Université PARIS DESCARTES
Centre universitaire des Saints-Pères
UFR DE MATHÉMATIQUES ET INFORMATIQUE

Thèse présentée pour l'obtention du grade de Docteur
de l'université PARIS DESCARTES
Spécialité : **Informatique**

Sujet de thèse :

Suivi d'objets en mouvement dans une séquence vidéo

NICOLAS VERBEKE

Soutenue le ... décembre 2007, devant le jury composé de :

M.	Maurice	Milgram	Rapporteurs
M.	Henri	Nicolas	
M.	Patrick	Bouthemy	Examinateurs
M.	Laurent	Cohen	
M.	Mohamed	Nadif	
Mme	Nicole	Vincent	
M.	Philippe	Mouttou	Invité

Remerciements

Résumé

Table des matières

Table des figures

Chapitre 1

Introduction générale

De nos jours, le stockage de grands volumes de données est devenu possible et abordable. Parallèlement, la puissance de calcul des microprocesseur a décuplé, et les caméras numériques sont devenues extrêmement performantes pour un coût de plus en plus bas. Ainsi, depuis que le traitement des données vidéo en temps réel est devenu sérieusement envisageable, pour des problématiques aussi diverses que l'analyse statistique de la fréquentation d'un lieu, la sécurisation de l'accès à des bâtiments, la surveillance de malades épileptiques dans des hôpitaux, ou encore la facturation des véhicules aux péages des autoroutes, les industriels proposent de plus en plus de solutions techniques basées sur l'acquisition numérique de séquences vidéo, alors qu'auparavant, des solutions plus mécaniques auraient été privilégiées.

Nous allons, dans cette introduction, situer dans son contexte industriel la problématique qui a fait l'objet de de ces trois années de thèses, puis décrire les enjeux et les principales difficultés qui seront les nôtres.

1.1 Problématique et contexte industriel

Ces dernières années, la sécurité des personnes est devenue une thématique omniprésente dans l'actualité et porteuse d'enjeux politiques majeurs. Par ailleurs, les progrès réalisés par les fabricants de composants électroniques et informatiques ont fait que la mise en place d'importants réseaux de caméras de vidéosurveillance est devenu abordable pour les compagnies industrielles qui gèrent des sites réputés sensibles. Parmi les plus gros consommateurs de caméras de surveillance, on retrouve notamment les aéroports, les gares ferroviaires, et les sociétés de transports en commun qui voient passer quotidiennement des milliers, voire des millions de passagers. Par exemple, dans un aéroport international important, on compte plus de 4000 caméras de surveillance.

Le domaine des transports n'est pas le seul à produire de tels réseaux de caméras. Dans la grande distribution également, la vidéo numérique est utilisée de manière intensive. L'objectif n'est pas nécessairement lié à la sécurité, mais de plus en plus souvent au marketing. Ainsi, on peut souhaiter analyser le parcours type d'un client afin d'optimiser la disposition de certains rayons

de supermarché, où encore compter le nombre de personnes qui fréquentent quotidiennement un centre commercial.

Quel que soit le domaine d'application, chaque caméra produit en continu des données vidéo, c'est-à-dire 25 à 30 images numériques par seconde, qui ont la particularité d'être particulièrement volumineuses, ce qui interdit de les stocker en totalité. Actuellement, le seul moyen de trouver une utilité à ces données est le contrôle visuel par un opérateur humain. Évidemment, cette méthode a ses limites, car la dimension des réseaux de caméra croît plus rapidement que les moyens humains disponibles pour analyser les données produites. C'est pourquoi, depuis que la puissance de calcul des ordinateurs le permet, on cherche à automatiser les tâches réalisées juqu'à présent par des opérateurs humains.

Dans tous les cas, il s'agit de détecter les objets en mouvement, et d'analyser leurs déplacements. Même si les objets d'intérêt sont le plus souvent des êtres humains, il arrive que des applications aient pour objectif l'analyse des mouvements des véhicules. On pourra penser à la sécurisations des parcs de stationnement par exemple, ou encore à la facturation automatique à la sortie des autoroutes. En outre, les sciences médicales produisent de plus en plus de données où la dimension temporelle est présente. Ce dernier domaine d'application ne sera pas abordé dans cette thèse, mais le lecteur intéressé pourra se référer aux récents travaux de [Genovesio, 2005] pour plus de détails sur cette problématique.

Ainsi, les applications se distinguent par le type d'objets que l'on souhaite suivre, mais également par l'objectif de l'analyse. En effet, la difficulté des tâches est très variable. Parmi les besoins que l'on rencontre, on pourra citer le comptage de personnes ou de véhicules empruntant un lieu de passage identifié, le relevé automatique des immatriculations des véhicules entrant dans un parking ou sur une autoroute, le calcul de la durée de stationnement d'un véhicule à un endroit donné, la détection d'événements précisément décrits (comme l'abandon d'un bagage dans une gare), voire la détection d'événements « anormaux » (agression, mouvement de foule, ...) pourvu qu'il existe une description permettant de caractériser la normalité d'une situation.

Cette variabilité des besoins explique pourquoi, jusqu'à présent, la plupart des solutions informatiques proposées pour l'analyse de séquences vidéo ont un caractère dédié. Ainsi, une application de suivi de personnes ne sera pas adaptable à un problème de suivi de véhicules ; une application qui détecte le passage d'êtres humains par la forme caractéristique de leur silhouette lorqu'ils sont vus de face, ne fonctionnera plus si l'on décide de fixer la caméra au plafond afin d'obtenir une vue de dessus ; une application prévue pour l'analyse de scènes d'intérieur sera incapable de gérer les changements de luminosité que l'on observe en extérieur.

Afin d'améliorer la réutilisabilité des systèmes d'analyse de séquences vidéo mis au point, il faut absolument introduire un certain degré de généricité dans ces systèmes. C'est ce caractère générique qui intéresse particulièrement l'entreprise MKL System qui a financé ce travail de thèse. En tant qu'éditeur

de composants logiciels destinés à l'industrie, la pérennisation de l'entreprise passe nécessairement par le développement de composants au moins partiellement réutilisables.

1.2 Définition des tâches

Comme nous l'avons évoqué précédemment, sous l'appellation « analyse de séquences vidéo » sont regroupés des besoins très divers. Pour obtenir la généricité qui manque tant aux systèmes existants, nous pouvons commencer par rechercher ce qu'il y a de commun entre les différentes applications visées.

Parmi les exemples les plus simples, on pourra considérer le cas où l'on souhaite compter le nombre de personnes qui empruntent un couloir filmé par une caméra. Cela peut être réalisé en détectant les régions de l'image représentant une personne en mouvement, en associant entre elles les régions qui représentent la même personne détectée à des instants différents, en en déduisant la trajectoire de la personne suivie, et en incrémentant un compteur lorsque cette personne quitte le champ de vision. Cette tâche peut être très simple si aucun mouvement parasite ne vient perturber la détection, et si les personnes ne se déplacent pas en groupes compacts.

Considérons maintenant un cas très actuel : la détection de bagages abandonnés dans les halls de gare. Même si ce problème semble à première vue très différent du précédent, et bien plus complexe, les deux applications ont néanmoins en commun une partie importante du traitement. En effet, pour détecter l'abandon d'un bagage par un voyageur, il est d'abord nécessaire de détecter la présence du voyageur lui-même dans chaque image de la séquence vidéo, et de faire le lien entre chacune de ces détections. Il faut ensuite reconstruire la trajectoire du voyageur et détecter à quel moment il laisse le bagage — soit par analyse de sa silhouette, soit par modélisation et soustraction de l'arrière-plan.

On constate donc qu'on peut identifier deux étapes nécessaires à toute application d'analyse vidéo pour le suivi d'objets :

- La première étape consiste à détecter les régions de l'image où a eu lieu un mouvement. Celle-ci doit être exécutée pour chaque image de la vidéo, ou pour chaque durée élémentaire de la séquence à analyser. On produit ainsi, à intervalles réguliers, une liste de zones représentant un ou plusieurs objets en mouvement. La partie de l'image n'appartenant à aucune de ces zones peut être considérée comme représentant l'arrière-plan de la scène. Il s'agit donc de réaliser une *segmentation* entre objets mobiles et arrière-plan.
- Afin d'assigner une trajectoire à chaque objet, nous devons établir des correspondances entre zones de mouvement détectées à des instant différents. Cette étape doit également permettre de clarifier les ambiguïtés qui se produisent lorsque plusieurs objets se déplacent dans une même région de l'image. Il s'agit ici de *modéliser les déplacements* d'objets qui ont lieu dans la scène filmée.

L'exploitation des trajectoires ainsi obtenues dépend de l'application finale et ne peut pas être générique. Le cadre de cette étude se limitera donc aux deux étapes identifiées ci-dessus. Si un industriel possède deux briques logicielles génériques correspondant à ces tâches, lorsqu'un nouveau besoin d'application basée sur de l'analyse de séquences vidéo se présentera, tous les efforts de développement pourront être concentrés sur l'interprétation des déplacements extraits, ce qui représente un gain de temps conséquent.

1.3 Plan et approche choisie

La détection automatique du mouvement dans les séquences vidéo est un sujet très actif depuis le début des années 1980. Dans une première partie, nous allons passer en revue les différentes approches proposées dans la littérature ainsi que les quelques états de l'art publiés à ce sujet (chapitre 2). Nous constaterons que la dimension temporelle des données vidéo (qui confère à ces données leur caractère volumineux) est généralement traitée de manière peu satisfaisante. En effet, la plupart des méthodes proposées dans la littérature consiste à modéliser l'arrière-plan de la scène à chaque instant — c'est-à-dire à synthétiser tout le passé de la séquence dans une image unique — puis à confronter l'image courante à ce modèle afin de définir pour tout point si celui-ci appartient à l'arrière-plan, ou s'il s'agit d'un objet mobile. Nous pensons qu'il est possible de conserver une connaissance moins synthétique du passé en prenant mieux en compte la nature tridimenionnelle des données vidéo.

Comme la difficulté majeure liée à ce type de données est leur importante dimensionnalité, nous étudierons dans un seconde partie de l'état de l'art (chapitre 3), les méthodes de réduction de dimension qui ont été appliquées avec succès dans le domaine de l'analyse d'images et de séquences vidéo.

Cette étude de l'existant nous amènera à proposer une nouvelle approche qui fera l'objet de la deuxième partie de ce document. La plus grande part de ce travail de thèse a été consacrée à proposer une nouvelle méthode de détection des régions en mouvement dans les images vidéo. L'approche proposée utilise une méthode d'analyse de données, l'analyse en composantes principales, pour représenter les données vidéo dans un espace de dimension réduite adapté au contenu de la scène, dans lequel il sera plus facile de réaliser la segmentation entre les zones statiques et les régions représentant du mouvement. La présentation de cette méthode fait l'objet du chapitre 4.

Enfin, la deuxième étape du traitement, qui consiste à modéliser les déplacements d'objets à partir de la segmentation obtenue, est présentée dans le chapitre 5. Nous proposons une approche à base de graphes, qui permet de modéliser les incertitudes afin de les clarifier lorsque les informations nécessaires seront disponibles.

Première partie

État de l'art

Chapitre 2

Détection de mouvement dans la littérature

Sommaire

La plupart des algorithmes de suivi d'objets prenant en entrée les images fournies par une caméra fixe effectuent une première étape de détection de mouvement afin de déterminer parmi les pixels de l'image courante lesquels appartiennent à l'arrière-plan de la scène, et lesquels représentent des objets mobiles. Ce domaine de recherche est très actif depuis les débuts de l'analyse

de séquences vidéo à la fin des années 1970. Depuis cette date, le nombre d'articles publiés chaque année sur ce sujet ne cesse de croître, et en particulier depuis le milieu des années 1990, lorsque la puissance des ordinateurs grand public a permis d'envisager sérieusement un traitement en temps réel des données vidéo. Face à cette multitude de méthodes proposées dans la littérature, plusieurs états de l'art ont été publiés afin de tenter d'établir une taxinomie des algorithmes existants. Dans la section 2.1, nous allons passer en revue les différentes classifications proposées.

2.1 Travaux antérieurs

Classer et hiérarchiser les différentes méthodes de détection de mouvement présentes dans la littérature n'est pas une tâche aisée. En effet, les auteurs utilisent souvent une terminologie différente pour désigner des méthodes ou algorithmes qui ont le même objectif. Par ailleurs, un même terme dans un article particulier peut désigner un petit module bien défini d'un algorithme, alors que dans un autre, il désignera la totalité d'un système informatique incluant ce module ainsi que beaucoup d'autres. Dans ces conditions, il est nécessaire de définir avec précision les termes rencontrés afin d'éviter toute confusion.

2.1.1 Glossaire

Détection de mouvement (*Motion detection*). Ce terme est le plus générique, il indique uniquement que l'on parle d'une méthode qui a pour objet de trouver en quels points de l'image un mouvement a eu lieu. Un algorithme ayant cet objectif fournit en sortie une variable quantitative (« quantité de mouvement ») ou qualitative (booléenne) pour tout pixel de chaque image d'entrée. Toutes les méthodes présentées ci-après rentrent dans cette catégorie.

Estimation du mouvement (*Motion estimation*). Cette notion inclut la précédente en y ajoutant la contrainte que le résultat fourni doit être quantitatif. Ce terme est surtout utilisé par les auteurs travaillant sur l'estimation du flot optique ; dans ce cas, le résultat de l'estimation est un champ de vecteurs à deux dimensions représentant la projection sur le plan de l'image du mouvement réel tridimensionnel ayant lieu dans la scène. Pour un état de l'art détaillé et une étude comparative des méthodes classiques de calcul du flot optique, voir [Barron *et al.*, 1994].

Modélisation de l'arrière-plan (*Background modeling*). Cette catégorie regroupe toutes les méthodes de détection de mouvement qui consistent à créer un modèle de l'arrière-plan de la scène filmée (sans aucun objet mobile). Ce modèle peut être une image créée à partir des pixels observés à différents

instants de la séquence vidéo, comme dans [Yang *et al.*, 2004], un modèle statistique décrivant la fonction de distribution des niveaux de gris ou des couleurs en tout point (c'est de loin l'approche la plus fréquente depuis [Wren *et al.*, 1997]), ou encore une base d'images caractéristiques qui constitue un sous-espace vectoriel dans lequel on considère que les pixels représentant l'arrière-plan vont se trouver [Oliver *et al.*, 2000].

Soustraction de l'arrière-plan (*Background subtraction*). La soustraction de l'arrière-plan est l'opération qui suit logiquement la modélisation de l'arrière-plan afin d'obtenir une détection de mouvement. Si le modèle de l'arrière-plan est une image, une différence en valeur absolue entre ce modèle et l'image courante est effectuée afin d'obtenir une détection de mouvement. Quand il s'agit d'un modèle statistique, on calcule la probabilité que chaque pixel appartienne à l'arrière-plan en testant la valeur observée dans le modèle ; l'importance du mouvement observé varie dans le sens opposé à la probabilité calculée.

Segmentation de (par le) mouvement (*Motion[-based] segmentation*). Cette tâche va au-delà de la détection de mouvement puisqu'il s'agit de segmenter chaque image en régions qui présentent une homogénéité du mouvement apparent. Cette opération est généralement réalisée à part d'une estimation du flot optique [Weiss et Adelson, 1996] ou des dérivées spatio-temporelles de l'intensité lumineuse [Odobez et Bouthemy, 1998].

Dans cet état de l'art, nous ne nous intéresserons qu'aux méthodes de détection de mouvement car l'estimation de la direction et de l'amplitude du mouvement apparent (flot optique) n'est pas forcément nécessaire aux modules suivants, et la segmentation des zones en mouvement est considérée comme faisant partie du module de suivi (*tracking*), et sera donc traitée ultérieurement.

2.1.2 Domaines couverts

La plupart des états de l'art publiés jusqu'à présent sur la détection de mouvement, sont fortement orientés vers un domaine d'application précis, ou alors inclus dans un état de l'art plus vaste traitant des systèmes complets de suivi d'objets mobiles.

Par exemple, [Moeslund *et al.*, 2006] constitue un état de l'art très complet sur l'interprétation du mouvement des êtres humains. La détection de mouvement y est évoquée comme une technique pouvant servir à segmenter chaque image entre les personnes et le terrain, au même titre que la segmentation basée sur le mouvement ou sur un modèle morphologique ou chromatique de personne.

En 2003, un état de l'art sur le même domaine d'application [Wang *et al.*, 2003] avait été présenté, avec là aussi, une part importante consacrée à la détection de mouvement et à la segmentation de régions en mouvement.

Dans [Hu *et al.*, 2004] est présenté un état de l'art sur les différents systèmes de vidéosurveillance. La détection de mouvement y est présentée comme la première étape de tout système de vidéosurveillance.

En 2005, un panorama des méthodes de modélisation et de soustraction de l'arrière-plan appliquées à des scènes de trafic routier est proposé dans [Cheung et Kamath, 2005].

Sans se limiter à un domaine d'application particulier, Piccardi [Piccardi, 2004] passe en revue lui aussi plusieurs méthodes de soustraction de l'arrière-plan et les compare en termes de vitesse, d'espace mémoire nécessaire et de précision.

Dans [Toyama *et al.*, 1999], les auteurs présentent leur propre système de vidéosurveillance comportant un module de détection de mouvement, un module d'extraction de régions d'intérêt et un module de suivi, et comparent leur algorithme de détection de mouvement à neuf autres issus de la littérature, ce qui fournit un bon aperçu des méthodes connues avant 1999. Par ailleurs, les auteurs identifient sept types de difficultés qui mettent à mal la robustesse des algorithmes les plus connus, et en déduisent cinq principes généraux que, selon eux, tout algorithme de modélisation de l'arrière-plan devrait respecter pour pouvoir gérer ces difficultés :

1. la diférenciation sémantique des objets ne doit pas être gérée par le module de détection de mouvement ;
2. la segmentation des objets doit être correcte dès leur apparition dans la scène ;
3. il est nécessaire de définir des invariants à l'échelle du pixel pour caractériser les valeurs appartenant à l'arrière-plan ;
4. le modèle de l'arrière-plan doit s'adapter aussi bien à des changements brusques qu'à des changements progressifs de l'apparence du fond ;
5. le modèle de l'arrière-plan doit prendre en considération les changements qui peuvent avoir lieu à différentes échelles d'observation (pixel, région, image).

Dans [Pless, 2005] sont présentées cinq méthodes de modélisation locale de l'arrière-plan dans le contexte de la surveillance de scènes d'extérieur dans lesquelles l'arrière-plan peut présenter des mouvements que l'on ne souhaite pas détecter. Par « modélisation locale », les auteurs entendent que chaque pixel possède son propre modèle d'arrière-plan.

[Yilmaz *et al.*, 2006] constitue à ce jour l'état de l'art le plus exhaustif sur le suivi automatique d'objets. Les auteurs proposent une description hiérarchique convaincante de cette tâche dans laquelle la soustraction de l'arrière-plan est présentée comme une technique de détection d'objets, tout comme l'extraction de points d'intérêt, l'apprentissage de l'apparence des objets à partir de plusieurs vues, ou la segmentation d'images statiques.

Dans [Radke *et al.*, 2005], on parle de « détection de changement » entre deux vues de la même scène. Cette tâche inclut la détection de mouvement,

lorsque les deux vues sont prises à des instants consécutifs, mais peut aller au-delà lorsque ce n'est pas le cas. Dans cette dernière situation, les techniques utilisées ne sont pas forcément adaptées au traitement de séquences vidéo en raison des contraintes de temps réel imposées dans ce cas. Les méthodes de détection de mouvement présentées sont toutes des méthodes de modélisation statistique de l'arrière-plan.

2.1.3 Différentes taxinomies

Les états de l'art sur la détection de mouvement publiés dans la littérature ne proposent pas tous une taxinomie des méthodes présentées. C'est particulièrement le cas pour les articles qui ont pour objet de réaliser une étude comparative de ces méthodes. Par exemple, dans [Toyama *et al.*, 1999], les auteurs ont choisi neuf algorithmes considérés comme classiques (en fait, huit sont issus de la littérature, et le neuvième est celui qu'ils proposent) qu'ils ont implémentés et testés sur un jeu de séquences vidéo illustrant les difficultés les plus fréquemment rencontrées. Il s'agit d'une « classification par l'exemple » sans pour antant que soit précisé ce qui rend chaque exemple représentatif d'une classe bien spécifique d'algorithmes.

De même, dans les articles dont l'objet ne se limite pas à l'étude des méthodes de détection de mouvement, une taxinomie n'est pas forcément proposée, comme dans [Radke *et al.*, 2005] où quelques exemples d'algorithmes de modélisation de l'arrière-plan sont considérés comme représentatifs du domaine, et présentés succinctement de manière à offrir un aperçu des différentes options possibles.

Dans [Moeslund *et al.*, 2006], plutôt que de proposer une classification en familles d'algorithmes, les auteurs préfèrent mettre en avant les caractéristiques qui différencient les principaux algorithmes. En premier lieu, la manière de représenter l'arrière-plan est considérée comme étant distinctive. Sous l'appellation « représentation de l'arrière-plan », les auteurs incluent l'espace couleur utilisé mais aussi le type de modèle d'arrière-plan. Sont notamment cités les modèles statistiques, les sous-espaces vectoriels, et le modèle à base de chaîne binaire codant le voisinage introduit dans [Heikkilä et Pietikäinen, 2006]. La seconde caractéristique notée par les auteurs est la méthode de validation des pixels étiquetés comme n'appartenant pas à l'arrière-plan. Il s'agit en fait d'un post-traitement à la détection de mouvement. Plusieurs méthodes existent, dont la morphologie mathématique, les champs de Markov, le seuillage par hystérésis ou encore l'utilisation d'un classifieur. Le troisième élément distinctif est la méthode de mise à jour du modèle de l'arrière-plan, que ce soit par addition pondérée des nouvelles observations, ou par l'ajout de nouveaux modes dans un modèle statistique multimodal. Enfin, les différentes méthodes se distinguent par la manière dont le modèle est initialisé. Les méthodes les plus minimalistes utilisent la première image observée tandis que d'autres vont calculer une médiane sur un certain nombre d'images d'apprentissage, ou encore bâtir le modèle au fur et à mesure du traitement en acceptant les erreurs

inévitables au démarrage du système.

La taxinomie la plus simple est proposée par [Piccardi, 2004]. Il distingue les méthodes dites basiques des méthodes plus élaborées. Parmi les méthodes basiques, on retrouve la différence absolue entre images consécutives, la modélisation de l'arrière-plan par la moyenne ou la médiane (exacte ou mobile) des dernières images observées. Elles peuvent être agrémentées d'un opérateur de sélection permettant de ne mettre à jour le modèle qu'avec les points dont on est sûr qu'ils n'appartiennent pas à un objet [Yang *et al.*, 2004]. Les méthodes présentées comme plus élaborées sont la modélisation de l'arrière-plan par une gaussienne simple [Wren *et al.*, 1997], par un mélange de gaussiennes [Stauffer et Grimson, 1999], ou par estimation non paramétrique des densités ponctuelles [Elgammal *et al.*, 2000]. Sont également présentées la modélisation par un algorithme *Mean-Shift* [Han *et al.*, 2004] et celle par un sous-espace vectoriel [Oliver *et al.*, 2000].

Dans [Wang *et al.*, 2003] et [Hu *et al.*, 2004], les auteurs présentent des méthodes dites de « segmentation de l'arrière-plan », parmi lesquelles ils distinguent les algorithmes de modélisation statistique de l'arrière-plan, les algorithmes basés sur la différence entre deux ou trois images consécutives, et les algorithmes d'estimation du flot optique.

Les auteurs de [Cheung et Kamath, 2005] proposent également une classification binaire, les auteurs distinguent les algorithmes récursifs des algorithmes non récursifs. Un algorithme est dit non récursif s'il stocke un buffer des dernières images observées et que la détection de mouvement est le résultat de statistiques calculées sur l'ensemble des images contenues dans ce buffer. Pour les auteurs, le principal intérêt de ces méthodes est leur grande réactivité car les observations plus anciennes que la dernière image du buffer ne viennent pas perturber l'estimation de l'arrière-plan courant. Leur défaut majeur est l'espace mémoire nécessaire au stockage du buffer. Parmi ces méthodes, on compte la différence absolue entre images consécutives (cas extrême où le buffer est réduit à une seule image), la modélisation de l'arrière-plan par la médiane des dernières images observées, le filtrage prédictif linéaire [Toyama *et al.*, 1999] et l'estimation non paramétrique des densités de probabilité d'appartenance à l'arrière-plan. Ces méthodes sont opposées aux algorithmes dits récursifs, dont le modèle de l'arrière-plan n'est pas un buffer des dernières observations. Ils ont le mérite de nécessiter moins d'espace de stockage, mais font courir le risque de voir des erreurs demeurer dans le modèle pendant plus longtemps. Les modèles statistiques paramétriques font partie de cette famille, ainsi que le lissage par filtrage de Kalman [Karmann et von Brandt, 1990].

Dans [Pless, 2005], l'auteur ne parle pas explicitement de détection de mouvement, mais propose plutôt un ensemble de familles de méthodes de modélisation du mouvement de l'arrière-plan. Il distingue tout d'abord les modèles où l'arrière-plan possède une intensité connue, en prenant l'exemple de la technique de « l'écran bleu » utilisée sur les plateaux de télévision. Vient ensuite le modèle d'arrière-plan à intensité constante mais initialement inconnue. La troisième famille de modèles autorise de légères variations de l'intensité de

l'arrière-plan, c'est le cas de la modélisation par une gaussienne unique. L'auteur considère ensuite le cas où les gradients spatio-temporels sont également modélisés par une gaussienne. La catégorie suivante consiste à remplacer les modèles gaussiens unimodaux par des mélanges de gaussiennes. Puis sont évoqués les modèles à flot optique constant. C'est le seul état de l'art qui considère cette famille de modèles qui est surtout intéressante pour les séquences acquises par une caméra mobile. La dernière famille présentée est celle des modèles prédictifs linéaires [Toyama *et al.*, 1999].

