la taille des données).

6.2.2 Flot maximal par poussage de flot

Les algorithmes par poussage de flot utilisent des pré-flots dont la définition est donnée ci-dessous.

Définition 12 (Pré-flot)
Un pré-flot est un flot (fonction $f : \mathcal{E}^* \rightarrow \mathbb{R}$) ne respectant pas la contrainte de préservation de flot, c'est-à-dire la loi de Kirchoff :

$$\forall\ p \in \mathcal{V}\backslash\{\mathcal{S},\mathcal{T}\},\ \ |\sum_{q|(p,q)\in\mathcal{E}} f(p,q) - \sum_{q|(q,p)\in\mathcal{E}} f(q,p)| \geq 0$$

Le principe du poussage de flot est alors d'envoyer autant de flot que possible à partir de la source. Bien entendu, certains nœuds reçoivent un excès de flot. Ces sommets sont dits actifs. Le flot excessif dans ces nœuds est alors poussé vers d'autres nœuds disponibles. Une étiquette, évoluant au cours des itérations, est associée à chaque nœud. On notera l_p l'étiquette associée au sommet p. Le principe des méthodes par poussage de flot est détaillé dans l'algorithme 8.

Algorithme 8 Algorithme par poussage de flot

- Initialisation :
  Étiqueter les nœuds du graphe (le graphe contient $|\mathcal{V}|$ nœuds) :
  - $l(\mathcal{S}) = |\mathcal{V}|$
  - $l(\mathcal{T}) = 0$
  - $\forall\ p \in \mathcal{V}\backslash\{\mathcal{S},\mathcal{T}\}\ l_p = 1$

  $\forall\ p \in \mathcal{V}\backslash\{\mathcal{S},\mathcal{T}\},\ f(\mathcal{S},p) = w(\mathcal{S},p)$
- Tant Que il reste des sommets actifs, choisir un de ces sommets, noté p.

  Tant Que p est actif

  si $\exists\ q \in \mathcal{V}$ tel que $(p,q) \in \mathcal{E}$, $l(p) = l(q) + 1$ et $w(p,q) - f(p,q) > 0$
  alors $f(p,q) \leftarrow \min(w(p,q) - f(p,q), \sum_{q|(p,q)\in\mathcal{E}} f(p,q))$

  (pousser le flot excessif vers l'autre nœud)

  sinon ré-étiqueter p :

$$l_p \leftarrow \text{argmin}_{\substack{q|(p,q)\in\mathcal{E} \\ w(p,q)-f(p,q)>0}} (l_q + 1)$$

  Fin TantQue

  Fin TantQue
- Retourner f

Les étiquettes attribuées aux nœuds lors de la phase d'initialisation peuvent aussi avoir comme valeur la distance au puits. Cela évite des étapes de ré-étiquetage inutiles. Le plus célèbre des algorithmes par poussage de flot est l'algorithme "Push relabel" [Goldberg 86]. D'autres méthodes existent aussi [Goldberg 89, Schrijver 03].

6.3 Minimisation d'énergies

Cette section commence par décrire les énergies que l'on cherche à minimiser. Nous énonçons ensuite un certain nombre d'algorithmes ayant pour but la minimisation d'énergies et terminons en décrivant le principe de la minimisation d'énergies par coupe minimale/flot maximal.

Pour simplifier les notations nous considérerons, dans toute la suite du chapitre, que $\mathcal{V}$ est un ensemble contenant tous les nœuds d'un graphe exceptés les terminaux.

6.3.1 Présentation des énergies considérées

Certains problèmes en vision par ordinateur peuvent être formulés en terme d'étiquetage de pixels et de minimisation d'énergies. Une étiquette l_s appartenant à l'ensemble des étiquettes $\mathcal{L}$ doit être associée à chaque nœud $s \in \mathcal{V}$ d'un graphe. Le plus souvent, $\mathcal{V}$ représente l'ensemble des pixels $\mathcal{P}$ de l'image I. Dans ce cas, un nœud est associé à chaque pixel. L'étiquetage $\mathcal{L}$ doit dépendre des propriétés de l'image. Par exemple pour les applications de segmentation les étiquettes doivent respecter les contours forts et dépendre des intensités des régions que l'on souhaite segmenter. Ces problèmes d'étiquetage peuvent être résolus en minimisant des énergies. Un nombre important de fonctions d'énergies existe dans la littérature. Elles se décomposent le plus souvent en deux termes :

$$E(\mathcal{L}) = E_{donnees}(\mathcal{L}) + E_{lissage}(\mathcal{L}) \ . \tag{6.4}$$

Le premier terme, $E_{donnees}$, est un terme unaire qui représente l'attache aux données. Il encourage les étiquettes à respecter les informations observées dans l'image. Généralement, le terme d'attache aux données s'écrit :

$$E_{donnees}(\mathcal{L}) = \sum_{s \in \mathcal{V}} D_s(l_s) \ . \tag{6.5}$$

La fonction D_s indique si, connaissant les données observées, le label l_s est bien approprié pour le nœud s. Le deuxième terme, $E_{lissage}$, est un terme binaire de régularité portant sur des groupes de pixels voisins. Il encourage les étiquettes à respecter les discontinuités de l'image. Le terme de régularité peut avoir plusieurs formes. Nous nous concentrons ici sur les termes de lissage décrits par l'équation suivante :

$$E_{lissage}(\mathcal{L}) = \sum_{s \in < s,r >} B_{\{s,r\}}(l_s, l_r) \ , \tag{6.6}$$

où $< s, r >$ désigne tout arc de $\mathcal{E}$ reliant deux sommets s et r mutuellement voisins (l'ensemble des sommets voisins au pixel s sera noté $\mathcal{N}_s$). En général on utilise 4 ou 8 nœuds voisins (figure 6.3), appelés respectivement 4-voisinage ou 8-voisinage. La fonction $B_{\{s,r\}}$ encourage deux (ou parfois trois) pixels ayant des caractéristiques similaires à avoir la même étiquette.

6.3.2 Algorithmes d'étiquetage et de minimisation

Plusieurs algorithmes permettant de minimiser des énergies ou d'étiqueter les nœuds d'un graphe existent. Nous présentons ci-après une liste de ces méthodes.

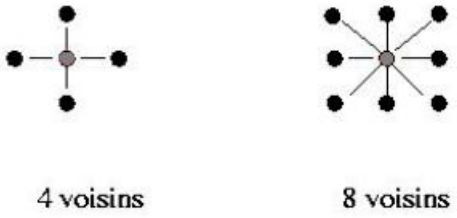


FIG. 6.3: Relation de voisinage (4 ou 8 plus proches voisins).

6.3.2.1 Marches aléatoires

Les algorithmes dits de marches aléatoires sont principalement dédiés à la segmentation d'image. L'idée est d'associer un graphe à l'image. Un nœud est placé en chaque pixel et les nœuds associés à des pixels voisins sont reliés par des arêtes. Le poids de ces arcs est d'autant plus grand que les pixels sont similaires. La marche aléatoire dans un graphe est basée sur l'assertion suivante : depuis un sommet, il est d'autant plus probable d'aller vers un autre nœud que l'arête les reliant ces deux sommets a un poids important.

Le but de la segmentation est d'associer une étiquette à chacun des pixels de l'image, cette étiquette dépendant de l'objet auquel le pixel appartient. Afin de segmenter une image avec un algorithme de marche aléatoire, l'utilisateur doit préalablement étiqueter quelques pixels à l'intérieur du ou des objets à segmenter, et à l'extérieur de ces objets. Ces étiquettes sont appelées graines. Ensuite, pour chaque pixel non étiqueté, une marche aléatoire est effectuée. Cette dernière s'arrête dès que l'une des graines est atteinte. Une probabilité d'atteindre chaque graine est définie en chaque pixel et l'étiquette finale du pixel sera la même que celle de la graine ayant la plus forte probabilité d'être atteinte. Pour de plus amples informations, nous renvoyons le lecteur à l'article [Grady 06].

6.3.2.2 Normalized Cuts

Le "Normalized Cuts" [Shi 00] est une méthode basée sur la théorie des graphes permettant de faire du clustering et de la segmentation. Le graphe est construit de la même façon que pour les algorithmes de marche aléatoire : un nœud est associé à chaque pixel et des arêtes relient les pixels, leurs poids étant d'autant plus grands que les pixels sont semblables. Il s'agit alors de trouver la coupe minimale dans ce graphe n'ayant pas de terminaux (pas de source ni de puits). La recherche de la solution optimale se reformule comme la recherche de vecteurs propres d'une matrice associée au graphe.

6.3.2.3 Champs de Markov

Avant de citer des algorithmes de minimisation d'énergie basées sur les champs de Markov, nous commençons par introduire la notion de champ de Gibbs. Un champ de Gibbs sur un graphe, pour une configuration donnée $\mathcal{L}$, est une distribution statistique π de la forme :

$$\pi(\mathcal{L}) \propto \exp\left[-E(\mathcal{L})\right] = \frac{1}{Z}\exp\left[-E(\mathcal{L})\right] \ , \ \ Z > 0 \ . \tag{6.7}$$

La distribution π est une distribution de probabilité sur l'ensemble des configurations possibles. Minimiser la fonction d'énergie revient donc à trouver la configuration la plus probable sous la distribution π. Les propriétés des champs de Markov permettent de remplacer la minimisation directe de l'énergie par un problème probabiliste de *Maximisation a Posteriori* (MAP) dans un champ de Markov. Nous allons maintenant donner quelques exemples de champs de Markov couramment utilisés. Nous ne présentons ici que le terme binaire, c'est-à-dire de régularité, des énergies considérées.

Modèle ferro-magnétique d'Ising
Le modèle d'Ising a été introduit en physique pour calculer l'aimantation de certains corps dans un champ magnétique. Dans les cas du modèle d'Ising pour les champs de Markov, les étiquettes sont binaires et peuvent prendre les valeurs -1 ou $+1$. La fonction d'énergie s'écrit :

$$E(l) = -\beta \sum_{s \in <s,r>} l_s l_r \ , \tag{6.8}$$

où β est une constante de couplage strictement positive qui mesure l'interaction entre deux pixels (ou deux "spins" en physique). Cette fonction d'énergie traduit le fait que deux spins voisins ont tendance à s'orienter dans la même direction : on dit alors que le champ est attractif.

Modèle de Potts
Le modèle de Potts est la généralisation du modèle d'Ising à un problème d'étiquetage non binaire. Chaque étiquette est choisie dans une ensemble de n étiquettes possibles. L'énergie s'écrit :

$$E(l) = -\beta \sum_{s \in <s,r>} \mathbf{1}_{l_s = l_r} + \beta \sum_{s \in <s,r>} \mathbf{1}_{l_s \neq l_r} \ . \tag{6.9}$$

Plusieurs algorithmes existent pour minimiser des énergies dans un champ de Markov. Ces algorithmes permettent de calculer le maximum *a posteriori* du champ de Markov. Les méthodes les plus connues sont l'algorithme de "Belief Propagation" ou de "Loopy Belief Propagation" [Felzenszwalb 06], l'algorithme "Tree-Reweighted" [Kolmogorov 05] et l'algorithme de recuit simulé [Geman 84].

6.3.3 Minimisation d'énergies par coupe minimale/flot maximal

Nous présentons maintenant la minimisation d'énergie par coupe minimale/flot maximal. Dans le début de cette sous-section, nous commençons par présenter l'algorithme de coupe minimale le plus utilisé en vision par ordinateur. Cet algorithme se restreint à des énergies ayant quelques propriétés particulières que nous détaillerons ensuite.

6.3.3.1 Algorithme "Expansion Move"

L'algorithme le plus connu et le plus utilisé pour minimiser des fonctions d'énergies en utilisant les Graph Cuts est l'algorithme appelé "Expansion Move" [Boykov 01b]. Il s'agit d'un algorithme itératif, valable pour le cas binaire ou pour le cas multi-étiquettes, qui repose sur des changements d'étiquetage appelés "α-expansions". Étant donné un étiquetage courant $\mathcal{L}$ et une

étiquette particulière α, l'"α-expansion" est définie comme le passage de l'étiquetage $\mathcal{L}$ à un étiquetage $\mathcal{L}'$ tel que si $l'_s \neq \alpha$ (avec $s \in \mathcal{V}$) alors $l'_s = l_s$. Autrement dit, l'étiquette d'un nœud ne peut être modifiée que si elle est différente de α. Le nœud prend alors l'étiquette α si cela fait décroître la fonction d'énergie. Ainsi le passage de $\mathcal{L}$ à $\mathcal{L}'$ augmente le nombre de pixels ayant pour étiquette α. Si le passage de $\mathcal{L}$ à $\mathcal{L}'$ diminue la valeur de l'énergie alors $\mathcal{L}'$ devient le nouvel étiquetage courant. L'étiquetage final est obtenu quand l'énergie ne peut plus être minimisée. La convergence de cet algorithme se fait en temps polynômial.

Depuis le début du chapitre nous ne nous sommes intéressés qu'à l'étiquetage binaire des pixels. Une seule source et un seul puits étaient considérés pour le graphe. Dans de nombreuses applications, plusieurs étiquettes sont nécessaires. Il faut alors se ramener à un graphe ayant autant de terminaux que d'étiquettes. Minimiser une fonction d'énergie dans un environnement multi-étiquettes revient à trouver la coupe minimale séparant chacun des terminaux. Ce type de coupe est appelé "multiway" [Boykov 01b]. Pour trouver une telle coupe minimale, le principe est de se ramener au cas binaire [Dahlhaus 92, Boykov 01b], itérativement et jusqu'à convergence, en trouvant la coupe minimale C_i séparant un terminal $\mathcal{S}_i$ de tous les autres. Chaque coupe est obtenue en appliquant un algorithme de type "Expansion Move". Il est ensuite possible de générer une coupe "multiway" C, en considérant $C = \cup_{\mathcal{S}_i \neq \mathcal{S}_{max}} C_i$, où $\mathcal{S}_{max} = \arg\max_{\mathcal{S}_i} |C_i|$ est le terminal dont la capacité de la coupe est la plus grande.

6.3.3.2 Propriétés des fonctions d'énergies

Nous énonçons maintenant les propriétés que doivent respecter les fonctions d'énergies pour qu'elles puissent être minimisées en utilisant l'algorithme "Expansion Move". Ces propriétés sont bien décrites dans [Kolmogorov 04]. Nous nous focalisons de nouveau ici sur le cas binaire. Une énergie E, fonction de n variables binaires, peut être représentée par un graphe s'il existe un graphe $\mathcal{G} = (\mathcal{V} \cup \{\mathcal{S}, \mathcal{T}\}, \mathcal{E})$ avec des terminaux $\mathcal{S}$ et $\mathcal{T}$ et un ensemble de n sommets intermédiaires $\mathcal{V} = (v_1, \ldots, v_n)$ tel que, pour toute configuration $l_1, \ldots, l_n$, la valeur de l'énergie $E(l_1, \ldots, l_n)$ soit égale à une constante plus la capacité d'une coupe. Le sommet v_i appartient à S si $l_{v_i} = 0$ et réciproquement, $v_i \in T$ si $l_{v_i} = 1$. Si la constante est nulle, on dit que l'énergie est exactement représentable par un graphe.

Les énergies étudiées ici sont de la forme de celles décrites par l'équation (6.4) :

$$E(\mathcal{L}) = \sum_{s \in \mathcal{V}} D_s(l_s) + \sum_{s \in <s,r>} B_{\{s,r\}}(l_s, l_r) \ . \tag{6.10}$$

Le graphe correspondant, s'il existe, a la forme suivante : il contient un ensemble de sommets $\mathcal{V} \cup \{\mathcal{S}, \mathcal{T}\}$ et un ensemble d'arcs $\mathcal{E}$ qui peut être divisé en deux sous-ensembles $\mathcal{E}_1$ et $\mathcal{E}_2$.

Le premier sous-ensemble, $\mathcal{E}_1$, contient les arcs reliant les terminaux à chacun des sommets de $\mathcal{V}$. Les capacités de ces arcs sont prises en compte dans le premier terme de la fonction d'énergie : les arcs $(\mathcal{S}, s)$, $s \in \mathcal{V}$, ont pour capacité $D_s(1)$ et les arcs $(s, \mathcal{T})$, $D_s(0)$. Cela s'explique très facilement en prenant un graphe composé d'une source, d'un puits et d'un seul sommet intermédiaire s (figure 6.4).

Fixer les capacités des deux arcs de manière à ce que la valeur de la coupe soit égale à l'énergie de l'étiquetage considéré est trivial. En effet, les deux configurations possibles sont :

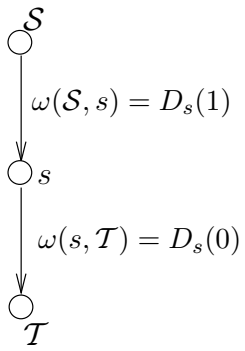


FIG. 6.4: Capacité des arcs reliant les terminaux aux nœuds du graphe.

- $l_s = 0$, c'est-à-dire $E(\mathcal{L}) = D_s(0)$. Cela revient à couper l'arc $(s, \mathcal{T})$ (la capacité de la coupe est donc égale à $\omega(s, \mathcal{T})$), et implique donc $\omega(s, \mathcal{T}) = D_s(0)$,
- $l_s = 1$, c'est-à-dire $E(\mathcal{L}) = D_s(1)$. Cela revient à couper l'arc $(\mathcal{S}, s)$ (la capacité de la coupe est donc égale à $\omega(\mathcal{S}, s)$), et implique donc $\omega(\mathcal{S}, s) = D_s(1)$.

Les capacités des arcs devant être positives, on s'assurera que la fonction D_s respecte cette propriété.

Le deuxième sous-ensemble, $\mathcal{E}_2$, contient l'ensemble des arcs (s, r) reliant deux sommets s et r de $\mathcal{V}$ mutuellement voisins. Les capacités de ces arcs sont prises en compte dans le second terme (terme binaire) de la fonction d'énergie. En effet, soit un graphe composé d'un source, d'un puits et de deux sommets intermédiaires s et r (figure 6.5). Nous connaissons déjà les capacités des arcs

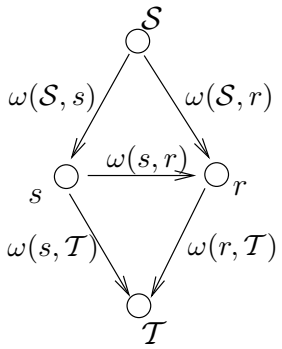


FIG. 6.5: Capacité des arcs reliant les sommets mutuellement voisins.

reliant les terminaux aux nœuds s et r :

$$\omega(\mathcal{S}, s) = D_s(1)\ ,\ \ \omega(\mathcal{S}, r) = D_r(1)\ ,\ \ \omega(s, \mathcal{T}) = D_s(0)\ ,\ \ \omega(r, \mathcal{T}) = D_r(0)\ .$$

Quatre coupes sont possibles dans ce graphe. Pour trouver la valeur de la capacité $\omega(s, r)$ de manière à ce que la valeur d'une coupe soit égale à l'énergie de l'étiquetage considéré, il nous faut résoudre le système suivant :

$$\left\{\begin{array}{l} \omega(s, \mathcal{T}) + \omega(r, \mathcal{T}) = D_s(0) + D_r(0) + B_{\{s,r\}}(0, 0)\ \ ,\ \ (l_s = 0 \text{ et } l_r = 0)\ , \\ \omega(\mathcal{S}, s) + \omega(r, \mathcal{T}) + \omega(s, r) = D_s(1) + D_r(0) + B_{\{s,r\}}(1, 0)\ \ ,\ \ (l_s = 1 \text{ et } l_r = 0)\ , \\ \omega(s, \mathcal{T}) + \omega(\mathcal{S}, r) = D_s(0) + D_r(1) + B_{\{s,r\}}(0, 1)\ \ ,\ \ (l_s = 0 \text{ et } l_r = 1)\ , \\ \omega(\mathcal{S}, s) + \omega(\mathcal{S}, r) = D_s(1) + D_r(1) + B_{\{s,r\}}(1, 1)\ \ ,\ \ (l_s = 1 \text{ et } l_r = 1)\ . \end{array}\right.$$

On obtient alors la capacité de l'arc :

$$\omega(s,r) = B_{\{s,r\}}(1,0) + B_{\{s,r\}}(0,1) - B_{\{s,r\}}(0,0) - B_{\{s,r\}}(1,1) \ .$$

Comme les capacités des arcs doivent être positives, les énergies, décrites par l'équation (6.10), sont représentables par un graphe si les fonctions $B_{\{s,r\}}$ sont régulières (ou sous-modulaires) [Kolmogorov 04] :

Définition 13 (Fonction régulière)
On dit que $B_{\{s,r\}}$ est régulière si :

$$B_{\{s,r\}}(0,0) + B_{\{s,r\}}(1,1) < B_{\{s,r\}}(0,1) + B_{\{s,r\}}(1,0) \ . \tag{6.11}$$

Récemment des algorithmes permettant d'appliquer les "Graph Cuts" à un ensemble plus large de fonctions d'énergies ont été proposés. Nous ne détaillons pas ici ces algorithmes et renvoyons le lecteur à [Kolmogorov 07].

6.4 Applications à la segmentation d'images

La minimisation d'énergies par coupe minimale/flot maximal a été appliquée à de nombreux problèmes de vision par ordinateur : restauration d'images, stéréo, synthèse de textures, mosaïques digitales et segmentation. Nous nous attardons ici sur l'application à la segmentation d'images dans le cas binaire (c'est-à-dire que l'on ne cherche à segmenter qu'un seul objet).

6.4.1 Segmentation d'images par coupe minimale/flot maximal

Le principe de la segmentation d'image par coupe minimale/flot maximal est assez simple. Un graphe $\mathcal{G} = (\mathcal{P} \cup \{\mathcal{S}, \mathcal{T}\}, \mathcal{E})$ est appliqué à l'image. Un sommet est associé à chaque pixel et chaque sommet est relié par un arc à ses 4 ou 8 sommets voisins. Les capacités des arcs sont d'autant plus faibles que le gradient de l'image est fort au voisinage de l'arête. On place ensuite, soit manuellement soit automatiquement, un certain nombre de graines à l'intérieur de l'objet à segmenter et un certain nombre de graines à l'extérieur. Placer des graines à l'intérieur de l'objet revient à étiqueter les pixels des graines à 1 (reliés au puits). Au contraire, placer des graines à l'extérieur (fond) de l'objet revient à étiqueter les pixels des graines à 0 (reliés à la source). La segmentation de l'objet est alors obtenue en trouvant la coupe minimale dans ce graphe.