La taxinomie la plus convaincante est certainement celle présentée dans [Yilmaz *et al.*, 2006]. La distinction entre les différentes méthodes de détection de mouvement se fait sur la base de l'échelle d'observation utilisée pour bâtir les modèles. Les auteurs distinguent en premier lieu les méthodes de modélisation locale, c'est-à-dire les méthodes qui créent un modèle statistique en tout point de l'image sans faire intervenir la notion de voisinage spatial. Cette famille comprend les méthodes basées sur la différence entre images consécutives et celles qui bâtissent un modèle gaussien ou multi-gaussien en tout point de l'image. Le second type de méthodes pourrait être qualifié de semi-local. Dans ce cas, le voisinage des pixels est pris en compte durant le processus de construction du modèle local d'arrière-plan. C'est notamment le cas des méthodes utilisant un modèle de texture pour caractériser les points de l'image [Heikkilä et Pietikäinen, 2006; Nguyen *et al.*, 2007]. Dans une moindre mesure, les articles proposant une étape de post-traitement visant à régulariser le résultat de la détection en uniformisant les mesures obtenues dans une même région [Toyama *et al.*, 1999] ou à supprimer les faux positifs par analyse des composantes connexes obtenues après seuillage [Elgammal *et al.*, 2000] peuvent être classés dans cette catégorie. La troisième famille est celle des méthodes globales de modélisation (appelées méthodes « holistiques » dans [Yilmaz *et al.*, 2006]). Il existe peu d'exemples de ce type dans la littérature, à part la méthode des *eigenbackgrounds* introduite dans [Oliver *et al.*, 2000] et qui consiste à bâtir un sous-espace vectoriel d'images dans lequel l'arrière-plan est mieux représenté que les objets en mouvement. Cette méthode sera discutée plus en détails dans la section 3.2.2. Les auteurs placent dans une catégorie supplémentaire les méthodes utilisant des modèles de Markov cachés pour faire évoluer des pixels ou blocs de pixels [Rittscher *et al.*, 2000], ou encore l'ensemble de la scène [Stenger *et al.*, 2001] entre différents états. Ces méthodes auraient néanmoins pu être classées dans une des trois premières catégories suivant le caractère local, semi-local ou global des états considérés. Les méthodes utilisant des modèles linéaires autorégressifs de séries temporelles sont également considérées comme des cas particuliers car elles ont comme objectif principal de modéliser la dynamique de l'arrière-plan. Puisque dans les articles cités par les auteurs ([Monnet *et al.*, 2003; Zhong et Sclaroff, 2003] par exemple), un modèle est créé par région de l'image, ces méthodes auraient très bien pu être classées dans la catégorie des méthodes de modélisation semi-locales.

En résumé, parmi les articles qui proposent une taxinomie des méthodes de détection de mouvement, on observe une grande disparité due essentiellement à

l'objectif principal de l'article. Lorsque le but est d'effectuer une étude comparative sur un ensemble de séquences de test, le nombre de méthodes présentées est nécessairement limité et il n'est pas possible d'établir une classification hiérarchisée.

Dans les articles qui visent à documenter un domaine d'application précis, comme l'analyse de séquences de trafic routier ou l'interprétation des mouvements humains, les catégories proposées sont souvent liées aux résultats fournis par les différents algorithmes sur les séquences d'intérêt. La taxinomie la plus générale est proposée dans [Yilmaz *et al.*, 2006], et elle servira de base à celle présentée dans la section 2.2.

2.2 Les différentes méthodes de détection de mouvement

2.2.1 Taxinomie

L'ensemble des états de l'art vus précédemment nous permet d'établir une classification hiérarchisée des différentes méthodes de détection de mouvement. Nous distinguerons quatre grandes familles de méthodes en fonction de la modélisation de l'arrière-plan.

Détection sans modélisation de l'arrière-plan. Ces méthodes consistent à détecter le mouvement par le calcul en tout point de l'image d'une quantité mathématique qui est fonction de l'intensité ou de la couleur de l'ensemble des pixels et qui est censée refléter l'importance du mouvement visible dans la scène. La dérivée temporelle de l'intensité lumineuse, l'entropie spatio-temporelle de l'image et la norme du flot optique en sont des exemples que nous présenterons dans la suite de ce document.

Modélisation locale de l'arrière-plan. Ces méthodes consistent à associer à tout point de l'image une valeur ou une fonction permettant de modéliser l'apparence de l'arrière-plan en ce point. Le modèle d'apparence de l'arrière-plan en un point ne dépend que des observations qui ont eu lieu en ce point. Les autres pixels de l'image n'interviennent pas. La grande majorité des méthodes présentées dans la littérature bâtissent un modèle statistique (ensemble de paramètres d'une loi, ou ensemble d'échantillons), mais il peut s'agir d'un processus stochastique, d'un filtre prédictif ou simplement d'une valeur d'intensité.

Modélisation semi-locale de l'arrière-plan. Ces méthodes sont très semblables à celles de la catégorie précédente, à la différence près que la modélisation de l'arrière-plan en un point dépend des observations qui ont eu lieu dans un certain voisinage de ce point, ou dans la région de l'image à laquelle il appartient.

Modélisation globale de l'arrière-plan. Ces méthodes utilisent à chaque instant l'ensemble des observations pour construire un modèle de l'ensemble de l'arrière-plan.

Cette taxinomie est résumée dans la figure 2.1.

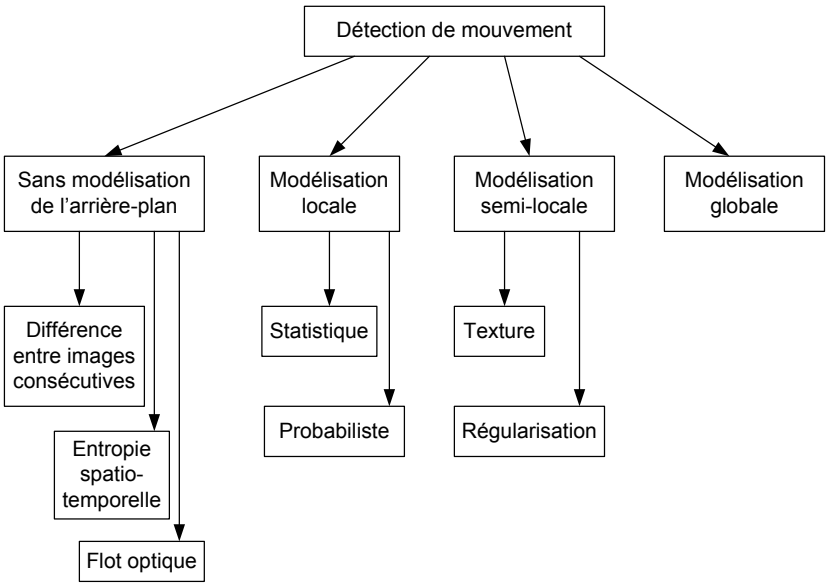


FIG. 2.1 – Taxinomie proposée pour les méthodes de détection de mouvement.

2.2.2 Détection sans modélisation

Dans l'ensemble de cette section et dans le reste du document, nous désignerons par $I(t)$ l'image présente dans le flux d'entrée au temps t et par $I(x, y, t)$ le pixel de coordonnées $\mathbf{x} = (x, y)$ dans cette même image. Nous appellerons $\mathbb{E}$ le domaine de définition des coordonnées des pixels. Les dérivées spatiales horizontale et verticale de l'image seront notées respectivement $I_x(x, y, t)$ et $I_y(x, y, t)$. La dérivée temporelle sera notée $I_t(x, y, t)$. Le gradient de l'image au point (x, y) et au temps t sera quant à lui abusivement noté $\nabla I(x, y, t) = [I_x(x, y, t)\ I_y(x, y, t)]^T$.

2.2.2.1 Dérivée temporelle

La manière la plus intuitive de détecter les zones en mouvement dans un champ de vision est de mesurer le changement d'apparence des pixels entre deux trames consécutives, soit la dérivée temporelle en tout point. La première utilisation de cette méthode dans l'analyse de séquence vidéo est généralement attribuée à Jain et Nagel [Jain et Nagel, 1979]. L'intensité lumineuse d'un pixel

étant un signal discret à une dimension, l'estimation de la dérivée temporelle instantanée du signal au temps t est donnée par

$$\forall (x,y) \in \mathbb{E} \quad \forall t > 0 \quad I_t(x,y,t) \approx |I(x,y,t) - I(x,y,t-1)| \,. \tag{2.1}$$

Comme l'ont observé de nombreux auteurs [Tian et Hampapur, 2005], cette méthode se montre peu robuste face à des phénomènes tels que les mouvements lents ou saccadés, les arrêts brefs d'un objet en mouvement, ou encore la présence de trames redondantes dans certaines séquences vidéo. Il convient donc d'effectuer un lissage temporel de la séquence, c'est-à-dire d'appliquer un opérateur de moyenne mobile à la mesure obtenue. Ceci peut être fait à l'aide d'une matrice de même taille que les trames d'entrée, appelée accumulateur, et que nous noterons A.

$$\forall (x,y) \in \mathbb{E} \begin{cases} A(x,y,0) = 0 \\ \forall t > 0 \quad A(x,y,t) = w_A A(x,y,t-1) + (1-w_A) I_t(x,y,t) \end{cases}, \tag{2.2}$$

avec $0 \leq w_A \leq 1$. Dans l'équation 2.2, le terme w_A pondère la contribution des mesures passées par rapport à la dernière mesure en date. Avec un w_A faible, les effets du lissage temporel sont peu visibles, et les problèmes qui avaient motivé l'utilisation de cette méthode risquent d'apparaître tout de même. Avec un w_A élevé, le lissage est important et un effet de persistance des entités détectées peut se produire (effet « fantôme »).

2.2.2.2 Entropie spatio-temporelle

L'entropie est une mesure issue de la thermodynamique, associée au degré de désordre d'un système. Dans notre cas, il s'agira de mesurer en chaque point, la « variabilité » de la grandeur mesurée. Concrètement, plus l'intensité lumineuse (ou la couleur, ou le gradient, etc.) aura pris de valeurs dissemblables en un point pendant un certain intervalle de temps, plus l'entropie sera élevée en ce point. Dans [Ma et Zhang, 2001], les auteurs proposent, pour calculer l'entropie, une méthode à base d'histogrammes spatio-temporels.

Il s'agit, en chaque pixel désigné par (x,y,t), de créer un histogramme à partir des points (u,v,τ) d'un voisinage spatio-temporel de diamètre W dans le domaine spatial, et de longueur L dans le domaine temporel. Ainsi, pour une image à Q niveaux de gris, on note $H_{x,y,t}(q)$ la fréquence de la q-ème classe de l'histogramme $(0 \leq q \leq Q-1)$ associée au point (x,y,t). Si on normalise cet histogramme, on obtient une fonction de densité de probabilité.

$$\forall (x,y) \in \mathbb{E} \quad \forall t \geq L \quad P_{x,y,t}(q) = \frac{H_{x,y,t}(q)}{W^2 L} \text{ avec } \sum_{q=0}^{Q-1} P_{x,y,t}(q) = 1 \tag{2.3}$$

Cette expression nous permet de déterminer en tout point l'entropie associée à la répartition des niveaux de gris au voisinage dudit point.

$$E(x,y,t) = -\sum_{q=0}^{Q-1} P_{x,y,t}(q) \log\left(P_{x,y,t}(q)\right) \tag{2.4}$$

L'entropie ainsi calculée peut être quantifiée en 256 niveaux, et être représentée sous la forme d'une image appelée STEI (*Spatial Temporal Entropy Image*) par les auteurs. Dans cette image, sont mis en évidence les points où les variations d'intensité lumineuse sont importantes au cours du temps, et dans une moindre mesure, au sein d'un voisinage spatial. Les zones de mouvement sont donc bien mises en évidence, mais les contours (zones de fort gradient) le sont également. On peut atténuer cet effet en donnant plus d'importance aux variations observées dans le domaine temporel que dans le domaine spatial, mais ce phénomène persistera toujours.

Tandis que Ma et Zhang proposent de débruiter la STEI à l'aide de filtres morphologiques afin d'éliminer les faux positifs dus aux contours, les auteurs de [Guo *et al.*, 2004] suggèrent d'utiliser comme donnée d'entrée, non plus les trames de la séquences, mais la différence entre trames consécutives telle que calculée dans la section 2.2.2.1. L'image ainsi obtenue est appelée DSTEI (*Difference-based Spatial Temporal Entropy Image*).

2.2.2.3 Flot optique

Alors que la dérivée temporelle quantifie la variation de l'aspect de chaque pixel considéré individuellement, le flot optique est un champ de vecteurs à deux dimensions représentant la projection sur le plan image du mouvement réel observé (tridimensionnel). De nombreuses méthodes ont été proposées depuis l'article précurseur de Horn et Schunck [Horn et Schunck, 1981], celles-ci sont détaillées dans plusieurs états de l'art. Dans [Barron *et al.*, 1994], neuf algorithmes sont étudiés et comparés selon des critères de précision et de la densité du champ obtenu, mais aucune mention n'est faite de la complexité algorithmique. Les travaux de [Liu *et al.*, 1998] permettent de combler cette lacune en mesurant les rapports précision/temps de calcul de ces méthodes. Quelle que soit la méthode choisie, le calcul du flot optique reste une opération très coûteuse en temps de calcul. Le temps-réel peut néanmoins être atteint en sous-échantillonnant les trames et en choisissant un algorithme rapide.

L'un des algorithmes les plus rapides et les plus populaires est celui de [Lucas et Kanade, 1981]. Il s'agit de calculer au temps t, le déplacement $\mathbf{d} = (d_x, d_y)^T$ du point $\mathbf{x} = (x, y)^T$. L'hypothèse sur laquelle se base la méthode est celle de la conservation de l'intensité lumineuse (2.5).

$$\forall \mathbf{x} \in \mathbb{E} \quad \forall t > 0 \quad I(\mathbf{x} + \mathbf{d}, t + 1) - I(\mathbf{x}, t) = 0. \tag{2.5}$$

Ainsi, l'estimation du flot optique au point $\mathbf{x}$ est le vecteur $\mathbf{d}^*$ qui minimise la fonction d'erreur quadratique ξ calculée sur un voisinage $\mathcal{N}(\mathbf{x})$ et définie par

$$\xi(\mathbf{d}) = \sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \left[I(\mathbf{x}' + \mathbf{d}, t + 1) - I(\mathbf{x}', t) \right]^2 . \tag{2.6}$$

La minimisation peut être obtenue de manière récursive (2.7).

$$\begin{cases} \mathbf{d}_0 = \mathbf{0} \\ \mathbf{d}_{n+1} = \mathbf{d}_n + \left(\sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \nabla I(\mathbf{x}' + \mathbf{d}_n, t+1)^T I_t(\mathbf{x}', t+1) \right) \cdot \\ \qquad \left(\sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x})} \nabla I(\mathbf{x}' + \mathbf{d}_n, t+1) \nabla I(\mathbf{x}' + \mathbf{d}_n, t+1)^T \right)^{-1} . \end{cases} \tag{2.7}$$

2.2.3 Détection par modélisation locale

2.2.3.1 Modélisation par une image

Le modèle d'arrière-plan le plus simple serait une image représentant la scène dépourvue d'objets. Cette méthode présente l'avantage de détecter aussi bien les mouvements lents que les mouvements rapides. Par ailleurs, même les objets momentanément immobiles sont détectés. Cependant, en environnement extérieur, les variations d'intensité lumineuse rendent rapidement obsolète un tel modèle, et il est nécessaire de mettre à jour cette image de l'arrière-plan. Par ailleurs, il n'est pas toujours possible d'obtenir d'une image de la scène totalement vide. Dans ces conditions, il est nécessaire de mettre à jour l'image de l'arrière-plan.

Par exemple, dans [Yang *et al.*, 2004], les auteurs proposent d'utiliser la différence entre images consécutives pour y parvenir. Ils considèrent la première image de la séquence comme une première approximation du modèle de l'arrière-plan. Ensuite, à chaque nouvelle trame, la différence par rapport à l'image précédente est calculée, et les pixels où aucun mouvement n'est détecté sont utilisés pour mettre à jour le modèle du fond. Pour plus de robustesse, les auteurs préconisent de ne considérer que les points auxquels la dérivée temporelle a été négligeable pendant un certain intervalle de temps.

Dans [Cutler et Davis, 1998], le modèle de l'arrière-plan est constitué en chaque point de la valeur médiane des niveaux de gris observés durant les N dernières trames. Dans le cas d'images en couleur, ils utilisent la couleur (artificielle) constituée de la médiane de chacun des canaux R, G, B. L'utilisation de l'information couleur a été améliorée dans [Cucchiara *et al.*, 2003] avec l'utilisation du médoïde à la place de la médiane.

2.2.3.2 Modélisation statistique

Le problème de la modélisation de l'arrière-plan peut être exprimé d'un point de vue statistique. Il s'agit, pour chaque pixel, d'estimer la probabilité d'y observer telle ou telle couleur (ou niveau de gris) en se basant sur un modèle appris, censé représenter l'arrière-plan de la scène. Le modèle consiste en un ensemble de fonctions de densité de probabilité : une par pixel de l'image. Les mesures dont la probabilité d'être observées est élevée correspondent à

des pixels qui seront étiquetés comme arrière-plan, tandis que celles dont la probabilité d'être observées est faible correspondent à des pixels qui seront étiquetés comme avant-plan.

Dans [Wren *et al.*, 1997], les auteurs proposent de modéliser l'intensité des points de l'arrière-plan par une distribution gaussienne. En tout point, la moyenne et l'écart-type sont mis à jour récursivement, et chaque nouvelle observation est déclarée comme appartenant à l'arrière-plan si elle se situe suffisamment près de la moyenne courante, compte tenu de l'estimation courante de l'écart-type. Le même procédé peut être utilisé sur des images en couleur [McKenna *et al.*, 2000]. Ce modèle permet d'obtenir de bons résultats pour des scènes d'intérieur où l'arrière-plan est parfaitement statique, mais en environnement extérieur, des phénomènes périodiques tels que l'ondulation d'une surface d'eau ou le balancement d'une branche d'arbre peuvent le rendre totalement inopérant car la distribution de l'apparence de l'arrière-plan est alors multimodale.

Dans [Stauffer et Grimson, 1999], chaque pixel de l'arrière-plan est modélisé par un mélange de k gaussiennes. La probabilité d'observer en $\mathbf{x}$ la valeur $\mathbf{z}$ au temps t est donc

$$P_{\mathbf{x}}(\mathbf{z}_t) = \sum_{i=1}^{k} \omega_{i,t}\, \eta(\mathbf{z}_t; \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t}), \tag{2.8}$$

où $\omega_{i,t}$ est le poids accordé à la i-ème gaussienne au temps t, $\eta(\cdot)$ est la fonction gaussienne de densité de probabilité, $\boldsymbol{\mu}_{i,t}$ et $\boldsymbol{\Sigma}_{i,t}$ sont respectivement la moyenne et la matrice de variance-covariance de la i-ème gaussienne au temps t. Dans [Stauffer et Grimson, 1999], k est compris entre 3 et 5, et les composantes couleur sont considérées comme indépendantes et de même variance, soit $\boldsymbol{\Sigma}_{i,t} = \sigma_{i,t}^2 \mathbf{I}$.

Pour chaque nouvelle observation $\mathbf{z}_t$, la gaussienne qui est susceptible d'expliquer $\mathbf{z}_t$ avec la plus forte probabilité est mise à jour. On considère que la gaussienne i explique $\mathbf{z}_t$ si $\|\mathbf{z}_t - \boldsymbol{\mu}_{i,t}\| \leq 2{,}5\sigma_{i,t}$. La mise à jour de la gaussienne i se déroule de la manière suivante :

$$\omega_{i,t} = (1-\alpha)\omega_{i,t-1} + \alpha; \tag{2.9}$$

$$\boldsymbol{\mu}_{i,t} = (1-\rho)\boldsymbol{\mu}_{i,t-1} + \rho\, \mathbf{z}_t; \tag{2.10}$$

$$\sigma_{i,t}^2 = (1-\rho)\sigma_{i,t-1}^2 + \rho(\mathbf{z}_t - \boldsymbol{\mu}_{i,t})^T(\mathbf{z}_t - \boldsymbol{\mu}_{i,t}), \tag{2.11}$$

avec $\rho = \alpha\, \eta(\mathbf{z}_t; \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t})$, la constante de temps $1/\alpha$ déterminant la vitesse à laquelle les paramètres sont mis à jour.

Pour décider parmi les gaussiennes appartenant au mélange lesquelles modélisent l'arrière-plan et lesquelles modélisent les objets en mouvement, les distributions sont ordonnées selon leur rapport poids/écart-type, et les b premières distributions telles que $\sum_{i=1}^{b} \omega_{i,t} > \theta$ sont attribuées à l'arrière-plan, θ étant un seuil fixé empiriquement.

Dans l'approche de [Stauffer et Grimson, 1999], le choix du nombre de modes de la distribution modélisée est problèmatique. Pour cette raison, dans

[Elgammal *et al.*, 2000], les auteurs proposent de calculer les densités de probabilité de manière non paramétrique à partir des dernières valeurs observées en tout point. En notant $\{\mathbf{z}_i\}_{i=1}^{N}$ les N dernières observations en un point donné, la probabilité d'observer la valeur $\mathbf{z}_t$ est donnée par une somme de noyaux gaussiens centrés sur chacun des échantillons (2.12).

$$\Pr\{I(x,y,t) = \mathbf{z}_t\} = \frac{1}{N}\sum_{i=1}^{N} \eta(\mathbf{z}_t; \mathbf{z}_i, \mathbf{\Sigma}_t) \tag{2.12}$$

Ici, le paramètre le plus délicat à déterminer est $\mathbf{\Sigma}_t$. Dans [Elgammal *et al.*, 2000], les auteurs considèrent que les composantes couleur sont indépendantes ($\mathbf{\Sigma}_t$ est diagonale), et estiment la variance en se basant sur les dernières différences entre valeurs consécutives observées au point considéré.

Plus récemment, l'algorithme *Mean-Shift*, bien connu en reconnaissance des formes pour ses capacités d'analyse de données multimodales, a été pour la première fois utilisé pour la modélisation de l'arrière-plan dans [Han *et al.*, 2004]. Cet algorithme permet de détecter les modes d'une distribution complexe en se basant uniquement sur un ensemble d'échantillons et sans que le nombre de modes n'ait besoin d'être connu. Malheureusement, cette technique étant très coûteuse en temps de calcul et en espace mémoire, elle ne peut être utilisée que pour initialiser la modélisation, la mise à jour se faisant par une méthode heuristique de propagation/fusion/éclatement des modes mis en évidence par l'initialisation.

2.2.3.3 Modélisation predictive

Une autre approche, assez semblable à l'approche statistique, consiste à utiliser un filtre de Wiener ou de Kalman pour prédire la prochaine valeur que l'on devrait observer en chaque point. C'est l'écart entre la prédiction et l'observation qui sera utilisé pour estimer l'amplitude du mouvement. Ces méthodes permettent de gérer les problèmes d'arrière-plans non statiques, et les perturbations intervenant à intervalles réguliers (par exemple, la bande horizontale qui apparaît lorque l'on filme un écran à tube cathodique).

Dans [Toyama *et al.*, 1999], les auteurs proposent une méthode à trois niveaux sémantiques (local, semi-local, global). La segmentation au niveau local est effectuée à l'aide d'un filtrage prédictif de Wiener. Le filtre de Wiener permet de construire une valeur estimée $\hat{\mathbf{z}}_t$ de la valeur que l'on devrait observer à l'instant t, à partir d'un échantillon de N mesures bruitées $\{\mathbf{z}_i\}_{i=1}^{N}$.

$$\hat{\mathbf{z}}_t = \sum_{i=1}^{N} \omega_i \, \mathbf{z}_{t-i}, \tag{2.13}$$

où les ω_i sont des coefficients pondérateurs calculés de manière à minimiser l'erreur quadratique moyenne d'estimation $\xi(t)$.

$$\xi(t) = E\Big[\, |\mathbf{z}_t - \hat{\mathbf{z}}_t|^2 \,\Big]. \tag{2.14}$$

Les coefficients ω_i sont calculés par rapport à la covariance des valeurs de l'échantillon. Le détail des calculs peut être trouvé dans [Hayes, 1996]. Les auteurs utilisent un échantillon de 50 valeurs pour calculer 30 coefficients de prédiction. La masse de calculs à réaliser est donc difficilement compatible avec des contraintes de temps réel.

Plus fréquemment, le filtrage prédictif est réalisé à l'aide d'un filtre de Kalman. Un didacticiel sur l'utilisation des filtres de Kalman peut être trouvé dans [Welch et Bishop, 1995]. La méthode suppose que la meilleure information que l'on puisse avoir sur l'état d'un système est obtenue par le calcul d'une estimation qui fait explicitement mention du bruit enregistré lors de l'observation.

De nombreuses variantes ont été proposées pour la modélisation du fond ; elles diffèrent essentiellement par le vecteur d'état utilisé pour décrire le système. Dans [Wren *et al.*, 1997], le filtre de Kalman est utilisé pour estimer les paramètres de la gaussienne qui modélise l'intensité du fond (cf. section 2.2.3.2). La version la plus simple est décrite dans [Boult *et al.*, 1999] et utilise simplement l'intensité lumineuse comme vecteur d'état. Le modèle proposé dans [Karmann et von Brandt, 1990] et enrichi dans [Ridder *et al.*, 1995] utilise l'intensité lumineuse et sa dérivée temporelle pour former le vecteur d'état, tandis que les dérivées spatiales sont utilisées à la place de la dérivée temporelle dans [Koller *et al.*, 1993]. Le schéma le plus populaire reste celui de [Karmann et von Brandt, 1990].

2.2.4 Détection par modélisation semi-locale

2.2.4.1 Détection par région

Etant donnée la nature progressive des mouvements généralement observés dans les séquences à analyser, et compte tenu des imprécisions dues aux caméras vidéo utilisées, certains auteurs préconisent de prendre en compte l'ensemble des pixels d'un voisinage au lieu de chercher à détecter les mouvements en un point donné sans se préoccuper des pixels alentour.

Par exemple, dans [Rittscher *et al.*, 2000], les auteurs considèrent l'ensemble des zones carrées 3×3 non recouvrantes de l'image et bâtissent un modèle de Markov caché (MMC) [Rabiner, 1989] pour décider si cette zone appartient à l'arrière-plan, à un objet mobile, ou à une ombre. Le MMC possède donc 3 états : arrière-plan, objet mobile, ombre. Le nombre de symboles observables est 256, puisqu'on observe une intensité lumineuse. Afin de prendre en compte l'intensité moyenne de la zone ainsi que son homogénéité, deux observations sont faites à chaque instant. Ces observations sont obtenues par application d'un filtre de convolution moyenneur 3×3 au centre de la zone et par le calcul de la norme du gradient de Sobel. Les auteurs déclarent que ces deux mesures sont statistiquement indépendantes. Les probabilités initiales et de transition sont initialisées en annotant manuellement une séquence de test et en mesurant en tout point la durée moyenne du passage par chacun des états et la proportion de temps passé dans chaque état. Les probabilités d'émission sont

différentes pour chaque état. Quand le système est dans l'état « objet mobile », la distribution des intensités est considérée comme uniforme car on ne peut pas préjuger de l'apparence des objets en mouvement. Dans les états « arrière-plan » et « ombre », les distributions utilisées sont des gaussiennes dont les paramètres sont appris à la manière de [Wren *et al.*, 1997]. Les paramètres du modèle sont ensuite optimisés de manière incrémentale par l'algorithme de Baum-Welch qui est une version généralisée de l'algorithme EM (*Expectation Maximization*). La classification des régions entre arrière-plan, objet mobile ou ombre se fait en sélectionnant la probabilité d'émission la plus élevée.

2.2.4.2 Caractérisation par la texture

La prise en compte du voisinage des points peut également être réalisée en calculant en tout point un vecteur caractérisant la texture à cet endroit, et en utilisant l'ensemble des vecteurs calculés comme espace de représentation des données.

Les auteurs de [Heikkilä et Pietikäinen, 2006] utilisent le codage LBP (*Local Binary Patterns*), introduit dans leurs précédents travaux, pour caractériser la texture des pixels de l'image. Le code LBP du point (x, y) est le mot binaire obtenu en concaténant l'intensité seuillée de tous les pixels situés dans un voisinage de (x, y). Le voisinage est constitué de P points $\{(u_i, v_i)\}_{i=1}^{P}$ uniformément répartis sur un cercle de rayon R centré en (x, y). La valeur du seuil utilisé est l'intensité du point (x, y).

$$\mathrm{LBP}_{P,R}(x,y) = \sum_{i=1}^{P} s\left(I(u_i,v_i) - I(x,y)\right) 2^{i-1}, \text{ avec } s(g) = \begin{cases} 1 & \text{si } g \geq 0, \\ 0 & \text{sinon.} \end{cases} \tag{2.15}$$

D'une manière assez similaire à [Stauffer et Grimson, 1999], les auteurs conçoivent le modèle d'arrière-plan en un lieu donné comme un ensemble de modes dotés d'un poids. Ici, les modes ne sont pas des distributions gaussiennes, mais des histogrammes de codes LBP calculés sur des blocs carrés partiellement recouvrants. Le nombre K d'histogrammes utilisés pour modéliser une région de l'arrière-plan est choisi empiriquement. Pour un bloc donné, à chaque nouvelle image, un nouvel histogramme $\mathbf{h}$ est calculé et comparé à chacun des K histogrammes $\{\mathbf{m}^i\}_{i=1}^{K}$ qui constituent le modèle. La mesure de similarité utilisée est l'intersection d'histogrammes (2.16).

$$\cap(\mathbf{h}, \mathbf{m}^i) = \sum_{j=1}^{N} \min(h_j, m_j^i), \tag{2.16}$$

où N est le nombre de classes des histogrammes. Si aucun des histogrammes du modèle n'est suffisamment proche de $\mathbf{h}$, l'histogramme du modèle qui possède le poids le plus faible est remplacé par $\mathbf{h}$.

Si, en revanche, le modèle contient un histogramme $\mathbf{m}$ suffisamment similaire à $\mathbf{h}$, $\mathbf{m}$ est mis à jour (2.17) ainsi que le poids de l'ensemble des histo-

grammes du modèle (2.18).

$$\mathbf{m} = \alpha\mathbf{h} + (1-\alpha)\mathbf{m} \tag{2.17}$$

$$\forall i \in [\![1, K]\!] \quad \omega_i = \beta M_i + (1-\beta)\omega_i, \tag{2.18}$$

où ω_i est le poids de l'histogramme $\mathbf{m}^i$, M_i est un indicateur valant 1 si $\mathbf{m}^i$ est l'histogamme du modèle le plus similaire à $\mathbf{h}$ et 0 sinon, et α et β sont des coefficients choisis dans l'intervalle $[0, 1]$ de manière à régler la vitesse de mise à jour.

Comme dans [Stauffer et Grimson, 1999], la segmentation entre avant-plan et arrière-plan est réalisée en classant les histogrammes du modèle par poids décroissants et en attribuant à l'arrière-plan les b premiers histogrammes dont la somme des poids dépasse un certain seuil.

2.2.4.3 Régularisation *a posteriori*

Plusieurs auteurs [Toyama *et al.*, 1999; Elgammal *et al.*, 2000; Tian et Hampapur, 2005] proposent une application dite « multi-couches », c'est-à-dire, à plusieurs niveaux sémantiques. Le niveau le plus bas bâtit un modèle local de l'arrière-plan (section 2.2.3) qui permet de réaliser une première estimation de la segmentation des objets mobiles. Ensuite, un second procédé analyse les résultats obtenus et les régularise de manière à augmenter la consistance des régions détectées dans un voisinage spatial et/ou temporel.

Après avoir segmenté l'arrière-plan à l'aide d'une modélisation locale, les auteurs de [Toyama *et al.*, 1999] considèrent que parmi les points détectés, ceux où la dérivée temporelle est importante appartiennent nécessairement à des objets mobiles. Dans ce cas, un histogramme normalisé des régions trouvées est construit, et les régions sont utilisées comme germes par un algorithme de segmentation par croissance de région utilisant l'histogramme comme critère d'arrêt.

Dans [Elgammal *et al.*, 2000] est également utilisée la notion de voisinage spatial pour supprimer les fausses détections du résultat fourni par la modélisation locale. Les auteurs attribuent les fausses détections à de légers mouvements de l'arrière-plan ou de la caméra qui n'auraient pas été modélisés pendant la phase d'apprentissage. Afin de les supprimer, ils proposent de calculer en tout point où un mouvement a été détecté, la probabilité que la valeur observée appartienne à la distribution d'arrière-plan d'un des points avoisinants. Cette probabilité sera élevée si le mouvement est dû à un léger déplacement de l'arrière-plan. Malheureusement, elle sera également élevée pour de vraies détections dont l'apparence est similaire à l'arrière-plan de pixels voisins. Avant d'éviter ce phénomène, les auteurs ne suppriment ces détections que si l'ensemble de la composante connexe qui les contient a subi un tel mouvement. Il s'agit donc d'une régularisation qui utilise deux échelles de voisinage.

Dans [Tian et Hampapur, 2005], la détection locale du mouvement est une simple dérivée temporelle lissée. Le coefficient de lissage est choisi de manière à avoir plus de faux positifs que de faux négatifs. Les auteurs estiment ensuite

le flot optique sur l'ensemble de l'image, et les points où le flot optique a trop fluctué dans un passé proche (c'est-à-dire, dans un voisinage temporel) sont supprimés du résultat.

2.2.5 Détection par modélisation globale

2.2.5.1 Basculement entre plusieurs modèles

Pour prendre en compte la totalité de la scène dans le processus de segmentation entre avant-plan et arrière-plan, les auteurs de [Toyama *et al.*, 1999] décident de garder en mémoire k modèles de l'arrière-plan. La détection de mouvement au niveau local est effectuée avec chacun des modèles, et celui qui détecte le moins de pixels en mouvement est retenu pour la décision finale. Initialement, les k modèles sont acquis en exécutant un algorithme *k-means* sur les images d'une séquence d'apprentissage. En cours de traitement, les modèles peuvent être mis à jour si une large majorité des pixels de l'image sont détectés comme étant en mouvement.

Dans [Stenger *et al.*, 2001] est présenté un modèle de Markov caché qui modifie automatiquement sa topologie quand il est confronté à un environnement dynamique. Les auteurs utilisent cet outil pour gérer les difficultés liées aux changements soudains d'éclairage. Typiquement, le MMC aura deux états (jour/nuit, par exemple) et à chaque état sera associé un modèle statistique local de l'arrière-plan. La nature précise du modèle statistique n'est cependant pas précisée.

2.2.5.2 Espaces vectoriels

Une autre manière de prendre en compte la globalité de l'espace image pour détecter les mouvements est de considérer les pixels comme des dimensions d'un espace de représentation, et les images successives comme des individus dans cet espace. Les méthodes d'analyse de données permettent alors de considérer tous les pixels de l'image dans une approche globale pour définir de nouvelles caractériqtiques que l'on pourra appliquer en tout point pour y détecter d'éventuels mouvements. La méthode des *eigenbackgrounds* (« arrière-plans propres ») introduite dans [Oliver *et al.*, 2000] constitue la première application de l'analyse de données à la détection de mouvement. Ce domaine, bien qu'en pleine expansion n'a pas à ce jour fait l'objet d'un état de l'art spécifique, et c'est ce que nous nous proposons de réaliser dans le chapitre suivant.

Chapitre 3

Analyse de données pour la détection de mouvement

Sommaire

Avant d'aborder la manière dont l'ACP a été appliquée à la détection de mouvement dans la littérature, il est nécessaire de présenter préalablement la théorie sous-jacente à cette méthode.

3.1 Théorie

Dans cette section, nous exposerons succinctement les bases théoriques de l'analyse en composantes principales. Nous considérerons que nous voulons traiter un ensemble de n observations, qui sont des réalisations d'une variable aléatoire à p dimensions $\mathbf{x} = (x_1 \ \cdots \ x_p)^T$. Ces observations seront disposées dans un tableau de données $\mathbf{X}$ à n lignes et p colonnes dont chaque ligne est constituée par une observation.

3.1.1 Algorithme de base

L'ACP a été développée au début du XIX$^{\text{e}}$ siècle pour analyser des données issues des sciences humaines. C'est une technique statistique qui vise à simplifier un ensemble de données en l'exprimant dans un nouveau système de

coordonnées de manière à ce que les plus grandes variances soient observées sur les premières coordonnées. Cela permet de réduire la dimensionnalité de l'espace de recherche en ne conservant que les premières dimensions de l'espace de projection obtenu. C'est la meilleure technique de réduction de dimension au sens des moindres carrés [Jolliffe, 2002].

Comme la variance d'une donnée dépend de son échelle, il est de coutume de centrer et de réduire les données avant de les traiter de manière à ce que toutes les variables soient exprimées dans des unités comparables. Nous considérerons donc que $\mathbf{X}$ contient des données ainsi normalisées. Si l'on appelle $\mathbf{C}$ la matrice de variance-covariance associée à $\mathbf{X}$,

$$\mathbf{C}_{p\times p} = \frac{1}{n}\mathbf{X}^T\,\mathbf{X}, \tag{3.1}$$

alors les directions des axes principaux sont données par les vecteurs propres de $\mathbf{C}$. L'axe sur lequel on observe la plus grande variance est défini par tout vecteur propre associé à la plus grande valeur propre en valeur absolue, l'axe sur lequel on observe la seconde plus grande variance est celui caractérisé par tout vecteur propre associé à la seconde plus grande valeur propre, et ainsi de suite. Comme seul l'ordre des valeurs propres nous intéresse, le facteur $\frac{1}{n}$ n'apporte aucune information. On utilisera donc par la suite la matrice de dispersion $\mathbf{S} = \mathbf{X}^T\,\mathbf{X}$ en lieu et place de $\mathbf{C}$.

Lorsque le nombre de variables p est grand ($p \gg n$), l'espace mémoire nécessaire pour stocker la matrice de dispersion $\mathbf{S}$ peut être impossible à obtenir. La décomposition en valeurs singulières permet de contourner ce problème. La décomposition en valeurs singulières de $\mathbf{X}$ revient à trouver trois matrices $\mathbf{U}$, $\mathbf{\Sigma}$ et $\mathbf{V}$ telles que

$$\mathbf{X} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T, \tag{3.2}$$

où $\mathbf{U}$ est une matrice orthogonale de dimension $(n \times n)$, $\mathbf{\Sigma}$ est une matrice symétrique de dimension $(n \times p)$ dont les éléments $\{\sigma_{ii}\}_{i\leq\min(n,p)}$ sont appelés les « valeurs singulières » de $\mathbf{X}$, et $\mathbf{V}$ est une matrice orthogonale de dimension $(p \times p)$. Les colonnes de $\mathbf{U}$ et $\mathbf{V}$ sont appelées respectivement les « vecteurs singuliers à gauche » et les « vecteurs singuliers à droite » de $\mathbf{X}$. En remplaçant $\mathbf{X}$ dans (3.1) par sa décomposition en valeurs singulières, et en utilisant le fait que $\mathbf{U}$ est orthogonale et $\mathbf{\Sigma}$ symétrique, on obtient

$$\begin{aligned}
\mathbf{S} &= (\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T)^T(\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T) && (3.3)\\
&= \mathbf{V}\,\mathbf{\Sigma}^T\,\mathbf{U}^T\mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T && (3.4)\\
&= \mathbf{V}\,\mathbf{\Sigma}^T\mathbf{\Sigma}\,\mathbf{V}^T && (3.5)\\
&= \mathbf{V}\,\mathbf{\Sigma}^2\mathbf{V}^T. && (3.6)
\end{aligned}$$

Autrement dit, les valeurs singulières de $\mathbf{X}$ et les vecteurs singuliers à droite associés sont égaux respectivement à la racine carrée des valeurs propres de $\mathbf{S}$ et aux vecteurs propres associés.

Généralement, quand p est grand, les variables mesurées sont très corrélées linéairement, si bien que le rang r de la matrice $\mathbf{X}$ est inférieur à p. Dans ce

cas, il est possible de calculer uniquement les r colonnes de $\mathbf{V}$ correspondant aux valeurs singulières significatives, ce qui permet de résoudre le problème de complexité spatiale évoqué précédemment.

3.1.2 ACP incrémentale

Que l'on utilise la matrice de variance-covariance ou la décomposition en valeurs singulières, l'algorithme original de l'ACP suppose que la totalité des données soient disponibles avant le traitement. Si l'on souhaite intégrer les observations une à une dans le calcul, soit parce qu'elles sont trop nombreuses, soit parce qu'elles ne sont pas toutes disponibles en même temps, il faut utiliser une version incrémentale de l'ACP. Parmi les méthodes proposées dans la littérature, celle de [Hall *et al.*, 1998] est l'une des plus fréquemment utilisées.

On peut partir de l'espace de représentation fourni par une ACP classique sur un petit nombre de données. Cet espace est entièrement décrit par $\Omega = (\boldsymbol{\mu}, \mathbf{V}, \boldsymbol{\Lambda})$, où $\boldsymbol{\mu}$ est la moyenne des observations, $\mathbf{V}$ est une matrice dont l'ensemble des colonnes $\{\mathbf{v}_j\}_{j=1}^{r}$ consitue la base $\mathcal{B}$ des axes principaux révélés par l'ACP, et $\boldsymbol{\Lambda}$ est une matrice diagonale composée des valeurs propres associées. On suppose que le tableau de données est de rang r, donc $\boldsymbol{\mu}$ est de taille $(1 \times p)$, $\mathbf{V}$ de taille $(p \times r)$ et $\boldsymbol{\Lambda}$ de taille $(r \times r)$.

Pour intégrer une nouvelle observation $\mathbf{x}_{n+1}$, on commence par mettre à jour la moyenne.

$$\boldsymbol{\mu}' = \frac{1}{n+1}(n\,\boldsymbol{\mu} + \mathbf{x}_{n+1}), \tag{3.7}$$

puis on projette cette nouvelle observation dans l'espace de représentation courant. Soit $\mathbf{y} = (\mathbf{x}_{n+1} - \boldsymbol{\mu})\mathbf{V}$ la projection obtenue. En rétro-projetant $\mathbf{y}$ dans l'espace d'observation des données, on peut déduire un vecteur résiduel $\mathbf{h}$ qui représente la partie de $\mathbf{x}$ qui est orthogonale à l'espace de représentation courant.

$$\mathbf{h} = (\mathbf{y}\,\mathbf{V}^T + \boldsymbol{\mu}) - \mathbf{x}_{n+1}, \tag{3.8}$$

La norme de $\mathbf{h}$ représente la proportion de la nouvelle observation que notre espace de représentation ne peut pas modéliser. On peut ainsi créer un vecteur résiduel unitaire $\hat{\mathbf{h}}$.

$$\hat{\mathbf{h}} = \begin{cases} \frac{\mathbf{h}}{\|\mathbf{h}\|} & \text{si } \|\mathbf{h}\| > 0\,; \\ \mathbf{0}^T & \text{sinon.} \end{cases} \tag{3.9}$$

On distingue alors deux cas.

(i) Si la norme de $\mathbf{h}$ est faible, c'est que la dimension actuelle de l'espace de représentation est suffisante : on intègre la nouvelle observation en opérant une rotation sur la base des vecteurs propres.

$$\mathbf{V}' = \mathbf{V}\,\mathbf{R}, \tag{3.10}$$

où la matrice de rotation $\mathbf{R}$ est obtenue par diagonalisation d'une matrice auxiliaire $\mathbf{D}$ de taille $(r \times r)$.

$$\mathbf{D} = \frac{n}{n+1}\boldsymbol{\Lambda} + \frac{n}{(n+1)^2}\mathbf{y}^T\mathbf{y}, \tag{3.11}$$

Les valeurs propres obtenues lors de la diagonalisation de $\mathbf{D}$ constituent les nouvelles valeurs propres du modèle Ω. Ce sont les valeurs situées sur la diagonale de $\mathbf{\Lambda}'$.

$$\mathbf{D}\,\mathbf{R} = \mathbf{R}\,\mathbf{\Lambda}'. \tag{3.12}$$

(ii) Si, en revanche, la norme de $\mathbf{h}$ est élevée, c'est que la dimension de l'espace de représentation actuel est trop faible pour modéliser correctement toutes les données. Dans ce cas, on ajoute le vecteur résiduel unitaire $\hat{\mathbf{h}}$ à la base des vecteurs propres, faisant ainsi passer la dimension de l'espace de représentation de r à $r+1$.

$$\mathbf{V}' = [\mathbf{V} \quad \hat{\mathbf{h}}]\,\mathbf{R}^+, \tag{3.13}$$

où $\mathbf{R}^+$ est une matrice de rotation obtenue par diagonalisation de la matrice auxiliaire $\mathbf{D}^+$ de taille $(r+1)\times(r+1)$.

$$\mathbf{D}^+ = \frac{n}{n+1}\begin{bmatrix}\mathbf{\Lambda} & \mathbf{0}\\ \mathbf{0}^T & 0\end{bmatrix} + \frac{n}{(n+1)^2}\begin{bmatrix}\mathbf{y}^T\mathbf{y} & \gamma\,\mathbf{y}^T\\ \gamma\,\mathbf{y} & \gamma^2\end{bmatrix}, \tag{3.14}$$

avec $\gamma = \hat{\mathbf{h}}^T(\mathbf{x}_{n+1} - \boldsymbol{\mu})$. Les nouvelles valeurs propres du modèle Ω sont alors les valeurs propres de $\mathbf{D}^+$ situées sur la diagonale de $\mathbf{\Lambda}^+$.

$$\mathbf{D}^+\,\mathbf{R}^+ = \mathbf{R}^+\,\mathbf{\Lambda}^+. \tag{3.15}$$

3.1.3 ACP robuste

Comme nous l'avons vu dans la section 3.1.1, l'ACP fournit une solution au problème de la réduction de dimension qui est optimale au sens des moindres carrés. Cependant, lorsque l'ensemble des échantillons étudiés contient des valeurs aberrantes, le fait de minimiser l'erreur quadratique moyenne ne garantit pas que la base obtenue représentera au mieux la distribution sous-jacente des observations [Huber, 1981]. Afin de présenter les solutions alternatives, il convient de reformuler l'ACP comme un problème de minimisation d'énergie. En reprenant les notations précédentes, si la matrice de données $\mathbf{X}$ est préalablement centrée, le problème consiste à minimiser

$$\sum_{i=1}^{n} \|\mathbf{x}_i\,\mathbf{V}\,\mathbf{V}^T - \mathbf{x}_i\|_2^2 = \sum_{i=1}^{n} \|\mathbf{h}_i\|_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{p} h_{ij}^2, \tag{3.16}$$

où $\mathbf{h}_i$ est l'erreur de reconstruction obtenue pour l'échantillon $\mathbf{x}_i$. Cette énergie étant basée sur le paradigme des moindres carrés, la méthode n'est pas robuste aux observations aberrantes qui peuvent biaiser la solution.

L'ACP robuste consiste à utiliser un M-estimateur [Huber, 1981], c'est-à-dire que l'on va chercher à minimiser une fonction objectif plus générale que la somme des carrés des résidus, soit

$$\min \sum_{i=1}^{n}\sum_{j=1}^{p} \rho(h_{ij}), \tag{3.17}$$

où ρ est une fonction symétrique définie positive, admettant un unique minimum en 0. Typiquement, on choisira une fonction ρ qui croît moins vite que la fonction $t \mapsto t^2$.

Ici, l'ensemble des paramètres à définir est $\boldsymbol{\pi} = \{v_{ij}\}_{1 \leq i \leq p, 1 \leq j \leq r}$, c'est-à-dire la matrice des vecteurs propres $\mathbf{V}$ de taille $(p \times r)$. Le M-estimateur de $\boldsymbol{\pi}$ basé sur la fonction ρ est le vecteur $\boldsymbol{\pi}$ qui est solution des $p \times r$ équations suivantes :

$$\forall k \in [\![1, p \times r]\!] \quad \sum_{i=1}^{n} \sum_{j=1}^{p} \frac{\partial \rho}{\partial h_{ij}}(h_{ij}) \frac{\partial h_{ij}}{\partial \pi_k} = 0, \tag{3.18}$$

où les dérivées de ρ sont appelée *fonctions d'influence* car elles mesurent l'influence d'une donnée sur la valeur du paramètre estimé. Par exemple, dans le cas des moindres carrés, on a $\rho(t) = t^2$, donc sa dérivée par rapport à t, notée ψ, est définie par $\psi(t) = 2t$. Autrement dit, l'influence d'une donnée sur la valeur de l'estimation croît linéairement par rapport à l'importance de l'erreur, ce qui est rarement le cas, et ce qui confirme que les moindres carrés ne constituent pas un estimateur robuste. On définit également une fonction de pondération w par

$$w(h_{ij}) = \frac{\psi(h_{ij})}{h_{ij}}. \tag{3.19}$$

Ainsi, le système d'équations 3.18 devient

$$\forall k \in [\![1, p \times r]\!] \quad \sum_{i=1}^{n} \sum_{j=1}^{p} w(h_{ij}) h_{ij} \frac{\partial h_{ij}}{\partial \pi_k} = 0. \tag{3.20}$$

Zhang [Zhang, 1995] montre que cela revient à résoudre le problème des moindres carrés repondérés [Holland et Welsch, 1977] suivant :

$$\min \sum_{i=1}^{n} \sum_{j=1}^{p} w(h_{ij}^{(k-1)}) h_{ij}^2, \tag{3.21}$$

où l'exposant $^{(k)}$ indique ici le numéro d'itération.

Il existe un grand nombre de M-estimateurs disponibles dans la littérature, dont une liste pourra être trouvée dans [Zhang, 1995]. Ce sont généralement des fonctions définies à un paramètre près, celui-ci devant être optimisé de manière itérative. Dans [De la Torre et Black, 2001] est proposé un algorithme calculant une ACP robuste à l'aide du M-estimateur de Geman-McClure basé sur la fonction

$$\rho(t; \sigma) = \frac{t^2}{t^2 + \sigma^2}, \tag{3.22}$$

où σ est un paramètre qui contrôle la convexité de la fonction ρ.

Les M-estimateurs ne constituent pas la seule manière de rendre l'ACP plus robuste. Par exemple, Xu et Yuille [Xu et Yuille, 1995] ont proposé de

généraliser l'expression de la fonction objectif (3.16) en y ajoutant des variables booléennes qui valent 0 pour les échantillons considérés comme aberrants.

$$\min \sum_{i=1}^{n} \left(M_i \|\mathbf{x}_i \, \mathbf{V} \, \mathbf{V}^T - \mathbf{x}_i\|_2^2 + \alpha(1 - M_i) \right), \tag{3.23}$$

où chaque M_i de $\mathbf{M} = [M_1 \; M_2 \; \cdots \; M_n]$ est une variable booléenne aléatoire. Le second terme de l'équation (3.23) est un terme de pénalité visant à empêcher l'algorithme de minimisation de choisir la solution triviale où tous les M_i valent zéro : si pour un échantillon $\mathbf{x}_i$, la norme de l'erreur résiduelle est inférieure au seuil α, alors il vaudra mieux le considérer comme une valeur acceptable en fixant M_i à 1. Les auteurs proposent une méthode combinant optimisation discrète et continue pour résoudre le problème de minimisation.

Cette formulation est satisfaisante pour ne pas prendre en compte les échantillons qui sont jugés aberrants. Cependant, lorsque les échantillons sont des images, on a plutôt affaire à des régions aberrantes : on préfèrerait donc pondérer chaque pixel plutôt que chaque image. Nous verrons par la suite comment l'ACP robuste a été adaptée au domaine de l'analyse d'images.

3.2 Applications au traitement d'images et de séquences vidéo

3.2.1 Applications au traitement d'images fixes

Les différentes méthodes d'analyse de données sont utilisées depuis de nombreuses années dans le domaine de l'analyse d'images statiques. En particulier, l'ACP est très fréquemment utilisée depuis la fin des années 1980 dans le domaine de la reconnaissance de visages [Sirovich et Kirby, 1987]. Les algorithmes proposés sont nombreux, mais il s'agit toujours de créer une matrice de données à partir d'un ensemble d'images d'apprentissage et d'y appliquer une ACP afin de dégager une base de vecteurs propres (*eigenfaces*) dont on peut ne conserver que ceux qui expliquent le mieux la variance de la base d'apprentissage. Chaque nouvelle image est ensuite projetée dans cet espace de dimension réduite, et une mesure de distance est utilisée pour retrouver le plus proche voisin dans la base d'apprentissage afin de procéder à l'identification. Les différents algorithmes se différencient surtout par la mesure de distance utilisée et par le nombre d'axes principaux retenus. Le lecteur intéressé pourra se référer à [Yambor *et al.*, 2002] pour une étude comparative de ces méthodes. Dans ce même domaine, d'autres auteurs préfèrent utiliser l'analyse discriminante (ou analyse de Fisher) [Belhumeur *et al.*, 1997] ou encore l'analyse en composantes indépendantes [Bartlett *et al.*, 1998]. Par ailleurs, dans [Artač *et al.*, 2002] est proposée la première application de l'ACP incrémentale de [Hall *et al.*, 1998] à la reconnaissance d'objets à partir d'imagettes.