Nous détaillons ici le premier algorithme donnant une solution au problème de la segmentation des objets avec les "Graph Cuts" [Boykov 01a]. L'utilisateur commence par placer des graines à l'intérieur et à l'extérieur de l'objet à segmenter (figure 6.6). Tous les pixels appartenant à ces graines sont alors forcés (dans l'étiquetage final) à appartenir soit à l'objet "Obj" (source) s'ils sont à l'intérieur de celui-ci, soit au fond "Bkg" (puits) s'ils sont à l'extérieur. Pour garantir le placement de ces pixels dans le bon ensemble, on procède comme suit. Si s est un pixel d'une graine à l'intérieur de l'objet, le terme d'attache aux données pour ce pixel est défini ainsi :

$$D_s(l_s) = \begin{cases} 0 \ \text{ si } l_s = 1, \\ \kappa \ \text{ sinon} \ , \end{cases} \tag{6.12}$$

avec

$$\kappa = 1 + \max_{s \in <s,r>} \beta_{\{s,r\}}(l_s, l_r) \ , \tag{6.13}$$

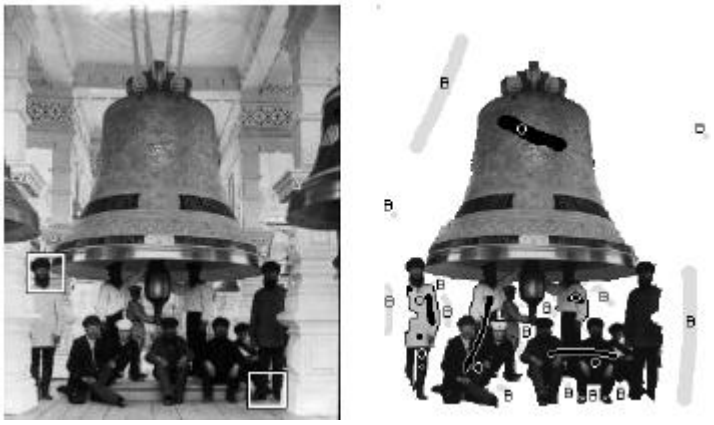


FIG. 6.6: Graines placées à l'extérieur ("B") et à l'intérieur de l'objet ("O") à segmenter : à gauche, l'image originale, à droite, le résultat de la segmentation et les graines. Cette figure provient de [Boykov 01a].

où $\beta_{\{s,r\}}$ est une fonction définie telle que $B_{\{s,r\}}(l_s, l_r) = \beta_{\{s,r\}}(l_s, l_r)\delta(l_s, l_r)$. La fonction δ est donnée, comme dans le chapitre 5, par

$$\delta(l_s, l_r) = \begin{cases} 1 \text{ si } l_s \neq l_r \\ 0 \text{ sinon ,} \end{cases} \tag{6.14}$$

De cette façon l'arête reliant le puits au pixel s, de capacité κ, ne sera jamais saturée et ne pourra jamais appartenir à la coupe. De la même manière, si le pixel s appartient à une graine placée à l'extérieur de l'objet, le terme d'attache aux données est défini par :

$$D_s(l_s) = \begin{cases} \kappa \text{ si } l_s = 1, \\ 0 \text{ sinon .} \end{cases} \tag{6.15}$$

Afin de définir le terme d'attache aux données pour tous les autres pixels, un histogramme est calculé sur tous les pixels des graines à l'intérieur de l'objet et un autre sur tous les pixels des graines à l'extérieur. Le terme d'attache aux données est alors donné par la probabilité (notée Pr) qu'un pixel appartienne à une distribution ou à l'autre :

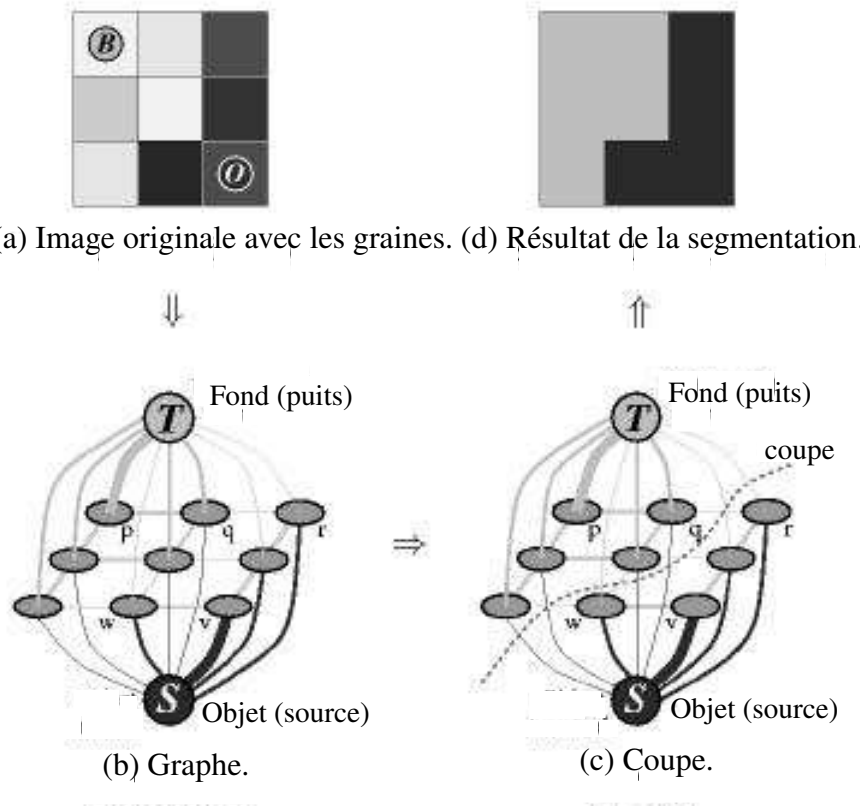
$$D_s(l_s) = \begin{cases} -\ln \Pr(z_t^{(G)}(s)|"Obj") \text{ si } l_s = 0 \\ -\ln \Pr(z_t^{(G)}(s)|"Bkg") \text{ sinon .} \end{cases} \tag{6.16}$$

Nous rappelons que $z^{(G)}(s)$ est la valeur de l'intensité au pixel s dans l'image à l'instant t, I_t.

Le terme de régularité dépend des gradients de l'image. Il s'écrit :

$$B_{\{s,r\}}(l_s, l_r) \propto \frac{1}{\text{dist}(s,r)} \exp\Big(- \frac{(z_t^{(G)}(s) - z_t^{(G)}(r))^2}{2\sigma^2} \Big) \delta(l_s, l_r) \ , \tag{6.17}$$

où dist est une fonction de distance et σ est un paramètre permettant de plus ou moins lisser les résultats. Ce terme de régularité pénalise la discontinuité des étiquettes entre des pixels ayant



(a) Image originale avec les graines. (d) Résultat de la segmentation.

(b) Graphe. (c) Coupe.

FIG. 6.7: Exemple de graphe et de segmentation. La capacité de chaque arc est traduite par son épaisseur. Cette figure provient de [Boykov 01a].

des intensités similaires ($|z_t^{(G)}(s) - z_t^{(G)}(r)| < \sigma$). Au contraire si les pixels ont des intensités éloignées, ce terme de pénalisation est faible. Un exemple de graphe est montré sur la figure 6.7.

De nombreuses autres méthodes ont été mises en place depuis cet article dans le but de segmenter des objets [Xu 03, Rother 04, Blake 04, Zabih 04] mais nous ne les détaillons pas ici.

6.4.2 Conlcusion : application à la détection et à la segmentation des objets en mouvement

Dans la partie précédente (partie I, chapitre 5), nous avons montré une application des algorithmes de coupe minimale/flot maximal à la segmentation des objets en mouvement. Le principe utilisé est très proche de celui de la méthode décrite ci-dessus. Dans notre cas les graines à l'intérieur de l'objet sont directement les pixels appartenant au cluster en mouvement à segmenter. Aucune graine représentant le fond n'est disponible et la distribution pour celui-ci est alors apprise sur l'ensemble de l'image. Au lieu d'utiliser des histogrammes sur la couleur, nous avons utilisés des mélanges de gaussiennes sur la couleur et sur le mouvement.

Dans cette partie nous allons développer une méthode de suivi utilisant les algorithmes de coupe minimale/flot maximal. Avant d'expliquer cette méthode, nous présentons dans le chapitre

suivant un rapide état de l'art sur les méthodes de suivi.

Chapitre 7

État de l'art sur le suivi d'objets

Ce chapitre présente un état de l'art des méthodes de suivi. Le but des méthodes de suivi est d'estimer au fil du temps des paramètres d'une cible (ou plusieurs) présente dans le champ de vision de la caméra et initialement détectée par un moyen quelconque. Les paramètres peuvent être divers : position dans l'image, à laquelle peuvent s'ajouter la taille et l'orientation apparente, l'attitude, l'apparence, etc. Pour qu'une méthode de suivi soit complètement automatique, il faut non seulement suivre la ou les cibles mais aussi les initialiser automatiquement par une méthode de détection et gérer leurs éventuels arrêts et/ou sorties du champ de la caméra.

De nombreuses méthodes de suivi existent. Une bonne et récente revue de ces techniques est [Yilmaz 06]. Dans cet article, les méthodes de suivi sont divisées en trois principales catégories : suivi de points, suivi de silhouette et suivi de fenêtres englobantes. Chacune de ces catégories a ses propres avantages et inconvénients. Dans ce chapitre nous ne nous intéresserons pas au suivi de points, puisque nous cherchons à suivre des objets préalablement détectés. Cependant certaines méthodes de suivi de points peuvent s'étendre directement au suivi d'objets puisqu'elles reposent sur la mise en correspondance entre des objets détectés et les objets suivis. Nous gardons donc globalement la classification de [Yilmaz 06] et divisons les méthodes de suivi en trois catégories nommées : suivi par appariement de détection ("detect-before-track"), suivi par segmentation dynamique et suivi déterministe par détection séquentielle.

7.1 Suivi par appariement de détections

Dans le cas où, à chaque instant, toutes les cibles visibles peuvent être détectées de façon fiable et rapide, le problème de suivi peut être défini comme un problème d'appariement de détections entre images successives. Les cibles candidates à chaque instant sont appelées observations. Elles sont généralement obtenues par soustraction de fond mais peuvent aussi être le résultat de méthodes de détection plus complexes (telle que la méthode de la partie I). Le suivi se fait alors en assignant les observations aux pistes en cours d'estimation. Ces méthodes, dites "detect-before-track" sont très populaires en pistage sonar et radar. Elles peuvent être déterministes ou probabilistes. On est confronté dans les deux cas à un problème combinatoire d'association.

7.1.1 Méthodes déterministes

Le principe des méthodes déterministes est d'associer les observations aux pistes en cours de suivi en minimisant une distance calculée sur certaines caractéristiques de l'objet. Les caractéristiques des objets couramment utilisées sont la proximité (hypothèse de déplacement limité) et l'apparence (similarité de forme et/ou de contenu photométrique et/ou de mouvement). Les modèles de l'objet basés sur l'apparence peuvent être des densités (histogrammes de couleur ou de contour), une carte de contours (contour ouvert ou fermé de l'objet) ou une combinaison de ces modèles.

Par exemple, dans [Haritaoglu 00], une information sur les contours contenus dans les objets est utilisée pour suivre des personnes. La mesure de distance entre les objets suivis et les observations est une mesure de corrélation calculée sur les contours de l'objet. Afin de rendre les caractéristiques invariantes aux translations, rotations et changements d'échelles, il est possible d'utiliser des histogrammes de couleurs ou de contours. Différentes mesures de distance entre les histogrammes de l'objet suivi et ceux d'une observation peuvent convenir : corrélation, distance de Bhattacharya ou divergence de Kullback-Leibler, les deux dernières donnant les meilleurs résultats. Enfin notons que la mise en correspondance peut aussi être basée sur le mouvement, calculé par des méthodes de flot optique [Sato 04].

Les méthodes citées ci-dessus permettent de mettre en correspondance des objets avec des observations entre deux images consécutives. Afin d'obtenir la trajectoire d'un objet tout le long d'une séquence et de gérer les occultations, on aura généralement recourt à un graphe treillis. Un objet à l'instant précédent est relié à un petit nombre d'observations à l'instant courant sur la base d'une mesure de distance comme celles utilisées par les méthodes décrites ci-dessus. Les liaisons entre observations et objet sont pondérées par la distance associée à cette mesure de similarité. L'idée est alors de trouver, pour une initialisation donnée, le meilleur chemin (celui de coût minimum) jusqu'à l'instant courant. Celui-ci est extrait par programmation dynamique (algorithme de Viterbi). Le suivi d'une seule cible peut être étendu au suivi multi-cible en rendant inter-dépendantes les extractions de pistes. Cela peut être fait séquentiellement : la meilleure piste, c'est-à-dire celle de coût minimum, est d'abord extraite et le graphe modifié pour accroître le coût des chemins passant sur les nœuds de cette piste. La seconde piste est alors extraite, et ainsi de suite [Pitié 05]. Cette approche permet de gérer les occultations d'un objet par un autre mais n'est pas adaptée à un suivi en ligne, c'est-à-dire à la volée. On parle alors de méthode "batch". Par ailleurs, les occultations par un élément non préalablement suivi ne sont pas gérées. Ce dernier point peut être amélioré en introduisant à chaque instant un nœud sans mesure permettant la prise en compte d'occultations ou de défaut de détection sur un intervalle de temps inférieur à un seuil fixé [Shafique 03].

7.1.2 Méthodes probabilistes

Les observations obtenues par un algorithme de détection sont très souvent corrompues par du bruit. De plus, le mouvement ou l'apparence d'un objet peut légèrement varier entre deux images consécutives. Les méthodes probabilistes permettent de gérer ces variations en ajoutant une incertitude au modèle de l'objet et aux modèles des observations. Le suivi d'une seule cible est alors obtenu par des méthodes de filtrage (filtres de Kalman, filtrage particulaire). Le suivi de plusieurs objets peut lui aussi se faire avec ces méthodes de filtrage mais une étape préalable d'association de l'objet avec l'observation la plus probable doit être ajoutée. Il existe de nombreuses méthodes

statistiques d'association [BarShalom 87, Cox 93]. Les deux techniques les plus utilisées sont le MHT ("Multiple Hypothesis Tracking") et le PDAF ("Probability Data Association Filtering").

L'approche MHT [Reid 79, Cox 93] est basée sur l'idée suivante : si une correspondance n'a été établie que pendant deux images il est possible que cette association soit incorrecte. Elle considère à chaque instant un ensemble d'hypothèses, chacune étant un ensemble d'association pistes-observations. Pour une hypothèse à l'instant précédent, chacune des pistes est prédite par filtrage de Kalman. Puis, pour chaque piste prédite, les observations suffisamment proches de la prédiction sont répertoriées. Ainsi, pour une initialisation donnée, on obtient un ensemble de pistes possibles, ces pistes utilisant les hypothèses définies à chaque instant. La meilleure piste sera alors celle ayant donnée les meilleures associations sur toute la période de temps. Là encore, il s'agit d'une méthode batch.

L'autre approche couramment utilisée est le PDAF (une seule piste) ou le JPDAF (plusieurs pistes). L'approche PDAF fait intervenir un état caché sur la position et la vitesse de la cible, état dont la loi *a posteriori* est estimée séquentiellement. Il s'agit donc d'une méthode bayésienne généralement vue comme une extension du filtrage de Kalman. Le JPDAF ajoute la possibilité de suivre plusieurs cibles en appliquant un PDAF par cible.

7.1.3 Avantages et limitations

L'ensemble des méthodes déterministes décrites dans cette section ont un faible coût de calcul (qui dépend bien entendu du temps de calcul de l'étape de détection) et gèrent l'apparition de nouvelles cibles et l'arrêt ou la disparition de cibles existantes. Certaines ne peuvent cependant pas être appliquées en ligne et traitent les séquences "off-line". L'inconvénient majeur de ces méthodes est qu'elles dépendent exclusivement de la qualité des détections. En cas de fausses ou de mauvaises détections, le suivi sera détérioré. De plus, l'hypothèse très restrictive qu'une cible génère au plus une observation à chaque instant doit souvent être faite. Or un objet devrait pouvoir être associé à plusieurs observations ou au contraire plusieurs objets devraient pouvoir être associés à une seule observation.

7.2 Segmentation dynamique

Les méthodes de suivi par segmentation dynamique sont utilisées lorsque l'on souhaite extraire la silhouette de la cible à chaque instant, et ce sans connaissance *a priori* sur sa forme. Ses approches reposent sur une succession de segmentations. Elles font généralement évoluer le contour de l'objet à l'instant précédent jusqu'à sa nouvelle position à l'instant courant. Ce type de suivi peut se faire avec des méthodes utilisant des modèles d'état ou avec des méthodes minimisant des fonctions d'énergie sur le contour.

7.2.1 Suivi avec une représentation explicite du contour

Un certain nombre de méthodes de suivi par segmentation dynamique cherche à représenter le contour par un ensemble de paramètres avant de le suivre par une méthode de filtrage. Il s'agit en fait de définir un modèle du contour de l'objet (généralement défini par la forme de l'objet) qui servira de modèle d'état à l'algorithme de filtrage. Les paramètres de mouvement de son contour sont mis à jour à chaque instant en maximisant une loi *a posteriori*.

Dans [Terzopoulos 93] des points de contrôle sont placés le long du contour à suivre. Le modèle d'état est alors représenté par leur dynamique. Cette dynamique donne une prédiction de ces points de contrôle (filtre de Kalman) qui est ensuite corrigée en utilisant des mesures de gradient de l'image. Une autre méthode, décrite dans [Isard 98], consiste à définir l'état comme un ensemble de splines et de paramètres de mouvement. Les mesures utilisées sont les contours de l'image calculés dans la direction normale à la silhouette. Les auteurs utilisent un algorithme de filtrage particulaire pour mettre à jour l'état. Cette méthode a été étendue au suivi de plusieurs objets et à la gestion des occultations dans [MacCormick 00].

7.2.2 Minimisation d'une fonction d'énergie sur contour : représentation implicite

Les méthodes décrites ci-dessus utilisent une représentation explicite du contour, et ne peuvent donc pas suivre les changements topologiques, la division ou la fusion de régions. Des méthodes basées sur la minimisation d'une fonctionnelle d'énergie permettent de suivre un contour ou une région en tenant compte des changements de topologie. La région occupée par la cible peut être définie par un masque binaire [Paragios 99b, Criminisi 06] ou par la courbe de niveau zéro d'une fonction à support continue [Paragios 99a, Shi 05]. La fonction d'énergie est définie en utilisant de l'information sous la forme de gradient temporel (flot optique) [Bertalmio 00, Mansouri 02, Cremers 03] ou en se basant sur l'apparence de l'objet et du fond (modélisation des intensités à l'intérieur et/ou à l'extérieur de l'objet) [Ronfard 94, Yilmaz 04]. Généralement cette fonction encourage les formes régulières s'appuyant sur des contours forts de l'image. L'avantage des méthodes utilisant les gradients temporels d'intensité de l'image est qu'elles permettent, de part le calcul du flot optique, de prédire la position du contour avant de procéder à la minimisation de la fonctionnelle, c'est-à-dire à la segmentation. Au contraire, les méthodes se basant sur des modélisations de l'apparence de l'image utilisent comme initialisation le contour de l'objet à l'instant précédent. Elles ne gèrent donc pas les grands mouvements des objets.

Cependant, toutes les méthodes décrites précédemment n'incluent pas de véritable cohérence temporelle et ne gèrent donc pas bien les occultations. Elles se servent tout au plus d'une prédiction basée sur le calcul du flot optique. Un filtrage stochastique utilisant une loi d'évolution temporelle et un processus de mesure permet de rendre le suivi plus robuste au bruit de l'image et aux occultations partielles [Niethammer 04]. Une méthode batch, utilisant une dynamique de l'objet, existe aussi pour le suivi de contour [Papadakis 07]. Elle permet un suivi robuste d'une cible et gère les occultations partielles et totales de la cible. La difficulté avec ces méthodes est de définir la loi dynamique.

Dans [Xu 02], les algorithmes de coupe minimale/flot maximal ont été utilisés pour minimiser une fonction d'énergie. Le principe est le suivant. Le contour de l'objet à l'instant précédent est dilaté. Un graphe est alors construit sur cette bande dilatée, un nœud étant associé à chaque pixel de la bande. Puis, tous les pixels sur le contour extérieur de la bande sont contraints à appartenir à un unique puits et tous ceux sur le contour intérieur à une unique source (figure 7.1). La fonction d'énergie associée à ce graphe ne comporte que des termes de régularité basés sur les gradient d'intensité de l'image. Les algorithmes de coupe minimale/flot maximal ont aussi été utilisés dans [Criminisi 06] pour faire de la segmentation de mouvement, c'est-à-dire pour segmenter successivement un objet au cours du temps en utilisant une information de mouvement.

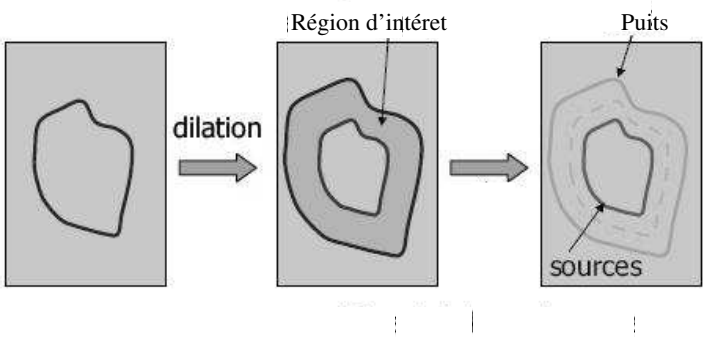


FIG. 7.1: Dilatation du contour et placement de la source et du puits [Xu 02]. Cette figure provient de [Xu 02].

7.2.3 Avantages et limitations

L'avantage principal des méthodes de suivi par segmentation dynamique est de fournir directement les segmentations de chaque objet suivi. Elles permettent de suivre tous types d'objets avec des formes diverses et s'adaptent relativement bien aux objets déformables. Les autres avantages et limites de ces méthodes dépendent de la représentation du contour. Une représentation explicite du contour permet de suivre un objet avec un coût de calcul assez faible et, si la dynamique utilisée est bonne, de gérer les occultations. Par contre elles ne s'adaptent pas bien aux changements de topologie. Les méthodes basées sur une représentation implicite sont robustes aux changements de topologie des objets mais ne sont pas robustes aux occultations. De plus la minimisation des fonctions d'énergie est généralement très coûteuse et la convergence vers un minimum global n'est pas nécessairement assurée. L'utilisation des algorithmes de coupe minimale/flot maximal donne une solution à ces deux derniers problèmes en assurant la convergence vers des minima globaux et cela avec un faible coût de calcul.

7.3 Suivi déterministe de fenêtre englobante par détection séquentielle

La dernière catégorie de méthodes évoquée dans ce chapitre regroupe les méthodes de suivi d'imagettes. Une imagette est une boîte (en général un rectangle mais parfois aussi une ellipse) entourant ou à l'intérieur de l'objet à suivre. Il s'agit en fait d'une petite portion de l'image. Les techniques de suivi considérées dans cette section sont basées sur la conservation de l'apparence (généralement couleur et/ou luminance) de l'objet pendant au moins deux instants consécutifs. Nous distinguons les algorithmes faisant l'hypothèse de conservation de l'apparence localement (en chaque point de l'objet) et ceux utilisant une caractérisation globale de la cible (hypothèse globale de conservation de l'apparence).

7.3.1 Suivi différentiel d'imagettes

Les méthodes décrites ci-dessous font l'hypothèse de conservation de la luminance d'un point physique visible entre deux instants consécutifs. L'approche la plus simple consiste à chercher la portion d'image la plus proche dans l'image courante de celle à l'instant précédent. La recherche se fait souvent autour de la position de l'objet à l'instant précédent. Ces approches sont appelées "block matching" ou "template matching". Elles font appel à une mesure de similarité telle que la corrélation ou la SSD ("sum of square differences"). Les intensités et les couleurs étant sensibles aux changements d'illumination, il est aussi possible d'exploiter les gradients de l'image. Le coût de calcul de ce type de méthodes peut être élevé. Il dépend de la taille du voisinage dans lequel l'imagette est recherchée.

En ajoutant une hypothèse de mouvement affine de l'imagette, l'approche KLT [Shi 94] permet de trouver l'objet à l'instant courant en minimisant la SSD par une méthode de type Gauss-Newton. Elle est restreinte au suivi d'imagette de petite taille (20x20 pixels). Pour suivre de plus grands objets, on peut appliquer la méthode précédente à un ensemble d'imagettes contenues dans la cible. Cela s'apparente à une estimation paramétrique de mouvement sur la base d'un champ épars de déplacements. Une autre approche pour suivre une région de forme prédéfinie est de calculer son déplacement (champ dense) par une méthode de type flot optique.

Plutôt que d'utiliser l'apparence de la cible à l'instant précédent, certains préféreront recourir à une imagette de référence, cette dernière pouvant être mise à jour au cours du temps. Cela permet de rendre ces méthodes plus robustes aux occultations. La remise à jour de l'imagette de référence doit de préférence être progressive ce qui n'est pas toujours un problème trivial [Jepson 03, Nguyen 04].

7.3.2 Suivi de distributions

Les problèmes de remise à jour évoqués ci-dessus sont moins cruciaux lorsque l'on se donne une caractérisation globale de l'apparence de la cible. La recherche de la cible à l'instant courant se base alors sur des distributions (histogrammes ou mélanges de gaussiennes) des couleurs dans une région de géométrie simple. La méthode la plus utilisée dans cette catégorie est le suivi par mean shift [Comaniciu 03c]. La distribution est un histogramme de couleur. L'algorithme consiste alors à déplacer une fenêtre d'analyse (noyau) de manière à trouver l'histogramme contenu dans la fenêtre qui coïncide le mieux avec l'histogramme de référence. La similarité entre l'histogramme de référence et l'histogramme candidat est mesurée par le coefficient de Battacharyya. Le déplacement du noyau se fait par montée de gradient itérative. Plus récemment, une méthode permettant le suivi de distributions sans calculer explicitement les distributions a été proposée dans [Boltz 07]. Elle consiste à ajouter, en plus de la couleur, une information sur la géométrie des objets. L'estimation de densités de probabilité en grande dimension n'étant pas triviale, les auteurs proposent une méthode permettant de calculer une distance de Kullback-Leibler entre deux densités de probabilité sans estimer explicitement ces densités.