3.2.2 *Eigenbackgrounds* : modélisation de l'arrière-plan par ACP

Dans le domaine de l'analyse de séquences vidéo, les techniques d'analyse de données n'ont que très récemment commencé à être utilisées. En effet, ces algorithmes ont pour point commun d'utiliser comme donnée d'entrée une matrice de données, c'est-à-dire un ensemble de mesures booléennes et/ou réelles obtenues à partir d'un ensemble d'échantillons ou d'une population. Implicitement, ce type d'analyse se fait en deux temps : il faut d'abord effectuer l'ensemble des mesures à analyser et ensuite seulement, appliquer l'algorithme correspondant à la méthode d'analyse de données choisie. Cette particularité semble a priori difficilement compatible avec la nature séquentielle de l'acquisition de données vidéo. Par ailleurs, les méthodes d'analyse de données s'utilisent bien sûr sur des matrices de taille finie, alors qu'un flux vidéo peut avoir une durée infinie, si l'on se place dans le cas d'un programme traitant en temps réel les images fournies par une caméra vidéo. Néanmoins, dans [Oliver *et al.*, 2000] est proposée une première utilisation de l'ACP pour la modélisation de l'arrière-plan de scènes vidéo.

La construction du modèle de l'arrière-plan est réalisée à partir de N images d'apprentissage prises à des instants non consécutifs. A partir de ces images est calculée une image moyenne $\boldsymbol{\mu}_b$ et une matrice de variance-covariance $\mathbf{C}_b$. Pour calculer $\mathbf{C}_b$, il est nécessaire de réarranger les images sous la forme d'un vecteur-colonne, puis de calculer la matrice de variance-covariance comme dans l'équation 3.1. Celle-ci est ensuite diagonalisée pour obtenir une base de vecteurs propres $\boldsymbol{\Phi}_b$ et les valeurs propres associées disposées dans la matrice diagonale $\mathbf{L}_b$. Pour réduire la dimensionnalité du modèle, seuls les vecteurs propres associés aux M plus grandes valeurs propres sont conservés. La sous-matrice des vecteurs propres conservés est notée $\boldsymbol{\Phi}_{Mb}$.

Comme les objets en mouvement apparaissent à des endroits différents dans les images d'apprentissage, et comme ils sont généralement de petite taille, leur contribution au modèle est faible. Par conséquent, la base de représentation obtenue constitue un modèle robuste de la fonction de distribution de probabilité de l'arrière-plan, mais pas des objets en mouvement.

Une fois que le modèle est construit, on peut projeter chaque nouvelle image $\mathbf{I}_i$ dans l'espace de représentation pour modéliser les parties statiques de la scène. Les objets mobiles sont détectés en calculant la distance euclidienne entre l'image d'entrée et l'image reconstruite à partir de sa projection.

$$\mathbf{D}_i = \left\| (\mathbf{I}_i - \boldsymbol{\mu}_b)\boldsymbol{\Phi}_{Mb}\,\boldsymbol{\Phi}_{Mb}^T + \boldsymbol{\mu}_b - \mathbf{I}_i \right\|_2 , \tag{3.24}$$

où $\mathbf{D}_i$ est une carte de détection de mouvement appelée *distance-from-feature-space* (DFFS), terminologie introduite dans [Moghaddam et Pentland, 1995]. Les auteurs annoncent que les résultats sont aussi satisfaisants que ceux obtenus avec la méthode de [Wren *et al.*, 1997] (voir section 2.2.3.2) pour un moindre coût en temps de calcul. Ils précisent qu'il est aisé de rendre la méthode adaptative afin de compenser les évolutions de l'arrière-plan, sans pour

autant préciser comment y parvenir.

Dans [Rymel *et al.*, 2004], les auteurs proposent une version évolutive et orientée région de l'algorithme précédent. Tout d'abord, l'image est partitionnée en régions rectangulaires. Un modèle d'arrière-plan va être bâti et mis à jour pour chacune de ces régions. En partant de l'hypothèse que toute région contient à la fois des points appartenant à l'arrière-plan et à des objets mobiles, et que l'arrière-plan représente une majorité de ces points, les auteurs proposent de sous-échantillonner chaque région en ne considérant que M pixels parmi les N qu'elle contient ($M \ll N$). Les pixels retenus sont choisis aléatoirement dans leur région en suivant une distribution uniforme. Nous noterons $\{i_k\}_{k=1}^{M}$ les indices des pixels ainsi retenus ($\forall k \in [\![1, M]\!], 1 \leq i_k \leq N$). En supposant que les pixels choisis proviennent de l'arrière-plan, si $\mathbf{x}$ est le vecteur contenant tous les niveaux de gris d'une région donnée, on construit une version sous-échantillonnée de $\mathbf{x}$ notée $\mathbf{x}'$ de taille M qui constitue le vecteur caractéristique de l'arrière-plan de cette région. Si le vecteur moyen est $\boldsymbol{\mu}$ et la base de l'espace de représentation associée à $\mathbf{V}$, on construit également leurs versions sous-échantillonnées $\boldsymbol{\mu}'$ et $\mathbf{V}'$. Dans ce cas, $\mathbf{V}'$ n'est plus orthogonale, donc la représentation du bloc dans l'espace décrit par $\mathbf{V}'$ est le vecteur $\mathbf{b}$.

$$\mathbf{b} = (\mathbf{x}' - \boldsymbol{\mu}')\,\mathbf{V}'\,[\mathbf{V}'\mathbf{V}'^{T}]^{-1} \tag{3.25}$$

Le nombre de points choisis dans chaque bloc (la taille du vecteur $\mathbf{x}'$) est choisi de manière à sur-déterminer l'estimation de $\mathbf{b}$ afin d'augmenter la robustesse dans le cas où certains des pixels choisis proviendraient d'éléments en mouvement. La mise à jour de l'espace de représentation proposée dans [Rymel *et al.*, 2004] est identique à celle décrite dans [Hall *et al.*, 1998] (voir section 3.1.2). Commme la dimension de l'espace de représentation est incrémentée à chaque itération, il faut tronquer le modèle à partir d'un certain point. Les auteurs suggèrent de supprimer la dimension associée à la plus petite valeur propre du modèle dès que le nombre de dimensions excède une valeur optimale.

Dans le domaine de la navigation autonome de robots, les auteurs de [Vieira Neto et Nehmzow, 2005] proposent une version légèrement simplifiée de l'ACP incrémentale introduite dans [Hall *et al.*, 1998] et appliquée au traitement d'images dans [Artač *et al.*, 2002]. Dans ce cas, il s'agit de décider pour un petit nombre de régions d'intérêt, si elles représentent une « nouveauté » ou non. L'ACP incrémentale est comparée à un réseau de neurones souvent utilisé en robotique pour ce genre de tâches, et permet d'obtenir de bien meilleurs résultats.

Li *et al.* [Li *et al.*, 2003] proposent d'intégrer dans un même algorithme les notions d'ACP incrémentale et d'ACP robuste. L'ACP incrémentale décrite est simplifiée par rapport à la méthode de [Hall *et al.*, 1998] (voir section 3.1.2), mais elle fait explicitement appel à la matrice de variance-covariance des données, ce qui pose problème lorsque le nombre de dimensions est important. Pour contourner le problème, les auteurs ont recours à une approximation supplémentaire. La robustesse est obtenue par l'utilisation d'un M-estimateur

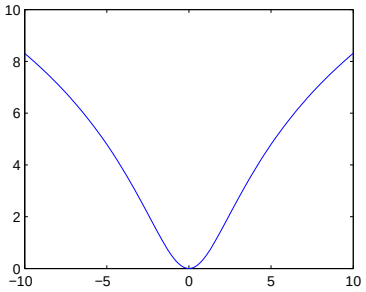
(cf. équation 3.17) basé sur la fonction lorentzienne définie par

$$\rho(t) = \frac{c^2}{2} \log\left(1 + \left(\frac{t}{c}\right)^2\right), \tag{3.26}$$

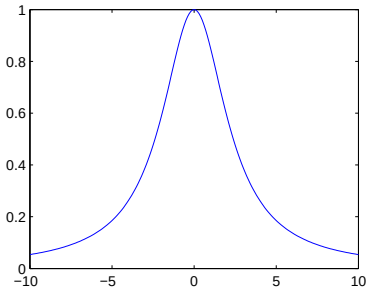
où c est un paramètre qui contrôle la convexité de la fonction. Cela leur permet d'introduire une fonction de pondération w définie par

$$w(t) = \frac{1}{t}\frac{\mathrm{d}\rho}{\mathrm{d}t}(t) = \frac{1}{1 + (t/c)^2}. \tag{3.27}$$

Cette fonction permet d'attribuer un poids à chaque élément d'observation, qui sera faible si l'erreur de reconstruction qui lui incombe est élevée. Les formes de la fonction ρ et de la fonction de pondération utilisées par [Li *et al.*, 2003] sont représentées sur la figue 3.1.



(a) Fonction ρ utilisée dans [Li *et al.*, 2003], dite lorentzienne ou de Cauchy.



(b) Fonction de pondération correspondante.

FIG. 3.1 – Un exemple de fonction robuste et de fonction de pondération correspondante.

Ainsi, les auteurs définissent une variable d'observation corrigée $\mathbf{z}_i$ dont les éléments valent

$$z_{ij} = \sqrt{w(h_{ij})}\, x_{ij}. \tag{3.28}$$

On peut montrer que la minimisation de la fonction objectif revient à minimiser

$$\sum_{i=1}^{n} \left\| \mathbf{z}_i \, \mathbf{V} \, \mathbf{V}^T - \mathbf{z}_i \right\|_2^2. \tag{3.29}$$

Il reste à déterminer c, le paramètre qui contrôle la convexité de la fonction ρ (équation 3.26). Dans la littérature, les paramètres des fonctions robustes sont calculés de manière itérative, ce qui est très coûteux en temps de calcul. Li *et al.* proposent une méthode simplifiée. Ils commencent par estimer σ_j, l'écart-type de la j-ème variable observée, de la manière suivante :

$$\sigma_j = \max_{i=1}^{p} \sqrt{\lambda_i}\, |v_{ij}|, \tag{3.30}$$

où λ_i est la i-ème valeur propre du modèle courant, et v_{ij} est le j-ème élément du i-ème vecteur propre considéré dans la base du nouvel espace. Pour la j-ème variable observée, on utilisera le paramère $c_j = \beta\,\sigma_j$ pour calculer le M-estimateur.

Conclusion sur l'état de l'art

L'étude des différentes méthodes de détection de mouvement proposées dans la littérature nous a permis de constater que la plupart des solutions proposées consistent à considérer les séquences vidéo comme des successions d'images, classées généralement en deux catégories : l'image courante, et le passé. L'approche la plus fréquente consiste à construire un modèle plus ou moins compact censé représenter tout le passé, et à confronter l'image courante à ce modèle afin de décider en tout point, si celui-ci représente l'arrière-plan ou un objet mobile. Nous pensons qu'il serait moins restrictif de considérer la scène dans le contexte spatio-temporel de l'instant que l'on considère. Cela nous oblige à prendre pleinement en considération la nature tridimensionnelle des données vidéo, et nous confronte au problème délicat du traitement en temps réel de données de dimensionnalité élevée.

C'est pourquoi nous avons vu, dans le second chapitre de l'état de l'art, qu'il existe des méthodes d'analyse de données qui permettent d'opérer une réduction de dimension. Certaines de ces techniques ont déjà été utilisées avec succès dans le domaine du traitement d'images, et dans une moindre mesure, de l'analyse de séquences vidéo. Nous considérons que ce type de méthodes est bien adapté à la problématique qui nous intéresse, et c'est cette solution que nous avons retenue pour proposer notre approche personnelle dans la partie suivante.

Deuxième partie

Nouvelle approche

Introduction

Comme nous l'avions évoqué dans l'introduction générale, tout système de suivi d'objets mobiles, quel que soit son domaine d'application, doit forcément posséder un module de détection de mouvement, et un module de modélisation des déplacements. L'étude de la littérature sur la détection de mouvement nous a permis de constater que la dimension temporelle des données vidéo était trop occultée dans les méthodes existantes. Nous allons proposer, dans le premier chapitre de cette partie, notre approche consistant à utiliser l'analyse de données pour réduire la dimension des données vidéo, afin de baser la détection de mouvement sur des *séquences élémentaires* plutôt que sur l'image courante uniquement.

Afin de valider les résultats obtenus par notre méthode de détection de mouvement, nous proposerons dans le deuxième chapitre de cette partie, une méthode de modélisation des déplacements qui exploite les résultats obtenus.

Chapitre 4

Détection de mouvement

Sommaire

Comme nous l'avons vu dans la section 3.2, les applications de l'analyse de données à la détection de mouvement ont toujours été réalisées selon le modèle de la méthode des *eigenbackgrounds* [Oliver *et al.*, 2000]. En effet, dans la littérature, on considère toujours que la variable aléatoire observée est l'image (représentée sous la forme d'un vecteur de niveaux de gris ou de couleurs), et que chaque nouvelle trame de la séquence vidéo est une réalisation de celle-ci. Cette vision du problème semble assez naturelle lorsque l'on considère la séquence vidéo comme une succession d'images statiques. Cependant, les données vidéo étant, par nature, tridimensionnelles (deux dimenions spatiales, et une dimension temporelle), il existe nécessairement d'autres manières de formaliser le problème de l'analyse des données vidéo.

4.1 Représentation des données vidéo

De manière à choisir la représentation des données la plus adaptée à la résolution de notre problème, il convient d'étudier et de comparer les différentes représentations possibles, ce que nous allons faire dans la section suivante au travers de l'étude d'un exemple de séquence vidéo. Cette étude nous permettra par la suite de proposer notre propre système de représentation.

4.1.1 Étude d'un exemple

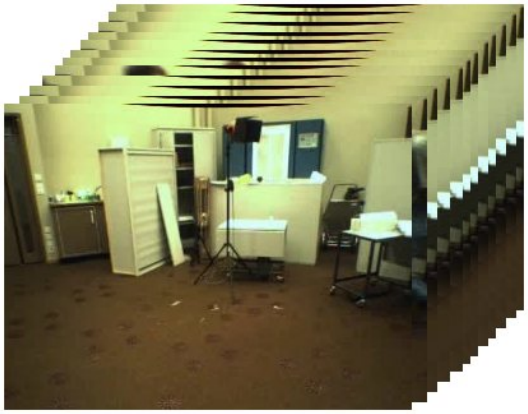
Afin d'étudier les différentes manières dont on peut se représenter une séquence vidéo, considérons l'exemple d'une scène comportant une personne traversant le champ de vision de la caméra de la droite vers la gauche. La séquence vidéo correspondant à cette scène serait constituée des images prises à des instants successifs telles que représentées sur la figure 4.1a[1]. Pour obtenir une vue compacte de la séquence étudiée, on peut empiler toutes ces images afin d'obtenir un volume tel que celui de la figure 4.1b. Dans ce cas, nous utilisons la troisième dimension pour représenter l'axe du temps. Dans toutes les méthodes de détection de mouvement par modélisation de l'arrière-plan de la scène (cf. chapitre 2), et notamment dans la méthode des *eigenbackgrounds* [Oliver *et al.*, 2000] qui nous intéresse plus particulièrement, les données vidéo sont traitées comme une telle succession d'images bidimensionnelles que l'on confronte à un modèle de l'arrière-plan afin de décider en tout point si celui-ci dénote ou non un mouvement apparent. Ce formalisme quasi unanime s'explique par plusieurs facteurs.

- La manière dont les données vidéo sont acquises incite naturellement à considérer les différentes images comme des entités indépendantes. En effet, une caméra vidéo statique fonctionne comme un appareil photographique prenant des vues d'une même scène à intervalles réguliers (25 fois par seconde), et les données obtenues sont transmises et stockées sous la forme d'une succession d'images numériques.
- Une coupe dans le plan (x, y) est une image statique qui est facile à se représenter, tandis qu'un échantillon de valeurs prises le long de l'axe du temps, ou une région du plan (x, t) par exemple, sont des entités plus abstraites auxquelles il est difficile d'attribuer du sens.
- Les deux dimensions spatiales d'une séquence vidéo sont connues et finies, tandis que la dimension temporelle est *a priori* infinie.

Cependant, si l'on choisit d'ignorer le caractère discret de la dimension temporelle induit par le fonctionnement de la caméra vidéo, toute séquence vidéo peut être considérée comme un volume dense 2D+T tel que représenté sur la Figure 4.2a, et l'on peut en tirer des informations porteuses de sens autrement qu'en pratiquant des coupes dans le plan (x, y).

[1]La séquence vidéo utilisée pour créer les figures 4.1, 4.2 et 4.3 a été fournie par le laboratoire CVLAB de l'École Polytechnique Fédérale de Lausanne (Suisse).

(a) Une séquence de 200 trames vue comme une succession d'images 2D. (Seules les trames 1, 27, 93, 151 et 200 sont représentées.)



(b) Vue compacte de la même séquence obtenue en empilant toutes les images le long de la troisième dimension.

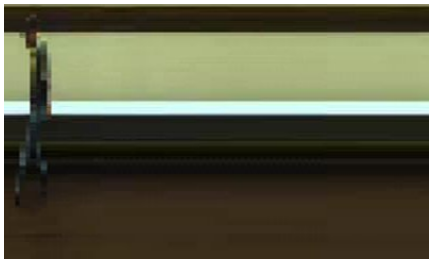
FIG. 4.1 – Exemple de séquence vidéo avec sa représentation en 2D (a) et volumique (b).

Par exemple, la figure 4.2b représente une coupe dans le plan (x, t) (c'est-à-dire que l'on fixe y à une certaine valeur) qui nous fournit différentes informations. On remarque tout d'abord des bandes colorées verticales. Comme une coupe dans le plan (x, t) représente la superposition verticale de toutes les observations que l'on obtiendrait en ne regardant qu'une ligne de pixels à hauteur fixe pendant toute la durée de la séquence, les lignes verticales de couleur uniforme représentent les points de l'image dont l'apparence reste constante, c'est-à-dire ceux où aucun mouvement n'a lieu. On remarque ensuite un motif entrelacé de couleur bleue orienté diagonalement dans le plan (x, t). Il s'agit du motif spatio-temporel créé par les jambes du personnage (la coupe a été pratiquée à cette hauteur) lorsque celui-ci passe dans le champ de vision de la caméra. Ce motif indique très clairement que le personnage apparaît à l'écran en arrivant par la droite peu après le début de la séquence, qu'il traverse le champ de vision pour en sortir par la gauche, puis réapparaît à l'endroit où il était sorti en se dirigeant vers la droite. On observe que la séquence s'arrête lorsque le personnage a parcouru le premier quart gauche du champ de vision.

La figure 4.2c représente une coupe dans le plan (y, t), c'est-à-dire que l'on fixe x à une certaine valeur (qui est ici le point le plus à droite de l'image). Elle correspond à ce que verrait un observateur en regardant la scène à travers une fente verticale située à droite du champ de vision. Les observations succes-

(a) La séquence de la figure 4.1 peut être vue comme un volume 2D+T. On peut voir qu'un objet mobile est ici représenté par une région uniforme tridimensionnelle.

(b) Coupe dans le plan (x, t) pratiquée dans la séquence précédente à la hauteur des jambes du personnage.



(c) Coupe dans le plan (y, t) pratiquée dans la séquence précédente à l'extrémité droite du champ de vision.

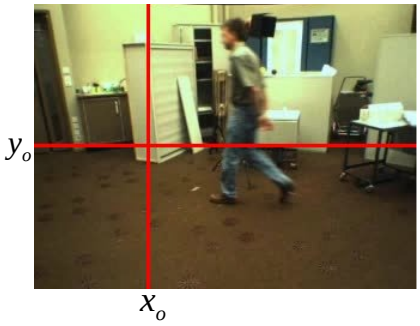
FIG. 4.2 – Interprétation du volume 2D+T que constitue une séquence vidéo.

sives sont ici empilées horizontalement. Dans ce cas, les éléments statiques de la scène sont représentés par des bandes colorées horizontales. Cette représentation nous permet de savoir précisément à quel moment le personnage passe dans la partie du champ de vision que l'on considère.

Si l'on décide de fixer x et y à des valeurs données, i.e. si l'on observe un point (x_o, y_o) de l'image (figure 4.3a), les différentes observations obtenues au cours du temps peuvent être entreposées dans un vecteur dont la dimension est égale à la durée de la séquence considérée. La figure 4.3b constitue un exemple d'un tel vecteur. On constate que sur ce vecteur, toutes les valeurs sont égales, sauf pendant un bref intervalle de temps où la couleur observée est plus sombre, ce qui correspond à l'occlusion du décor par le personnage.

L'étude de cet exemple simple illustre bien le fait que la représentation des données vidéo comme une succession d'images 2D n'est pas la méthode la plus simple de préparer l'extraction des informations relatives au mouvement. Cependant, cette même représentation est presque unanimement utilisée dans

la littérature. On notera néanmoins quelques exceptions, comme les méthodes utilisant le flot optique, qui prennent davantage en considération la dimension temporelle des séquences vidéo, mais se restreignent à un intervalle de temps de taille 2. Par ailleurs, dans les travaux de [Ma et Zhang, 2001], on considère que les zones en mouvement sont celles où l'entropie spatio-temporelle de la séquence est maximale. Contrairement aux méthodes de modélisation de l'arrière-plan, la dimension temporelle est pleinement prise en considération par un algorithme d'analyse semi-locale dans le volume 2D+T que constitue la séquence vidéo. Les résultats obtenus sont assez proches d'une dérivée temporelle lissée par un opérateur de moyenne mobile. Notre étude se situe dans ce cadre : nous cherchons à détecter les objets mobiles en analysant le volume 2D+T sans utiliser de manière systématique le découpage en images, mais plutôt en volumes élémentaires.



(a) On observe un point (x_o, y_o) fixé au cours du temps.

(b) Les valeurs observées en (x_o, y_o) peuvent être entreposées dans un vecteur. La partie sombre représente le passage du personnage à cet endroit pendant un intervalle de temps.

FIG. 4.3 – Vecteur obtenu en observant un point donné pendant toute la durée de la séquence.

4.1.2 Représentation proposée

Les données vidéo sont initialement représentées par une fonction définie dans un espace à trois dimensions : deux dimensions spatiales (x, y) et une temporelle (t). A chaque point de cet espace est associé un niveau de gris (ou un vecteur de composantes couleur) $I(x, y, t)$ en un point (x, y) à l'instant t. Les différentes entités sémantiques (arrière-plan, objets mobiles) sont donc des sous-ensembles de points de cet espace.

Afin de les identifier, il convient de les agréger en classes de points présentant des caractéristiques communes. Il va sans dire que le nombre de points à considérer est très important, surtout si l'on veut prendre en compte plus

de deux trames pour détecter les objets en mouvement. C'est pourquoi l'approche consistant à bâtir un modèle de l'arrière-plan est si usuelle : les seuls points à considérer sont ceux de la trame courante, tandis que le modèle de l'arrière-plan est censé résumer toutes les observations passées. Nous pensons qu'il est préférable de conserver une connaissance moins synthétique du passé car l'information pertinente à en extraire n'est pas toujours la même.

Nous envisageons donc de choisir un espace de représentation adapté davantage à la séquence elle-même plus qu'à chacune des trames et qui permette de prendre en compte le mouvement sans modifier l'information initiale. Les techniques d'analyse de données permettent de réaliser une réduction de dimension adaptative par rapport aux données étudiées. Il s'agit toujours d'étudier un ensemble d'*individus* au travers d'un certain nombre de *variables observées*. Ces individus sont représentés sous la forme d'un vecteur de nombres réels dans le cas d'une analyse quantitative.

Parmi les différentes méthodes de représentation des données vidéo présentées dans la section précédente, l'observation d'un lieu fixé de l'image le long d'un intervalle de temps (figure 4.3b) fournit une représentation compacte dans laquelle l'information de mouvement est facile à extraire. Nous nous orienterons donc vers une représentation prenant la forme d'un ensemble de vecteurs dont les éléments décrivent l'évolution d'un lieu donné au cours du temps.

Nous désignerons par $\mathcal{V} = \{I(1), I(2), \ldots, I(\tau)\}$ l'ensemble des données contenues dans le volume vidéo, où τ est la durée de la séquence correspondante.

Pour un intervalle de temps $[1, \tau]$, le vecteur représentant l'évolution de l'apparence du lieu $\mathbf{x} = (x, y)$ pendant la durée de la séquence étudiée sera noté $\mathbf{u}_{\mathbf{x},\tau}$ et défini par

$$\mathbf{u}_{\mathbf{x},\tau} = [I(\mathbf{x}, 1) \quad I(\mathbf{x}, 2) \quad \cdots \quad I(\mathbf{x}, \tau)]^T. \tag{4.1}$$

Lui est associée la base canonique usuelle $\mathcal{B} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_\tau\}$ avec

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \ldots \quad \mathbf{e}_\tau = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \tag{4.2}$$

Dans la perspective de l'utilisation de techniques d'analyse de données, on peut bâtir une matrice de données (ou tableau de données), dont chaque ligne est un individu — c'est-à-dire la représentation d'un lieu $\mathbf{x}$ du plan-image —, et chaque colonne est une variable descriptive — c'est-à-dire l'apparence d'un point de l'image à un instant donné. Il existe une telle matrice pour chaque durée τ, que l'on note $\mathbf{X}_\tau$ de taille $n \times \tau$ (n est le nombre de pixels du champ de vision), et qui est définie par

$$\mathbf{X}_\tau = [\mathbf{u}_{\mathbf{x}_1,\tau} \quad \mathbf{u}_{\mathbf{x}_2,\tau} \quad \cdots \quad \mathbf{u}_{\mathbf{x}_n,\tau}]^T. \tag{4.3}$$

On s'intéresse davantage aux variations d'apparence dans l'image qu'à la valeur des pixels en elle-même. De manière à mettre en évidence cette variation, on ne perd aucune information en considérant l'ensemble $\mathcal{V}' = \{I(1), I(2) - I(1), \ldots, I(\tau) - I(\tau - 1)\}$ au lieu de $\mathcal{V}$. On remplace ainsi les vecteurs $\mathbf{u}_{\mathbf{x},\tau}$ de dimension τ par les vecteurs $\mathbf{u}'_{\mathbf{x},\tau}$ composés de la dérivée discrète par rapport au temps de l'apparence des pixels considérés. Pour ne pas perdre d'information, nous avons ajouté dans l'ensemble $\mathcal{V}'$ la première image de la séquence $I(1)$ qui fournit la valeur initiale des pixels. La dérivée discrète est obtenue par un opérateur de convolution sur $\mathbf{u}_{\mathbf{x},\tau}$, soit

$$\mathbf{u}'_{\mathbf{x},\tau} = \mathbf{u}_{\mathbf{x},\tau} \otimes [1 \quad -1]^T \tag{4.4}$$

$$= [I(\mathbf{x},1) \quad I(\mathbf{x},2) - I(\mathbf{x},1) \quad \cdots \quad I(\mathbf{x},\tau) - I(\mathbf{x},\tau-1)]^T, \tag{4.5}$$

où $\otimes$ est l'opérateur de convolution.

Nous nous intéressons au mouvement relatif, donc la première composante des vecteurs $\mathbf{u}'_{\mathbf{x},\tau}$ n'est pas indispensable à l'analyse, d'où l'utilisation de la transformation p définie en tout point $\mathbf{x}$ par

$$\begin{array}{rccc} p: & \mathbb{R}^\tau & \longrightarrow & \mathbb{R}^{\tau-1} \\ & \begin{pmatrix} I(\mathbf{x},1) \\ \vdots \\ I(\mathbf{x},\tau) \end{pmatrix} & \longmapsto & \begin{pmatrix} I(\mathbf{x},2) - I(\mathbf{x},1) \\ \vdots \\ I(\mathbf{x},\tau) - I(\mathbf{x},\tau-1) \end{pmatrix} \end{array} \tag{4.6}$$

Les vecteurs ainsi transformés seront notés $\tilde{\mathbf{u}}'_{\mathbf{x},\tau} = p(\mathbf{u}_{\mathbf{x},\tau})$. Si l'on utilise les vecteurs $\tilde{\mathbf{u}}'_{\mathbf{x},\tau}$ pour construire la matrice de données, on notera celle-ci $\mathbf{X}'_\tau$.

On peut considérer qu'on a un volume des dérivées temporelles désigné par l'ensemble $\tilde{\mathcal{V}}' = \{I(2) - I(1), \ldots, I(\tau) - I(\tau - 1)\}$. La figure 4.4 constitue un exemple d'ensemble $\tilde{\mathcal{V}}'$ construit à partir d'une séquence vidéo courte.

Plus le délai est important, plus le mouvement sera visible. Le cumul des dérivées est alors à envisager. D'où l'introduction de $\mathcal{V}'' = \{I(2) - I(1), I(3) - I(1), \ldots, I(\tau) - I(1)\}$. Les ensembles $\mathcal{V}'$ et $\mathcal{V}''$ représentent la même information. Les vecteurs caractéristiques des points de $\mathcal{V}'$ s'expriment par l'équation 4.6 et ceux de $\mathcal{V}''$, notés $\tilde{\mathbf{u}}''_{\mathbf{x},\tau}$ sont définis par

$$[0 \quad \tilde{\mathbf{u}}''_{\mathbf{x},\tau}]^T = \mathbf{u}^T_{\mathbf{x},\tau} - I(\mathbf{x},1) \cdot \mathbf{1}^T_\tau, \tag{4.7}$$

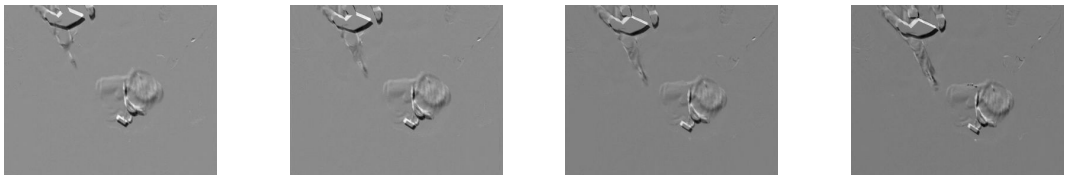
où $\mathbf{1}_\tau$ est un vecteur de dimension τ dont tous les éléments valent 1. Par conséquent, les vecteurs $\tilde{\mathbf{u}}''_{\mathbf{x},\tau}$ s'expriment par

$$\tilde{\mathbf{u}}''_{\mathbf{x},\tau} = [I(\mathbf{x},2) - I(\mathbf{x},1) \quad I(\mathbf{x},3) - I(\mathbf{x},1) \quad \cdots \quad I(\mathbf{x},\tau) - I(\mathbf{x},1)]^T. \tag{4.8}$$

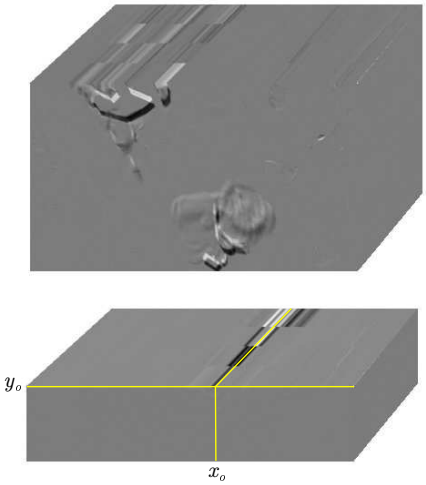
La figure 4.5 constitue un exemple d'ensemble $\tilde{\mathcal{V}}''$ construit à partir de la même séquence vidéo que précédemment.

Si l'on compare les vecteurs caractéristiques d'un même lieu (x_o, y_o) construits à partir des images de $\tilde{\mathcal{V}}'$ (figure 4.4d) et à partir des images de $\tilde{\mathcal{V}}''$ (figure 4.5c), on constate que dans le premier cas le mouvement n'est apparent

(a) Une séquence vidéo de 5 images. Celles-ci correspondent aux éléments de $\mathcal{V}$.



(b) Les 4 éléments de $\tilde{\mathcal{V}}'$ correspondant à cette même séquence vidéo.



(c) L'ensemble $\tilde{\mathcal{V}}'$ vu comme un volume 2D+T.

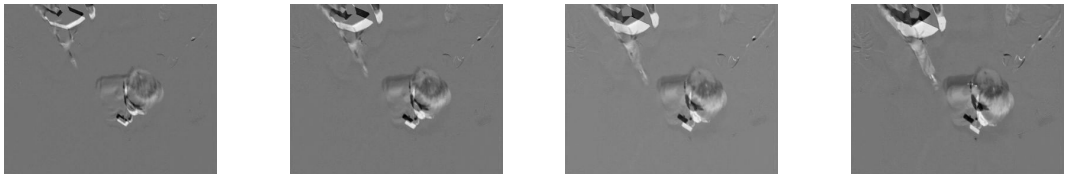
(d) Le vecteur de dimension 4 représentant le pixel (x_o, y_o) dans $\tilde{\mathcal{V}}'$.

FIG. 4.4 – Représentation d'une séquence de 5 images dans l'ensemble $\tilde{\mathcal{V}}'$.

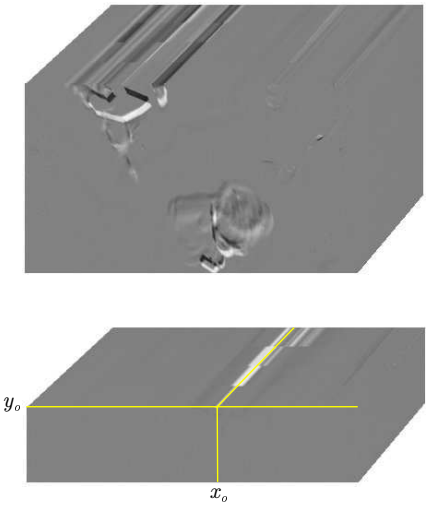
que sur les composantes correspondant au passage des contours de l'objet, alors que dans le deuxième cas le mouvement est apparent durant toute la période où l'objet est visible en (x_o, y_o). Ceci est dû à la propagation des informations de mouvement vers les dernières composantes des vecteurs construits à partir de $\tilde{\mathcal{V}}''$.

Si l'on utilise les vecteurs $\tilde{\mathbf{u}}''_{\mathbf{x},t}$ pour construire la matrice de données, on notera celle-ci $\mathbf{X}''_\tau$.

Pour passer de $\tilde{\mathcal{V}}'$ à $\tilde{\mathcal{V}}''$, nous avons utilisé la transformation de matrice $\mathbf{T}$



(a) Les 4 éléments de $\tilde{\mathcal{V}}''$ correspondant à la séquence vidéo de la figure 4.4a.



(b) L'ensemble $\tilde{\mathcal{V}}''$ vu comme un volume 2D+T.

(c) Le vecteur de dimension 4 représentant le pixel (x_o, y_o) dans $\tilde{\mathcal{V}}''$.

FIG. 4.5 – Représentation d'une séquence de 5 images dans l'ensemble $\tilde{\mathcal{V}}''$.

définie par

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 1 & 0 \\ 1 & \cdots & 1 & 1 \end{pmatrix}, \tag{4.9}$$

ce qui revient à faire dans l'espace de représentation des caractéristiques des pixels de $\tilde{\mathcal{V}}'$ un changement de base. Si $\{\mathbf{e}'_1, \mathbf{e}'_2, \ldots, \mathbf{e}'_{\tau-1}\}$ est la base de représentation dans $\tilde{\mathcal{V}}'$, la base de représentation dans $\tilde{\mathcal{V}}''$ s'écrit $\{\mathbf{e}''_1, \mathbf{e}''_2, \ldots, \mathbf{e}''_{\tau-1}\}$ avec

$$\begin{cases} \mathbf{e}''_{\tau-1} &= \mathbf{e}'_{\tau-1} \\ & \vdots \\ \mathbf{e}''_i &= \mathbf{e}''_{i+1} - \mathbf{e}''_i \\ & \vdots \\ \mathbf{e}''_1 &= \mathbf{e}''_2 - \mathbf{e}''_1 \end{cases}. \tag{4.10}$$

Nous avons ainsi à notre disposition trois systèmes pour représenter les données à étudier. Nous allons présenter dans la section suivante comment les données pourront être modélisées de manière efficace dans un espace de dimension réduite. Nous allons commencer par chercher une base encore mieux adaptée à la résolution de notre problème.

4.2 Modélisation concise des données

Compte tenu des contraintes de temps réel ou approché auxquelles sont soumises la plupart des applications de traitement de séquences vidéo, nous souhaitons réduire la dimension des vecteurs caractéristiques utilisés. Dans un premier temps nous allons présenter comment, grâce à une analyse en composantes principales, nous parvenons à concentrer l'information pertinente sur les premières composantes des vecteurs. En outre, nous comparerons les différents systèmes de représentation dans ce cadre. Dans un second temps, nous chercherons à réduire la dimension des vecteurs ainsi modifiés.

4.2.1 Changement de base adapté aux données

En utilisant le formalisme présenté dans la section précédente, nous avons à notre disposition une masse de données constituées de vecteurs de dimension *a priori* infinie, qui représentent l'évolution au cours du temps de l'apparence d'un lieu (x, y) du plan image. Dans la perspective de l'utilisation des techniques d'analyse des données, la séquence n'est plus considérée comme une fonction mais comme un ensemble d'individus : les pixels que nous observons quand nous regardons la séquence. Dans cette phase, les relations spatiales entre les pixels sont donc ignorées. Pour éviter de devoir faire une analyse fine, ce ne sont pas les objets que l'on suit mais c'est une position fixe que l'on considère sur la surface de l'image. De chaque pixel on va retenir plusieurs valeurs de niveaux de gris au cours du temps.

Nous noterons p le nombre de valeurs retenues. Pour une séquence de τ images, p vaudra τ si les pixels sont représentés dans l'ensemble $\mathcal{V}$, ou $\tau - 1$ si les pixels sont représentés dans $\tilde{\mathcal{V}}'$ ou $\tilde{\mathcal{V}}''$. Chaque pixel devient alors un individu caractérisé par un ensemble de paramètres. Les individus sont repérés dans un espace de dimension p. Comme notre méthode traite p trames à la fois, nous pouvons nous permettre d'être p fois plus lent que si nous traitions chaque trame individuellement, et donc d'utiliser des techniques plus coûteuses en temps de calcul.

Néanmoins, pour rester dans des temps de traitement raisonnables, presque temps réel, nous devons faire une réduction de la masse des informations. Il existe de nombreuses méthodes de réduction de dimension, telles que l'analyse en composantes principales (ACP), l'analyse factorielle des correspondances (AFC), toute la famille des méthodes d'analyse en composantes indépendantes (ACI), ou encore les algorithmes à base de réseaux neuronaux tels que les

cartes de Kohonen ou les architectures en diabolo. Le lecteur intéressé pourra se référer à [Fodor, 2002] pour obtenir un panorama détaillé. L'ACP étant connue pour être la meilleure technique linéaire de réduction de dimension au sens des moindres carrés, nous avons choisi cette méthode dans le but de ne préserver que les informations qui permettront au mieux de discriminer les points et de construire des classes — ici, les objets en mouvement et l'arrière-plan de la scène. La théorie sous-jacente à cette méthode a été présentée dans la section 3.1.

On peut penser que les pixels correspondant au fond ont des composantes à peu près toutes égales alors que les pixels correspondant au passage d'un objet mobile comportent un changement. C'est ce changement que l'on veut mettre en évidence. Pour cela il est intéressant de trouver l'axe, c'est-à-dire la bonne base dans l'espace de représentation, où la variance du facteur est la plus grande.

Dans le cas d'une séquence vidéo, la matrice des données $\mathbf{X}$ contient donc l'ensemble des caractéristiques des points à considérer. Par la suite, nous noterons n le nombre de lignes de $\mathbf{X}$, c'est le nombre de pixels de l'image, et p son nombre de colonnes, c'est le nombre de caractéristiques retenues pour chaque pixel, lié à la longueur de la séquence étudiée. Les deux premières coordonnées peuvent prendre un nombre fini de valeurs — le domaine de définition $\mathcal{D}_P$ des pixels est borné. En revanche, le domaine de définition de la troisième coordonnée (le temps) est *a priori* non borné. Il convient donc de choisir une plage de valeurs qui devra contenir toute l'information pertinente. Nous proposons d'utiliser le domaine $\mathcal{D}_t = \{t - p + 1, \ldots, t\}$ où t est le temps courant. L'influence sur les résultats du paramètre p sera étudiée par la suite.

Nous choisissons de remplir la matrice $\mathbf{X}$ en considérant qu'une donnée (une ligne) est un point (x, y), et qu'une variable (une colonne) est un ensemble de niveaux de gris observés à chaque instant de $\mathcal{D}_t$. La nouvelle base de l'espace de représentation est alors associée aux vecteurs propres de la matrice de covariance $\mathbf{C}$ des données :

$$\mathbf{C} = \frac{1}{p}\bar{\mathbf{X}}^T\,\bar{\mathbf{X}}, \tag{4.11}$$

où $\bar{\mathbf{X}}$ est la matrice des données centrées. Comme nous n'avons aucune information supplémentaire, on suppose que chaque variable devrait présenter une variance comparable, et nous utilisons une ACP dite « simple » (données centrées) plutôt qu'une ACP « standard » (données centrées-réduites).

Les axes principaux sont obtenus par diagonalisation de la matrice de covariance, c'est-à-dire que l'on recherche une matrice orthogonale $\mathbf{V}$ dont les colonnes sont des vecteurs propres de $\mathbf{C}$ et une matrice diagonale $\mathbf{\Lambda}$ dont les éléments diagonaux sont les valeurs propres de $\mathbf{C}$ telles que

$$\mathbf{C} = \mathbf{V}\,\mathbf{\Lambda}\,\mathbf{V}^T. \tag{4.12}$$

Si l'on réordonne les vecteurs propres dans l'ordre décroissant des valeurs propres associées, on notera $\{\mathbf{v}_1, \ldots, \mathbf{v}_p\}$ l'ensemble des vecteurs propres choisis

et $\{\lambda_1, \ldots, \lambda_p\}$ l'ensemble des valeurs propres associées. L'ensemble des vecteurs propres constitue une base de représentation des données $\mathcal{B}_{ACP} = \{\mathbf{v}_j\}_{j=1}^{p}$ telle que la variance des données projetées sur chacun des axes décroît lorsque le rang de l'axe augmente. De manière plus formelle :

$$\forall j \in [\![1, p-1]\!] \quad \forall k \in [\![j+1, p]\!] \quad \text{var}(\bar{\mathbf{X}}\,\mathbf{v}_j) \geq \text{var}(\bar{\mathbf{X}}\,\mathbf{v}_k), \tag{4.13}$$

où Var$(\cdot)$ désigne la variance des éléments d'un vecteur.

Considérons la séquence de 10 trames de 576 lignes par 720 colonnes, dont la première moitié est représentée sur la figure 4.4a. Si l'on représente la séquence par l'ensemble $\mathcal{V}$, la matrice $\mathbf{X}$ a donc 576×720 lignes et 10 colonnes. Chaque ligne de $\mathbf{X}$ est un des vecteurs $\mathbf{u}_i$ définis par l'équation 4.1.

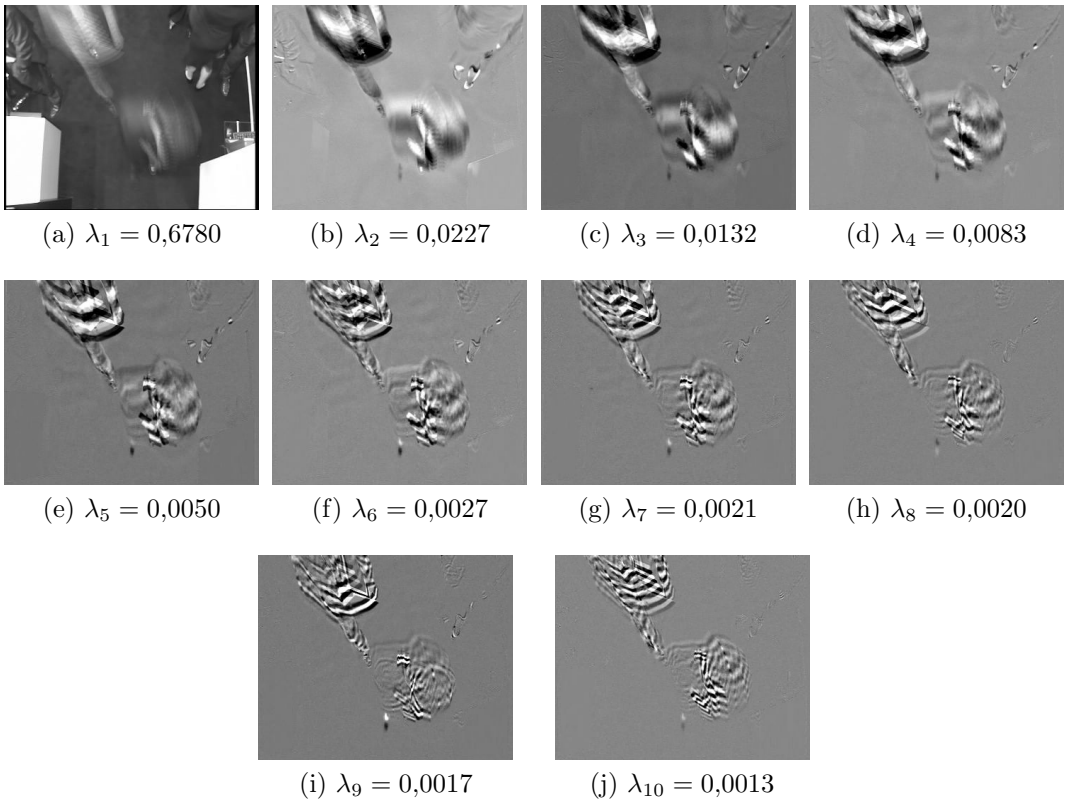
La figure 4.6 montre les dix projections de $\mathbf{X}$ sur les axes principaux issus de l'ACP. Plus précisément nous considérons le domaine de l'image et nous construisons une image dont le niveau de gris correspond à la valeur de la composante du vecteur de caractéristiques sur l'un des facteurs. Concrètement, la projection $\boldsymbol{\pi}_j = \bar{\mathbf{X}}\,\mathbf{v}_j$ de l'ensemble des données $\mathbf{X}$ sur l'axe $\mathbf{v}_j$ est transformée de manière à ce que tous ses éléments soient contenus dans l'intervalle $[0, 1]$. Cette transformation est obtenue par application de la fonction f définie par

$$\begin{array}{rccc} f: & \mathbb{R}^p & \longrightarrow & [0,1]^p \\ & \boldsymbol{\pi} & \longmapsto & \dfrac{\boldsymbol{\pi} - \min\limits_{i=1}^{n}(\pi_i)}{\max\limits_{i=1}^{n}(\pi_i) - \min\limits_{i=1}^{n}(\pi_i)} \end{array} \tag{4.14}$$

La figure 4.7 représente la composition de chacun des facteurs principaux, c'est-à-dire la valeur de chaque composante pour chacun des vecteurs propres de $\mathbf{C}$. Chaque ensemble de 10 barres correspond à une direction propre, et la i-ème barre de l'histogramme correspondant au j-ème vecteur propre représente la valeur de la i-ème composante de ce vecteur.

Comme nous le laisse supposer la figure 4.6a, la projection des données sur le premier axe principal n'est autre que la moyenne des images de la séquence. Ceci est confirmé lorsque l'on regarde la composition du premier facteur principal sur la figure 4.7, dont toutes les composantes ont des valeurs très similaires. On peut donc en déduire qu'en représentant les données dans l'ensemble $\mathcal{V}$, le premier facteur principal n'apporte aucune information en termes de mouvement. En revanche, les projections des données sur les axes suivants révèlent très clairement les zones où un mouvement s'est produit.

On observe également sur ce graphique que la fonction qui associe les poids affectés aux différentes trames pour définir chacun des axes factoriels, a une allure sinusoïdale dont la fréquence augmente avec le rang de l'axe. La base obtenue fait donc pense à celle qui est introduite dans la transformée de Fourier. Nous avons ainsi, par le biais d'un changement de base, adapté l'espace de représentation au contenu d'un domaine tridimensionnel. Comme il existait une disparité dans la signification des trois dimensions d'origine (deux spatiales et une temporelle), nous ne pouvions pas avoir recours directement à une transformée de Fourier 3D. Par contre, la vitesse se traduisant par la modification



(a) $\lambda_1 = 0{,}6780$ (b) $\lambda_2 = 0{,}0227$ (c) $\lambda_3 = 0{,}0132$ (d) $\lambda_4 = 0{,}0083$

(e) $\lambda_5 = 0{,}0050$ (f) $\lambda_6 = 0{,}0027$ (g) $\lambda_7 = 0{,}0021$ (h) $\lambda_8 = 0{,}0020$

(i) $\lambda_9 = 0{,}0017$ (j) $\lambda_{10} = 0{,}0013$

FIG. 4.6 – Projections de $\mathbf{X}$ sur chacun des axes mis en évidence par l'ACP dans la base $\mathcal{B}$ et valeurs propres associées à chacun des facteurs.

de l'apparence des pixels, nous sommes en train d'introduire par ce choix de la base de l'espace de représentation, une étude fréquentielle du volume étudié.

Pour construire les ensembles $\tilde{\mathcal{V}}'$ et $\tilde{\mathcal{V}}''$, nous avons pris le parti d'ignorer la valeur de référence des pixels (constituée par la première image de la séquence) afin de se focaliser sur le mouvement relatif. Nous pouvons donc espérer qu'en exprimant les données dans ces espaces de représentation, le premier axe factoriel sera celui qui présente le plus d'information concernant le mouvement dans la scène.

Nous commencerons par exprimer les données dans $\tilde{\mathcal{V}}'$. Dans ce cas, la matrice de données $\mathbf{X}'$ comporte 576×720 lignes et 9 colonnes. Chacune de ses lignes est un vecteur $\tilde{\mathbf{u}}'$ tel que défini par l'équation 4.6. Sa matrice de covariance $\mathbf{C}'$ est une matrice carrée de dimension 9 qui est construite de la même manière selon l'équation 4.11. Par diagonalisation, on obtient 9 valeurs propres $\{\lambda'_1, \ldots, \lambda'_9\}$ rangées par ordre croissant et disposées sur la diagonale d'une matrice $\mathbf{\Lambda}'$, et 9 vecteurs propres associés $\{\mathbf{v}'_1, \ldots, \mathbf{v}'_9\}$ qui constituent les colonnes d'une matrice $\mathbf{V}'$.

La figure 4.8 montre les neuf projections de $\mathbf{X}'$ sur les axes principaux issus de l'ACP. Les projections sont représentées sous forme d'images en utilisant la formule de l'équation 4.14.

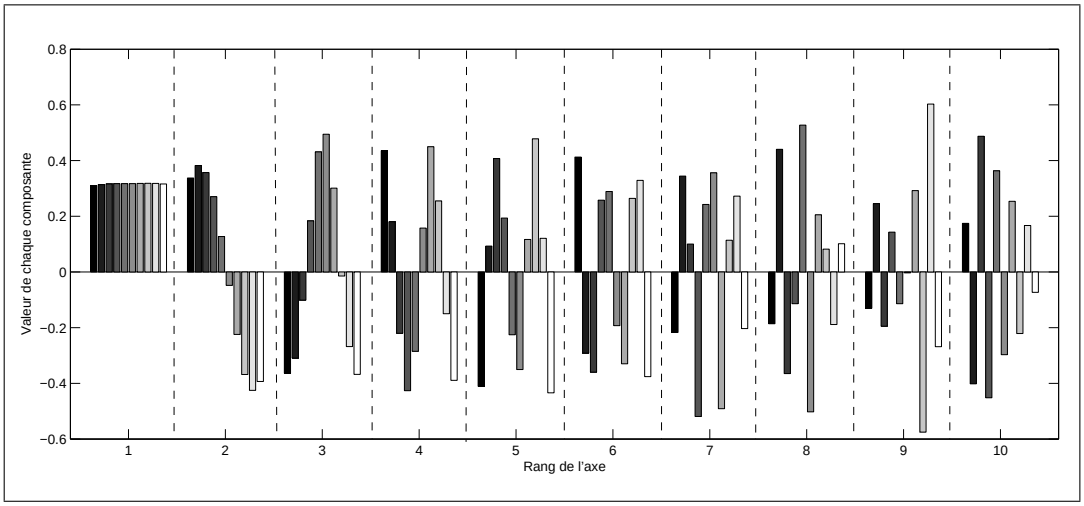
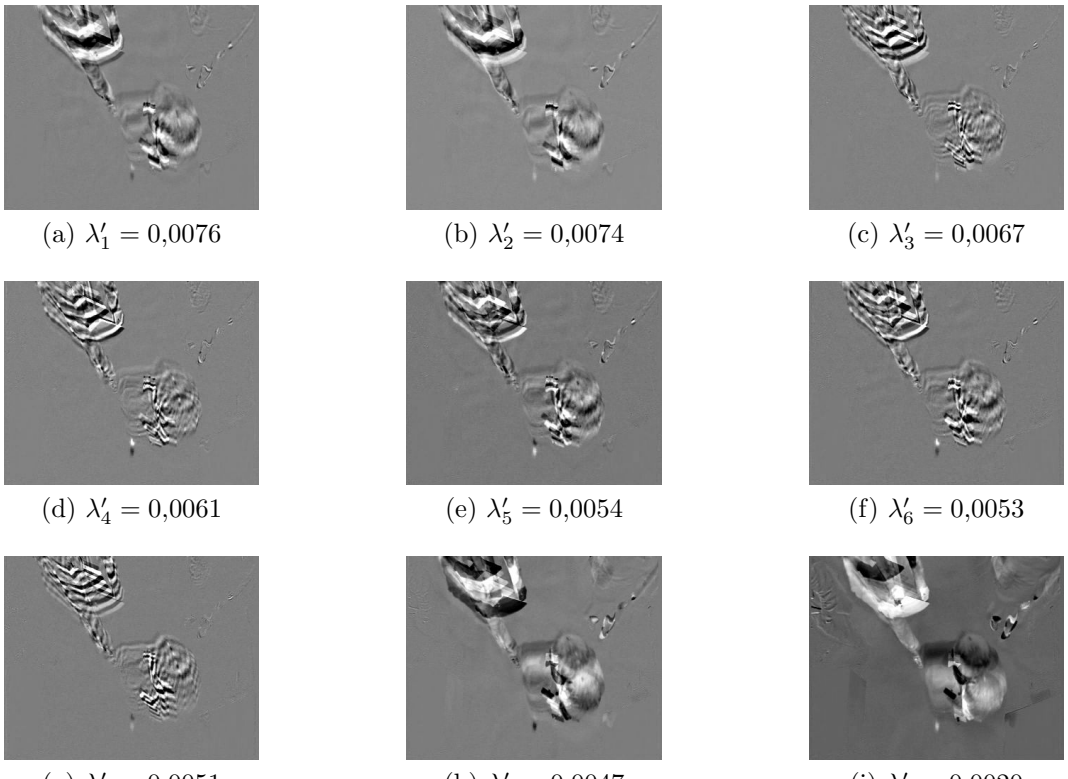


FIG. 4.7 – Composition des facteurs principaux quand les données sont exprimées avec l'ensemble $\mathcal{V}$.