7.3.3 Avantages et limitations

Les méthodes décrites dans cette section ont l'avantage d'exploiter des informations sur l'apparence photométrique des objets ce qui les rend robustes aux changements d'illumination. L'utilisation de distributions permet de rendre moins critique la phase de remise à jour de l'imagette de référence. Les techniques de suivi résultantes sont invariantes aux rotations et, dans une certaine

mesure, aux changements d'échelles. De plus, les algorithmes de suivi par mean shift sont peu coûteux. Cependant, la description d'une cible par des distributions n'est pas adaptée aux objets de petite taille. Enfin, toutes les méthodes de cette section ne gèrent pas l'initialisation automatique de nouveaux objets ni leurs arrêts et sorties du champ de la caméra.

7.4 Conclusion

Ce chapitre a passé en revu les principales techniques permettant de suivre un ou plusieurs objets dans une séquence d'images. Le but était de donner un bref apperçu des types de méthodes existants et non de faire une liste exhaustive de toute la littérature sur le sujet. Chaque catégorie de suivi a ses propres avantages et ses inconvénients. Il ressort de ce chapitre des points importants permettant de définir la qualité d'un algorithme de suivi.

Les caractéristiques d'un bon algorithme de suivi sont les suivantes. Tout d'abord la méthode doit être capable d'initialiser automatiquement les cibles et doit gérer les arrêts et les sorties du champ de la caméra. Elle doit de plus être robuste aux changements d'illumination et aux éventuels changements de topologie. La topologie n'est importante que dans le cas où une segmentation de la cible est exigée. Enfin, la méthode doit permettre de continuer à suivre la cible même en cas d'occultations partielles ou totales par un autre objet ou par le fond. La première caractéristique ne peut être obtenue qu'avec l'ajout d'une méthode de détection des objets, des observations étant apportées à chaque instant ou à des instants espacés par un faible pas de temps. Les changements d'illumination ou de photométrie ne sont bien pris en compte que par les méthodes utilisant des histogrammes ou des mélanges de gaussiennes sur l'intensité ou la couleur. Ces distributions doivent régulièrement être mises à jour. Enfin, la gestion des occultations n'est bien considérée qu'avec l'utilisation d'observations ou en ajoutant une loi dynamique au processus de suivi, la difficulté étant alors de déterminer cette loi. La plupart du temps les objets peuvent avoir un mouvement quelconque qu'aucune loi ne peut caractériser.

Dans le chapitre suivant, nous présentons une méthode de suivi à la volée basée sur la minimisation d'une fonction d'énergie par coupe minimale/flot maximal. Elle utilise des observations extérieures fournies par un module de détection (soustraction de fond ou méthode décrite dans la partie I de ce document). Ainsi, elle gère bien l'arrivée de nouveaux objets. Des distributions de couleurs et de mouvement rendent le suivi robuste aux changements d'illumination et permet de gérer les occultations partielles.

Chapitre 8

Suivi et segmentation par coupe minimale/flot maximal

Dans ce chapitre, nous mettons en place un algorithme de suivi d'objets utilisant des observations apportées par un module extérieur de détection d'objets en mouvement. Nous souhaitons que l'algorithme de suivi donne simultanément les cartes de segmentation des objets. En effet, si nous utilisons le détecteur décrit dans la première partie du document, il serait intéressant de pouvoir combiner la phase de segmentation et la phase de suivi. Les observations extérieures pourraient ainsi être les clusters en mouvement obtenus à la fin de la phase de partitionnement des points de la grille. Nous ne nous concentrons pas uniquement sur le détecteur de la partie I et dédions notre méthode de suivi à n'importe quelle méthode de détection d'objets. Nous montrerons, par exemple, des résultats pour lesquels les détections ont été obtenues par soustraction de fond.

Nous ne voulons pas ajouter de phase d'association entre les observations et les objets à l'algorithme de suivi. En effet, l'association pour le suivi multicibles peut s'avérer difficile, en particulier si nous utilisons le détecteur de la partie I. Dans ce cas, les observations seraient en effet des clusters en mouvement tandis que les pistes sont des objets segmentés. Au contraire, la méthode que nous mettons en place dans ce chapitre fait directement cette association : il s'agit d'un résultat de l'algorithme. Ce résultat d'association permet, de plus, de gérer automatiquement l'apparition de nouvelles pistes.

Le chapitre est organisé comme suit. La première section explique le principe général de l'algorithme et introduit les notations. L'algorithme de suivi utilise les algorithmes de coupe minimale/flot maximal expliqués au chapitre 6. Il repose donc sur des minimisations d'énergie. Il se décompose en deux étapes principales. La première, décrite dans la section 8.2 est dédiée à la segmentation et au suivi de chaque objet indépendamment. Une deuxième étape est rajoutée afin de séparer les objets ayant fusionné au cours de la première étape. Elle est expliquée dans la section 8.3. Dans la dernière section (section 8.4), des résultats expérimentaux sont présentés, d'une part sur une séquence relativement simple où les observations sont obtenues par soustraction de fond, et d'autre part sur des séquences plus complexes ayant un fond dynamique. Pour ces dernières, les observations sont le résultat du détecteur de la partie I. Il s'agit ainsi des résultats finaux de cette thèse traitant de la détection, la segmentation et le suivi d'objets mobiles dans des scènes complexes.

8.1 Principe

La méthode de suivi proposée se décompose en deux parties principales : une phase de segmentation et de suivi, et une phase de séparation des objets ayant fusionné. Les deux parties se basent sur une minimisation d'énergie par Graph Cuts. Avant d'expliquer le rôle de ces deux phases et leurs principes, nous présentons les caractéristiques utilisées pour représenter les objets et les observations.

8.1.1 Notations et caractérisation des observations

Une observation, provenant d'un module externe de détection, est définie comme un masque de pixels. On suppose que, à chaque instant t, m_t observations sont disponibles. La j^{eme} observation à l'instant t sera notée $\mathcal{M}_t^{(j)}$, $\mathcal{M}_t^{(j)} \subset \mathcal{P}$. Nous rappelons que $\mathcal{P}$ est l'ensemble des pixels de l'image. De la même façon, l'état de la i^{eme} piste à l'instant t, $\mathcal{O}_t^{(i)}$, est un masque de pixels, $\mathcal{O}_t^{(i)} \subset \mathcal{P}$, qui doit correspondre à l'objet $\mathcal{O}_{t-1}^{(i)}$ de l'instant $t-1$. À chaque instant, on suppose que k_t objets sont suivis.

Comme dans tout le document, chaque pixel de l'image à l'instant t est décrit par son vecteur de caractéristiques $\mathbf{z}_t(s) = (\mathbf{z}_t^{(G)}(s), \mathbf{z}_t^{(M)}(s), \mathbf{z}_t^{(C)}(s))$. Nous considérons pour nos résultats un champ épars de mouvement (calcul détaillé au chapitre 2), mais la méthode s'applique tout aussi bien à un champ dense.

Chaque observation $\mathcal{M}_t^{(j)}$ est caractérisée par le vecteur moyenne :

$$\overline{\mathbf{z}}_t^{(j)} = \frac{\sum_{s \in \mathcal{M}_t^{(j)}} \mathbf{z}_t(s)}{|\mathcal{M}_t^{(j)}|} \ , \tag{8.1}$$

tandis que les objets seront caractérisés par des distributions définies plus loin.

8.1.2 Principe général

Nous proposons dans ce chapitre une méthode de suivi d'objets basée sur la minimisation d'une fonction d'énergie par une approche de coupe minimale/flot maximal. Les caractéristiques sur lesquelles repose le suivi sont la couleur et le mouvement, regroupées dans le terme d'apparence. Chaque objet est suivi indépendamment, un graphe étant construit pour chacun d'entre eux. Nous avons choisi de suivre indépendamment chaque objet afin de distinguer les occultations partielles ou totales d'une cible par une autre de la fusion de plusieurs cibles en une seule. En effet, ces deux cas de figure peuvent s'interpréter de la même façon en terme d'étiquetage des pixels et doivent donc être traités différemment : supposons que l'on souhaite minimiser une énergie multi-étiquettes, où chaque étiquette correspond à un objet ou au fond, et que chaque pixel de l'image ne puisse être associé qu'à une seule de ces étiquettes. Nous considérons ici des objets déformables, leur topologie pouvant changer au cours du temps. Considérons les deux cas suivants (figure 8.1). Tout d'abord, supposons deux objets d'apparence (photométrie et mouvement) très proche se retrouvant juxtaposés dans l'image à l'instant t (de part la déformation de l'un des objets ou à cause d'un changement de profondeur de la caméra) alors qu'ils étaient séparés à l'instant $t-1$. Comme ils sont similaires, le gradient de couleur, d'intensité ou de mouvement entre les deux objets est très faible. Si le terme de régularisation est basé sur les gradients, il va encourager tous

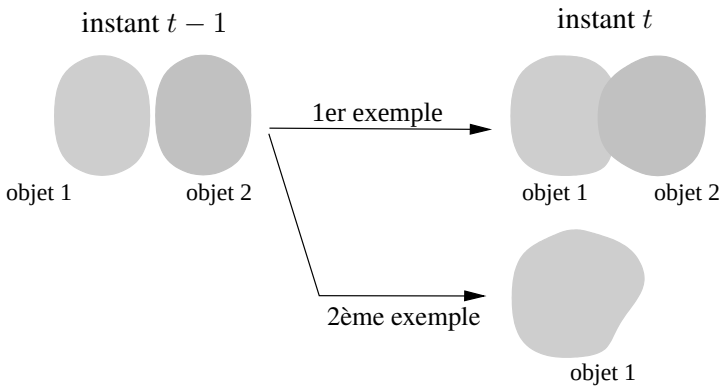


FIG. 8.1: Fusion ou occultation ?

les pixels des deux objets à avoir la même étiquette. Ainsi, tous les pixels représentant les deux objets risquent d'avoir la même étiquette, celle-ci ne correspondant qu'à un seul des deux objets. L'autre objet n'étant plus représenté par aucun pixel peut alors être supposé occulté. Considérons maintenant un deuxième exemple. Supposons qu'au temps t un objet occulte totalement une autre cible (d'apparence similaire ou non au premier). Après minimisation de l'énergie, tous les pixels concernés seront associés (de part leur étiquette) à l'objet qui occulte, et l'objet occultant ne sera plus représenté par aucun pixel. Ainsi, dans les deux cas, l'étiquetage obtenu fait apparaître un objet dominant occultant le second objet.

Afin de différencier ces deux cas, nous proposons de suivre indépendamment chaque objet en considérant une fonction d'énergie par piste. Chaque pixel peut ainsi être associé à k_t étiquettes à chaque instant t. Le résultat pour le deuxième exemple (occultation) est le même qu'avec une énergie multi-étiquettes. En revanche, pour le premier exemple (juxtaposition de deux objets), le résultat est différent. En effet, les pixels représentant une partie du premier objet sont associés à l'étiquette correspondant à cet objet, tandis que les pixels représentant une partie du second objet sont associés à l'étiquette correspondant au second objet. Ainsi, suivre chaque objet indépendamment permet de différencier les deux cas. Bien entendu, ce résultat n'est toujours pas satisfaisant. En effet, dans le cas du premier exemple, les deux objets se retrouvent égaux à l'union des deux cibles et nous n'obtenons donc pas la véritable segmentation de chaque objet. Pour les différencier, nous ajouterons à notre algorithme une deuxième étape de minimisation d'énergie qui aura pour rôle la séparation des objets ayant pu fusionner.

8.1.3 Principe de la méthode de suivi

Le principe de l'algorithme de suivi est le suivant. Supposons que l'on cherche à suivre le i^{eme} objet de l'instant $t-1$, $\mathcal{O}_{t-1}^{(i)}$. La première étape de l'algorithme consiste à faire une prédiction, $\mathcal{O}_{t|t-1}^{(i)}$, de cet objet. Comme les objets et les observations, la prédiction est un masque de pixels $\mathcal{O}_{t|t-1}^{(i)} \subset \mathcal{P}$. Elle est calculée en déplaçant chaque pixel du masque avec un vecteur $\mathbf{d}_{t-1}^{(i)}$ égal à la

moyenne des vecteurs de mouvement $\mathbf{d}_{t-1}^{(i)}$ sur l'objet à l'instant précédent :

$$\mathbf{d}_{t-1}^{(i)} = \frac{\sum_{s \in \mathcal{O}_{t-1}^{(i)}} \mathbf{z}_{t-1}^{(M)}(s)}{|\mathcal{O}_{t-1}^{(i)}|} \ . \tag{8.2}$$

Nous utilisons la moyenne des vecteurs de mouvement car nous considérons ici un champ épars de mouvement. Cependant, avec un champ dense, chaque pixel pourrait être déplacé indépendamment par son propre vecteur de mouvement.

Les valeurs du ou des vecteurs de déplacement sont arrondies à l'entier le plus proche afin que la prédiction de chaque pixel tombe sur la grille des pixels. La prédiction est finalement définie par :

$$\mathcal{O}_{t|t-1}^{(i)} = \{s + \mathbf{d}_{t-1}^{(i)}, s \in \mathcal{O}_{t-1}^{(i)}\} \ . \tag{8.3}$$

En utilisant cette prédiction, les observations et certaines caractéristiques de l'objet $\mathcal{O}_{t-1}^{(i)}$, un graphe et une fonction d'énergie sont créés. La minimisation de la fonction d'énergie par un algorithme de coupe minimale/flot maximal [Boykov 01b] permet d'obtenir le nouvel objet à l'instant t, $\mathcal{O}_t^{(i)}$. L'utilisation des "Graph Cuts" permet non seulement de trouver la position de l'objet à l'instant t mais aussi d'obtenir sa segmentation. De plus, l'ajout des observations dans la fonction d'énergie a pour conséquences l'amélioration de la segmentation, la correction de la prédiction et donne directement l'association entre chaque observation et l'objet à l'instant précédent. Cette association permettra la création de nouveaux objets à suivre.

Le principe de l'algorithme est résumé sur le schéma de la figure 8.2.

8.1.4 Principe de la méthode de séparation des objets fusionnés

À la fin de la phase de suivi, certains objets peuvent avoir fusionné. Cela se traduit par une superposition, complète ou partielle, des masques de pixels des objets : $\cap_{i=1\ldots k_t} \mathcal{O}_t^{(i)} \neq \emptyset$. La fusion est généralement dûe aux caractéristiques similaires des objets. Afin de continuer à suivre chaque objet indépendamment, il est nécessaire de séparer les objets fusionnés. Pour cela, une seconde fonction d'énergie est ajoutée. Il s'agira cette fois-ci d'un problème multi-étiquettes, le graphe contenant autant de sources qu'il y a d'objets fusionnés.

8.2 Fonction d'énergie pour le suivi de chaque objet

Cette section présente l'algorithme de suivi et de segmentation mis en place. Nous considérons ici que l'on suit l'objet i de l'instant $t-1$. La méthode décrite doit en effet être successivement appliquée à chacun des objets suivis.

Nous nous plaçons dans le cadre de la minimisation d'énergie par coupe minimale/flot maximal. Le but de la méthode est d'attribuer une étiquette "fond" ou "objet" à chaque pixel de l'image à l'instant t. Pour cela, un graphe associé à une fonction d'énergie est construit. Cette section décrit dans un premier temps la forme du graphe puis détaille la fonction d'énergie. À la fin de la section, nous montrons comment cette technique permet de gérer automatiquement la création de nouvelles pistes.

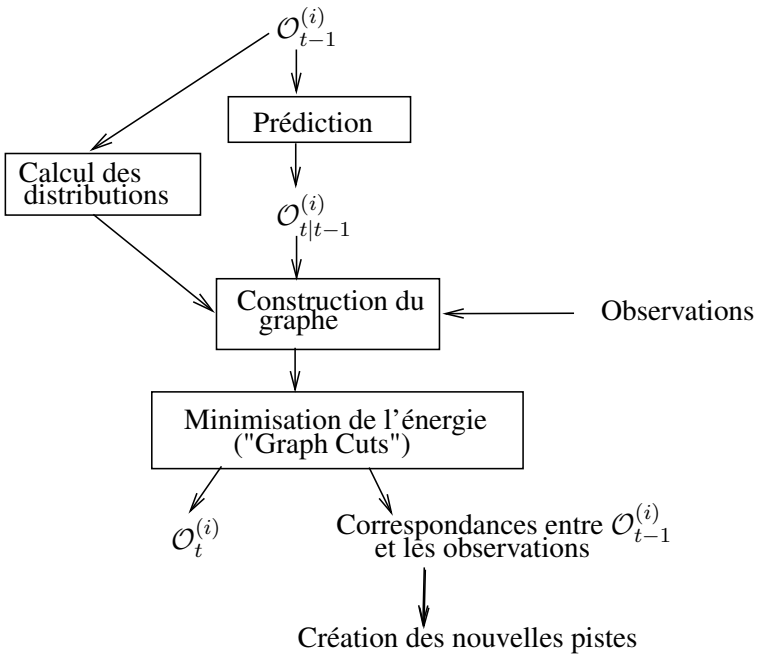


FIG. 8.2: Principe de l'algorithme de suivi.

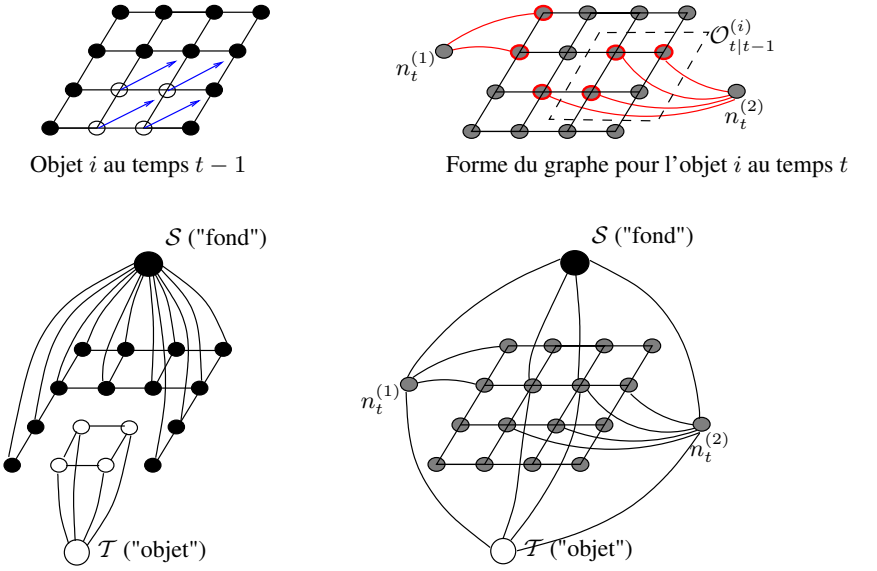
8.2.1 Forme du graphe

Le graphe non orienté $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ est composé d'un ensemble de nœuds (ou sommets) $\mathcal{V}_t$ et d'un ensemble d'arcs (ou arêtes) $\mathcal{E}_t$. La figure 8.3 montre la forme du graphe décrit ci-après. L'ensemble des sommets se divisent en trois groupes. Le premier, $\{\mathcal{S}, \mathcal{T}\}$, contient les deux terminaux (la source et le puits). Le second contient les N nœuds correspondant à la grille de pixels $\mathcal{P}$. Enfin, le troisième groupe représente la nouveauté majeure de la méthode de suivi. En effet, il permet d'inclure dans le graphe les observations obtenues par le module de détection. Ce groupe contient m_t sommets $n_t^{(j)}$, $j = 1 \dots m_t$, chacun associé à une observation $\mathcal{M}_t^{(j)}$. Chaque nœud est appelé **nœud d'observation**. L'ensemble complet des nœuds du graphe s'écrit finalement :

$$\mathcal{V}_t = \{\mathcal{S}, \mathcal{T}\} \bigcup \mathcal{P} \bigcup_{j=1}^{m_t} n_t^{(j)} \quad . \tag{8.4}$$

L'ensemble des arcs du graphe peut être lui aussi décomposé en trois groupes. Le premier, $\mathcal{E}_{term}$, contient l'ensemble des arcs reliant les terminaux à tous les autres nœuds. Le deuxième, $\mathcal{E}_{\mathcal{P}}$, regroupe les arêtes (s, r) non orientées reliant les sommets voisins de $\mathcal{P}$. On considère ici un 8-voisinage, mais la méthode s'étend très bien à d'autres types de voisinage. Notons que, pour plus de clarté, l'ensemble des graphes représentés sur les figures montrent seulement des 4-voisinages. Le troisième ensemble d'arcs, $\mathcal{E}_{\mathcal{M}_t^{(j)}}$, contient les arêtes $(s, n_t^{(j)})$, avec $s \in \mathcal{M}_t^{(j)}$, reliant les nœuds d'observation aux pixels de l'image appartenant aux détections. Finalement, l'ensemble des arcs du graphe s'écrit :

$$\mathcal{E}_t = \mathcal{E}_{term} \bigcup \mathcal{E}_{\mathcal{P}} \bigcup_{j=1}^{m_t} \mathcal{E}_{\mathcal{M}_t^{(j)}} \quad . \tag{8.5}$$



Objet i au temps $t-1$ — Forme du graphe pour l'objet i au temps t

Graphe final (coupé) de l'objet i au temps $t-1$ — Graphe initial pour l'objet i au temps t
(Pour plus de clarté, tous les arcs ne sont pas dessinés)

FIG. 8.3: Présentation du graphe utilisé. (a) Résultat de la minimisation d'énergie à l'instant $t-1$. Seuls les sommets correspondant aux pixels de l'image sont montrés. L'étiquette des nœuds blancs est "objet" et celle des nœuds noirs est "fond". En bleu est montré en chaque pixel de l'objet le vecteur de mouvement moyen qui permet la prédiction. (b) Graphe coupé correspondant à (a) avec les terminaux. Les arcs coupés ne sont pas dessinés. (c) Forme du graphe à l'instant t. Les terminaux et leurs arcs associés ne sont pas montrés. Les nœuds ($n_t^{(1)}$ et $n_t^{(2)}$) associés aux deux observations détectées $\mathcal{M}_t^{(1)}$ et $\mathcal{M}_t^{(2)}$ sont ajoutés à l'ensemble des sommets correspondant aux pixels. Les sommets en rouge représentent les pixels appartenant aux observations. Les arcs reliant ces sommets aux sommets correspondant aux observations apparaissent en rouge. Cette figure montre également les sommets appartenant à la prédiction. (d) Graphe avec les terminaux associés à (c). Pour des raisons de lisibilité, tous les arcs entre les terminaux et les autres sommets ne sont pas montrés.

Le suivi et la segmentation d'un objet i passent par l'étiquetage de l'ensemble $\mathcal{V}_t \backslash \{\mathcal{S}, \mathcal{T}\}$ des nœuds du graphe. Nous rappelons que chaque objet est suivi indépendamment, ce qui entraîne la création d'un graphe par objet. Un étiquetage binaire (une seule source et un seul puits) est considéré ici, les deux étiquettes étant "fond" et "objet". Pour l'objet i, l'étiquette en un pixel s à l'instant t sera notée $l_{s,t}^{(i)}$, et celle en un nœud d'observation $n_t^{(j)}$ sera $l_{j,t}^{(i)}$. Enfin, l'ensemble des étiquettes à l'instant t sera $L_t^{(i)}$.

8.2.2 Fonction d'énergie

Pour trouver la coupe minimale dans le graphe défini précédemment, la capacité de chacun de ces arcs doit être définie. Dans le chapitre 6, nous avons montré qu'il y a équivalence entre coupe

minimale et flot maximal dans un graphe et que trouver la coupe minimale peut se traduire par la minimisation d'une fonction d'énergie.

Nous considérons une fonction d'énergie de forme classique contenant un terme unaire d'attache aux données $R_{s,t}^{(i)}$ et un terme binaire de régularisation $B_{s,r,t}^{(i)}$:

$$E_t^{(i)}(L_t^{(i)}) = \sum_{s \in \mathcal{V}_t \setminus \{\mathcal{S},\mathcal{T}\}} R_{s,t}^{(i)}(l_{s,t}^{(i)}) + \sum_{(s,r) \in \mathcal{E}_t \setminus \mathcal{E}_{term}} B_{s,r,t}^{(i)} \delta(l_{s,t}^{(i)}, l_{r,t}^{(i)}) \ . \tag{8.6}$$

La fonction δ est la même que celle définie par l'équation (6.14).

8.2.2.1 Terme d'attache aux données

Le rôle du terme d'attache aux données est de prendre en compte l'apparence de l'objet et de traduire les observations et la prédiction dans la fonction d'énergie. Une information sur les données ne sera ajoutée que pour les sommets $n_t^{(j)}$, $j = 1 \dots m_t$, associés aux détections et pour les sommets $s \in \mathcal{P}$ appartenant à la prédiction $\mathcal{O}_{t|t-1}^{(i)}$. Aucune information *a priori* n'est donnée aux autres nœuds. Leur étiquette sera simplement déterminer par le terme binaire de régularisation. Il s'agit en fait d'un problème d'évolution d'un objet prédit en fonction d'observations externes et des contours de l'image. Le terme d'attache aux données se décompose donc en deux parties :

$$\sum_{s \in \mathcal{V}_t} R_{s,t}^{(i)}(l_{s,t}^{(i)}) = \sum_{s \in \mathcal{O}_{t|t-1}^{(i)}} -\ln(p_1^{(i)}(s, l_{s,t}^{(i)})) + \sum_{j=1}^{m_t} -\ln(p_2^{(i)}(j, l_{j,t})) \ . \tag{8.7}$$

Nous détaillons ci-après les distributions p_1 et p_2 de l'équation précédente. L'objet à l'instant t doit avoir une apparence proche de celle de l'objet à l'instant $t-1$. L'apparence regroupe les caractéristiques de couleur et de mouvement. On va donc définir les distributions de couleur, $p_{t-1}^{(i,C)}$, et de mouvement, $p_{t-1}^{(i,M)}$, pour l'objet $\mathcal{O}_{t-1}^{(i)}$. Ces distributions sont des mélanges de gaussiennes calculés avec un algorithme EM (Expectation Maximization) sur l'ensemble de vecteurs $\{\mathbf{z}_{t-1}^{(C)}(s)\}_{s \in \mathcal{O}_{t-1}^{(i)}}$ pour la distribution de couleur et sur les vecteurs $\{\mathbf{z}_{t-1}^{(M)}(s)\}_{s \in \mathcal{O}_{t-1}^{(i)}}$ pour la distribution de mouvement. Bien entendu, dans le cas où le champ de mouvement est épars, seuls les pixels sur lesquels un vecteur de mouvement a été calculé sont pris en compte. Nous considérons les informations de couleur et de mouvement indépendantes. Cela engendre finalement la distribution de l'objet suivante :

$$p_{t-1}^{(i)}(\mathbf{z}_t(s)) = p_{t-1}^{(i,C)}(\mathbf{z}_t^{(C)}(s))\ p_{t-1}^{(i,M)}(\mathbf{z}_t^{(M)}(s)) \ . \tag{8.8}$$

Si le mouvement est épars, on ne considère, pour les pixels où le mouvement n'est pas défini, que la distribution de couleur :

$$p_{t-1}^{(i)}(\mathbf{z}_t(s)) = p_{t-1}^{(i,C)}(\mathbf{z}_t^{(C)}(s)) \ .$$

Avec les distributions précédentes, les probabilités d'appartenance de chaque pixel et observation à l'objet peuvent être estimées. Ces probabilités vont être comparées à la probabilité d'appartenance au fond. Ainsi les distributions d'apparence du fond doivent également être calculées. Les deux distributions pour la couleur et le mouvement sont notées $q_{t-1}^{(i,C)}$ et $q_{t-1}^{(i,M)}$ respectivement. Là encore il s'agit de mélanges de gaussiennes calculés avec un algorithme EM. Pour la couleur,

le mélange est calculé sur l'ensemble des vecteurs $\{\mathbf{z}_{t-1}^{(C)}(s)\}_{s\in\mathcal{P}\setminus\mathcal{O}_{t-1}^{(i)}}$, tandis que pour le mouvement c'est l'ensemble $\{\mathbf{z}_{t-1}^{(M)}(s)\}_{s\in\mathcal{P}\setminus\mathcal{O}_{t-1}^{(i)}}$ qui est considéré. La distribution finale pour le fond est :

$$q_{t-1}^{(i)}(\mathbf{z}_t(s)) = q_{t-1}^{(i,C)}(\mathbf{z}_t^{(C)}(s))\; q_{t-1}^{(i,M)}(\mathbf{z}_t^{(M)}(s)) \;\; . \tag{8.9}$$

Nous pouvons maintenant définir la distribution p_1 appliquée aux sommets appartenant à la prédiction :

$$p_1^{(i)}(s,l) = \begin{cases} p_{t-1}^{(i)}(\mathbf{z}_t(s)) \text{ si } l = \text{“objet”}, \\ q_{t-1}^{(i)}(\mathbf{z}_t(s)) \text{ si } l = \text{“fond”} \;\; . \end{cases} \tag{8.10}$$

Une observation ne doit apporter de l'information que si son apparence est proche de l'objet en cours de suivi. La forme de la distribution p_2 aura donc la même forme que p_1. Cependant une observation n'est caractérisée que par son vecteur moyen $\overline{\mathbf{z}}_t^{(j)}$ (équation 8.1). La distribution p_2 appliquée à chaque nœud d'observation est finalement définie par :

$$p_2^{(i)}(j,l) = \begin{cases} p_{t-1}^{(i)}(\overline{\mathbf{z}}_t^{(j)}) \text{ si } l = \text{“objet”}, \\ q_{t-1}^{(i)}(\overline{\mathbf{z}}_t^{(j)}) \text{ si } l = \text{“fond”} \;\; . \end{cases} \tag{8.11}$$

L'utilisation simple de la moyenne pour les observations peut paraître surprenante. Ce choix a été fait car un seul nœud est défini pour chaque observation, et ceci pour permettre la création des nouvelles pistes (sous-section 8.2.3). Cette représentation simpliste nous a permis d'obtenir des résultats satisfaisants, mais un travail futur devrait se concentrer sur une caractérisation plus précise des observations, sans pour autant ajouter des nœuds supplémentaires au graphe.

8.2.2.2 Terme de régularisation

Le terme binaire de régularisation sert à lisser la segmentation de l'objet recherché. Nous avons choisi d'utiliser la forme du terme binaire la plus classique [Boykov 01a] en segmentation d'image (équation 6.17). Elle s'appuie sur le gradient d'intensité de l'image. En introduisant le paramètre σ_T traduisant en quelque sorte la quantité de lissage voulue, le terme binaire aura pour rôle la pénalisation de la discontinuité des étiquettes entre deux pixels s et r ayant des intensités similaires ($|\mathbf{z}_t^{(G)}(s) - \mathbf{z}_t^{(G)}(r)| < \sigma_T$).

Nous définissons différemment le terme binaire appliqué aux arcs de $\mathcal{E}_\mathcal{P}$ et aux arcs de $\bigcup_{j=1}^{m_t} \mathcal{E}_{\mathcal{M}_t^{(j)}}$. La différence vient une nouvelle fois de la caractérisation des observations par une moyenne. Ainsi, le terme binaire est donné pour un arc $(s,r) \in \mathcal{E}_\mathcal{P}$ par :

$$B_{s,r,t}^{(i)} = \lambda_1 \frac{1}{\text{dist}(s,r)} e^{-\frac{\|\mathbf{z}_t^{(G)}(s) - \mathbf{z}_t^{(G)}(r)\|^2}{\sigma_T^2}} \;\; , \tag{8.12}$$

où $dist$ est une fonction de distance et par :

$$B_{s,n_t^{(j)},t}^{(i)} = \lambda_2 \; e^{-\frac{\|\mathbf{z}_t^{(G)}(s) - \overline{\mathbf{z}}_t^{(j,G)}\|^2}{\sigma_T^2}} \tag{8.13}$$

pour un arc $(s, n_t^{(j)}) \in \mathcal{E}_{\mathcal{M}_t^{(j)}}$. Comme dans [Blake 04], le paramètre σ_T vaut

$$\sigma_T = 4 \cdot \langle (\mathbf{z}_t^{(G)}(s) - \mathbf{z}_t^{(G)}(r))^2 \rangle \ , \tag{8.14}$$

où $\langle . \rangle$ représente la moyenne sur une boîte englobant l'objet. Les paramètres λ_1 et λ_2 ont une influence non négligeable sur les résultats. Leur valeur sera discutée dans la section des résultats (section 8.4).

8.2.2.3 Minimisation de l'énergie

L'étiquetage final des sommets du graphe est obtenu en minimisant l'énergie avec l'algorithme "Expansion Move" de [Boykov 01b] (chapitre 6) :

$$\hat{L}_t^{(i)} = \arg\min_{L_t^{(i)}} E_t^{(i)}(L_t^{(i)}). \tag{8.15}$$

Cet étiquetage donne directement le nouvel objet i à l'instant t correspondant à l'objet i de l'instant $t-1$:

$$\mathcal{O}_t^{(i)} = \{s \in \mathcal{P} : \hat{l}_{s,t}^{(i)} = \text{“objet”}\}. \tag{8.16}$$

8.2.3 Création automatique de nouvelles pistes

Comme nous l'avons déjà évoqué, un avantage de notre méthode est qu'elle permet de prendre en compte des observations extérieures sans avoir à associer préalablement les observations avec les objets suivis. L'utilisation de nœuds d'observation permet de plus la création automatique de nouvelles pistes de suivi. En effet, l'étiquetage final des nœuds d'observation permet de savoir si chaque observation au temps courant correspond ou non à au moins un objet de l'instant précédent. Si après la minimisation d'énergie pour un objet i, un sommet $n_t^{(j)}$ a pour étiquette "objet" ($l_{j,t}^{(i)}$ ="objet"), cela signifie que l'observation j correspond à l'objet i de l'instant $t-1$: sa position et son apparence concordent avec la position et l'apparence de l'objet. Au contraire si le nœud a pour étiquette "fond", alors l'objet et l'observation ne sont pas associés.

Si pour tous les objets ($i = 1, \ldots, k_{t-1}$), l'observation est associée au fond après minimisation des énergies, alors l'observation ne correspond à aucun objet et une nouvelle piste peut être créée. Le nombre de pistes devient alors $k_t = k_{t-1} + 1$, et le nouvel objet d'indice k_t est défini par $\mathcal{O}_t^{(k_t)} = \mathcal{M}_t^{(j)}$.

Dans la pratique, la création d'une nouvelle piste ne sera validée que si le nouvel objet créé est mis en correspondance avec au moins une observation dans l'image suivant la création, c'est-à-dire si $\exists\, j \in \{1 \ldots m_{t+1}\}$ tel que $l_{j,t+1}^{(i)} =$ "objet".

8.3 Séparation des objets ayant fusionné

Après l'utilisation de la première fonction d'énergie, il est possible que les masques de segmentation de plusieurs objets se superposent sans pour autant que ces objets ne s'occultent. Nous appelons cela le problème de fusion d'objets. Ces fusions peuvent avoir plusieurs raisons : i) une seule observation correspond à plusieurs objets ; ii) les objets sont juxtaposés dans l'image ; iii)

les apparences des objets sont trop proches. Un exemple de ce problème est illustré par la figure 8.7.

Il est souhaitable de continuer à suivre chaque objet séparément. Pour cela nous ajoutons à notre algorithme une phase de séparation des objets fusionnés utilisant elle aussi une minimisation d'énergie par une technique de coupe minimale/flot maximal. Comme dans la section précédente nous commençons par présenter la forme du graphe et détaillons ensuite la fonction d'énergie qui sera minimisée.

8.3.1 Forme du graphe

Supposons qu'à l'instant t des objets aient fusionné :

$$\cap_{i=1\ldots k_t} \mathcal{O}_t^{(i)} \neq \emptyset \ .$$

L'hypothèse faite ici est qu'un pixel ne peut appartenir au plus qu'à un seul objet. On va donc chercher à trouver à quel objet un pixel a le plus de chance d'appartenir. Pour cela, on introduit un nouveau graphe $\tilde{G}_t = (\tilde{\mathcal{V}}_t, \tilde{\mathcal{E}}_t)$. Notons

$$\mathcal{F} = \{i \in \{1 \ldots k_t\} | \exists j \neq i \text{ tel que } \mathcal{O}_t^{(i)} \cap \mathcal{O}_t^{(j)} \neq \emptyset\}. \tag{8.17}$$

L'ensemble des nœuds $\tilde{\mathcal{V}}_t$ du graphe se décompose en deux groupes. Le premier groupe correspond aux terminaux. Le nombre de terminaux est égal $|\mathcal{F}|$. Ils sont notés $\mathcal{S}_i$, avec $i = 1, \ldots, |\mathcal{F}|$. Le deuxième groupe contient l'ensemble des pixels appartenant aux masques des objets fusionnés : $\cup_{i\in\mathcal{F}} \mathcal{O}_t^{(i)}$. L'ensemble des nœuds du graphe s'écrit donc :

$$\tilde{\mathcal{V}}_t = \{\mathcal{S}_i\}_{i=1\ldots|\mathcal{F}|} \bigcup_{i\in\mathcal{F}} \mathcal{O}_t^{(i)} \ . \tag{8.18}$$

L'ensemble des arêtes du graphe regroupe les arcs reliant les terminaux à tous les nœuds de $\cup_{i\in\mathcal{F}} \mathcal{O}_t^{(i)}$ et les arcs reliant ces derniers à leurs 8 nœuds voisins. Un exemple de graphe, dans lequel un système de 4 voisins est représenté, est montré sur la figure 8.4.

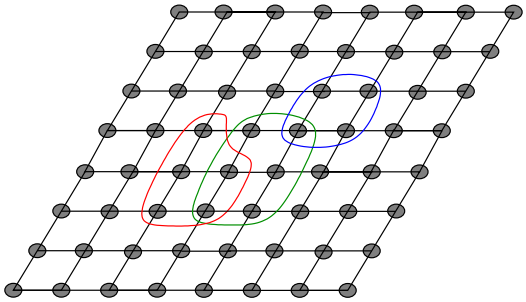
8.3.2 Énergie multi-étiquettes

Le but du deuxième graphe et de sa fonction d'énergie est d'associer chaque pixel des objets fusionnés à un seul objet. Il s'agit d'un problème multi-étiquettes (le graphe a plusieurs sources) où chaque étiquette "objet i" correspond à un des objets de l'ensemble $\mathcal{F}$. Une étiquette ψ_s doit être attachée à chaque sommet, l'étiquetage de l'ensemble des sommets s'écrivant $\tilde{L}_t = \{\psi_s, s \in \tilde{\mathcal{V}}_t\}$. L'énergie à minimiser pour obtenir l'étiquetage final contient un terme unaire d'attache aux données et un terme binaire de régularisation. Elle s'écrit :

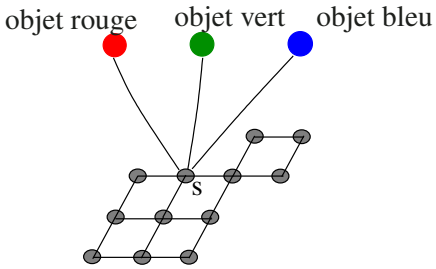
$$\tilde{E}_t(\tilde{L}_t) = \sum_{s\in\tilde{\mathcal{V}}_t} -\ln(p_3(s,\psi_s)) + \lambda_3 \sum_{(s,r)\in\tilde{\mathcal{E}}_t} \frac{1}{\mathrm{dist}(s,r)} e^{-\frac{\|\mathbf{z}_t^{(G)}(s) - \mathbf{z}_t^{(G)}(r)\|^2}{\sigma_3^2}} (1 - \delta(\psi_s, \psi_r)) \ , \tag{8.19}$$

où le paramètre σ_3 est égal à :

$$\sigma_3 = 4 \cdot \langle (\mathbf{z}_t^{(C)}(s) - \mathbf{z}_{r,t}^{(C)}(s))^2 \rangle \ , \ (s,r) \in \tilde{\mathcal{E}} \ . \tag{8.20}$$



(a) Résultat de la phase de suivi.
3 objets ont fusionnés.



(b) Forme du graphe correspondant.
(Tous les sommets sont, comme le nœud s, reliés aux trois sources)

FIG. 8.4: Présentation du graphe utilisé. (a) Résultat de l'algorithme de suivi. Trois objets se superposent. (b) Forme du graphe correspondant à la deuxième fonction d'énergie. Il contient 3 terminaux (3 sources), chacun relié à l'ensemble des pixels. Pour plus de clarté, tous les arcs ne sont pas montrés.

Les objets à l'instant t doivent avoir une apparence proche de celle des objets correspondant à l'instant $t-1$. Cependant, la distribution p_3 ne peut pas simplement dépendre de l'apparence. En effet, des objets se sont retrouvés fusionnés après la minimisation de la première fonction d'énergie principalement parce qu'ils avaient des apparences similaires. Il faut donc trouver un autre critère permettant de distinguer les objets. Nous avons choisi pour cela de privilégier la prédiction et définissons la distribution p_3 par :

$$p_3(s,\psi) = \begin{cases} p_{t-1}^{(\psi)}(\mathbf{z}_t(t)) & \text{si } s \notin \mathcal{O}_{t|t-1}^{(\psi)}, \\ 1 & \text{sinon .} \end{cases} \tag{8.21}$$

Ainsi, un pixel appartenant à la prédiction d'un objet suivi est fortement encouragé à appartenir au masque de segmentation de cet objet dans l'étiquetage final. Si, par contre, il n'appartient pas à la prédiction, c'est l'apparence qui permettra de décider son étiquette finale.

L'énergie multi-étiquettes définie ci-dessus est minimisée en utilisant l'algorithme "Expansion Move" [Boykov 98, Boykov 01b] en se ramenant itérativement dans un cas binaire (chapitre 6) : la coupe minimale est obtenue en séparant successivement chaque source de toutes les autres.

8.4 Résultats Expérimentaux

Dans cette section, les résultats de la méthode de suivi sont présentés. Dans un premier temps, nous considérerons que les observations sont obtenues par soustraction de fond. Dans un deuxième temps, la méthode de suivi sera combinée avec la méthode de détection des objets mobiles de la première partie de ce document. Dans tous les résultats, une couleur est associée à chaque objet suivi. Elle ne dépend que de l'ordre dans lequel les pistes sont initialisées.

8.4.1 Suivi d'objets détectés par soustraction de fond

Dans un premier temps, nous montrons des résultats de notre méthode sur une séquence de données de PETS 2006 (séquence 1, caméra 4). Les observations sont obtenues par simple soustraction de fond (avec une image de référence). Chaque observation équivaut à un élément connexe de la carte de détection résultant de la soustraction de fond (figure 8.5). Les éléments connexes sont obtenus en utilisant la méthode dite "gap/mountain" décrite dans [Wang 00] et en ignorant les petits objets.

(a) (b) (c)

FIG. 8.5: Observations obtenues par soustraction de fond. (a) Image de référence. (b) Image courante. (c) Résultat de la soustraction de fond (les pixels en noir sont les pixels ayant changé) et observations déduites (régions entourées d'une boîte englobante rouge).

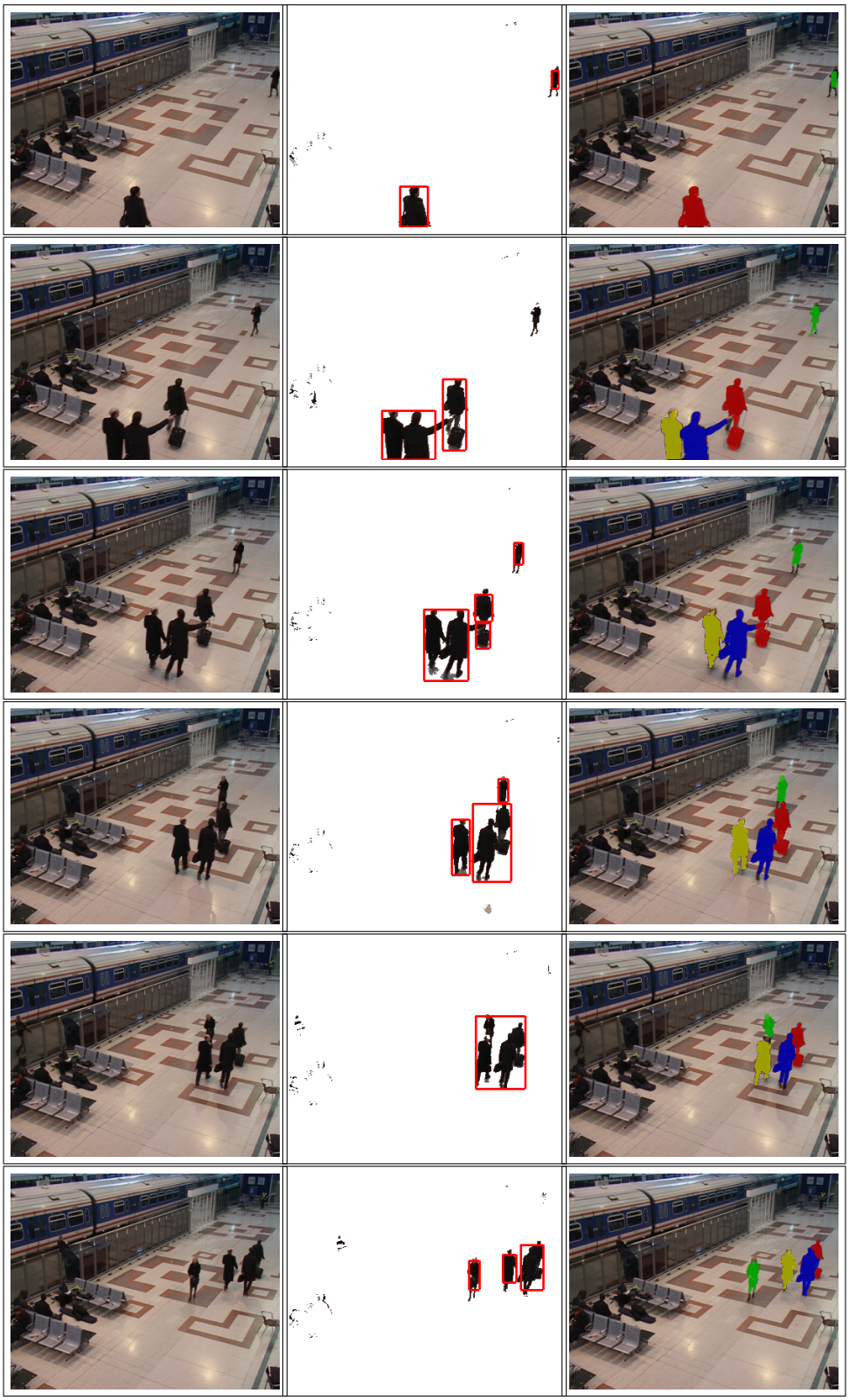
Les mêmes paramètres ont été utilisés pour obtenir tous les résultats de cette section. Comme dans [Blake 04], le paramètre λ_3 (équation 8.19) est fixé à 20. Les autres paramètres λ_1 (équation 8.12) et λ_2 (équation 8.13) sont plus critiques. Différents tests nous ont conduits à les fixer à $\lambda_1 = 10$ et $\lambda_2 = 2$. Pour finir, toutes les distributions évoquées dans ce chapitre sont des mélanges de 10 gaussiennes.

Notons que chaque objet segmenté a une couleur différente, qui ne dépend que de l'ordre arbitraire dans lequel les objets ont commencé à être suivi.

Nous commençons par présenter un premier résultat (figure 8.6) montrant la capacité de notre méthode à bien suivre plusieurs objets sur la séquence relativement simple de données de PETS 2006 (séquence 1, caméra 4). Sur la première image (image 81), deux pistes sont initialisées en utilisant les observations. Les observations étant des masques de pixels à l'intérieur d'une boîte englobante, elles ne donnent pas une segmentation parfaite de l'objet (par exemple, les jambes de la personne suivie en rouge ne sont pas détectées). Cependant notre méthode de suivi et de segmentation arrive dès l'image suivante à récupérer un masque respectant bien la forme et les contours de l'objet.

La méthode proposée gère bien les entrées et sorties des objets du champ de la caméra. Ainsi, dès qu'une nouvelle personne entre dans la scène, un nouvel objet (nouvelle piste) est créé automatiquement par la méthode.

Cette séquence souligne bien la capacité de notre méthode à suivre séparément des objets pouvant avoir fusionné. En effet, toutes les personnes à suivre dans cette séquence portent des vêtements foncés, de même couleur. De plus les trois personnes les plus en bas marchent quasiment dans la même direction et à la même vitesse. Ainsi, lorsque ces personnes se retrouvent juxtaposées, il est très difficile de les suivre séparément. En particulier, les deux personnes au centre en



(a) (b) (c)

FIG. 8.6: Résultats sur la séquence PETS 2006 (images 81, 116, 146, 176, 206 et 248). (a) Images originales. (b) Observations. (c) Résultats du suivi.

bas (jaune et bleu) correspondent au départ à deux observations différentes, puis, dès l'image 102, elles ne sont plus détectées que comme un seul objet. Sans la seconde fonction d'énergie, ces deux personnes ne peuvent plus être suivies indépendamment (figure 8.7).

(a) (b) (c)

FIG. 8.7: Séparation avec la seconde fonction d'énergie des objets ayant fusionné (images 101 et 102 de la séquence PETS 2006). (a) Observations. (b) Résultat du suivi et de la segmentation en n'utilisant que la première fonction d'énergie. (c) Résultat obtenu en rjoutant la deuxième énergie.