(a) $\lambda'_1 = 0{,}0076$ (b) $\lambda'_2 = 0{,}0074$ (c) $\lambda'_3 = 0{,}0067$

(d) $\lambda'_4 = 0{,}0061$ (e) $\lambda'_5 = 0{,}0054$ (f) $\lambda'_6 = 0{,}0053$

(g) $\lambda'_7 = 0{,}0051$ (h) $\lambda'_8 = 0{,}0047$ (i) $\lambda'_9 = 0{,}0020$

FIG. 4.8 – Projections de $\mathbf{X}'$ sur chacun des axes mis en évidence par l'ACP et valeurs propres associées à chacun des facteurs.

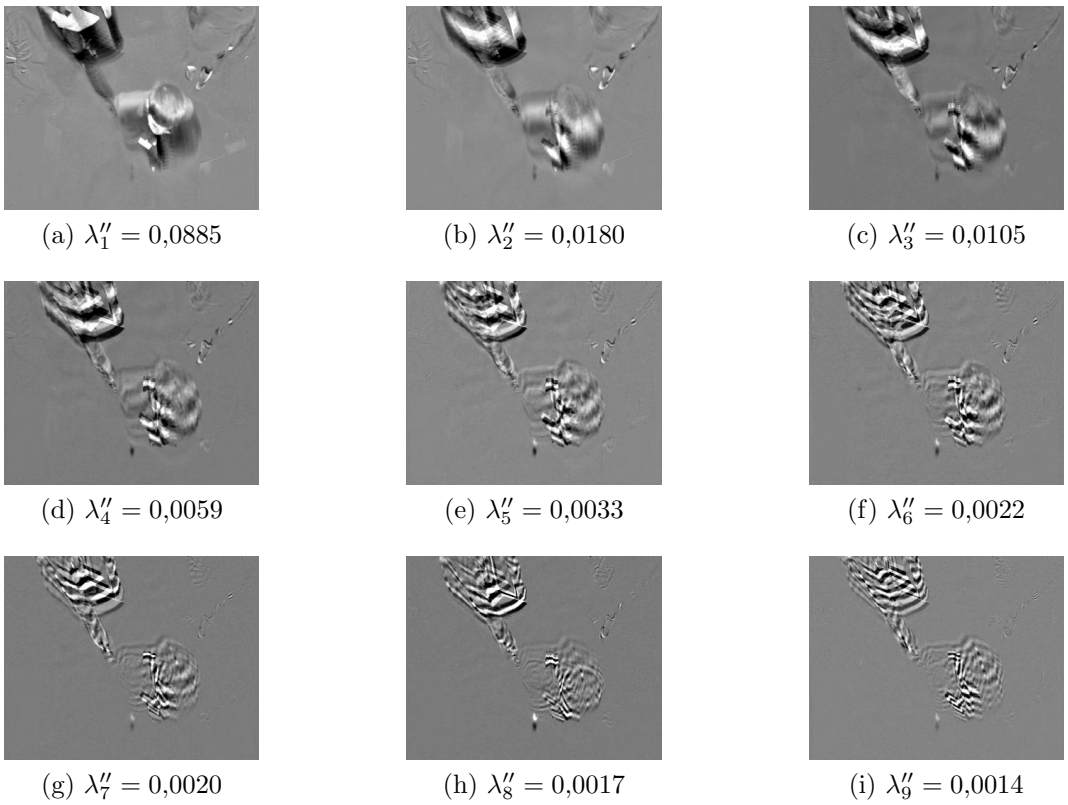
Comme nous l'avions prévu, dans ce cas, toutes les projections sur les axes principaux représentent des informations de mouvement, ce qui constitue un progrès par rapport aux résultats obtenus lorsque les données sont représentées

dans $\mathcal{V}$.

Cependant, en ce qui concerne notre problème, l'ordre des facteurs principaux n'est pas satisfaisant. En effet, lorsque les données sont représentées dans $\mathcal{V}$ (figure 4.6), si l'on ignore le premier facteur principal qui constitue une mesure de référence des valeurs des pixels, les facteurs suivants sont rangés dans l'ordre décroissant de la quantité de mouvement visible sur les données projetées. En revanche, lorsque les données sont exprimées dans $\tilde{\mathcal{V}}'$, l'ordre des facteurs principaux est sans rapport avec la quantité de mouvement perçue dans les projections associées, ce qui rend difficile le traitement automatique de ce type de résultat.

Comme les informations contenues dans $\tilde{\mathcal{V}}'$ et $\tilde{\mathcal{V}}''$ sont équivalentes, nous allons maintenant réaliser une ACP sur les données exprimées dans $\tilde{\mathcal{V}}''$ afin de vérifier si le problème précédent est toujours présent. Comme dans le cas précédent, la matrice de données $\mathbf{X}''$ comporte 576×720 lignes et 9 colonnes. Chacune de ses lignes est un vecteur $\tilde{\mathbf{u}}''$ tel que défini par l'équation 4.8. On notera $\mathbf{C}''$ sa matrice de covariance dont les valeurs propres sont $\{\lambda''_1, \ldots, \lambda''_9\}$ et les vecteurs propres associés $\{\mathbf{v}''_1, \ldots, \mathbf{v}''_9\}$.

La figure 4.9 montre les neuf projections de $\mathbf{X}''$ sur les axes principaux issus de l'ACP. Les projections sont représentées sous forme d'images en utilisant la formule de l'équation 4.14.



(a) $\lambda''_1 = 0{,}0885$ (b) $\lambda''_2 = 0{,}0180$ (c) $\lambda''_3 = 0{,}0105$

(d) $\lambda''_4 = 0{,}0059$ (e) $\lambda''_5 = 0{,}0033$ (f) $\lambda''_6 = 0{,}0022$

(g) $\lambda''_7 = 0{,}0020$ (h) $\lambda''_8 = 0{,}0017$ (i) $\lambda''_9 = 0{,}0014$

FIG. 4.9 – Projections de $\mathbf{X}''$ sur chacun des axes mis en évidence par l'ACP et valeurs propres associées à chacun des facteurs.

On constate que le fait de représenter les données dans $\tilde{\mathcal{V}}''$ permet de combiner les avantages des deux autres sytèmes de représentation :

- Les facteurs principaux sont rangés par ordre décroissant de la quantité de mouvement observable sur les données projetées sur l'axe correspondant. C'est un avantage que l'on avait obtenu en exprimant les données dans $\mathcal{V}$ mais pas dans $\tilde{\mathcal{V}}'$.
- Les données projetées sur les axes principaux traduisent uniquement des informations de mouvement, sans que les valeurs de référence des pixels ne viennent polluer la représentation. C'est un avantage que l'on avait obtenu en exprimant les données dans $\tilde{\mathcal{V}}'$ mais pas dans $\mathcal{V}$.

Pour cette raison, nous allons de préférence exprimer les données dans l'espace décrit par $\tilde{\mathcal{V}}''$ dans la suite de notre étude.

Maintenant que nous avons modifié l'espace de représentation des données de manière à ce que les informations porteuses de sens soient concentrées sur les premières coordonnées des individus traités, nous pouvons envisager de réduire la dimensionnalité de cet espace de manière automatique. Nous allons présenter dans la section suivante de quelle manière nous proposons d'y parvenir.

4.2.2 Réduction de dimension

D'après la Figure 4.9, les zones en mouvement apparaissent clairement lorsqu'on projette la matrice $\mathbf{X}''$ sur les premiers axes principaux. La différence entre une zone statique et une zone en mouvement est accentuée sur ces axes. On peut donc raisonnablement penser que l'on peut choisir de réduire la dimension de l'espace de représentation obtenu par l'ACP sans perdre trop d'information. Pour accomplir une réduction de dimension automatisée, le problème qui se pose est celui du choix du nombre de composantes principales que l'on doit garder. Il s'agit donc de trouver un nombre m ($m < p$) de composantes de manière à préserver au maximum la variance du nuage initial. Nous allons présenter dans cette section différentes manières de déterminer ce nombre.

La variance de la k-ème variable principale (variance des composantes de la projection des données sur le k-ème axe factoriel) est égale à la k-ème valeur propre de la matrice de covariance des données $\mathbf{C}$, et est notée λ_k. La figure 4.10 montre les variances des variables principales obtenues par ACP sur les ensembles $\mathcal{V}$ (a), $\tilde{\mathcal{V}}'$ (b) et $\tilde{\mathcal{V}}''$ (c).

Le critère le plus intuitif est certainement celui du pourcentage de variance cumulée. Nous pouvons par exemple décider que nous souhaitons que la variance des données projetées dans l'espace de dimension réduite présente une variance au moins égale à 80% de la variance des données. Le nombre de composantes nécessaires est donc la plus petite valeur de m telle que le pourcentage voulu soit atteint. Dans [Jolliffe, 2002], le pourcentage de variance cumulée

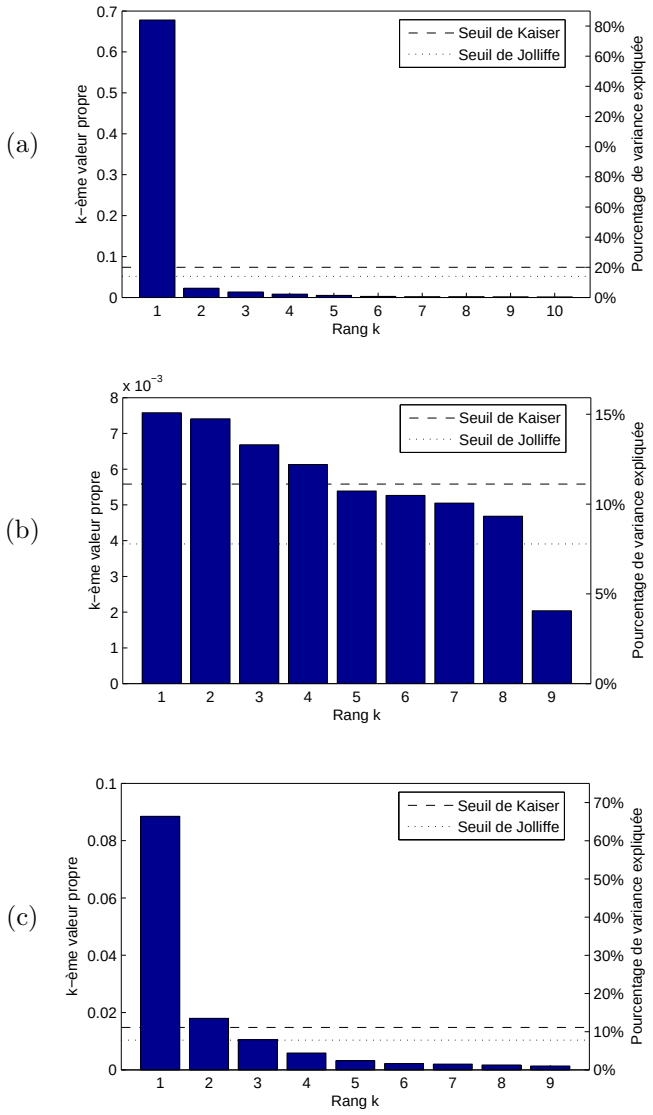


FIG. 4.10 – Histogrammes des valeurs propres obtenues en calculant une ACP lorsque les données sont exprimées dans $\mathcal{V}$ (a), $\tilde{\mathcal{V}}'$ (b) et $\tilde{\mathcal{V}}''$ (c).

obtenue avec les m premières composantes principales est noté t_m et défini par

$$t_m = 100 \frac{\sum_{k=1}^{m} \lambda_k}{\sum_{j=1}^{p} c_{jj}} \tag{4.15}$$

$$= 100 \frac{\sum_{k=1}^{m} \lambda_k}{\sum_{j=1}^{p} \lambda_j} \tag{4.16}$$

$$= \frac{100}{p} \sum_{k=1}^{m} \lambda_k, \tag{4.17}$$

où λ peut être remplacé par λ' ou λ'' si l'on exprime les données dans $\tilde{\mathcal{V}}'$ ou $\tilde{\mathcal{V}}''$. La méthode consiste donc à choisir un seuil t^* situé entre 70% et 90%

et à retenir m composantes où m est le plus petit entier tel que $t_m > t^*$. La détermination de la valeur idéale de t^* reste néanmoins empirique et dépend de plusieurs facteurs, tels que la dimension de l'espace de représentation original ou d'éventuelles spécificités du type de données étudiées.

La règle de Kaiser [Kaiser, 1960] est une possibilité pour automatiser le choix de m. Cette règle consiste à ne retenir que les composantes dont la valeur propre associée est supérieure à la moyenne des valeurs propres de la matrice de covariance. Dans [Jolliffe, 2002], il est observé que cette règle a tendance à sous-estimer le nombre de variables principales à retenir, ce qui est parfois préjudiciable. L'auteur propose de plutôt garder toutes les composantes k telles que

$$\lambda_k \geq \frac{0{,}7}{p} \sum_{j=1}^{p} \lambda_j. \tag{4.18}$$

Sur la figure 4.10 sont représentés les seuils de Kaiser et de Jolliffe calculés pour chaque espace de représentation des données.

Bien que basés sur les mêmes données, les trois résultats d'ACP sont très différents du point de vue de la variance expliquée par les différents facteurs principaux. La variance expliquée par un facteur est définie par le rapport entre la valeur propre associée à ce facteur, et la somme des valeurs propres de la matrice de covariance des données. Dans le cas où les données sont exprimées dans $\mathcal{V}$ (figure 4.10a), le premier facteur principal explique 92% de la variance totale du nuage initial. Que l'on décide de suivre le critère du pourcentage de variance cumulée, le seuil de Kaiser, ou celui de Jolliffe, on devrait considérer dans ce cas que la projection des données sur le premier axe principal suffit pour résumer toutes les données. Ceci s'explique par le fait que dans $\mathcal{V}$, les données sont exprimées en termes de valeur des pixels. Bien que des personnages se déplacent, la majorité de la scène reste immobile, donc une image moyenne constitue une estimation de bonne qualité de chacune des trames de la séquence. Les informations de mouvement qui nous intéressent sont ici considérées comme un bruit de faible amplitude.

Lorsqu'elles sont exprimées dans $\tilde{\mathcal{V}}'$ (figure 4.10b), les données traitées sont constituées de la dérivée temporelle de la séquence étudiée. Chaque échantillon fait apparaître les contours des objets mobiles à l'endroit où ils se trouvent à l'instant considéré (cf. figure 4.4a). Comme les personnages se déplacent à vitesse constante selon une trajectoire rectiligne, les seules données non nulles des vecteurs initiaux se situent à des emplacements différents à chaque instant. Dans ces conditions, l'ACP considère que les données sont complètement décorrélées. La conclusion qui s'impose au vu de la figure 4.10b est que les données ne sont presque pas compressibles par une ACP et qu'il faut conserver de nombreux facteurs principaux (4 à 8 selon le critère considéré) afin de ne pas perdre d'information.

Si les données sont exprimées dans $\tilde{\mathcal{V}}''$ (figure 4.10c), le mouvement apparaît de manière incrémentale sur les images traitées. Contrairement au cas précédent, la corrélation linéaire entre les différentes variables de l'espace de

représentation original est facile à observer. De ce fait, l'ACP va être capable de concentrer une grande partie de la variance du nuage sur les premiers axes principaux. Par ailleurs, cette variance concerne bien les informations de mouvement car la valeur de référence des pixels a été supprimée pour construire l'ensemble $\tilde{\mathcal{V}}''$. Les différents critères permettant de choisir le nombre de facteurs à conserver que nous avons présentés jusqu'à présent nous permettent de conclure que seulement deux facteurs principaux suffisent pour représenter les données sans perdre trop d'information. C'est-à-dire qu'avec uniquement 20% de la masse de données initialement présente, on parvient à préserver 80% de la variance originale. Ceci confirme ce que nous avions pressenti en étudiant les données projetées sur les différents axes principaux (cf. section 4.2.1), à savoir que le fait d'exprimer les données dans $\tilde{\mathcal{V}}''$ nous permet d'obtenir de meilleurs résultats qu'avec les autres systèmes de représentation proposés.

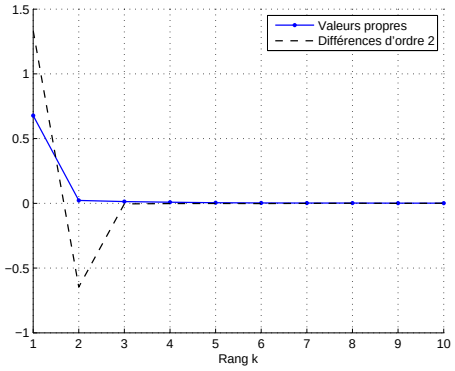
Les règles de décision que nous venons de présenter font appel à la subjectivité de la personne qui interprète les résultats. La méthode d'éboulis des valeurs propres (*scree graph*) est moins subjective puisqu'elle consiste à observer le graphe de décroissance des valeurs propres lorsque le rang de l'axe associé augmente, et à rechercher un « coude » dans ce graphe, c'est-à-dire un rang k à gauche duquel la pente de la courbe est importante et à droite duquel la pente est faible. Mathématiquement, il s'agit de détecter le premier changement de signe important dans la suite des différences d'ordre 2 entre valeurs propres consécutives (dérivée seconde discrète). La figure 4.11 montre les graphes d'éboulis des valeurs propres pour les trois systèmes de représentation des données. Les différences d'ordre 2 sont également représentées en pointillés. Celles-ci sont obtenues par un opérateur de convolution sur la suite des valeurs propres, soit

$$[\delta^2(\lambda_1) \quad \cdots \quad \delta^2(\lambda_p)] = [\lambda_1 \quad \cdots \quad \lambda_p] \otimes [-1 \quad 2 \quad -1]. \tag{4.19}$$

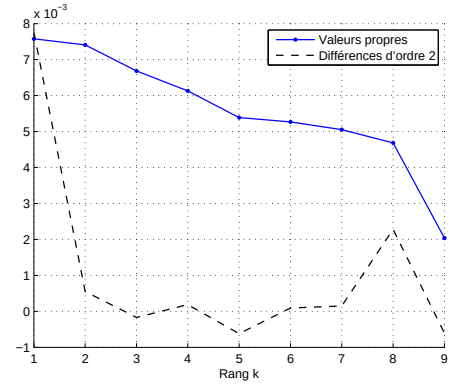
La figure 4.11a représente l'éboulis des valeurs propres quand les données sont exprimées dans $\mathcal{V}$. On observe un coude très net au niveau de la seconde valeur propre, qui est confirmé par un brusque changement de signe de la différence d'ordre 2. D'après ce critère, il faudrait conserver deux facteurs principaux pour compresser les données sans perte significative d'information, alors que les critères précédents conduisaient à n'en garder qu'un. Ceci s'explique par la définition même de l'éboulis des valeurs propres : comme on cherche rang k tel que la pente du graphe soit beaucoup plus importante à sa gauche qu'à sa droite, ce rang ne peut être ni le premier ni le dernier de la série.

En exprimant les données dans $\tilde{\mathcal{V}}'$, le graphe d'éboulis des valeurs propres obtenu est celui de la figure 4.11b. Celui-ci ne laisse apparaître aucun coude significatif, la pente du graphe varie peu. On notera néanmoins quelques changements de signe de la dérivée seconde, mais ceux-si sont de faible amplitude. Ce graphe confirme les observations précédentes : l'ACP n'est pas capable de trouver des axes de projection sur lesquels on observerait la majeure partie de la variance du nuage initial.

(a)



(b)



(c)

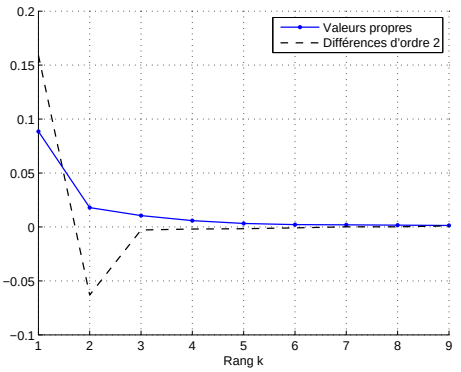


FIG. 4.11 – Éboulis des valeurs propres lorsque les données sont exprimées dans $\mathcal{V}$ (a), $\tilde{\mathcal{V}}'$ (b) et $\tilde{\mathcal{V}}''$ (c).

Pour des données exprimées dans $\tilde{\mathcal{V}}''$, le critère de l'éboulis des valeurs propres est en accord avec les précédents critères présentés. Dans ce cas, deux facteurs principaux suffisent pour représenter les données en minimisant la perte d'information due à la compression. Cet espace de représentation est donc bien adapté à la recherche de directions principales sur lesquelles se concentre la variance du nuage de données.

Dans [Besse, 1992], l'ACP est présentée comme le résultat de l'estimation d'un modèle. Ainsi, les auteurs définissent un critère de stabilité du sous-espace

de représentation par un risque moyen quadratique qui évalue la qualité de l'estimation. Ce risque est défini comme l'espérance d'une distance entre le « vrai » modèle et l'estimation qui en est faite. Celle-ci est obtenue par approximation de l'estimateur *jackknife* qui a pour expression

$$\hat{R}_m = \frac{1}{n-1} \sum_{k=1}^{m} \sum_{j=m+1}^{p} \frac{\frac{1}{n} \sum_{i=1}^{n} (\bar{\mathbf{x}}_i \mathbf{v}_k)^2 (\bar{\mathbf{x}}_i \mathbf{v}_j)^2}{(\lambda_j - \lambda_k)^2}. \tag{4.20}$$

La figure 4.12 montre la stabilité du sous-espace de représentation en fonction de la dimension m pour l'ACP des données exprimées dans chacun des ensembles $\mathcal{V}$ (a), $\tilde{\mathcal{V}}'$ (b) et $\tilde{\mathcal{V}}''$ (c).

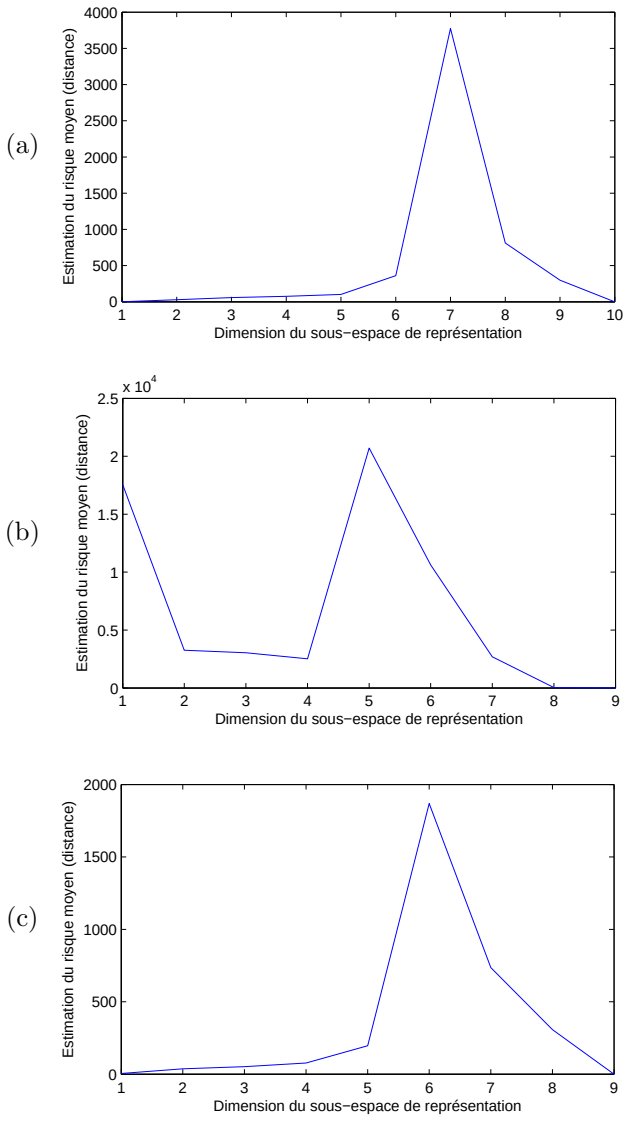


FIG. 4.12 – Stabilité des sous-espaces de représentation quand les données sont exprimées dans $\mathcal{V}$ (a), $\tilde{\mathcal{V}}'$ (b) et $\tilde{\mathcal{V}}''$ (c).