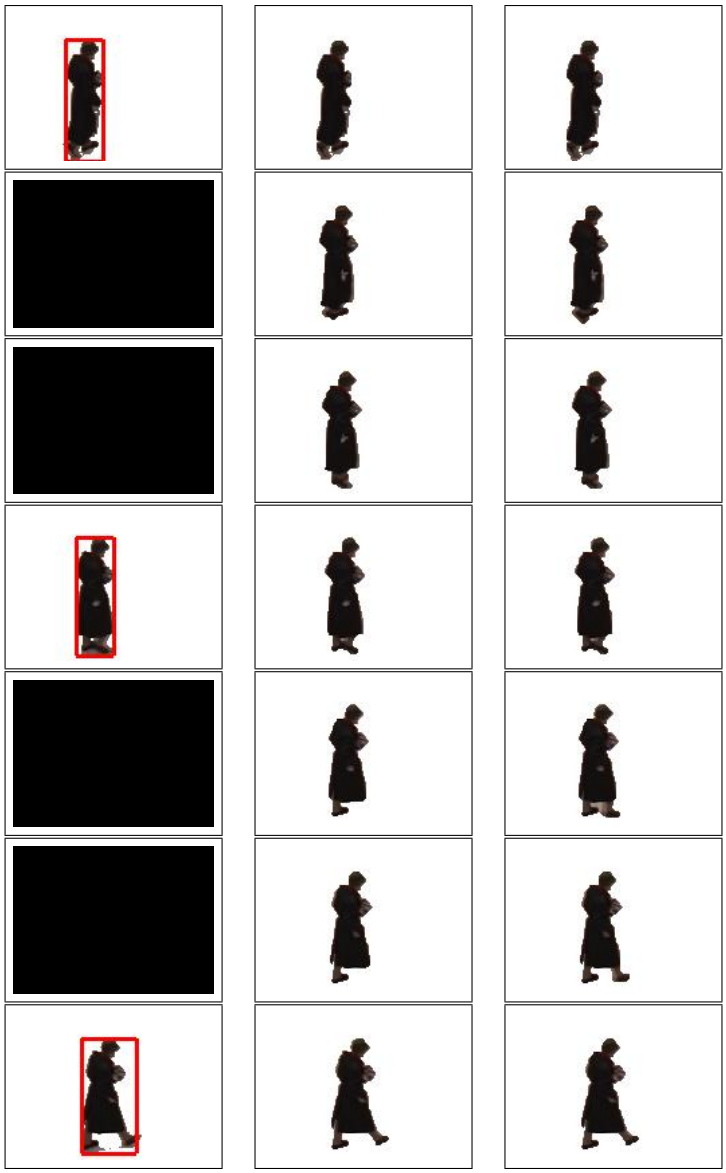
À la fin de la séquence, la personne suivie en vert est partiellement occultée par les autres. Même si tous ces objets ont des couleurs très proches, l'occultation est relativement bien gérée par notre algorithme de suivi.

Nous allons maintenant montrer le comportement de notre algorithme en cas d'absence d'observation. Pour cela, nous avons volontairement décidé de n'appliquer la détection par soustraction de fond pour pour une image sur trois. Le résultat est visible sur la figure 8.8. Les masques des objets obtenus en cas de manque d'observations sont comparés aux masques obtenus si des observations sont disponibles à chaque instant. Grâce à l'utilisation de la prédiction, les segmentations ne sont que très peu détériorées. Ainsi, une jambe manque au masque de la personne pour les images 105 et 106, mais elle est récupérée dès l'obtention de nouvelles observations. Ce résultat montre aussi que l'ajout des observations dans la phase de suivi permet d'obtenir de meilleures segmentations que si l'on n'utilisait que la prédiction.

8.4.2 Détection, segmentation et suivi d'objets dans des scènes complexes

Pour terminer ce document, nous allons présenter les résultats obtenus en combinant l'algorithme de détection de la première partie avec la méthode de suivi de ce chapitre. Il s'agit des résultats finaux de cette thèse.

Les observations sont les clusters en mouvement obtenus après division de la grille de points en clusters avec l'algorithme mean shift. Ainsi, l'étape finale du détecteur (chapitre 5) n'est pas appliquée, sauf lors de la création de nouvelles pistes de suivi. En effet, quand une observation n'a été associée à aucun objet et qu'elle devient un nouvel objet à suivre, nous initialisons cet objet avec son masque complet de segmentation, ceci afin d'éviter le suivi d'objets non valides. En outre, comme nous l'avons vu au chapitre 5, la segmentation à partir du cluster en mouvement permet d'obtenir le masque complet de l'objet mais aussi de valider cet objet (si à la fin de la



(a) (b) (c)

FIG. 8.8: Résultats de la méthode dans le cas où des observations sont manquantes (le détecteur n'est utilisé que pour une image sur trois entre les images 801 et 807 de la séquence PETS 2006) (a) Observations. (b) Résultat du suivi. (c) Résultats obtenus dans le cas où les observations sont disponibles.

phase de segmentation, le nombre de pixels appartenant au masque est égal au nombre de pixels du cluster initial, l'objet est déclaré non valide).

Nous reprenons ici la séquence de ski nautique ainsi que les deux séquences de conducteur déjà traitées dans la partie I. Pour chacune des images, nous présenterons les clusters en mouvement obtenus par le détecteur (servant d'observations), et les masques des objets suivis.

Les paramètres caractérisant le poids des termes binaires dans les fonctions d'énergie sont les

suivants. Comme pour les séquences à fond fixe de la section précédente, le paramètre λ_3 est fixé à 20 et le paramètre λ_1 (équation 8.12) à 10. Seul λ_2 (équation 8.13) est différent puisqu'il vaut ici 1.

Nous commençons donc par montrer les résultats de l'algorithme de suivi sur la séquence de ski nautique (figure 8.9). La méthode de suivi proposée dans ce chapitre permet de bien suivre

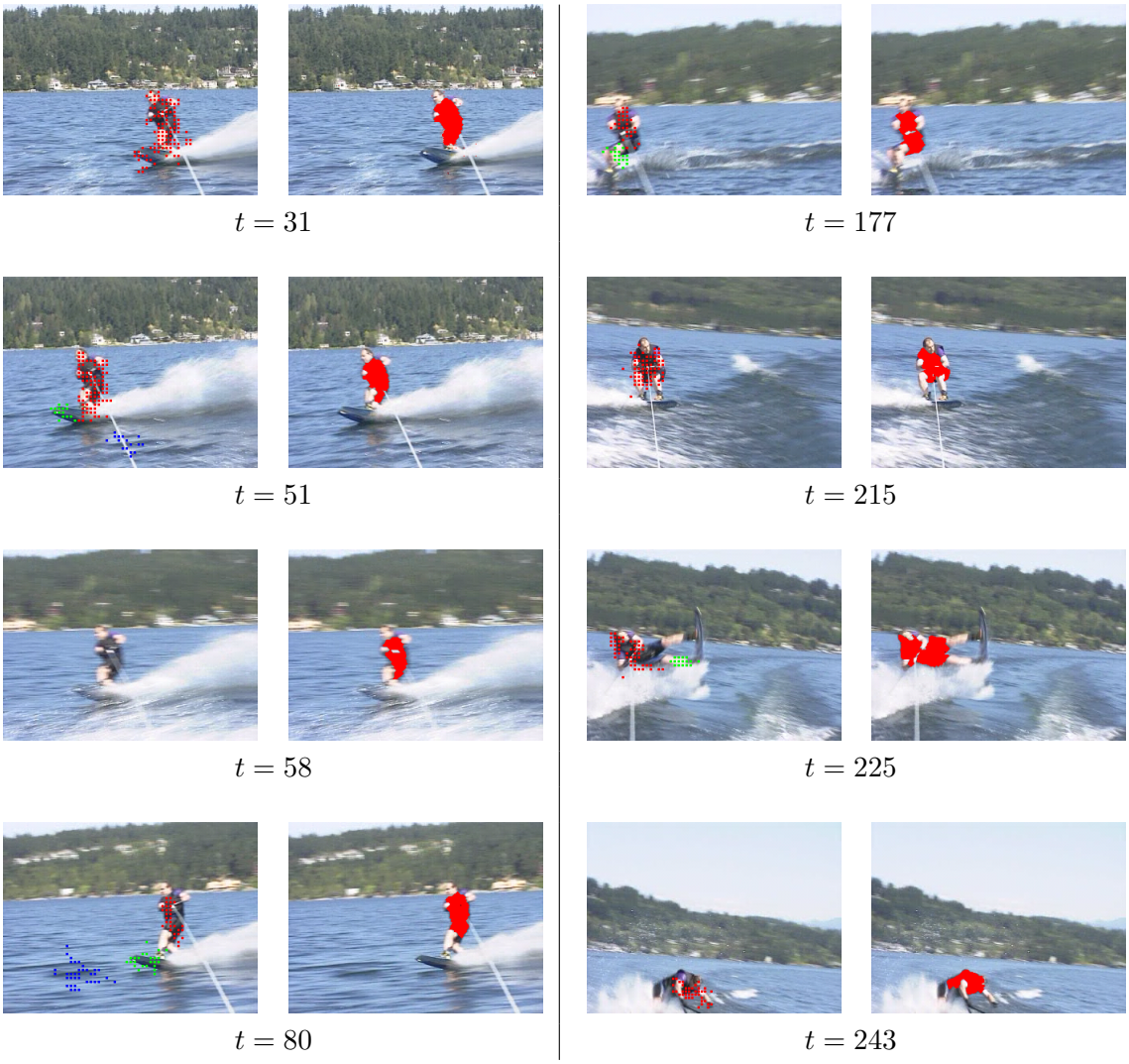


FIG. 8.9: Résultats de la méthode suivi sur la séquence de ski nautique. Les observations sont les clusters en mouvement obtenus avec le détecteur de la partie I. À chaque instant, les observations sont montrées sur l'image de gauche et le masque des objets suivis est visible sur l'image de droite.

le skieur (ou plus précisément la combinaison du skieur) malgré ses changements de trajectoire. Dans le chapitre 5, l'étude de cette séquence a montré que le détecteur de la première partie est capable de bien détecter le skieur, excepté quand il bouge trop rapidement ou ne bouge pas du tout relativement à la caméra. Cette séquence valide donc une nouvelle fois la robustesse de la méthode de suivi aux manques d'observations, puisque la combinaison continue à être suivie même lors de

sa non détection (image 58).

Par ailleurs, des clusters en mouvement sont parfois détectés dans l'eau. L'algorithme de suivi indique qu'une nouvelle piste doit être créée pour ces objets. Cependant, l'initialisation de nouvelles pistes de suivi se fait avec le masque complet des objets, c'est-à-dire que la phase de segmentation du chapitre 5 est appliquée au moment de la création des pistes. Si au moment de la segmentation l'objet est déclaré comme non valide, alors la nouvelle piste n'est pas créée et l'objet ne sera pas suivi. L'utilisation du masque complet de l'objet comme initialisation permet ainsi de supprimer les objets non valides. Ainsi, sur cette séquence, aucun objet dans l'eau n'est suivi.

Pour les deux résultats suivants, nous reprenons les deux séquences de conducteur présentées dans la première partie. Les résultats sur la première séquence sont visibles sur la figure 8.10. La

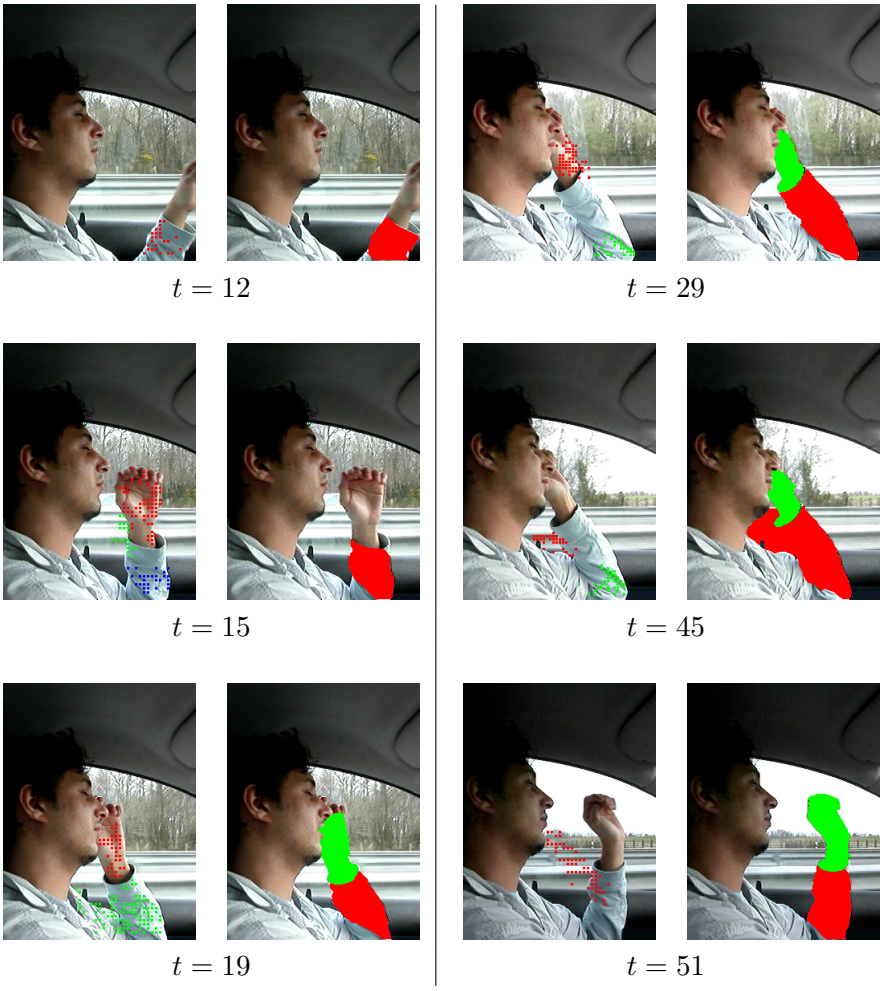


FIG. 8.10: Résultats de la méthode suivi sur la séquence d'un conducteur. Les observations sont les clusters en mouvement obtenus avec le détecteur de la partie I. À chaque instant, les observations sont montrées sur l'image de gauche et le masque des objets suivis est visible sur l'image de droite.

main et le bras sont bien détectés sur la majorité des images de la séquence. Une piste de suivi est

initialisée pour le bras dès sa première détection. Il est ainsi bien suivi tout au long de la séquence. On remarque néanmoins que sur l'image 45, la segmentation du bras empiète sur l'habitacle de la voiture et sur la barrière de sécurité. Cela est dû au fait que le contraste entre le bras et la barrière est faible. De plus le bras ne bouge quasiment pas à cet instant et a donc un mouvement très proche de celui de l'habitacle et de la barrière. Enfin, un cluster en mouvement est détecté sur la barrière et, comme il a des caractéristiques de mouvement et de couleur proches de celles du bras, il est associé à l'objet représentant le bras à l'instant précédent. Notons néanmoins que dès que le bras recommence à bouger (image 49), une bonne segmentation est à nouveau réalisée.

Contrairement au bras, la main n'est pas suivie dès sa première détection. Pour qu'une nouvelle piste soit définitivement créée, il faut que l'objet correspondant est été associé à une observation à l'instant suivant son initialisation, ce qui n'est pas le cas ici. Les vecteurs de mouvement sont très différents entre deux instants consécutifs lorsque que la main monte vers le visage. Il y a aussi de légers changements d'illumination. Ces deux difficultés empêchent la création d'une piste pour la main. Néanmoins, dès l'image 19, une piste est créée pour la main et est ensuite bien segmentée et suivie. Les observations sont simplement caractérisées par la moyenne des vecteurs de couleur et de mouvement. Nous pensons qu'une meilleure caractérisation des observations tenant compte de la disparité des clusters et éventuellement d'autres caractéristiques de l'objet pourrait probablement permettre une meilleure association entre objets suivis et observations, et améliorerait l'influence des observations sur la segmentation et le suivi. Nous pensons qu'ainsi la main pourrait être suivie dès sa détection.

Les résultats sur cette séquence sont très encourageants étant donnée la complexité de la vidéo. Ainsi, le détecteur ne trouve pas d'objets dans le fond très dynamique derrière les fenêtres, et permet une initialisation relativement bonne de la main et du bras du conducteur. Ces deux objets, une fois initialisés, sont bien suivis même en présence de changements d'illumination.

Nous terminons en montrant les résultats sur une deuxième séquence de conducteur (figure 8.11). Le premier objet détecté et suivi est le visage. Le suivi du visage montre une nouvelle fois la robustesse de la méthode au manque d'observations. En effet, même si à partir de l'image 19 le visage ne bouge plus et n'est donc plus détecté, l'algorithme de suivi continue à le suivre et à le segmenter correctement. Il est de nouveau détecté lorsque la conductrice commence à tourner la tête.

Le suivi du volant peut paraître surprenant. Cependant, il est détecté pendant deux images consécutives lorsqu'il tourne. Un post-traitement, ne conservant que les objets ayant des trajectoires intéressantes, pourrait certainement permettre de supprimer cet objet.

Le résultat le plus important sur cette séquence concerne la détection et le suivi des mains. Nous avions déjà discuté dans le chapitre 5 de la capacité du détecteur à bien détecter les deux mains séparément même lorsqu'elles se touchent (image 40 sur la figure 5.3 du chapitre 5, et 39 de la figure 8.11). Un tel résultat est principalement dû à l'utilisation du mouvement dans le descripteur. À la fin de la première minimisation d'énergie de l'algorithme de suivi, les masques des deux mains se superposent légèrement : les deux masques ont quelques pixels en commun. L'étape de séparation des objets fusionnés est donc appliquée. La minimisation de la deuxième fonction d'énergie permet de bien séparer les deux masques et de continuer à suivre et segmenter indépendamment les deux mains.

Les résultats de cette sous-section montrent la performance de l'algorithme de détection de

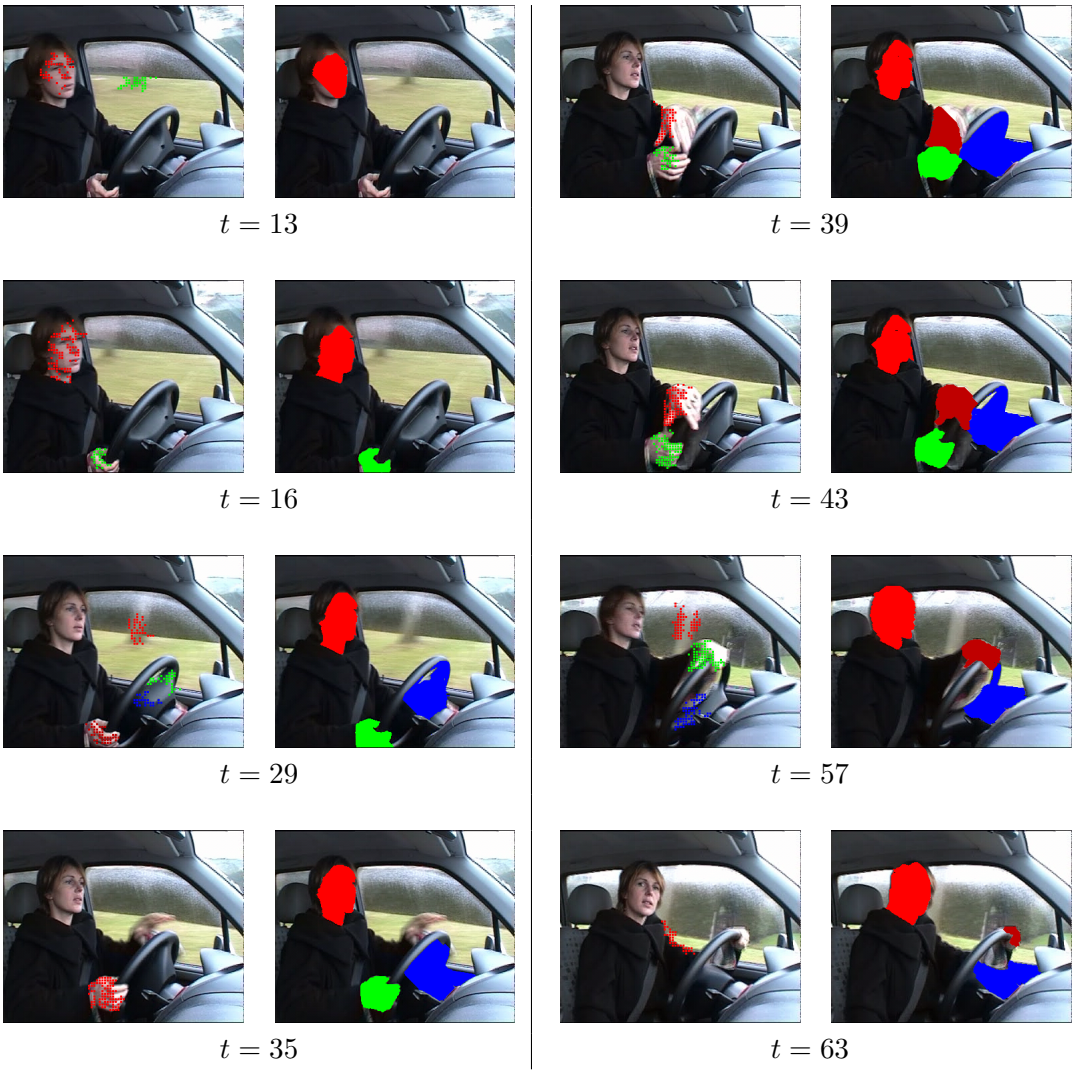


FIG. 8.11: Résultats de la méthode de suivi sur la séquence d'une conductrice. Les observations sont les clusters en mouvement obtenus avec le détecteur de la partie I. À chaque instant, les observations sont montrées sur l'image de gauche et le masque des objets suivis est visible sur l'image de droite.

la première partie et de l'algorithme de suivi décrit dans cette section. La combinaison des deux méthodes permet la détection, la segmentation et le suivi d'objets dans des scènes complexes. Les résultats de détection et de suivi du visage et des mains dans les séquences de conducteur indiquent que les méthodes développées au cours de cette thèse pourraient probablement être utilisées pour aider à analyser le comportement des conducteurs.

Conclusion

Dans cette deuxième partie, une méthode de segmentation et de suivi d'objets reposant sur la minimisation d'énergies par un algorithme de coupe minimale/flot maximal ("Graph Cuts") a été proposée. Elle combine les avantages de différentes méthodes de suivi existantes. Un état de l'art de ces méthodes a été donné au chapitre 7. L'approche proposée permet ainsi de simultanément suivre et segmenter des objets en prenant en compte des distributions de couleur et de mouvement et en utilisant des observations extérieures obtenues par un module de détection. Toutes ces informations sont regroupées dans une première fonction d'énergie. Une énergie est construite et minimisée pour chaque objet.

Les observations extérieures sont prises en compte en ajoutant au graphe des nœuds d'observation (un nœud par observation est ajouté). Cela permet de ne pas avoir à associer préalablement chaque objet avec une observation, ce qui est généralement indispensable pour toutes les autres techniques de suivi. Chaque observation n'est caractérisée que par un vecteur contenant la moyenne calculée sur l'ensemble de l'observation, et des vecteurs de couleur et de mouvement. Cette représentation est un peu simpliste et mériterait d'être améliorée. Il serait en effet préférable d'utiliser des distributions de couleur et de mouvement calculées sur toute l'observation. Il faudrait alors étudier comment ajouter de telles distributions au terme unaire de la (première) fonction d'énergie en ne conservant qu'un seul nœud par observation. Enfin, une étude future pourrait porter sur l'utilisation d'autres caractéristiques comme par exemple la forme ou la taille de l'objet. La difficulté serait alors de minimiser une énergie pour laquelle le terme binaire n'est pas forcément une fonction régulière.

Afin de suivre indépendamment chaque objet, une fonction d'énergie multi-étiquettes est ajoutée à la méthode de suivi. Elle permet de séparer les objets ayant fusionné après la minimisation de la première fonction d'énergie. Nous considérons que des objets ont fusionné si leur segmentation se superpose. Ils sont séparés en donnant plus d'importance à la prédiction.

Deux énergies sont donc pour l'instant nécessaires pour suivre indépendamment chaque objet. Cependant, il serait intéressant d'étudier si ces énergies ne pourraient pas être réécrites afin de n'effectuer qu'une minimisation pour tous les objets.

Comme expliqué dans le chapitre 6, les fonctions d'énergies minimisées par coupe minimale/flot maximal comportent en général deux termes principaux : un terme unaire d'attache aux données et un terme binaire de régularisation. Un paramètre constant permet de définir l'importance de chacun de ces deux termes dans l'énergie. Ce paramètre influence beaucoup les résultats. Pourtant, aucune méthode permettant de calculer ce paramètre et de l'adapter aux données n'existe. Des travaux futurs sur les "Graph Cuts" devraient donc porter sur une étude

approfondie de ce paramètre.

À la fin du chapitre 8, nous avons présenté de bons résultats, obtenus en regroupant l'algorithme de détection mis en place dans la première partie et l'algorithme de suivi de cette deuxième partie. La combinaison de ces deux algorithmes permet de détecter, segmenter et suivre des objets dans des scènes complexes.

Conclusion générale et perspectives

Dans ce document, nous nous sommes intéressés à la détection, la segmentation et le suivi d'objets en mouvement dans des scènes complexes (caméra mobile, fond dynamique, changements d'illumination ...) . La première partie a proposé une nouvelle méthode de détection et de segmentation d'objets mobiles et la deuxième partie une nouvelle méthode de suivi. L'application des ces deux méthodes, qui peuvent être couplées, a été montrée sur des séquences diverses, et notamment sur les vidéos complexes de conducteurs. Dans cette conclusion générale, nous faisons tout d'abord un bilan de ces travaux puis proposons un certain nombre de perspectives.

Synthèse des travaux effectués

Nous avons tout d'abord proposé une nouvelle méthode de détection des objets en mouvement, qui repose sur plusieurs techniques couramment utilisées en vision par ordinateur. Cette méthode se décomposent en trois étapes principales, contenant chacune différentes nouveautés. La première étape consiste à définir une grille de pixels en mouvement, c'est-à-dire une grille de points n'appartenant pas au mouvement de la caméra. L'utilisation d'un nombre restreint de points permet d'une part de rechercher les objets uniquement sur les régions en mouvement de l'image, et d'autre part de diminuer le coût de calcul de l'algorithme. Une description de ces points, reposant sur des informations de position, de mouvement et de photométrie, a ensuite été proposée. Une particularité de la description concerne la validation des vecteurs de mouvement, calculé avec un algorithme de flot optique. En effet, nous avons proposé de valider les vecteurs à l'aide d'un test statistique.

La deuxième étape consiste à diviser les points de la grille, à partir de leurs descripteurs, en un certain nombre de clusters. Nous avons choisi d'utiliser pour cela un algorithme mean shift. Comme les données considérées sont de dimension supérieure à trois, nous avons mis en place un nouvel algorithme nommé, "pseudo balloon mean shift", reposant sur l'estimateur à noyau balloon. Les résultats des algorithmes mean shift, qui sont basés sur l'estimation de la densité inconnue des données à l'aide d'un estimateur à noyau, dépendent beaucoup de la taille du noyau de l'estimateur. Comme les données sont ici particulières (elles sont multidimensionnelles et hétérogènes), une nouvelle méthode de sélection de la taille du noyau a dû être proposée. Cette sélection s'appuie sur un critère de validation de la stabilité des clusters obtenus. L'algorithme de sélection et l'algorithme "pseudo balloon mean shift" ont été en premier lieu appliqués à la segmentation d'images couleur. Nous avons ensuite montré comment utiliser ces méthodes pour la détection de clusters en mouvement.

La dernière étape, facultative, du détecteur d'objets a pour but l'obtention du masque de pixels complet des objets en mouvement, à partir des clusters en mouvement produits par l'étape précédente. Chaque masque est le résultat de la minimisation d'une fonction d'énergie grâce au

calcul de la coupe minimale, c'est-à-dire du flot maximal, dans un graphe. Cette étape utilise l'ensemble des pixels de l'image. L'originalité de cette approche est l'utilisation de données éparses sur le mouvement à l'intérieur de la fonction d'énergie.

La deuxième partie du document a traité du problème du suivi des objets en mouvement. Afin de coupler la dernière étape du détecteur (étape de segmentation) et le suivi des objets, nous avons proposé un nouvel algorithme basé sur la minimisation d'énergie par coupe minimale/flot maximal dans un graphe. Il permet de segmenter et de suivre directement les clusters en mouvement obtenus à la fin de la deuxième étape du détecteur. Le principe de la méthode proposée est de regrouper la prédiction de l'objet suivi, des distributions sur sa photométrie et son mouvement, ainsi que des observations extérieures, dans une fonction d'énergie. Les observations peuvent être de n'importe quel type, mais nous nous sommes concentrés, dans les expérimentions, sur des masques de pixels obtenus par soustraction de fond puis sur les clusters en mouvement donnés par le détecteur. Contrairement aux méthodes de suivi existantes, l'utilisation d'observations extérieures ne nécessite pas l'ajout préalable d'une étape d'association entre les observations et les objets suivis.

Dans l'algorithme mis en place, chaque objet est suivi indépendamment, pour permettre de bien distinguer la fusion de deux objets et l'occultation d'un objet par un autre. Il n'est cependant pas suffisant à lui seul pour suivre indépendamment chaque objet, et plus précisément pour séparer des objets ayant fusionné au cours du suivi. Nous avons donc rajouté une étape de séparation des objets fusionnés reposant sur la minimisation, cette fois multi-étiquettes, d'une seconde fonction d'énergie. La séparation repose en grande partie sur la confiance que nous accordons à la prédiction des objets.

Les algorithmes de suivi (et de segmentation) et de séparation des objets fusionnés ont été appliqués dans un premier temps à une séquence dans laquelle le fond est fixe. Nous avons alors utilisé des observations issues d'une soustraction de fond. La difficulté dans cette séquence était que les objets à suivre étaient des personnes portant des vêtements de même couleur et pouvant avoir des mouvements similaires. Néanmoins, l'utilisation de l'algorithme de séparation des objets fusionnés a permis de bien suivre et segmenter ces objets séparément. Nous avons ensuite appliqué ces méthodes aux séquences complexes déjà considérées dans la première partie du document. Les observations étaient alors les clusters en mouvement obtenus par le détecteur d'objets en mouvement. Ces résultats finaux sont très encourageants. Nous avons ainsi été capable de détecter la tête et/ou les mains d'un conducteur malgré le fond hautement dynamique présent derrière les fenêtres. Finalement, les méthodes développées au cours de cette thèse pourraient probablement être utilisées pour aider l'analyse du comportement des conducteurs.

Perspectives

Un certain nombre de perspectives (déjà évoquées dans les conclusions de parties) peuvent être issues de cette thèse. On peut distinguer des perspectives plus méthodologiques et des perspectives plus applicatives.

Description des points

Comme nous venons de le rappeler, le détecteur d'objets en mouvement repose sur la description (position, photométrie et mouvement) des points d'une grille. Les vecteurs de mouvement

sont validés par un test statistique. Une étude plus approfondie du test utilisé, c'est-à-dire la p-value, devrait être menée. En particulier, il serait intéressant de comparer plus en détail la p-value à un test de corrélation, afin de mieux mettre en avant ses atouts et son apport sur les résultats finaux de détection.

Sélection de la taille du noyau pour le clustering des points

La sélection de la taille du noyau pour les algorithmes mean shift est basée sur un critère de stabilité des clusters. Pour cela, un ensemble de noyaux est prédéfini et un clustering mean shift est appliqué aux données avec chacun de ces noyaux. La stabilité est alors calculée à l'aide d'une mesure de divergence entre les lois normales représentant chacun des clusters issus de ces clustering. Nous avons proposé d'utiliser les définitions traditionnelles de la moyenne et de la covariance pour calculer les lois normales, mais un calcul de covariance robuste caractérisant mieux la forme du cluster serait probablement plus approprié. Pour cela, on pourrait par exemple s'appuyer sur les travaux de [Pena 01] ou [Wang 02].

Segmentation multi-étiquettes des objets en mouvement

Dans la dernière étape du détecteur chaque objet est segmenté en minimisant une énergie par un algorithme de coupe minimale/flot maximal dans un graphe, afin d'obtenir le masque complet des objets en mouvement, . Nous avons proposé de segmenter chaque objet indépendamment : une fonction d'énergie est minimisée pour chaque objet. Les algorithmes de coupe minimale/flot maximal donnent la possibilité de minimiser des énergies multi-étiquettes. Nous avons d'ailleurs utilisé cette particularité au cours du suivi, dans l'algorithme de séparation des objets fusionnés. Une extension de la méthode pourrait alors consister à modifier notre algorithme afin que tous les objets soient segmentés simultanément. Cela permettrait d'une part d'améliorer le coût de calcul de l'algorithme, et d'autre part de prendre en compte les interactions entre objets.

Détection des petits objets / mouvement dominant

Le résultat présenté sur la figure 5.4 a mis en avant une limitation de la méthode de détection des objets en mouvement. En effet, sur cette figure le piéton au milieu à droite des images n'est presque jamais détecté comme un objet en mouvement. La majorité des pixels de ce piéton appartenant au mouvement dominant de l'image, trop peu de pixels en mouvement sont disponibles pour le détecter correctement. Des travaux futurs devraient se concentrer sur l'étude de la détection de petits objets appartenant majoritairement au mouvement dominant. On peut d'ailleurs se demander s'il est vraiment possible de les détecter tant que les pixels appartenant au mouvement dominant ne sont pas considéré.

Description (caractérisation) des observations dans l'algorithme de suivi

Les observations extérieures sont prises en compte dans l'algorithme de suivi en ajoutant au graphe des nœuds d'observations (un nœud par observation est ajouté). Chaque observation n'est caractérisée que par un vecteur contenant la moyenne des vecteurs de couleur et de mouvement, mais il serait préférable d'utiliser des distributions calculées sur toute l'observation.

Combiner le suivi et la séparation des objets fusionnés

Le suivi et la séparation des objets fusionnés sont réalisés avec des fonctions d'énergies différentes. Ainsi, deux énergies sont pour l'instant nécessaires pour suivre indépendamment chaque objet. Il serait intéressant d'étudier si ces énergies ne pourraient pas être réécrites afin de n'effectuer qu'une minimisation pour tous les objets.

Poids des termes binaires des fonctions d'énergie

Comme expliqué dans le chapitre 6, les fonctions d'énergies minimisées par coupe minimale/flot maximal comportent en général deux termes principaux : un terme unaire d'attache aux données et un terme binaire de régularisation. Un paramètre constant permet de définir l'importance de chacun de ces deux termes dans l'énergie. Ce paramètre influence beaucoup les résultats. Pourtant, aucune méthode permettant de calculer ce paramètre et de l'adapter aux données n'existent actuellement. Des travaux futurs sur les "Graph Cuts" devraient donc porter sur une étude approfondie de ce paramètre.

Ajout de contraintes sur la forme de l'objet suivi

Une étude future pourrait également porter sur l'utilisation d'autres caractéristiques (en plus de la couleur et du mouvement), comme par exemple la forme ou la taille de l'objet, dans l'algorithme de suivi. La difficulté serait alors de minimiser une énergie pour laquelle le terme binaire n'est pas forcément une fonction régulière. Des méthodes utilisant des termes globaux dans des fonctions d'énergie minimisées par coupe minimale/flot maximal dans un graphe ont déjà été développées dans [Yuille 03, Rihan 06, Rother 06].

Loi dynamique pour la prédiction

L'algorithme de suivi utilise la prédiction des objets à l'intérieur de la fonction d'énergie. Cette prédiction est simplement obtenue en déplaçant l'objet à l'instant précédent $t-1$ avec les vecteurs de flot optique calculés à $t-1$. Il s'agit donc simplement d'une succession de segmentation se basant sur la prédiction, les distributions et les observations. Afin d'obtenir une technique de suivi robuste aux occultations totales et adaptées à des dynamiques plus compliquées, il serait intéressant d'utiliser une loi dynamique plus sophistiquée pour la prédiction des objets. Cela a déjà été proposé pour le suivi de courbes dans [Cremers 01, Jackson 04, Niethammer 04].

Extension de l'algorithme de suivi à d'autres applications

Pour finir, la méthode de suivi pourrait probablement être étendue à d'autres applications. En particulier, comme les méthodes de coupe dans un graphe ont déjà été appliquées à la stéréo vision et à la reconstruction de scène à partir de plusieurs caméras, on peut penser qu'il serait possible, à l'aide des nœuds d'observations, de suivre conjointement le même objet dans des séquences de la même scène, prise avec des angles de vue différents. On pourrait alors directement obtenir la trajectoire 3D ou la reconstruction 3D de cet objet.

Annexes

Annexe A

Preuve de convergence de l'algorithme pseudo balloon mean shift

L'estimateur à noyau balloon est défini par :

$$\widehat{f}(\mathbf{x}) = \frac{c_k}{n} \sum_{i=1}^{n} \frac{1}{|\mathbf{H}(\mathbf{x})|^{1/2}} k(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2) \ . \tag{A.1}$$

Le preuve de convergence du filtrage mean shift utilisant cet estimateur est proche de celle proposée dans [Comaniciu 02] pour le filtrage mean shift avec l'estimateur sample point. Afin de montrer que $\mathbf{m}(\mathbf{x})$ converge, nous commençons par montrer que $\widehat{f}$ converge pour les points de trajectoires. Nous montrons donc que $\widehat{f}(\mathbf{y}^{(j)})$ converge quand j augmente.

Comme n est fini, l'estimée $\widehat{f}$ est bornée : $0 < \widehat{f}(x) \leq \frac{c_k}{n|\mathbf{H}(\mathbf{x})|^{1/2}}$. Il suffit donc de montrer que $\widehat{f}$ est strictement croissante ou décroissante. Comme nous supposons la bande passante $\mathbf{H}(\mathbf{x})$ constante pour tous les points de trajectoire $\mathbf{y}^{(j)}$ associés au point d'estimation $\mathbf{x}$, nous avons :

$$\begin{aligned}\widehat{f}(\mathbf{y}^{(j+1)}) - \widehat{f}(\mathbf{y}^{(j)}) \\ = \frac{c_k}{n|\mathbf{H}(\mathbf{x})|^{1/2}} \sum_{i=1}^{n} \left(k(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j+1)} - \mathbf{x}^{(i)})\|^2) - k(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j)} - \mathbf{x}^{(i)})\|^2) \right) \ .\end{aligned} \tag{A.2}$$

Le profile k étant convexe,

$$\forall (x_1, x_2) \in [0, +\infty) \ \ k(x_2) \geq k(x_1) + k'(x_1)(x_2 - x_1) \ .$$

Cela implique :

$$\begin{aligned}\widehat{f}(\mathbf{y}^{(j+1)}) - \widehat{f}(\mathbf{y}^{(j)}) \geq \frac{c_k}{n|\mathbf{H}(\mathbf{x})|^{1/2}} \sum_{i=1}^{n} k'(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j)} - \mathbf{x}^{(i)})\|^2) \\ \left(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j+1)} - \mathbf{x}^{(i)})\|^2 - \|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j)} - \mathbf{x}^{(i)})\|^2 \right).\end{aligned} \tag{A.3}$$

Supposons $\mathbf{H}(\mathbf{x})^T = \mathbf{H}(\mathbf{x})$. En développant le dernier terme de l'équation précédente et en utilisant la définition du vecteur mean shift (équation 3.29), il peut être montré que :

$$\widehat{f}(\mathbf{y}^{(j+1)}) - \widehat{f}(\mathbf{y}^{(j)}) \geq \frac{c_k}{n|\mathbf{H}(\mathbf{x})|^{1/2}} \sum_{i=1}^{n} k'(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j)} - \mathbf{x}^{(i)})\|^2)\Big(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j+1)} - \mathbf{y}^{(j)})\|^2\Big). \tag{A.4}$$

En additionnant maintenant les termes de cette équation pour les indices $j, j+1, \ldots, j+m-1$, et en introduisant

$$M = \text{argmin}_{l \geq 0}\, k(\|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(l)} - \mathbf{x}^{(i)})\|^2$$

nous obtenons le résultat suivant :

$$\widehat{f}(\mathbf{y}^{(j+m)}) - \widehat{f}(\mathbf{y}^{(j)}) \geq \frac{c_k}{n|\mathbf{H}(\mathbf{x})|^{1/2}} M \|\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{y}^{(j+m)} - \mathbf{y}^{(j)})\|^2 \geq 0. \tag{A.5}$$

Nous avons ainsi montré que la suite $\{\widehat{f}(\mathbf{y}^{(j)})\}_{j=1,2\ldots}$ est strictement croissante, bornée, et donc convergente. L'inégalité précédente implique que $\{\mathbf{y}^{(j)}\}_{j=1,2\ldots}$ est une suite de Cauchy par rapport à la distance de Mahalanobis et donc par rapport à la distance euclidienne (d'après l'équivalence des normes dans $\mathbb{R}^d$). Cela prouve la convergence des points de trajectoires vers le mode local de $\widehat{f}$. ■

Annexe B

Critère de stabilité pour la sélection de noyaux

Dans l'algorithme itératif de sélection de la taille du noyau défini au chapitre 4, la taille finale $\boldsymbol{\Upsilon}(\mathbf{x}^{(i)})$ sélectionnée en chaque point $\mathbf{x}^{(i)}$ correspond à la taille prédéfinie $\mathbf{H}^{(b^*)}$ ayant donné le cluster le plus stable :

$$\boldsymbol{\Upsilon}(\mathbf{x}^{(i)}) = \mathbf{H}^{(b^*)} \ .$$

Le paramètre b^* indique l'échelle pour laquelle le cluster est le plus stable.

Dans [Comaniciu 03a], en revanche, la taille finale est la covariance de la loi normale calculée sur le cluster le plus stable :

$$\boldsymbol{\Upsilon}(\mathbf{x}^{(i)}) = \boldsymbol{\Sigma}^{(b^*)}_{c(i,b^*)} \ .$$

L'auteur justifie son choix de la façon suivante. Il indique que la diminution du biais d'un estimateur à noyau est liée à la maximisation de la norme du gradient de la densité. Or il peut être facilement montré que maximiser la norme du gradient revient à maximiser le vecteur mean shift normalisé par le noyau. Le théorème suivant montre alors que ce vecteur est maximal lorsque la taille du noyau est égale à la covariance de la distribution que l'on cherche à estimer dès lors que celle-ci est normale :

Théorème 3
Supposons que la densité f est localement une loi normale $\mathcal{N}(\mu, \boldsymbol{\Sigma})$ et le mean shift à noyau fixe est calculé en utilisant le noyau normal K de taille $\mathbf{H}$. Le vecteur mean shift normalisé par $\mathbf{H}$ est maximal si la bande passante $\mathbf{H}$ est égale à la covariance $\boldsymbol{\Sigma}$.

Nous allons, dans cette annexe, comparer les deux critères de sélection que nous nommerons : sélection de la meilleure taille prédéfinie et sélection de la meilleure covariance. Ces deux critères seront utilisés dans l'algorithme itératif (algorithme 5). Le théorème précédent n'est pas applicable directement à cet algorithme. En outre, les itérations nécessitent l'utilisation d'un estimateur à noyau variable. Nous avons, dans ce document, pris le parti d'utiliser l'estimateur balloon. Le théorème précédent se réécrit très bien pour cet estimateur :

Théorème 4
Supposons que la densité f est localement une loi normale $\mathcal{N}(\mu, \boldsymbol{\Sigma})$ et le pseudo-balloon mean shift est calculé, en chaque point $\mathbf{x}$, en utilisant le noyau normal K de taille $\mathbf{H}(\mathbf{x})$. Le vecteur

mean shift normalisé par $\mathbf{H}(\mathbf{x})$ *est maximal si la bande passante* $\mathbf{H}(\mathbf{x})$ *est égale à la covariance* $\mathbf{\Sigma}$.

Ce théorème est démontré dans l'annexe C. L'utilisation du clustering mean shift basé sur l'estimateur sample point n'est pas possible ici. En effet, le théorème 3 ne peut pas se démontrer avec cet estimateur.

Comme dans [Comaniciu 03a], le théorème 4 conduit au critère de sélection de la meilleure covariance. Seule l'étape 2.2 de l'algorithme 5 change en utilisant ce critère. Elle devient : "La meilleure bande passante $\Upsilon_{\rho}^{(i)}$ est $\mathbf{\Sigma}_{c(i,b^*),\rho}^{(b^*)}$".

Au chapitre 4, nous avons préféré utiliser un critère différent (sélection de la meilleure taille prédéfinie) dans l'algorithme initial pour les deux raisons suivantes. Tout d'abord, nous n'avons pas réussi, théoriquement, à faire le lien entre la maximisation des vecteurs mean shift normalisés et le biais de l'estimateur. De plus, en utilisant le critère de sélection de la meilleure covariance, on ne peut plus considérer l'algorithme comme étant basé sur la stabilité des clusters. En effet, même si le noyau final correspond à la covariance du cluster de l'échelle la plus stable, il n'est pas forcément compris dans l'ensemble prédéfini. Ainsi, parfois, le noyau final est inférieur au noyau préfédini le plus faible ou supérieur au plus fort.

Cette annexe a pour but la comparaison expérimentale des deux critères de sélection. Tous les résultats ont été obtenus en utilisant 9 noyaux prédéfinis compris entre 10 et 30. Nous comparons les segmentations obtenues en appliquant l'algorithme de clustering "pseudo balloon mean shift" avec les noyaux obtenus par les deux critères de sélection précédents.

Les premiers résultats sont sur l'image d'un chalet (figure B.1). La segmentation de cette image en sélectionnant la meilleure taille prédéfinie donne 31 clusters. Le deuxième critère, "sélection de la meilleure covariance" permet d'en obtenir 45 et conserve plus de détails. En particulier, l'ombre sur les montagnes et la petite région de terre sur l'herbe devant le chalet sont toujours visibles après la segmentation. Au contraire, des détails sur le chalet sont perdus, le toit et la façade étant regroupés en une seule partition.

a) b) c)

FIG. B.1: Validation du critère de sélection sur l'image d'un chalet. a) Image originale ; b) Résultat de la segmentation avec le critère "sélection de la meilleure covariance" ; c) Résultat de la segmentation avec le critère "sélection de la meilleure taille prédéfinie".

Nous montrons également les résultats de segmentation sur l'image d'une main (figure B.2). Ce résultat conduit à la même conclusion : plus de détails sont conservés en utilisant la covariance pour bande passante finale. Les ongles sont encore visibles après la segmentation alors que le

critère initial ne permet que d'en garder un. La figure B.3 montre, pour chaque domaine les valeurs des bandes passantes obtenues avec les deux critères pour tous les pixels d'ordonnée égale à la moitié de la hauteur de l'image (ligne au milieu de l'image). Pour tous les domaines, excepté le premier correspondant à l'abscisse, les tailles finales de noyau sont plus faibles si l'on choisit la meilleure covariance plutôt que la meilleure bande passante prédéfinie. Ces figures expliquent pourquoi plus détails sont conservés avec le critère "sélection de la meilleure covariance".

a) b) c)

FIG. B.2: Validation du critère de sélection sur l'image d'une main. a) Image originale ; b) Résultat de la segmentation avec le critère "sélection de la meilleure covariance" ; c) Résultat de la segmentation avec le critère "sélection de la meilleure taille prédéfinie".

En conclusion les résultats expérimentaux de cette annexe montrent qu'utiliser un critère de sélection s'appuyant sur le théorème 4, c'est-à-dire si la taille du noyau en chaque point est choisie égale à la covariance de la distribution donnant le cluster le plus stable, permet de conserver plus de détails et d'obtenir un nombre plus important de clusters. Cependant, il n'y a pas de réelle justification théorique à l'utilisation d'un tel critère. De plus, il est difficile de vraiment juger une segmentation. En effet, garder plus de détails peut avoir des avantages pour certaines applications et au contraire peut n'être pas souhaitable pour d'autres. Dans notre application à la détection de cluster en mouvement, nous voulons que les clusters soient de tailles suffisamment grandes, afin qu'ils représentent bien les objets. C'est pourquoi, dans ce manuscrit, nous avons utilisé le critère consistant à sélectionner, en chaque point, la bande passante prédéfinie donnant la partition la plus stable.