Lorque la valeur du risque estimé est faible (quand $m \leq 5$ pour une ACP

sur $\mathcal{V}$, ou quand $m \leq 4$ pour une ACP sur $\tilde{\mathcal{V}}''$), on sait que la représentation des données dans un sous-espace de dimenion m issu de l'ACP est fiable, c'est-à-dire qu'elle est stable par rapport aux fluctuations de l'échantillon. Lorsque cette même valeur est très élevée (quand $m = 7$ pour une ACP sur $\mathcal{V}$, quand $m < 8$ pour une ACP sur $\tilde{\mathcal{V}}'$, ou quand $m = 6$ pour une ACP sur $\tilde{\mathcal{V}}''$), cela signifie que les axes considérés sont très sensibles à toute perturbation des données, et sont très probablement associés à du bruit. Ce dernier critère confirme lui aussi que la représentation des données dans $\tilde{\mathcal{V}}'$ n'est pas adaptée à une étude par ACP, tandis que les deux autres systèmes fournissent des sous-espaces de représentation dont la fiabilité est satisfaisante.

L'étude de plusieurs méthodes d'aide à la décision pour le choix du nombre de composantes principales à conserver nous permet d'établir plusieurs conclusions. Tout d'abord, lorsque les données sont exprimées dans $\mathcal{V}$, la plupart des critères s'accordent sur le fait qu'un seul facteur permettrait de résumer toutes les données sans perte significative. Or, le premier facteur constitue une valeur de référence pour l'ensemble des pixels. Par conséquent, l'ensemble $\mathcal{V}$ n'est pas adapté à l'étude du mouvement par la recherche de directions principales, car les valeurs des pixels y sont beaucoup plus visibles que leurs variations d'apparence. Ensuite, on observe que l'ACP ne parvient pas à trouver un espace vectoriel dans lequel la variance des données est concentrée sur les premiers axes si l'espace de représentation original est $\tilde{\mathcal{V}}'$. Cela s'explique par le fait que la séquence vidéo, une fois dérivée par rapport au temps, constitue un ensemble de vecteurs décorrélés de sorte qu'une méthode d'analyse de données linéaire telle que l'ACP n'est pas adaptée à son étude. Selon tous les critères passés en revue, il faudrait dans ce cas conserver la quasi totalité des facteurs principaux pour minimiser la perte d'information. Enfin, lorsque les données sont exprimées dans $\tilde{\mathcal{V}}''$, aussi bien l'étude des projections sur les axes principaux (cf. section 4.2.1) que les différents critères pour le choix du nombre de dimensions à conserver, nous permettent de conclure que cet espace est adapté à l'étude du mouvement, et que l'ACP trouve facilement des combinaisons linéaires indépendantes des données ainsi représentées. C'est pourquoi, dans la suite de notre étude, nous choisirons de représenter les données vidéo par leur projection dans le premier plan factoriel issu d'une ACP calculée sur la séquence représentée par l'ensemble $\tilde{\mathcal{V}}''$.

4.3 Détection de zones de mouvement cohérent

Maintenant que nous avons défini de quelle manière on peut représenter les données dans un espace de dimension réduite tout en accentuant au mieux les informations relatives au mouvement, nous allons voir comment nous pouvons exploiter cette représentation afin de réaliser une segmentation des données entre l'arrière-plan de la scène et les objets mobiles. Dans un premier temps, nous proposerons une solution triviale utilisant un seul facteur principal, et

nous expliquerons pourquoi nous ne considérons pas cette solution comme satisfaisante. Nous présenterons ensuite l'intérêt d'exploiter la projection des données dans le premier plan factoriel dans un contexte semi-local.

4.3.1 Solution globale

La représentation des données telle que dans la figure 4.13a (projection des données vidéo sur le premier axe factoriel issu de l'ACP sur $\tilde{\mathcal{V}}''$) permet de facilement détecter les mouvements au niveau local (en permettant d'étiqueter chaque pixel). En effet, il suffit de sélectionner les pixels dont la valeur absolue est élevée (les plus sombres et les plus clairs) pour obtenir une segmentation objets mobiles/arrière-plan. La figure 4.13b représente la segmentation automatique de la projection de $\mathbf{X}''$ sur le premier axe principal. Cette segmentation automatique est obtenue par l'algorithme d'Otsu [Otsu, 1979] qui consiste à chercher un seuil qui minimise la variance intra-classe des pixels (ce qui revient à maximiser la variance inter-classes), où le terme « classe » désigne les deux ensembles constitués des pixels dont les niveaux de gris sont situés de part et d'autre du seuil en question. L'utilisation de cette méthode est justifiée dans notre situation puisqu'elle est basée sur l'hypothèse que l'histogramme des niveaux de gris est bimodal — or nous cherchons à définir deux classes de pixels : ceux qui repésentent l'arrière-plan de la scène et ceux qui représentent les objets mobiles.

(a) Projection de $\mathbf{X}''$sur le premier axe factoriel issu de l'ACP sur $\tilde{\mathcal{V}}''$.

(b) Segmentation automatique de la valeur absolue de (a).

(c) Segmentation améliorée par des opérations morphologiques.

FIG. 4.13 – Segmentation objets mobiles/arrière-plan obtenue à partir d'une seule composante principale.

Nous nous retrouvons alors dans le cas de la plupart des méthodes présentes dans la littérature, une telle image binaire serait étiquetée en composantes connexes pour obtenir une détection des objets mobiles. Comme dans la plupart des cas, dans l'exemple de la figure 4.13, un prétraitement de l'image serait nécessaire pour supprimer les faux positifs et pour rétablir la connexité des objets (c). Une telle approche fournit une segmentation précise, mais le choix des opérations morphologiques à effectuer est souvent délicat et difficilement automatisable. Une erreur dans le choix d'un élément structurant pourrait effacer un objet intéressant, connecter deux objets différents, valider

un faux positif, etc. Une phase d'apprentissage est nécessaire pour adapter la méthode générale au cas particulier de la séquence étudiée. Nous préférons donc éviter d'avoir à effectuer une telle étape. Notre objectif reste néanmoins de définir des zones connexes associées à un unique objet mobile.

Détecter un mouvement revient à détecter que le comportement d'une zone de l'image est différent du comportement principal observé. Notre vision globale actuelle de l'image ne semble donc pas adaptée à une telle perception. Cependant, notre première étape nous a permis de nous placer dans un espace de dimension réduite créé de manière à être le mieux adapté à la scène. Dans la section suivante, nous allons étudier comment la notion de voisinage spatial peut être exploitée pour détecter les mouvements dans l'espace de représentation des données présenté dans la section précédente.

4.3.2 Voisinage spatial vu dans l'espace sélectionné

Nous disposons d'un système de représentation de séquences vidéos courtes dans un espace de dimension réduite où les caractéristiques de mouvement sont bien visibles. Nous avons vu que le fait d'examiner les pixels individuellement ne permet d'obtenir une segmentation du plan-image entre arrière-plan et objets mobiles sans une étape délicate de réglage de paramètres. En fait, nous ne cherchons pas à étiqueter chaque pixel, mais plutôt chaque « région » du plan-image, puisque nous pouvons raisonnablement supposer qu'un objet mobile intéressant sera toujours représenté par un ensemble connexe de plusieurs pixels. Il existe différentes manières de découper une image en régions selon des critères de contours ou d'homogénéité ; le lecteur intéressé par cet aspect du problème pourra se référer à [Pham *et al.*, 2000] pour un panorama détaillé. Dans le cadre de notre problème, nous ne chercherons pas pour l'instant à optimiser la segmentation en régions. Ainsi, nous nous contenterons dans un premier temps de découper le plan image en zones carrées de mêmes tailles. La figureé 4.14 constitue un exemple d'un tel découpage.

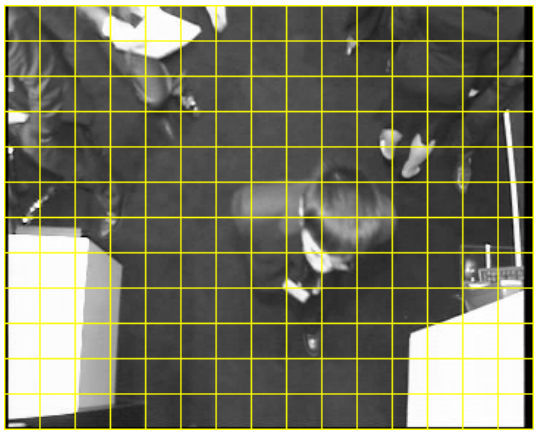


FIG. 4.14 – Découpage du plan-image en blocs carrés de 48 pixels de côté.

Le découpage du plan-image en blocs illustré par cette figure nous laisse

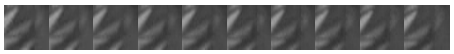
présentir que l'on pourra rencontrer différents types de mouvements selon le bloc considéré. Nous avons identifié quatre types de mouvements observables et nous allons tenter de voir de quelle manière ces phénomènes se traduisent dans l'espace de représentation que nous avons choisi.

- Le phénomène le plus simple est l'absence de mouvement. Pour l'étudier, nous allons observer une région correspondant à une partie de l'image qui reste statique pendant toute la durée de la séquence.
- La notion de bruit, bien connue dans l'analyse d'images statiques, existe également lorqu'on s'intéresse au mouvement. Il peut s'agir de bruit d'acquisition, que l'on observe généralement dans les parties sombres de l'image que les caméras numériques ont du mal à capturer. Le bruit peut également être dû à des mouvements réels mais non significatifs. En environnement extérieur, on pourra penser par exemple à l'oscillation de branches d'arbres sous l'effet du vent, ou encore à l'ondulation d'un plan d'eau. Dans le cas de notre exemple, on observera le personnage en haut à gauche de l'image qui ne se déplace pas mais qui n'est cependant pas totalement immobile.
- Lorsqu'un objet d'intérêt se déplace de manière significative, on s'intéressera à la manière dont sont représentées les régions qui se trouvent sur la surface de cet objet pendant la durée de la séquence étudiée. Ici, nous choisirons un bloc qui se trouve sur la trajectoire du personnage représenté au centre de l'image.
- Le dernier phénomène que nous avons identifié est celui des blocs situés sur les frontières spatio-temporelles des objets en mouvement, c'est-à-dire les régions qui, pendant un certain intervalle de temps, représentent uniquement l'arrière-plan de la scène, et à d'autres moments, représentent des objets en mouvement. Nous trouverons une telle région dans la séquence que nous étudions.

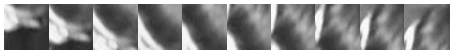
Ayant identifié ces quatre comportements types, nous choisissons dans la séquence étudiée, quatre régions du plan-image qui illustrent chacun d'eux. Il s'agit des blocs numérotés de 1 à 4 sur la figure 4.15. La figure 4.15a indique où se trouvent ces régions dans le plan-image. Le bloc 1 (b) correspond à une zone où le mouvement est presque imperceptible mais néanmoins présent ; il illustre le phénomène de bruit de mouvement. Le bloc 2 (c) représente le personnage du centre de la scène qui est en déplacement constant. Le bloc 3 (d) représente initialement l'arrière-plan de la scène, puis, après quelques intervalles de temps, un personnage en déplacement y apparaît. Ce bloc illustre le cas d'une frontière spatio-temporelle d'un objet mobile. Le bloc 4 (e) représente une zone statique de l'image dépourvue de toute forme de bruit. Il illustre une absence totale de mouvement.

Chacun des blocs présentés précédemment constitue une sous-population de vecteurs caractérisant un lieu du plan-image. Comme nous avons choisi de représenter les données dans $\tilde{\mathcal{V}}''$, nous pouvons projeter les vecteurs $\mathbf{u}''$ contenus dans chacun des blocs dans le premier plan factoriel issu de l'ACP calculée sur la globalité du plan-image. Les nuages de points bidimensionnels obtenus par

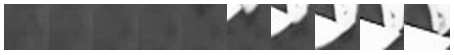
(a) Emplacement des quatre blocs étudiés sur la première image de la séquence.



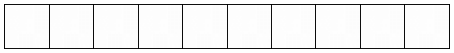
(b) Historique du bloc 1.



(c) Historique du bloc 2.



(d) Historique du bloc 3.



(e) Historique du bloc 4.

FIG. 4.15 – Historiques d'apparence de quatre régions du plan-image le long d'une séquence de dix trames.

projection sont représentés sur la figure 4.16.

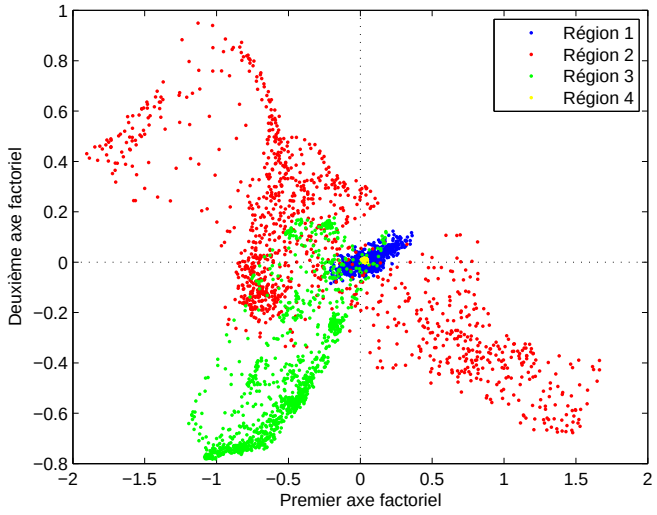


FIG. 4.16 – Projection des points des quatre régions étudiées dans le premier plan factoriel.

Les nuages de points obtenus par projection des différents types de régions étudiées dans le premier plan factoriel se différencient selon plusieurs aspects.

- La dispersion des points dans le plan factoriel varie selon la région considérée. Lorsque le mouvement est nul ou négligeable, comme dans les régions 1 et 4, les points obtenus se trouvent tous agglutinés dans une même zone du plan, à savoir, l'origine du repère. En revanche, lorsqu'un mouvement notable est visible dans la région considérée, comme pour les blocs 2 et 3, les points sont plus dispersés dans le plan.
- L'emplacement du centre du nuage dans le plan factoriel est également

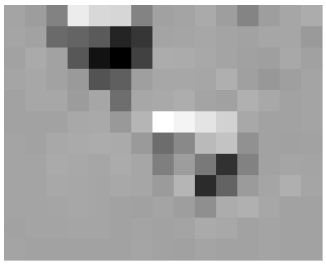
différent selon le bloc considéré. Parmi les quatre exemples étudiés, trois des nuages correspondants sont centrés dans le repère (régions 1, 2 et 4), tandis que le nuage obtenu par projection du bloc 3 se trouve excentré par rapport à l'origine du repère.
– Enfin, l'orientation principale du nuage peut également varier selon les régions. On remarquera en particulier que les directions principales des nuages correspondant aux blocs 2 et 3 sont pratiquement orthogonales dans le plan factoriel.

Ainsi, d'après la comparaison que l'on peut faire entre le mouvement perçu dans une région du plan-image et la projection des vecteurs $\mathbf{u}''$ correspondant dans le plan factoriel, il semblerait que l'on puisse qualifier le mouvement local grâce à certaines mesures statistiques classiques. En premier lieu, la « quantité » de mouvement perçu (son importance) est manifestement liée à la surface de la zone sur laquelle on rencontre les points obtenus par projection, qui est elle-même liée à la matrice de variance-covariance de l'ensemble de ces points. Ensuite, le centre de gravité du nuage est d'autant plus proche de l'origine du repère que le mouvement dans la région correspondante est uniforme. Le centre de gravité n'est autre que la moyenne des vecteurs 2D obtenus par projection. En ce qui concerne l'orientation principale du nuage obtenu, elle est également liée à la matrice de variance-covariance des points projetés, mais aucun lien entre cette caractéristique du nuage et le mouvement perceptible dans la région associée ne semble évident.

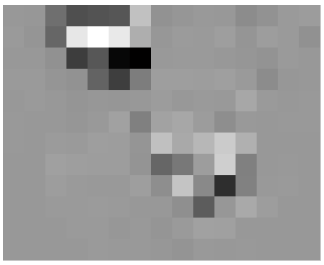
Nous pouvons donc caractériser le mouvement perçu dans une région à l'aide des statistiques d'ordre 1 et 2 du nuage de points 2D obtenus en projetant les vecteurs $\mathbf{u}''$ de cette région dans le plan factoriel. En effet, le moment du premier ordre est la moyenne du nuage, et il renseigne sur l'uniformité du mouvement pendant l'intervalle de temps considéré. Les moments du second ordre (moments d'inertie) constituent la matrice de variance-covariance des points projetés et renseignent sur l'importance du mouvement dans la région étudiée.

Avant de généraliser cette conclusion, nous allons observer la valeur de ces mesures pour l'ensemble des régions de la séquence étudiée. La figure 4.17 représente des mesures statistiques liées aux moments du premier ordre (sur la ligne du haut) et du second ordre (sur la ligne du bas). Ces mesures sont représentées sous forme d'image par normalisation des valeurs entre le minimum et le maximum des valeurs obtenues sur l'ensemble des régions considérées. La transformation permettant de passer d'une mesure statistique à un niveau de gris est similaire à celle décrite par l'équation 4.14.

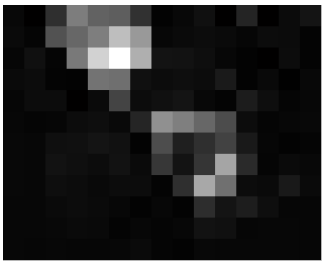
Pour illustrer les moments du premier ordre, nous avons choisi de représenter, pour chaque bloc, la moyenne des abscisses sur le premier axe factoriel (figure 4.17a), la moyenne des ordonnées sur le deuxième axe factoriel (b), et la distance du centre de gravité du nuage des points projetés (c), c'est-à-dire la norme euclidienne des deux premières mesures. Pour illustrer les moments du second ordre, nous avons représenté la variance des abscisses sur le premier axe factoriel (d), la variance des ordonnées sur le deuxième axe factoriel (e), et la



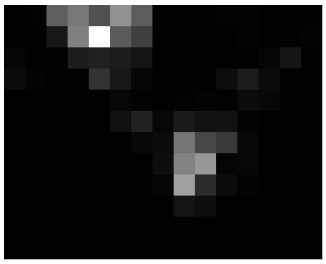
(a) Moyenne des abscisses sur le premier axe factoriel.



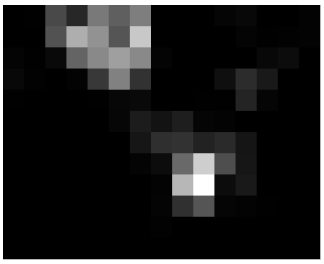
(b) Moyenne des ordonnées sur le deuxième axe factoriel.



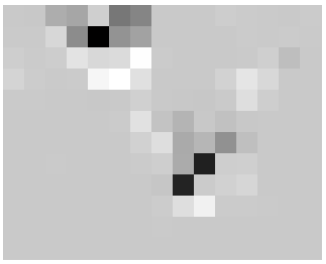
(c) Distance du centre de gravité du nuage au centre du repère.



(d) Variance des abscisses sur le premier axe factoriel.



(e) Variance des ordonnées sur le deuxième axe factoriel.



(f) Covariance des coordonnées sur les deux premiers axes factoriels.

FIG. 4.17 – Statistiques d'ordre 1 et 2 des projections de différentes régions du plan-image dans le premier plan factoriel.

covariance des valeurs des coordonnées sur les deux premiers axes factoriels (f).

Cette visualisation des moments d'inertie confirme bien que ceux-ci synthétisent les informations relatives au mouvement qui a lieu dans une région du plan-image. On constate que la valeur absolue de la moyenne des abscisses sur les deux premiers axes principaux est particulièrement élevée sur les contours (spatio-temporels) des objets en mouvement, tandis que la variance est plus élevée au centre des objets. C'est pourquoi nous pensons qu'il serait intéressant d'agréger ces deux types de mesures dans une seule représentation synthétique des nuages de points représentant les régions.

L'ellipse d'inertie d'un ensemble de points permet de résumer les moments géométriques des deux premiers ordres dans une représentation paramétrique légère et facile à visualiser. Une ellipse d'inertie est définie par un centre $\mathbf{g} = (g_1, g_2)$, un demi-grand axe a, un demi-petit axe b, et l'angle θ que forme le grand axe avec l'axe horizontal du repère utilisé. Le centre $\mathbf{g}$ de l'ellipse d'inertie est le centre de gravité des points. Si l'on note M_{11} la variance des abscisses sur l'axe horizontal, M_{22} la variance des ordonnées sur l'axe vertical, et M_{12} la covariance de ces deux coordonnées, les longueurs des demi-axes de

l'ellipse d'inertie sont définies par

$$a = \sqrt{s\lambda_1}, \tag{4.21}$$

$$b = \sqrt{s\lambda_2}, \tag{4.22}$$

où s est un scalaire contrôlant l'échelle de représentation de l'ellipse, et avec

$$\lambda_1 = \frac{M_{11} + M_{22} + \sqrt{(M_{11} - M_{22})^2 + 4M_{12}^2}}{2} \text{ et} \tag{4.23}$$

$$\lambda_2 = \frac{M_{11} + M_{22} - \sqrt{(M_{11} - M_{22})^2 + 4M_{12}^2}}{2}. \tag{4.24}$$

L'angle que forme le grand axe de l'ellipse avec l'axe horizontal du repère est défini par

$$\theta = \begin{cases} \theta_0 & \text{si } M_{11} < M_{22} \\ \theta_0 + \operatorname{sgn}(\theta_0)\dfrac{\pi}{2} & \text{sinon} \end{cases}, \text{ où } \theta_0 = \frac{1}{2}\arctan\frac{2M_{12}}{M_{22} - M_{11}}. \tag{4.25}$$

Ainsi, nous venons de définir une représentation concise des informations de mouvement présentes au sein d'une région spatio-temporelle du flux vidéo. Cela constitue un progrès par rapport aux représentations ponctuelles telles que présentées dans la section 4.3.1 car une *région* en mouvement est plus susceptible de représenter un objet qu'un *point* en mouvement. En se basant sur ce modèle semi-local de représentation du mouvement, nous allons entreprendre, dans la section suivante, de proposer une solution permettant de répondre au mieux à la question de la détection des *objets* en mouvement.

4.3.3 Solution semi-locale

Le fait de modéliser une région du plan-image par l'ellipse d'inertie des points obtenus par projection des vecteurs de $\tilde{\mathcal{V}}''$ associés à cette région nous permet de représenter les informations de mouvement de manière concise tout en prenant en compte le voisinage des pixels concernés. Nous avons vérifié sur un exemple de séquence que les moments d'ordre 1 et 2 révèlent les informations de mouvement des pixels. Ces mesures statistiques sont entièrement contenues dans la primitive simple que constitue une ellipse. Ainsi, une sous-séquence de 10 images telle que celle utilisée précédemment peut être représentée par un ensemble d'ellipses dans le premier plan factoriel issu de l'ACP globale calculée sur l'ensemble de la séquence. La Figure 4.18 montre un exemple d'un tel ensemble d'ellipses.

Les ellipses observées se différencient par leur position dans le plan, leur surface, et leur orientation. Dans la section précédente, nous avons observé que l'orientation des ellipses — c'est-à-dire la direction principale du nuage de points qu'elles modélisent — n'était pas directement associable à une notion de mouvement. En revanche, leur position dans le plan et leur surface sont caractéristiques du mouvement qui a lieu dans la région correspondante.

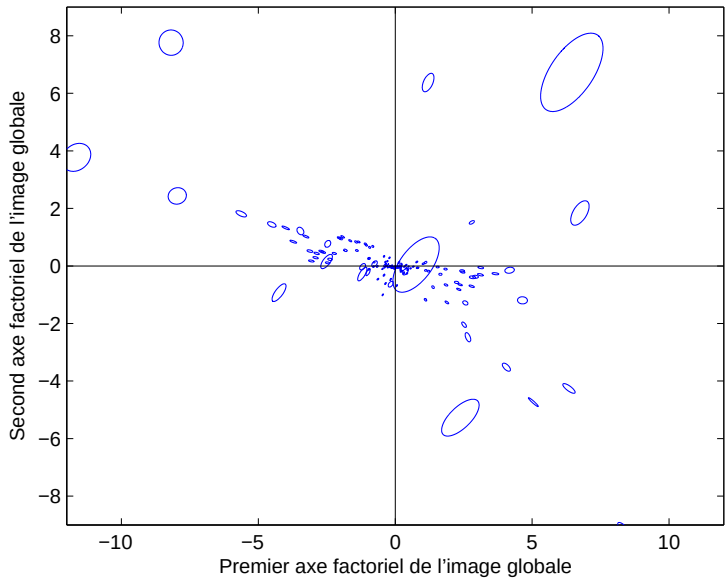


FIG. 4.18 – Chaque bloc tridimensionnel est modélisé par l'ellipse d'inertie des projections des vecteurs qui le composent, dans le premier plan factoriel de l'image globale.

La position d'une ellipse dans le plan factoriel est liée au centre de gravité du nuage qu'elle modélise. Nous avons constaté qu'un nuage excentré par rapport à l'origine du repère révélait une région dans laquelle le mouvement n'est pas constant le long de l'intervalle de temps considéré. Par conséquent, une ellipse qui se trouve éloignée de l'origine du reprère pourra être interprétée comme une région dans laquelle un objet mobile apparaît ou disparaît.

L'aire de la surface d'une ellipse de demi-grand axe de longueur a et de demi-petit axe de longueur b est égale à πab. Or, d'après les équations 4.21 à 4.24, ces deux longueurs sont fonction des moments d'inertie d'ordre 2 du nuage associé. Par ailleurs, nous avions observé qu'un nuage présentant une dispersion importante révélait une grande quantité de mouvement perçu dans la région correspondante. De ce fait, une ellipse dont la surface est grande pourra être interprétée comme une région dans laquelle un mouvement important a eu lieu.

Pour détecter à la fois le contour et l'intérieur des objets en mouvement, nous pouvons donc commencer par sélectionner les ellipses dont le centre se trouve au-delà d'une certaine distance de l'origine du repère et celles dont la surface excède un certain seuil. Autrement dit, nous voulons observer toutes les ellipses de centre $\mathbf{g} = (g_1, g_2)$, de demi-grand axe a et de demi-petit axe b telles que

$$(\sqrt{g_1^2 + g_2^2} > \alpha) \vee (ab > \beta), \tag{4.26}$$

où α et β sont deux seuils à définir.