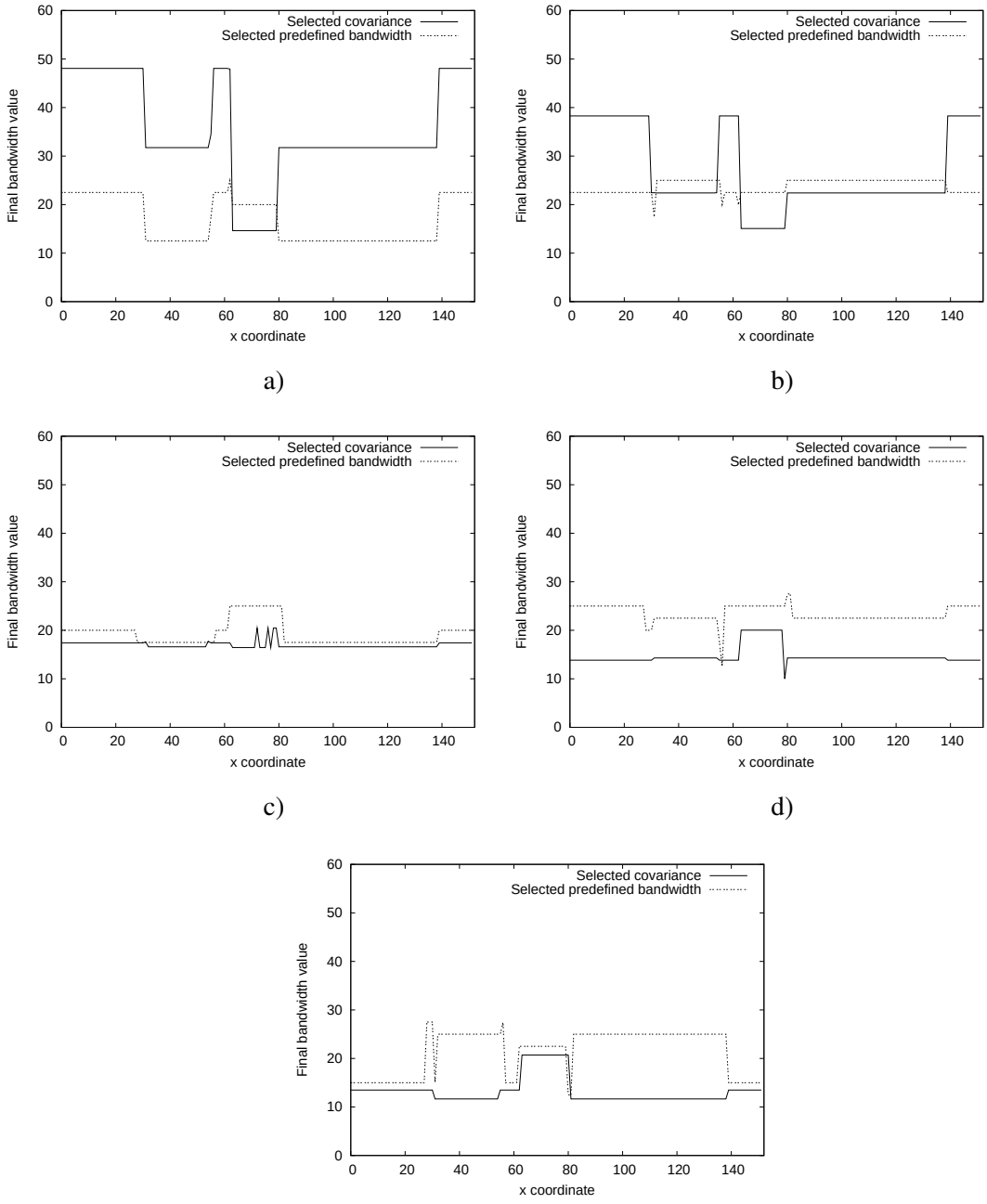


FIG. B.3: Valeurs finales des bandes passantes pour tous les points situés sur la ligne au milieu de l'image d'une main a) domaine correspondant aux abscisses b) domaine correspondant aux ordonnées c) canal rouge d) canal vert and e) canal bleu.

Annexe C

Preuve du théorème 4

Supposons que la densité f est localement une loi normale $\mathcal{N}(\mu, \mathbf{\Sigma})$ et le pseudo-balloon mean shift est calculé, en chaque point $\mathbf{x}$, en utilisant le noyau normal K de taille $\mathbf{H}(\mathbf{x})$. Le vecteur mean shift normalisé par $\mathbf{H}(\mathbf{x})$ est maximal si la bande passante $\mathbf{H}(\mathbf{x})$ est égale à la covariance $\mathbf{\Sigma}$.

Preuve La preuve de ce théorème a la même forme que celle du théorème 3 donnée dans [Comaniciu 03a].

Considérons l'estimateur à noyau "balloon" défini par :

$$\begin{aligned} \widehat{f}(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}(\mathbf{x})}(\mathbf{x}-\mathbf{x}^{(i)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|\mathbf{H}(\mathbf{x})|^{1/2}} K(\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{x}-\mathbf{x}^{(i)})) \quad . \end{aligned} \tag{C.1}$$

Quand la taille des données augmente, le vecteur mean shift $\mathbf{m}(\mathbf{x})$ converge vers sa limite en probabilité donnée par :

$$\text{plim}\ \mathbf{m}(\mathbf{x}) = \text{plim}\ (\mathbf{H}(\mathbf{x})\frac{\widehat{\nabla} f(\mathbf{x})}{\widehat{f}(\mathbf{x})}), \tag{C.2}$$

En utilisant le théorème suivant :

Théorème 5 (Slutsky's theorem)
Soient (X_n) et (Y_n) deux suites de nombre aléatoire. Si (X_n) converge en probabilité vers X et (Y_n) vers Y, et si $f : \mathbb{R} \to \mathbb{R}$ est une fonction continue,
alors

- *$(X_n + Y_n)$ converge en probabilité vers $X + Y$,*
- *$(X_n Y_n)$ converge en probabilité vers XY,*
- *$(X_n)/(Y_n)$ converge en probabilité vers X/Y,*
- *$f(X_n)$ converge en probabilité vers $f(X)$,*

l'équation (C.2) se réécrit :

$$\begin{aligned} \text{plim } \mathbf{m}(\mathbf{x}) &= \mathbf{H}\frac{\text{plim } (\widehat{\nabla} f(\mathbf{x}))}{\text{plim } (\widehat{f}(\mathbf{x}))} \\ &= \mathbf{H}\frac{E[\widehat{\nabla} f(\mathbf{x})]}{E[\widehat{f}(\mathbf{x})]}. \end{aligned} \tag{C.3}$$

L'espérance d'un l'estimateur à noyau peut être interprétée comme une convolution [Stoker 93]. L'opérateur de convolution sera noté $*$ dans la suite. Si la densité f est une loi normale $\mathcal{N}(\mu, \mathbf{\Sigma})$, alors l'espérance de $\widehat{f}$ est distribuée localement autour du point $\mathbf{x}$ selon une loi normale de moyenne μ et de covariance $\mathbf{\Sigma} + \mathbf{H}(\mathbf{x})$. En effet, on a :

$$\begin{aligned} E[\widehat{f}(x)] = & \int_{\mathbf{x}^{(i)}} K_{\mathbf{H}(\mathbf{x})}(\mathbf{H}_(\mathbf{x})^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)}))f(\mathbf{x}^{(i)})d\mathbf{x}^{(i)} \\ &= (K_{\mathbf{H}(\mathbf{x})} * f)(\mathbf{x}) \\ &\sim \mathcal{N}(\mathbf{x}; \mu, \mathbf{H}(\mathbf{x}) + \mathbf{\Sigma}). \end{aligned} \tag{C.4}$$

Comme l'opérateur gradient est linéaire, l'espérance du gradient de l'estimateur est :

$$\begin{aligned} E[\widehat{\nabla} f(\mathbf{x})] &= \frac{\partial}{\partial \mathbf{x}}\mathcal{N}(\mathbf{x}; \mu, \mathbf{H}(\mathbf{x}) + \mathbf{\Sigma}) \\ &= -(\mathbf{H}(\mathbf{x}) + \mathbf{\Sigma})^{-1}(\mathbf{x} - \mu)E[\widehat{f}(\mathbf{x})]. \end{aligned} \tag{C.5}$$

Ainsi, la limite en probabilité du vecteur mean shift est :

$$\text{plim } \mathbf{m}(\mathbf{x}) = -\mathbf{H}(\mathbf{x})(\mathbf{H}(\mathbf{x}) + \mathbf{\Sigma})^{-1}(\mathbf{x} - \mu) \tag{C.6}$$

et le vecteur mean shift normalisé est alors donné par :

$$m(\mathbf{x}; \mathbf{H}(\mathbf{x})) = \|\mathbf{H}(\mathbf{x})^{-1/2}\text{plim } \mathbf{m}(\mathbf{x})\| \ . \tag{C.7}$$

En suivant [Comaniciu 03a], on peut alors directement montrer que le vecteur mean shift normalisé est maximal si $\mathbf{H}(\mathbf{x}) = \mathbf{\Sigma}$. ■

Liste des figures

0.1 Exemples d'images de séquences de conducteur. (a) Séquence fournie par PSA PEUGEOT CITROËN, acquise dans un simulateur. (b) Séquence en conditions réelles. 8

0.2 Exemples de résultats obtenus sur une séquence fournie par PSA PEUGEOT CITROËN. Première ligne : résultat du suivi par un filtre de Kalman [Kalman 60] des objets détectés. En haut à gauche, résultat sur le conducteur filmé de face. En haut à droite, résultat sur le conducteur filmé de profil. Seconde ligne : Résultats intermédiaires de la détection des objets en mouvement. En bas à gauche, résultat de la détection d'objets en mouvement extraits avec la méthode dite "gap/mountain" décrite dans [Wang 00] appliquée à une carte binaire caractérisant l'appartenance de chaque pixel au mouvement de la caméra (voir chapitre 2). En bas à droite, représentation des objets extraits par des ellipses obtenues avec la méthode décrite dans [Zivkovic 04]. 9

1.1 Quelques images d'une séquence de conducteur. 23

2.1 Carte de détection des pixels en mouvement sur la séquence de ski nautique. (a)-(c) Images 107, 108 et 109 en niveau de gris. (d) Carte des poids W_{t-1} calculée entre t et $t-1$. (e) Carte des poids W_t calculée entre t et $t+1$. (f) Carte M_t de détection des pixels en mouvement. Les pixels blancs sont les pixels en mouvement. 30

2.2 Carte de détection des pixels en mouvement sur la séquence montrant une personne marchant au bord d'un fleuve.(a) Images en niveau de gris. (b) Carte M_t de détection des pixels en mouvement. Les pixels blancs sont les pixels en mouvement. 30

2.3 Résultat du détecteur de coin Harris sur a) l'image 16 de la séquence de conducteur b) l'image 108 de la séquence ski nautique. 31

2.4 Construction de la grille de points (deuxième ligne) et calcul du flot optique (troisième ligne) sur l'image 108 de la séquence de ski nautique. (a) Image initiale. (b) Carte des pixels en mouvement. (c) Grille restreinte uniquement aux pixels en mouvement (les vecteurs de mouvement sont montrés sur l'image (f)). (d) Grille finale obtenue après validation des vecteurs de mouvement par un test de corrélation (seuil = 0.5). (les vecteurs de mouvement sont montrés sur l'image (g)). (e) Grille finale obtenue après validation des vecteurs de mouvement par un test statistique (les vecteurs de mouvement sont montrés sur l'image (h)). 38

2.5 Construction de la grille de points (deuxième ligne) et calcul du flot optique (troisième ligne) sur l'image 16 de la séquence de conducteur. (a) Image initiale. (b) Carte des pixels en mouvement. (c) Grille restreinte uniquement aux pixels en mouvement (les vecteurs de mouvement sont montrés sur l'image (f)). (d) Grille finale obtenue après validation des vecteurs de mouvement par un test de corrélation (seuil = 0.5). (les vecteurs de mouvement sont montrés sur l'image (g)). (e) Grille finale obtenue après validation des vecteurs de mouvement par un test statistique (les vecteurs de mouvement sont montrés sur l'image (h)). 39
2.6 Espace de chrominance YUV a)-c) Les trois canaux rouge, vert et bleu pour l'image 16 de la séquence de conducteur. d)-f) Les trois canaux Y,U,V correspondant 41

4.1 Schéma représentant une itération de notre algorithme. 61
4.2 Validation de l'approche itérative sur l'image d'un chalet. a) Image originale ; b) Résultat de la segmentation avec une approche non itérative ; c) Résultat avec la méthode itérative. 64
4.3 Évolution des segmentations avec les échelles (correspondant aux indices des bandes passantes) pour l'approche non itérative. 65
4.4 Évolution des segmentations avec les échelles (correspondant aux indices des bandes passantes) et les espaces de caractéristiques pour l'approche itérative. . . 66
4.5 Validation de l'approche itérative sur l'image d'une main. a) Image originale ; b) Résultat de la segmentation avec une approche non itérative ; c) Résultat avec la méthode itérative. 67
4.6 Validation de l'approche itérative sur l'image d'un taureau. a) Image originale ; b) Résultat de la segmentation avec une approche non itérative ; c) Résultat avec la méthode itérative. 67
4.7 Validation du clustering "pseudo balloon mean shift" sur l'image d'un chalet. a) Image originale ; b) Résultat de la segmentation avec l'algorithme "variable bandwidth mean shift" ; c) Résultat avec l'algorithme "pseudo balloon mean shift". . . 67
4.8 Validation du clustering "pseudo balloon mean shift" sur l'image d'une main. a) Image originale ; b) Résultat de la segmentation avec l'algorithme "variable bandwidth mean shift" ; c) Résultat avec l'algorithme "pseudo balloon mean shift". . . 68
4.9 Validation du clustering "pseudo balloon mean shift" sur l'image d'un taureau. a) Image originale ; b) Résultat de la segmentation avec l'algorithme "variable bandwidth mean shift" ; c) Résultat avec l'algorithme "pseudo balloon mean shift". 68
4.10 Ordre de traitement des espaces de caractéristiques. Résultats sur l'image d'un chalet du clustering "pseudo balloon mean shift" avec les bandes passantes obtenues en traitant les domaines dans les ordres suivants : a) abscisse, ordonnée, canal rouge, canal vert, canal bleu ; b) canal bleu, canal vert, canal rouge, ordonnée, abscisse. 69
4.11 Ordre de traitement des espaces de caractéristiques. Résultats sur l'image d'un singe du clustering "pseudo balloon mean shift" avec les bandes passantes obtenues en traitant les domaines dans les ordres suivants : a) abscisse, ordonnée, canal rouge, canal vert, canal bleu ; b) canal bleu, canal vert, canal rouge, ordonnée, abscisse. 69
4.12 Résultats de la détection de clusters en mouvement sur l'image 109 de la séquence trafic. 71

5.1 Résultats de la détection sur la séquence de ski nautique pour les images 74, 124, 144, 214, 232, 236 et 242 78
5.2 Résultats de la détection sur la séquence de conducteur pour les images 15, 16, 17 et 48 79
5.3 Résultats de la détection sur la séquence d'une conductrice pour les images 16, 17, 40, 42 et 47 80
5.4 Résultats de la détection sur la séquence trafic. Seules les boîtes englobant les clusters en mouvement sont montrées. 82
5.5 Masques de détection obtenus par différentes méthodes pour la séquence montrant un piéton et un cycliste se déplaçant devant un fleuve. 84
5.6 Masques de détection obtenus par différentes méthodes pour la séquence montrant des piétons marchant au milieu d'arbres. 85

6.1 Exemple de graphe. 94
6.2 Comparaison de l'algorithme de saturation (algorithme 6) et de l'algorithme de Ford & Fulkerson (algorithme 7). En rouge sont montrés des chemins augmentant possibles et en vert est indiqué le flot dans chaque arc. a) Algorithme de saturation correspondant aussi à la première étape de l'algorithme de Ford & Fulkerson. b) Saturation des chemins dans le graphe résiduel (algorithme de Ford & Fulkerson). La figure montre le graphe résiduel avec ses capacités. Les arcs de capacité nulle ne sont pas représentés. c) Graphe résiduel final, dans lequel il n'existe plus de chemin augmentant. 101
6.3 Relation de voisinage (4 ou 8 plus proches voisins). 104
6.4 Capacité des arcs reliant les terminaux aux nœuds du graphe. 107
6.5 Capacité des arcs reliant les sommets mutuellement voisins. 107
6.6 Graines placées à l'extérieur ("B") et à l'intérieur de l'objet ("O") à segmenter : à gauche, l'image originale, à droite, le résultat de la segmentation et les graines. Cette figure provient de [Boykov 01a]. 109
6.7 Exemple de graphe et de segmentation. La capacité de chaque arc est traduite par son épaisseur. Cette figure provient de [Boykov 01a]. 110

7.1 Dilatation du contour et placement de la source et du puits [Xu 02]. Cette figure provient de [Xu 02]. 117

8.1 Fusion ou occultation ? 123
8.2 Principe de l'algorithme de suivi. 125

8.3 Présentation du graphe utilisé. (a) Résultat de la minimisation d'énergie à l'instant $t-1$. Seuls les sommets correspondant aux pixels de l'image sont montrés. L'étiquette des nœuds blancs est "objet" et celle des nœuds noirs est "fond". En bleu est montré en chaque pixel de l'objet le vecteur de mouvement moyen qui permet la prédiction. (b) Graphe coupé correspondant à (a) avec les terminaux. Les arcs coupés ne sont pas dessinés. (c) Forme du graphe à l'instant t. Les terminaux et leurs arcs associés ne sont pas montrés. Les nœuds ($n_t^{(1)}$ et $n_t^{(2)}$) associés aux deux observations détectées $\mathcal{M}_t^{(1)}$ et $\mathcal{M}_t^{(2)}$ sont ajoutés à l'ensemble des sommets correspondant aux pixels. Les sommets en rouge représentent les pixels appartenant aux observations. Les arcs reliant ces sommets aux sommets correspondant aux observations apparaissent en rouge. Cette figure montre également les sommets appartenant à la prédiction. (d) Graphe avec les terminaux associés à (c). Pour des raisons de lisibilité, tous les arcs entre les terminaux et les autres sommets ne sont pas montrés. 126
8.4 Présentation du graphe utilisé. (a) Résultat de l'algorithme de suivi. Trois objets se superposent. (b) Forme du graphe correspondant à la deuxième fonction d'énergie. Il contient 3 terminaux (3 sources), chacun relié à l'ensemble des pixels. Pour plus de clarté, tous les arcs ne sont pas montrés. 131
8.5 Observations obtenues par soustraction de fond. (a) Image de référence. (b) Image courante. (c) Résultat de la soustraction de fond (les pixels en noir sont les pixels ayant changé) et observations déduites (régions entourées d'une boîte englobante rouge). 132
8.6 Résultats sur la séquence PETS 2006 (images 81, 116, 146, 176, 206 et 248). (a) Images originales. (b) Observations. (c) Résultats du suivi. 133
8.7 Séparation avec la seconde fonction d'énergie des objets ayant fusionné (images 101 et 102 de la séquence PETS 2006). (a) Observations. (b) Résultat du suivi et de la segmentation en n'utilisant que la première fonction d'énergie. (c) Résultat obtenu en rjoutant la deuxième énergie. 134
8.8 Résultats de la méthode dans le cas où des observations sont manquantes (le détecteur n'est utilisé que pour une image sur trois entre les images 801 et 807 de la séquence PETS 2006) (a) Observations. (b) Résultat du suivi. (c) Résultats obtenus dans le cas où les observations sont disponibles. 135
8.9 Résultats de la méthode suivi sur la séquence de ski nautique. Les observations sont les clusters en mouvement obtenus avec le détecteur de la partie I. À chaque instant, les observations sont montrées sur l'image de gauche et le masque des objets suivis est visible sur l'image de droite. 136
8.10 Résultats de la méthode suivi sur la séquence d'un conducteur. Les observations sont les clusters en mouvement obtenus avec le détecteur de la partie I. À chaque instant, les observations sont montrées sur l'image de gauche et le masque des objets suivis est visible sur l'image de droite. 137
8.11 Résultats de la méthode de suivi sur la séquence d'une conductrice. Les observations sont les clusters en mouvement obtenus avec le détecteur de la partie I. À chaque instant, les observations sont montrées sur l'image de gauche et le masque des objets suivis est visible sur l'image de droite. 139

B.1 Validation du critère de sélection sur l'image d'un chalet. a) Image originale ; b) Résultat de la segmentation avec le critère "sélection de la meilleure covariance" ; c) Résultat de la segmentation avec le critère "sélection de la meilleure taille prédéfinie". 152
B.2 Validation du critère de sélection sur l'image d'une main. a) Image originale ; b) Résultat de la segmentation avec le critère "sélection de la meilleure covariance" ; c) Résultat de la segmentation avec le critère "sélection de la meilleure taille prédéfinie". 153
B.3 Valeurs finales des bandes passantes pour tous les points situés sur la ligne au milieu de l'image d'une main a) domaine correspondant aux abscisses b) domaine correspondant aux ordonnées c) canal rouge d) canal vert and e) canal bleu. . . . 154

Liste des algorithmes

1 Principe de l'algorithme de détection des objets en mouvement et plan de la partie 26
2 Filtrage mean shift 50
3 Clustering mean shift 51
4 Algorithme "pseudo balloon mean shift" 53
5 Algorithme itératif d'estimation de la taille du noyau 63
6 Algorithme par saturation de chemins 100
7 Algorithme de Ford & Fulkerson 101
8 Algorithme par poussage de flot 102

Bibliographie

[Aach 95] T. Aach, A. Kaup. – Bayesian algorithms for change detection in image sequences using markov random fields. *Signal Processing : Image Communication*, 7(2) :147–160, 1995.

[Ayer 95] S. Ayer, H. Sawhney. – Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. *Proc. Int. Conf. Computer Vision*, 1995.

[BarShalom 87] Y. Bar-Shalom. – *Tracking and data association*. – Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[Bertalmio 00] M. Bertalmio, G. Sapiro, G. Randall. – Morphing active contours. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(7) :733–737, 2000.

[Black 96] M.J. Black, P. Anandan. – The robust estimation of multiple motions : parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1) :75–104, 1996.

[Blake 04] A. Blake, C. Rother, M. Brown, P. Pérez, P. Torr. – Interactive image segmentation using an adaptive gmmrf model. – *Proc. Europ. Conf. Computer Vision*, 2004.

[Boltz 07] S. Boltz, E. Debreuve, M. Barlaud. – High-dimensional statistical distance for region-of-interest tracking : Application to combining a soft geometric constraint with radiometry. – *Proc. Conf. Comp. Vision Pattern Rec.*, 2007.

[Bouthemy 93] P. Bouthemy, P. Lalande. – Recovery of moving object masks in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32(6) :1205–1212, June 1993.

[Boykov 98] Y. Boykov, O. Veksler, R. Zabih. – Markov random fields with efficient approximations. – *Proc. Conf. Comp. Vision Pattern Rec.*, 1998.

[Boykov 01a] Y. Boykov, M. Jolly. – Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *Proc. Int. Conf. Computer Vision*, 2001.

[Boykov 01b] Y. Boykov, O. Veksler, R. Zabih. – Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(11) :1222–1239, 2001.

[Boykov 04] Y. Boykov, V. Kolmogorov. – An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9) :1124–1137, September 2004.

[Cavallaro 00] A. Cavallaro, T. Ebrahimi. – Video object extraction based on adaptive background and statistical change detection. *in Proc. of SPIE VCIP*, 2000.

[Cheng 95] Y. Cheng. – Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(8) :790–799, 1995.

[Comaniciu 99] D. Comaniciu, Peter Meer. – Mean shift analysis and applications. – *Proc. Int. Conf. Computer Vision*, pp. 1197–1203, 1999.

[Comaniciu 01] D. Comaniciu, V. Ramesh, P. Meer. – The variable bandwidth mean shift and Data-Driven scale selection. *Proc. Int. Conf. Computer Vision*, 1, 2001.

[Comaniciu 02] D. Comaniciu, P. Meer. – Mean shift : A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(5) :603–619, 2002.

[Comaniciu 03a] D. Comaniciu. – An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(2) :281–288, 2003.

[Comaniciu 03b] D. Comaniciu, P. Meer, D. Tyler. – Dissimilarity computation through low rank corrections. *Pattern Recognition Letters*, 24(1-3) :227–236, 2003.

[Comaniciu 03c] D. Comaniciu, V. Ramesh, P. Meer. – Kernel-based optical tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(5) :564–577, mai 2003.

[Cox 93] I. Cox. – A review of statistical data association for motion correspondence. *Int. J. Computer Vision*, 10(1) :53–66, 1993.

[Cremers 01] D. Cremers, C. Schnorr, J. Weickert. – Diffusion-snakes : Combining statistical shape knowledge and image information in a variational framework. *Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01)*, 2001.

[Cremers 03] D. Cremers, C. C. Schnörr. – Statistical shape knowledge in variational motion segmentation. *Image and Vision Computing*, 21(1) :77–86, 2003.

[Criminisi 06] A. Criminisi, G. Cross, A. Blake, V. Kolmogorov. – Bilayer segmentation of live video. *Proc. Conf. Comp. Vision Pattern Rec.*, 2006.

[Csurka 99] G. Csurka, P. Bouthemy. – Direct identification of moving objects and background from 2d motion models. *Proc. Int. Conf. Computer Vision*, 1999.

[Dahlhaus 92] E. Dahlhaus, D. Johnson, C. Papadimitriou, P. Seymour, M. Yannakakis. – The complexity of multiway cuts (extended abstract). *ACM Symp. on Theory of Computing*, 1992.

[Darrel 91] T. Darrel, A. Pentland. – Robust estimation of a multi-layered motion representation. *IEEE Workshop on Visual Motion*, 1991.

[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.

[Dimitrova 95] N. Dimitrova, F. Golshani. – Motion recovery for video content classification. *ACM Trans. Inf. Syst.*, 13(4) :408–439, 1995.

[Dinic 70] E. Dinic. – Algorithm for solution of a problem of maximum flow in networks with power estimation. *Soviet Math. Dokl.*, 11 :1277–1280, 1970.

[Doretto 03] G. Doretto, A. Chiuso, Y.N. Wu, S. Soatto. – Dynamic textures. *Int. J. Computer Vision*, 51(2) :91–109, 2003.

[Elgammal 00] A. Elgammal, D. Harwood, L. Davis. – Non-parametric model for background subtraction. *Proc. Europ. Conf. Computer Vision*, 2000.

[Felzenszwalb 06] P. Felzenszwalb, D. Huttenlocher. – Efficient belief propagation for early vision. *Int. J. Computer Vision*, 70 :41–54, 2006.

[Fleet 90] D. Fleet, A. Jepson. – Computation of component image velocity from local phase information. *Int. J. Computer Vision*, 5(1) :77–104, 1990.

[Ford 62] L. Ford, D. Fulkerson. – *Flows in Networks*. – Princeton University Press, Princeton, 1962.

[Friedman 97] N. Friedman, S. Russell. – Image segmentation in video sequences : A probabilistic approach. *Uncertainty in Artificial Intelligence*, pp. 175–181, 1997.

[Fukunaga 75] K. Fukunaga, L.D. Hostetler. – The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Information Theory*, 21(1) :32–40, 1975.

[Fukunaga 90] K. Fukunaga. – *Introduction to statistical pattern recognition (2nd ed.)*. – Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[Geman 84] S. Geman, D. Geman. – Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6(6) :721–741, 1984.

[Goldberg 86] A. Goldberg, R. Tarjan. – A new approach to the maximum flow problem. *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, 1986.

[Goldberg 89] A. Goldberg, E. Tardos, R. Tarjan. – Network flow algorithms. *Technical Report, Stanford University*, 1989.

[Grabmeier 02] J. Grabmeier, A. Rudolph. – Techniques of cluster algorithms in data mining. *Data Min. Knowl. Discov.*, 6(4) :303–360, 2002.

[Grady 06] L. Grady. – Random walks for image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(11) :1768–1783, November 2006.

[Greig 89] D. Greig, B. Porteous, A. Seheult. – Exact maximum a posteriori estimation for binary images. *J. Royal Statist. Soc.*, 51(2) :271–279, 1989.

[Grimson 98] Y. Grimson, C. Stauffer, R. Romano, L. Lee. – Using adaptive tracking to classify and monitor activities in a site. *Proc. Conf. Comp. Vision Pattern Rec.*, 1998.

[Gryn 05] J. Gryn, R. Wildes, J. Tsotsos. – Detecting motion patterns via direction maps with application to surveillance. *Proc. of the Seventh IEEE Workshops on Application of Comp. Vision*, 1 :202–209, 2005.

[Haritaoglu 00] I. Haritaoglu, D. Harwood, L. David. – W4 : Real-time surveillance of people and their activities. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8) :809–830, 2000.

[Harris 88] C. Harris, M. Stephens. – A combined corner and edge detector. *Alvey Vision Conference*, 1988.

[Heeger 88] D. Heeger. – Optical flow using spatio-temporal filters. *Int. J. Computer Vision*, 1(4) :279–302, 1988.

[Hérault 06a] R. Hérault, F. Davoine, F. Dornaika, Y. Grandvalet. – Suivis simultanées et robustes de visages et de gestes faciaux. *15eme Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, 2006.

[Hérault 06b] R. Hérault, F. Davoine, Y. Grandvalet. – Head and facial action tracking : Comparison of two robust approaches. *International Conference on Automatic Face and Gesture Recognition*, 2006.

[Horn 81] B. Horn, B. Schunck. – Determining optical flow. *Artif. Intell.*, 17(1-3) :185–203, 1981.

[Hsu 84] Y. Hsu, H. Nagel, G. Rekers. – New likelihood test methods for change detection in image sequences. *Comput. Vision, Graphics, Image Proc.*, 26(1) :73–106, 1984.

[Hubert 74] L. Hubert. – Approximate evaluation technique for the single-link and complete-link hierarchical clustering procedure. *J. of the American Statistical association*, 69 :968, 1974.

[Huwer 00] S. Huwer, H. Niemann. – Adaptive change detection for real-time surveillance applications. – *Third IEEE International Workshop on Visual Surveillance*, Dublin, 2000.

[Irani 98] M. Irani, P. Anandan. – A unified approach to moving object detection in 2d and 3d scenes. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(6) :577–589, 1998.

[Isard 98] M. Isard, A. Blake. – Condensation – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1) :5–28, 1998.

[Jackson 04] J. Jackson, A. Yezzi, S. Soatto. – Tracking deformable moving objects under severe occlusions. *IEEE Conference on Decision and Control*, 2004.

[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.

[Jain 99] A. K. Jain, M. N. Murty, P. J. Flynn. – Data clustering : a review. *ACM Comput. Surv.*, 31(3) :264–323, 1999.

[Jeffreys 46] H. Jeffreys. – An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London*, 186(1007) :453–461, March 1946.

[Jepson 93] A. Jepson, M. Black. – Mixture models for optical flow computation. *Proc. Conf. Comp. Vision Pattern Rec.*, 1993.

[Jepson 03] A. Jepson, D. Fleet, T. El-Maraghi. – Robust online appearance models for visual tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, 2003.

[Kalman 60] R. Kalman. – A new approach to linear filtering and prediction problems. *J. Basic Eng.*, 82 :35–45, 1960.

[Kanade 98] T. Kanade, R. Collins, A. Lipton, P. Burt, L. Wixson. – Advances in cooperative multi-sensor video surveillance, 1998.

[Karmann 90] K. Karmann, A. Brand. – *Time-varying image processing and moving object recognition*. – Elsevier Science Publish., 1990.

[Kim 05] K. Kim, D. Harwood, L. Davis. – Background updating for visual surveillance. *Int. Symposium on Visual Computing.*, 2005.

[Kjeldsen 96] R. Kjeldsen, J. Kender. – Finding skin in color images. *International Conference on Automatic Face and Gesture Recognition*, 1996.

[Koller 94] D. Koller, J. Weber, J. Malik. – Robust multiple car tracking with occlusion reasoning. – *Proc. Europ. Conf. Computer Vision*, 1994.

[Kolmogorov 04] V. Kolmogorov, R. Zabih. – What energy functions can be minimized via graph cuts ? *IEEE Trans. Pattern Anal. Machine Intell.*, 26(2) :147–159, 2004.

[Kolmogorov 05] V. Kolmogorov, M. Wainwright. – On the optimality of tree-reweighted max-product message-passing. *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, 2005.

[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.

[Konrad 00] J. Konrad. – *Handbook of image and Video processing*. – Academic press, 207–225p., 2000.

[Kullback 51] S. Kullback, R. A. Leibler. – On information and sufficiency. *Annals of Mathematical Statistics*, 22(1) :79–86, March 1951.

[Lance 67] G. Lance, W. Williams. – A general theory of classificatory sorting strategies. i. hierarchical systems. *Computer Journal*, (9) :373–380, 1967.

[Laptev 05] I. Laptev, S. Belongie, P. Pérez, J. Wills. – Periodic motion detection and segmentation via approximate sequence alignment. *Proc. Int. Conf. Computer Vision*, 2005.

[L.Breiman 77] W. Meisel L. Breiman, E. Purcell. – Variable kernel estimates of multivariate densities. *Technometrics*, 19 :135–144, 1977.

[Lebart 84] L. Lebart. – Correspondence analysis of graph structure. *Bulletin technique du CESIA, Paris*, pp. 5–19, 1984.

[Leung 00] Y. Leung, J. Zhang, X. Zong-Ben. – Clustering by scale-space filtering. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(12) :1396–1410, 2000.

[Lin 91] J. Lin. – Divergence measures based on the shannon entropy. *IEEE Trans. Information Theory*, 37(1) :145–151, January 1991.

[Lisani 03] J. Lisani, J. Morel. – Detection of major changes in satellite images. *Proc. Int. Conf. Image Processing*, 2003.

[Loftsgaarden 65] D. Loftsgaarden, C. Quesenberry. – A nonparametric estimate of a multivariate density function. *Annals of Mathematical Statistics*, 36 :1049–1051, 1965.

[Lowe 99] D.G. Lowe. – Object recognition from local scale-invariant features. *Proc. Int. Conf. Computer Vision*, 1999.

[Lucas 81] B.D. Lucas, T. Kanade. – An iterative technique of image registration and its application to stereo. *Proc. Int. Joint Conf. on Artificial Intelligence*, 1981.

[Lucchese 01] L. Lucchese, S. Mitra. – Color image segmentation : A state-of-the-art survey. *Proc. of the Indian National Science Academy*, 67 :207–221, 2001.

[MacCormick 00] J. MacCormick, A. Blake. – A probabilistic exclusion principle for tracking multiple objects. *Int. J. Computer Vision*, 39(1) :57–71, 2000.

[Mack 79] Y. Mack, M. Rosenblatt. – Multivariate k-nearest neighbor density estimates. *J. Multivariate Analysis*, 9 :1–15, 1979.

[Mansouri 02] A. Mansouri. – Region tracking via level set pdes without motion computation. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(7) :947–961, 2002.

[Meziane 02] A. Meziane. – Digital images segmentation : a state of art of the different methods. *Revue d'Information Scientifique et Technique*, 12(1) :105–120, 2002.

[Milligan 85] G. Milligan, M. Cooper. – An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50 :159–179, 1985.

[Mittal 04] A. Mittal, N. Paragios. – Motion-based background subtraction using adaptative kernel density estimation. *Proc. Conf. Comp. Vision Pattern Rec.*, 2004.

[Nelson 91] R. Nelson. – Qualitative detection of motion by a moving observer. *Proc. Conf. Comp. Vision Pattern Rec.*, 1991.

[Nguyen 04] H. Nguyen, A. Smeulders. – Fast occluded object tracking by a robust appearance filter. *IEEE Trans. Pattern Anal. Machine Intell.*, 2004.

[Niethammer 04] M. Niethammer, A. Tannenbaum. – Dynamic geodesic snakes for visual tracking. *Proc. Conf. Comp. Vision Pattern Rec.*, 2004.

[Odobez 95] J.-M. Odobez, P. Bouthemy. – Robust multiresolution estimation of parametric motion models. *J. Visual Com. and Image Representation*, 6(4) :348–365, December 1995.

[Odobez 97] J.-M. Odobez, P. Bouthemy. – *Separation of moving regions from background in an image sequence acquired with a mobile camera*. – Kluwer Academic Publisher, 283–311p., 1997.

[Pal 93] N. Pal, S. Pal. – A review on image segmentation techniques. *Pattern Recognition*, 26(9) :1277–1294, 1993.

[Papadakis 07] N. Papadakis, É. Mémin. – Variational optimal control technique for the tracking of deformable objects. – *Proc. Int. Conf. Computer Vision*, 2007.

[Paragios 99a] N. Paragios, R. Deriche. – Geodesic active regions for motion estimation and tracking. – *Proc. Int. Conf. Computer Vision*, 1999.

[Paragios 99b] N. Paragios, G. Tziritas. – Adaptive detection and localization of moving objects in image sequences. *Signal Processing : Image Communication*, 14 :277–296, 1999.

[Park 90] B. Park, J. Marron. – Comparison of data-driven bandwidth selectors. *J. of the American Statistical association*, 85(409) :66–72, 1990.

[Pena 01] D. Pena, F.J. Prieto. – Robust covariance matrix estimation and multivariate outlier detection. *Technometrics*, 43(3) :286–310, 2001.

[Pitié 05] F. Pitié, S. Berrani, R. Dahyot, A. Kokaram. – Off-line multiple object tracking using candidate selection and the viterbi algorithm. *Proc. Int. Conf. Image Processing*, 2005.

[Press 92] W. Press, S. Teukolsky, W. Vetterling, B. Flannery. – *Numerical Recipes in C : The Art of Scientific Computing*. – Cambridge University Press, New York, NY, USA, 1992.

[Pundlik 06] S. Pundlik, S. Birchfield. – Motion segmentation at any speed. *Proc. of the British Machine Vision Conf.*, 2006.

[Radke 05] R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam. – Image change detection algorithms : a systematic survey. *IEEE Transactions on Image Processing*, 14(3) :294–307, 2005.

[Reid 79] D Reid. – An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control*, 24(6) :843–854, 1979.

[Ren 03] Y. Ren, C. Chua, Y. Ho. – Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24(1-3) :183–196, January 2003.

[Rihan 06] J. Rihan, P. Kohli, P. Torr. – Objcut for face detection. *Indian Conf. on Computer Vision, Graphics and Image Proc.*, 2006.

[Ronfard 94] R. Ronfard. – Region-based strategies for active contour models. *Int. J. Computer Vision*, 13(2) :229–251, 1994.

[Rosin 98] P. Rosin. – Thresholding for change detection. *Proc. Int. Conf. Computer Vision*, pp. 274–279, 1998.

[Rother 04] C. Rother, V. Kolmogorov, A. Blake. – "grabcut" : interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3) :309–314, 2004.

[Rother 06] C. Rother, T. Minka, A. Blake, V. Kolmogorov. – Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. – *Proc. Conf. Comp. Vision Pattern Rec.*, 2006.

[Sahouria 97] E. Sahouria, A. Zakhor. – Motion indexing of video. *Proc. Int. Conf. Image Processing*, 1997.

[Sato 04] K. Sato, J. Aggarwal. – Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96(2) :100–128, 2004.

[Sawhney 96] H. Sawhney, S. Ayer. – Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(8) :814–830, 1996.

[Schrijver 03] A. Schrijver. – *Combinatorial Optimization - Polyhedra and Efficiency*. – Springer, 2003.

[Scott 92] D. Scott. – Multivariate density estimation. *Wiley-Interscience*, 1992.

[Shafique 03] K. Shafique, M. Shah. – A non-iterative greedy algorithm for multi-frame point correspondence. *Proc. Int. Conf. Computer Vision*, 2003.

[Sheather 91] S. Sheather, M. Jones. – A reliable data-based bandwidth selection method for kernel density estimation. *J. Royal Statist. Soc.*, 53 :683–690, 1991.

[Sheikh 05] Y. Sheikh, M. Sha h. – Bayesian modeling of dynamic scenes for object detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(11) :603–619, 2005.

[Shi 94] J. Shi, C. Tomasi. – Good features to track. *Proc. Conf. Comp. Vision Pattern Rec.*, 1994.

[Shi 00] J. Shi, J. Malik. – Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8) :888–905, 2000.

[Shi 05] Y. Shi, W. Karl. – Real-time tracking using level sets. *Proc. Conf. Comp. Vision Pattern Rec.*, 2005.

[Silverman 86] B. Silverman. – Density estimation for statistics and data analysis. *Chapman and Hall*, 1986.

[Simonoff 96] J. Simonoff. – Smoothing methods in statistics. *Springer-Verlag*, 1996.

[Singh 03] S. Singh, D. Chauhan, M. Vatsa, R. Singh. – A robust skin color based face detection algorithm. *Tamkang Journal of Science and Engineering*, 6(4) :227–234, 2003.

[Spinei 98] A. Spinei, D. Pellerin, J. Herault. – Spatiotemporal energy-based method for velocity estimation. *Signal Processing*, 65 :347–362, 1998.

[Stoker 93] T. M. Stoker. – Smoothing bias in derivative estimation. *J. of the American Statistical association*, 88(423) :855–863, 1993.

[Sullivan 06] J. Sullivan, S. Carlsson. – Tracking and labelling of interacting multiple targets. *Proc. Europ. Conf. Computer Vision*, 2006.

[Terrell 92] G. Terrell, D. Scott. – Variable kernel density estimation. *The Annals of Statistics*, pp. 1236–1265, 1992.

[Terrillon 00] J. Terrillon, S. Akamatsu. – Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images. *International Conference on Automatic Face and Gesture Recognition*, 2000.

[Terzopoulos 93] D. Terzopoulos, R. Szeliski. – Tracking with kalman snakes. *Active vision*, pp. 3–20, 1993.

[Thompson 93] W. Thompson, P. Lechleider, E. Stuck. – Detecting moving objects using the rigidity constraint. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(2) :162–166, 1993.

[Tian 05] Y.L. Tian, A. Hampapur. – Robust salient motion detection with complex background for real-time video surveillance. *Workshop on Motion and Video Computing*, 2005.

[Tomasi 91] C. Tomasi, T. Kanade. – *Detection and Tracking of Point Features*. – Rapport de Recherche nCMU-CS-91-132, Carnegie Mellon University, April 1991.

[Toyama 99] K. Toyama, J. Krumm, B. Brumitt, B. Meyers. – Wallflower : Principles and practice of background maintenance. – *Proc. Int. Conf. Computer Vision*, 1999.

[Tsybakov 03] A. Tsybakov. – Introduction à l'estimation non paramétrique. *Mathematiques et Applications*, 41, 2003.

[Ullman 79] S. Ullman. – The interpretation of visual motion. *MIT Press*, 1979.

[Veit 04] T. Veit, F. Cao, P. Bouthemy. – *An a contrario framework for motion detection*. – Rapport de Recherche n5313, INRIA, 2004.

[Veit 05a] 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 05b] T. Veit, F. Cao, P. Bouthemy. – A maximality principle applied to a contrario motion detection. *Proc. Int. Conf. Image Processing*, 2005.

[Vidal 04] R. Vidal, Y. Ma. – A unified algebraic approach to 2-d and 3-d motion segmentation. *Proc. Europ. Conf. Computer Vision*, 2004.

[Vidal 05] R. Vidal, D. Singaraju. – A closed form solution to direct motion segmentation. *Proc. Conf. Comp. Vision Pattern Rec.*, 2005.

[Wand 95] M. P. Wand, M. C. Jones. – *Kernel Smoothing*. – Chapman & Hall, London, U.K., 1995.

[Wang 94] J. Y. A. Wang, E. H. Adelson. – Representing moving images with layers. *IEEE Trans. on Image Processing Special Issue*, 3(5) :625–638, 1994.

[Wang 00] Y. Wang, J.F. Doherty, R.E. Van Dyck. – Moving object tracking in video. *Applied Imagery Pattern Recognition (AIPR) Annual Workshop*, 2000.

[Wang 02] N. Wang, Adrian E. Raftery. – Nearest neighbor variance estimation (nnve) : Robust covariance estimation via nearest neighbor cleaning. *J. of the American Statistical association*, 97(460) :994–, December 2002.

[Weiss 97] Y. Weiss. – Smoothness in layers : Motion segmentation using nonparametric mixture estimation. *Proc. Conf. Comp. Vision Pattern Rec.*, 1997.

[Wildes 98] R. Wildes. – A measure of motion salience for surveillance applications. *Proc. Int. Conf. Image Processing*, 1998.

[Wixson 00] L. Wixson. – Detecting salient motion by accumulating directionally-consistent flow. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8) :774–780, 2000.

[Wren 97] C.R. Wren, A. Azarbayejani, T. Darrell, A. Pentland. – Pfinder : Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(7) :780–785, 1997.

[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. – Accurate motion layer segmentation and matting. *Proc. Conf. Comp. Vision Pattern Rec.*, 2005.

[Xu 02] N. Xu, N. Ahuja. – Object contour tracking using graph cuts based active contours. *Proc. Int. Conf. Image Processing*, 2002.

[Xu 03] N. Xu, R. Bansal, N. Ahuja. – Object segmentation using graph cuts based active contours. *Proc. Conf. Comp. Vision Pattern Rec.*, 2003.

[Yilmaz 04] A. Yilmaz. – Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(11) :1531–1536, 2004.

[Yilmaz 06] A. Yilmaz, O. Javed, M. Shah. – Object tracking : A survey. *ACM Comput. Surv.*, 38(4) :13, 2006.

[Yuille 03] A. Yuille, A. Rangarajan. – The concave-convex procedure. *Neural Comput.*, 15(4) :915–936, 2003.

[Zabih 04] Ramin Zabih, Vladimir Kolmogorov. – Spatially coherent clustering using graph cuts. *Proc. Conf. Comp. Vision Pattern Rec.*, 2004.

[Zhang 06] Y. Zhang. – Advances in image and video segmentation. – *IRM Press*, 2006.

[Zhong 03] J. Zhong, S. Sclaroff. – Segmenting foreground objects from a dynamic textured background via a robust kalman filter. *Proc. Int. Conf. Computer Vision*, 2003.

[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.

[Zivkovic 04] Z. Zivkovic, B. Krose. – An em-like algorithm for color-histogram-based object tracking. *Proc. Int. Conf. Computer Vision*, 2004.

Liste des publications relatives aux travaux de thèse

Conférences internationales

1. A. Bugeau, P. Pérez. Joint Tracking and Segmentation of Objects using Graph Cuts. *Proc. Conf. Advanced Concepts for Intelligent Vision Systems (ACIVS' 07), Delft, the Netherlands*, August 2007.
2. A. Bugeau, P. Pérez. Detection and segmentation of moving objects in highly dynamic scenes. *Proc. Int. Conf. Computer Vision and Pattern Recog. (CVPR' 07), Minneapolis, MI*, June 2007.

Conférences nationales

1. A. Bugeau, P. Pérez. Sélection de la taille du noyau pour l'estimation à noyau dans des espaces multidimensionnels hétérogènes. *21ème colloque GRETSI sur le traitement du signal et des images, Troyes*, September 2007.

Rapports de recherche

1. A. Bugeau, P. Pérez. Detection and segmentation of moving objects in highly dynamic scenes. *Rapport de recherche IRISA*, No 6282, septembre 2007.
2. A. Bugeau, P. Pérez. Bandwidth selection for kernel estimation in mixed multi-dimensional spaces. *Rapport de recherche IRISA*, No 6286, septembre 2007.

Soumission

1. A. Bugeau, P. Pérez. Detection and segmentation of moving objects in highly dynamic scenes. *soumis à Computer Vision and Image Understanding.*
2. A. Bugeau, P. Pérez. Bandwidth selection for kernel estimation in mixed multi-dimensional spaces. *soumis à Journal of Mathematical Imaging and Vision.*

Résumé

De nombreuses applications en vision par ordinateur nécessitent la détection et le suivi des objets en mouvement dans une séquence d'images. La plupart des méthodes existantes ne donnent de bons résultats que pour des séquences avec des fonds peu changeants, ou si le fond et les objets sont rigides. Le but de cette thèse est de détecter et suivre les objets mobiles dans des séquences (telles que des séquences de conducteurs) ayant un fond dynamique, avec de forts changements d'illumination, de faibles contrastes et éventuellement acquises par une caméra mobile. Cette thèse est décomposée en deux parties. Dans la première, une méthode de détection est proposée. Elle repose sur la définition d'une grille de points décrits par leur mouvement et leur photométrie. Ces points sont ensuite regroupés en "clusters en mouvement" avec un algorithme mean shift à noyau variable et une méthode de sélection automatique de la taille des noyaux. La deuxième partie propose une méthode de suivi combinant des distributions de couleur et de mouvement, la prédiction de l'objet et des observations extérieures (pouvant être les clusters en mouvement) dans une fonction d'énergie minimisée par coupe minimale/flot maximal dans un graphe. Les algorithmes de détection et de suivi sont validés sur différentes séquences aux contenus dynamiques complexes.

Mots clés : Analyse du mouvement, détection de mouvement, suivi, mean shift, coupe minimale/flot maximal dans un graphe.

Abstract

Detecting and tracking moving objects in dynamic scenes is a hard but essential task in a large number of computer vision applications such as surveillance. Most existing methods only give good results in the case of persistent or slowly changing background, or if both the objects and the background can be characterized by simple parametric motions. This thesis aims at detecting, segmenting and tracking foreground moving objects in sequences (such as driver sequences) having highly dynamic backgrounds, illumination changes and low contrasts, and possibly shot by a moving camera. Two main steps compose the thesis. First, moving points, described by their motion and color, are selected within a sub-grid of image pixels. Clusters of points are then formed using a variable bandwidth mean shift with automatic bandwidth selection. In a second part, a tracking method is proposed. It combines color and motion distributions, the prediction of the tracked object and some external observations (which can be the clusters from the detector) into an energy function minimized with Graph Cuts. The algorithms are validated on several different sequences with complex dynamic contents.

Keywords : Motion analysis, moving objects detection, tracking, mean shift, Graph Cuts.