En reprenant l'exemple de séquence utilisé précédemment, en fixant le facteur d'échelle de représentation des ellipses à 6 (variable s dans les équations 4.21 et 4.22), et en choisissant comme valeurs de seuil $\alpha = 0{,}1$ et $\beta = \frac{3}{2}$, le critère défini par l'équation 4.26 nous amène à sélectionner les zones représentées en blanc sur la figure 4.19.

(a) Régions sélectionnées par le critère $\sqrt{g_1^2 + g_2^2} > 0{,}1$.

(b) Régions sélectionnées par le critère $ab > \frac{3}{2}$.

(c) Régions sélectionnées par la disjonction des deux critères précédents.

FIG. 4.19 – Régions sélectionnées par un critère basé sur les paramètres des ellipses d'inertie qui les modélisent.

Cette expérimentation confirme que les deux critères de l'équation 4.26 sont nécessaires pour détecter les zones en mouvement dans leur intégralité. Bien que cohérente avec le nombre d'objets mobiles présents dans la scène, la segmentation obtenue manque de précision du fait du découpage de l'image en régions carrées. Nous devons donc envisager de modifier la méthode de manière à trouver un compromis acceptable entre précision des contours, temps de calcul et généricité.

Au vu de la figure 4.19, le fait de considérer l'ensemble d'un voisinage spatial pour détecter le mouvement en un lieu de l'image sembe fournir de bons résultats. Cependant, comme toutes les régions définies sont mutuellement exclusives, le résultat de la segmentation est nécessairement imprécis et produit l'effet crénelé que nous pouvons observer.

Une possibilité pour éliminer cet effet indésirable serait de considérer une fenêtre glissante de la même taille que celles utilisées précédemment, que l'on centrerait en tout point du plan-image. La figure 4.20 présente le résultat obtenu par cette méthode. Comme on peut le constater, le résultat obtenu est beaucoup plus lisse et permet d'obtenir une segmentation plus précise. Malheureusement, si l'on note n le nombre de pixels du plan-image et w la largeur des régions considérées, une telle méthode nous amène à traiter nw^2 pixels, ce qui implique des temps de calcul inacceptables dans le contexte d'une application temps réel.

C'est pourquoi nous proposons une solution intermédiaire consistant à définir des régions recouvrantes par moitié. De cette manière chaque pixel — hormis ceux situés à moins de $\frac{w}{2}$ pixels d'un bord de l'image — intervient dans quatre régions de l'image segmentée. Afin de se rapprocher de l'aspect lisse des

(a) Distance du centre des ellipses à l'origine du repère.

(b) Aire des ellipses.

(c) Union des deux images précedentes binarisées automatiquement.

FIG. 4.20 – Segmentation obtenue par la même méthode que pour la figure 4.19, mais en utilisant cette fois une fenêtre glissante.

résultats de la figure 4.20 ainsi que d'améliorer la précision de la segmentation obtenue, nous proposons une méthode d'approximation.

Il s'agit de considérer que les mesures statistiques utilisées lors du seuillage (surface et norme du centre de l'ellipse d'inertie) sont affectées au pixel central de la région pour laquelle elles ont été calculées. Nous avons ainsi un grand nombre de valeurs manquantes que nous pouvons interpoler à partir des valeurs calculées. La méthode d'interpolation la plus précise parmi celles qui sont fréquemment utilisées dans le domaine du traitement d'images est l'interpolation bicubique.

Si l'on note $f : (x, y) \mapsto f(x, y)$ la fonction (inconnue) qui associe une valeur de niveau de gris aux coordonnées d'un pixel, le problème de l'interpolation consiste à considérer que l'on dispose des valeurs de f uniquement pour un ensemble $\mathcal{R}$ de $n_x^{\text{ref}} \times n_y^{\text{ref}}$ pixels de référence dont les abscisses seront notées $\{x_i^{\text{ref}}\}_{i=1}^{n_x^{\text{ref}}}$ et les ordonnées $\{y_i^{\text{ref}}\}_{i=1}^{n_y^{\text{ref}}}$. Le but est de déterminer la valeur de niveau de gris prise par f en un point (x, y) n'appartenant pas à $\mathcal{R}$. Considérons tout d'abord les quatre pixels de référence qui forment le plus petit rectangle contenant le point (x, y). Nous les noterons $\mathbf{p}_1 = (x_j^{\text{ref}}, y_k^{\text{ref}})$, $\mathbf{p}_2 = (x_{j+1}^{\text{ref}}, y_k^{\text{ref}})$, $\mathbf{p}_3 = (x_{j+1}^{\text{ref}}, y_{k+1}^{\text{ref}})$ et $\mathbf{p}_4 = (x_j^{\text{ref}}, y_{k+1}^{\text{ref}})$ avec

$$x_j^{\text{ref}} < x < x_{j+1}^{\text{ref}} \quad \text{et} \tag{4.27}$$

$$y_k^{\text{ref}} < y < y_{k+1}^{\text{ref}}. \tag{4.28}$$

Ils délimitent ce que l'on appellera le « bloc » dans lequel se trouve (x, y).

La technique d'interpolation la plus simple est l'interpolation bilinéaire. Elle consiste à calculer deux variables t et u de l'intervalle $[0, 1]$ définies par

$$t = \frac{x - x_j^{\text{ref}}}{x_{j+1}^{\text{ref}} - x_j^{\text{ref}}} \quad \text{et} \tag{4.29}$$

$$u = \frac{y - y_k^{\text{ref}}}{y_{k+1}^{\text{ref}} - y_k^{\text{ref}}}. \tag{4.30}$$

La valeur de f interpolée au point (x, y) est alors

$$f(x,y) = (1-t)(1-u)f(\mathbf{p}_1)+t(1-u)f(\mathbf{p}_2)+t\,u\,f(\mathbf{p}_3)+(1-t)u\,f(\mathbf{p}_4). \quad (4.31)$$

L'inconvénient de cette méthode est qu'à la frontière de deux blocs, la fonction f n'est pas forcément différenciable, ce qui risque de reproduire l'effet crénelé que nous cherchons à supprimer. L'interpolation bicubique permet de réduire cet effet. Dans ce cas, pour tous les pixels de référence, il faut non seulement connaître la valeur prise par la fonction f, mais également la valeur de son gradient $\overrightarrow{\nabla f} = (\partial f/\partial x, \partial f/\partial y)$ et celle de la dérivée seconde croisée $\partial^2 f/\partial x \partial y$. Les valeurs de ces dérivées peuvent être estimées par le calcul des dérivées partielles discrètes de la fonction f aux pixels de référence, soit

$$\frac{\partial f}{\partial x}(x_j^{\text{ref}}, y_k^{\text{ref}}) \approx \frac{f(x_{j+1}^{\text{ref}}, y_k^{\text{ref}}) - f(x_{j-1}^{\text{ref}}, y_k^{\text{ref}})}{x_{j+1}^{\text{ref}} - x_{j-1}^{\text{ref}}}, \quad (4.32)$$

$$\frac{\partial f}{\partial y}(x_j^{\text{ref}}, y_k^{\text{ref}}) \approx \frac{f(x_j^{\text{ref}}, y_{k+1}^{\text{ref}}) - f(x_j^{\text{ref}}, y_{k-1}^{\text{ref}})}{y_{k+1}^{\text{ref}} - y_{k-1}^{\text{ref}}}, \quad (4.33)$$

$$\frac{\partial^2 f}{\partial x \partial y}(x_j^{\text{ref}}, y_k^{\text{ref}}) \approx \frac{f(x_{j+1}^{\text{ref}}, y_{k+1}^{\text{ref}}) - f(x_{j+1}^{\text{ref}}, y_{k-1}^{\text{ref}}) - f(x_{j-1}^{\text{ref}}, y_{k+1}^{\text{ref}}) + f(x_{j-1}^{\text{ref}}, y_{k-1}^{\text{ref}})}{(x_{j+1}^{\text{ref}} - x_{j-1}^{\text{ref}})(y_{k+1}^{\text{ref}} - y_{k-1}^{\text{ref}})}. \quad (4.34)$$

Les valeurs de f et de ses dérivées peuvent ensuite être interpolées de la manière suivante :

$$f(x,y) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij} t^{i-1} u^{j-1}, \quad (4.35)$$

$$\frac{\partial f}{\partial x}(x,y) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij} (i-1) t^{i-2} u^{j-1}, \quad (4.36)$$

$$\frac{\partial f}{\partial y}(x,y) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij} (j-1) t^{i-1} u^{j-2}, \quad (4.37)$$

$$\frac{\partial^2 f}{\partial x \partial y}(x,y) = \sum_{i=1}^{4}\sum_{j=1}^{4} c_{ij} (i-1)(j-1) t^{i-2} u^{j-2}, \quad (4.38)$$

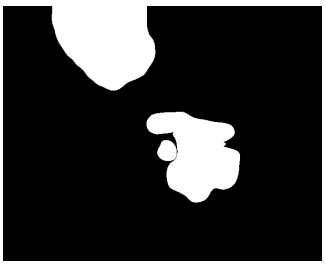
où t et u sont donnés par les équations 4.29 et 4.30, et où les 16 coefficients c_{ij} sont obtenus par résolution d'un système linéaire de 16 équations basées sur les valeurs de f et de ses dérivées aux quatre coins de la case dans laquelle se trouve le point (x, y). Les détails de la méthode de calcul des c_{ij} peuvent être trouvés dans [Kincaid et Cheney, 2001].

Grâce à cette méthode, nous parvenons à obtenir une segmentation présentant des contours très proches de ceux que nous avions eus en utilisant une

fenêtre glissante (figure 4.20). La figure 4.21 présente les résultats obtenus en choisissant des régions de même taille que précédemment, mais qui sont cette fois recouvrantes par moitié, ainsi que la version lissée obtenue grâce à une interpolation bicubique des mesures statistiques.

(a) Distance du centre des ellipses à l'origine du repère.

(b) Aire des ellipses.



(c) Union des deux images précedentes binarisées automatiquement.

FIG. 4.21 – En utilisant des régions recouvrantes par moitié et l'interpolation bicubique, on obtient des résultats très proches de ceux de la figure 4.20 pour un temps de calcul nettement inférieur.

Nous venons donc de définir une méthode qui permet d'obtenir une segmentation lisse des objets en mouvement, en prenant en compte un large voisinage spatial des points qui interviennent, et qui ne nécessite pas de temps de calcul trop importants. En effet, le rapport entre le temps de calcul nécessaire pour obtenir le résultat de la figure 4.20 et celui de la figure 4.21 est de l'ordre de 10^4.

Bien que le nombre de paramètres de l'algoritme soit peu important, ceux-ci nécessitent néanmoins d'être définis. Dans la section suivante, nous allons étudier l'influence de chacun de ces paramètres sur la qualité de la segmentation obtenue et sur les temps de calcul nécessaires.

4.4 Expérimentation

La méthode de détection de mouvement que nous venons d'exposer présente plusieurs avantages par rapport aux autres méthodes décrites dans la littérature, que nous avons passées en revue dans le chapitre 2. En premier lieu, notre méthode nécessite peu de paramétrage. En effet, seuls quatre paramètres sont à définir :

- la longueur de la séquence sur laquelle l'ACP doit être calculée, c'est-à-dire la durée d'observation de la scène prise en compte pour obtenir une segmentation ;
- la taille des régions du plan-image qui permettent de définir des blocs spatio-temporels au sein desquels les mesures statistiques seront relevées ;
- le seuil relatif à la surface des ellipses d'inertie qui modélisent les régions, permettant d'établir une segmentation des objets mobiles ;

– le seuil relatif à la distance de l'origine du repère aux centres des mêmes ellipses.

Aussi, nous allons étudier pour chacun de ces paramètres, leur influence sur la qualité des résultats et sur les temps de calcul nécessaires. Nous déterminerons également des règles de décision permettant à l'utilisateur final de fixer ces paramètres en fonction de l'application utilisant les résultats de la détection de mouvement.

Le second avantage de notre méthode est que la détection des objets mobiles se fait dans un espace de représentation qui est continuellement adapté au contenu des données à analyser. Par ailleurs, nous avons vu au chapitre 2 que les méthodes classiques de détection de mouvement réalisent l'analyse des données vidéo à une échelle qui est soit ponctuelle, soit semi-locale, soit globale. Par opposition, notre méthode utilise les données ponctuelles pour bâtir un espace de représentation global qui est ensuite exploité au niveau semi-local. Pour vérifier que ces caractéristiques conduisent à de meilleurs résultats que les méthodes plus classiques, nous avons choisi un ensemble de séquences vidéo présentant des difficultés variées afin de réaliser une étude comparative.

Mais préalablement à ces expérimentations, il est nécessaire de définir la méthodologie que nous allons mettre en œuvre pour étudier le paramétrage, ainsi que les métriques que nous utiliserons dans le cadre de notre étude comparative.

4.4.1 Méthodologie et métriques

Pour chacun des paramètres de l'algorithme, nous allons étudier son influence sur la qualité des résultats. Il faut donc définir précisément ce que l'on entend par « qualité ». La recherche de méthodologies pour l'évaluation des algorithmes d'analyse de séquences vidéo, et en particulier des algorithmes de détection de mouvement, est un domaine très actif depuis le début des années 2000 [Nascimento et Marques, 2004; Bashir et Porikli, 2006; Lazarevic-McManus *et al.*, 2006].

Un algorithme de détection de mouvement peut être vu comme un classifieur binaire. Typiquement, les méthodes d'évaluation proposées nécessitent de posséder une « réalité terrain » (*ground truth*), c'est-à-dire une version de la séquence vidéo pour laquelle chaque pixel aura été manuellement étiqueté par un expert comme appartenant à l'une des deux classes, que nous nommerons « fond » et « objet ». La comparaison des résultats de l'algorithme utilisé avec la réalité terrain fournit quatre mesures :

– Les vrais positifs (VP) sont les entités étiquetées comme « objet » par l'algorithme et par l'expert.
– Les vrais négatifs (VN) sont les entités étiquetées comme « fond » par l'algorithme et par l'expert.
– Les faux positifs (FP), ou fausses alarmes, sont les entités étiquetées comme « objet » par l'algorithme mais comme « fond » par l'expert.
– Les faux négatifs (FN), ou détections manquantes, sont les entités éti-

quetées comme « fond » par l'algorithme mais comme « objet » par l'expert.

Ces quatre mesures sont généralement consignées dans un tableau de contingence analogue au tableau 4.1.

Classification automatique	Classification manuelle	
	Objet	Fond
Objet	VP	FP
Fond	FN	VN

TAB. 4.1 – Tableau de contingence utilisé pour comparer une détection automatique à une réalité terrain.

Parmi les métriques les plus utilisées, on citera le rappel (ou taux de détection) ρ et la précision ν. Ces deux valeurs peuvent être synthétisées dans une seule mesure de performance globale communément utilisée, la F-mesure F. Ces métriques sont définies par

$$\rho = \frac{\#VP}{\#VP + \#FN} \tag{4.39}$$

$$\nu = \frac{\#VP}{\#VP + \#FP} \tag{4.40}$$

$$F = \frac{2 \times \rho \times \nu}{\rho + \nu}. \tag{4.41}$$

La plupart du temps, les entités comparées sont des pixels. L'adéquation entre les nombres d'objets ou de composantes connexes identifiés par l'algorithme et par l'expert n'est pas mesurée, mais le processus a le mérite d'être totalement objectif.

Parfois, comme dans [Nascimento et Marques, 2004], la comparaison se fait au niveau des composantes connexes trouvées par la segmentation automatique et par l'expert. Cette approche permet d'introduire de nouvelles mesures comme le nombre de fusions indésirables ou le nombre de divisions d'objets uniques. Néanmoins, cela nécessite de paramétrer le processus d'évaluation en fixant par exemple le taux de recouvrement minimum pour considérer que deux régions coïncident. En outre, cette méthode introduit un certain degré de subjectivité car lorsque deux objets se déplacent côte à côte, l'expert peut avoir tendance à forcer les deux régions correspondantes à ne pas être connexes alors que tout détecteur de mouvement considérerait une seule région en mouvement. Pour ces raisons, nous avons choisi d'effectuer l'évaluation de notre méthode en prenant comme unité graphique de comparaison le pixel.

En ce qui concerne l'étude des temps de calcul, ceux-ci sont forcément liés à la vitesse des micro-processeurs utilisés, au langage de programmation choisi, ainsi qu'à la dimension des trames des séquences vidéo étudiées. Aussi, quand nous comparerons plusieurs paramétrages de l'algorithme, nous exprimerons

les temps de calcul en pourcentage du temps requis pour exécuter l'algorithme dans sa configuration considérée comme optimale.

4.4.2 Durée d'observation

La longueur de la séquence élémentaire analysée a évidemment une grande influence sur les résultats de la segmentation. On peut comparer ce paramètre à la durée d'ouverture du diaphragme d'un appareil photographique : un temps de pose élevé permet de capturer plus d'information (lumière), mais peut produire un effet de persistance visuelle des objets en mouvement. Dans notre cas, un temps d'observation trop long risque de produire un effet « fantôme » tel qu'il a été décrit et analysé dans [Cucchiara *et al.*, 2003], ce qui peut empêcher de localiser précisément les objets mobiles tout en augmentant le risque de fusion non désirée entre régions détectées.

À l'inverse, un temps d'observation trop court aurait pour effet d'amoindrir la portée du processus de construction d'un espace de représentation adaptée à la séquence, car les informations de mouvement y seraient moins visibles. Le risque est de ne pas détecter correctement certains mouvements, et la connexité des régions détectées pourrait en pâtir.

Ce paramètre doit être choisi en fonction de la vitesse de déplacement des objets attendus dans la scène filmée. Pour illustrer le processus de choix de ce paramètre, nous allons utiliser des séquences mettant en scène des objets de vitesses différentes. La première séquence présentée est celle qui a servi de support à la description de la méthode dans les deux sections précédentes. Elle met en scène des personnes vues de dessus dont la vitesse de déplacement dans le repère lié à l'image varie entre 5 et 10 pixels par trame. La deuxième séquence utilisée présente la particularité de mettre en scène deux types d'objets mobiles : des personnes et des véhicules, les seconds se déplaçant bien sûr beaucoup plus rapidement que les premiers (7 pixels par trame contre 0,9). Ce support permettra d'illustrer le fait que la longueur de la séquence doit être choisie en fonction des objets que l'on souhaite détecter.

Nous allons étudier l'influence de ce paramètre à deux niveaux d'analyse. L'observation des résultats obtenus sur des séquences de test constitue une analyse qualitative, tandis que le calcul des métriques présentées précédemment constitue une analyse quantitative.

Analyse qualitative

La figure 4.22 présente les segmentations obtenues pour des séquences de 3, 5, 11 et 21 trames. Les seuils utilisés pour obtenir la binarisation finale ont été choisis automatiquement de manière à minimiser la variance intra-classe pour les deux classes « fond » et « objet » [Otsu, 1979]. Nous avons conservé la même taille de régions du plan-image que dans la section 4.3.

On constate que pour des durées d'observations courtes ($\tau = 3$ et $\tau = 5$), les objets détectés sont fragmentés et les régions correspondantes sont trop petites

(a) Séquence d'entrée. (b) Vérité terrain. (c) $\tau = 3$

(d) $\tau = 5$ (e) $\tau = 11$ (f) $\tau = 21$

FIG. 4.22 – Segmentations obtenues sur une séquence simple avec différentes durées d'observation.

par rapport à la vérité terrain. Pour une durée élémentaire de 21 images, l'effet fantôme évoqué plus haut est trop important pour que la détection soit acceptable, ce qui implique que la forme des régions obtenues ne correspond plus à la forme des objets dans la vidéo analysée. Pour cette séquence, le meilleur résultat est obtenu avec une durée d'observation de 11 images (figure 4.22e). On remarquera que, pour une durée d'observation supérieure ou égale à 11, le personnage en haut à droite est détecté comme ayant bougé alors qu'il n'était pas étiqueté comme tel sur la vérité terrain (figure 4.22b). Cela souligne la subjectivité du processus d'étiquetage de la vérité terrain et devra nous amener à relativiser les résultats de l'analyse quantitative.

La figure 4.23 représente les résultats obtenus en appliquant le même processus de test que précédemment, mais sur une séquence qui, cette fois, met en scène deux familles d'objets dont les vitesses moyennes sont très différentes : des personnes à pied et des véhicules sur une voie rapide.

Comme dans le cas précédent, nous observons que la connexité des régions détectées s'améliore lorsque la durée d'observation augmente, sauf dans le cas où τ vaut 21 (figure 4.23f) pour lequel la segmentation automatique nous fait perdre la voiture située dans la partie la plus inférieure de l'image. Les personnes présentes dans la scène se déplaçant lentement, elles ne sont presque pas détectées lorsque l'on considère des séquences élémentaires de 3 images (figure 4.23c). Par contre, à partir de 5 images, les régions correspondant aux véhicules les plus proches de l'horizon commencent à être interconnectées.

Cette analyse qualitative des résultats nous a donc appris que la vitesse des objets que l'on souhaite détecter aura une grande influence dans le choix du

(a) Séquence d'entrée. (b) Vérité terrain. (c) $\tau = 3$

(d) $\tau = 5$ (e) $\tau = 11$ (f) $\tau = 21$

FIG. 4.23 – Influence de la durée de la séquence élémentaire sur l'extraction des zones en mouvement.

paramètre τ.

Analyse quantitative

Nous avons calculé les métriques proposées dans la section 4.4.1 pour chacun des tests que nous venons de présenter. En ce qui concerne la deuxième séquence vidéo, nous distinguerons deux cas :

- **Scénario 1 :** L'expert a créé une réalité terrain dans l'optique de réaliser une application qui détecte tous les objets mobiles. Dans ce cas, les personnes et les véhicules sont étiquetés comme « objets ».
- **Scénario 2 :** L'application cible est un système de suivi de véhicules : seuls les véhicules sont étiquetés comme « objet » dans la réalité terrain, les personnes étant étiquetées comme « fond ».

C'est pourquoi nous avons utilisé deux couleurs pour désigner les régions de la vérité terrain de la figure 4.23b (blanc et gris). Les régions blanches sont étiquetées comme « objets » dans les deux cas de figure, et les régions grises ne sont étiquetées « objets » que dans le cas du scénario 1.

Le tableau 4.2 rassemble les valeurs du rappel, de la précision et de la F-mesure obtenus dans chacun des cas. On indique également pour chaque séquence et chaque scénario, la durée d'observation idéale τ^* au sens de la F-mesure.

Comme nous l'avions supposé en observant les résultats obtenus pour la séquence 1, la durée d'observation optimale au sens de la F-mesure est de 11 images. On constate que globalement, lorsque la durée d'observation aug-

Séquence	τ	ρ	ν	F	τ^*
Séquence 1	3	0,1621	1,0000	0,2789	11
	5	0,4656	0,9009	0,6139	
	11	0,9150	0,6341	**0,7491**	
	21	0,7925	0,4970	0,6109	
Séquence 2 Scénario 1	3	0,6941	0,7216	0,7076	5
	5	0,9855	0,5785	**0,7290**	
	11	0,9912	0,3885	0,5582	
	21	0,7626	0,2620	0,3900	
Séquence 2 Scénario 2	3	0,7662	0,6690	**0,7143**	3
	5	0,9883	0,4873	0,6527	
	11	0,9895	0,3258	0,4902	
	21	0,7173	0,2070	0,3213	

TAB. 4.2 – Performances mesurées lorsque la durée d'observation varie.

mente, le rappel augmente, tandis que la précision diminue. Cette observation est également vraie pour la séquence 2. Dans le cas où l'on souhaite détecter les personnes (scénario 1), la durée d'observation optimale est de 5 images. Dans ce cas, il faudra compter sur le module de niveau supérieur (suivi d'objets) pour gérer les régions fusionnées que l'on obtient pour les voitures les plus proches de l'horizon. Si l'on ne souhaite pas détecter les personnes (scénario 2), il est préférable de choisir des séquences élémentaires de durée 3. On aura dans ce cas affaire à des objets fractionnés, qui devront eux aussi être gérés par le module de suivi d'objets.

Nous avons également mesuré les temps d'exécution pour chacune des configurations présentées ci-dessus. Le temps d'exécution ne dépend que de la taille des images de la séquence utilisée. Comme nos deux séquences de test sont de mêmes dimensions (720×576 pixels), la courbe qui représente l'évolution du temps d'exécution en fonction de la durée d'observation est la même dans chaque cas. Celle-ci est représentée sur le graphe de la figure 4.24.

Assez logiquement, on observe que la relation entre la durée des séquences élémentaires et le temps de calcul nécessaire est quasi-linéaire. La majorité du temps d'exécution étant utilisée pour le calcul de l'ACP, on constate que lorsque la durée d'observation est courte, le temps de calcul se situe légèrement au-dessus de la droite par laquelle on pourrait approximer la courbe obtenue. En effet, dans ce cas, le calcul de l'ACP étant très rapide, les opérations annexes telles que l'interpolation bicubique ou le calcul des statistiques par bloc commencent à intervenir visiblement.

Dans cette section, nous avons montré que la durée d'observation optimale pouvait être déterminée expérimentalement en fonction d'une vérité terrain étiquetée par un expert. Le temps d'exécution de l'algorithme est une fonction linéarire de la longueur de la séquence élémentaire à analyser. Nous allons

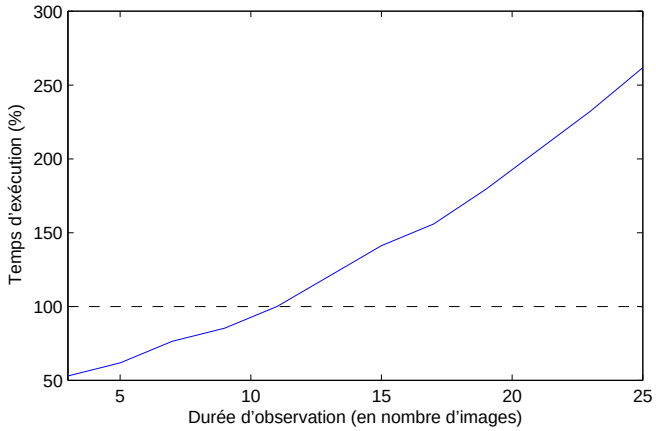


FIG. 4.24 – Temps d'exécution du traitement en fonction de la durée des séquences élémentaires.

maintenant étudier l'influence de la taille des régions sur les résultats.

4.4.3 Taille des régions

La taille des blocs spatio-temporels introduits à la section 4.3.2 est également un paramètre à fixer par l'utilisateur. Des blocs trop larges nuiraient à la précision des contours des objets détectés, tandis que des blocs trop petits impliqueraient des temps de calcul plus élevés, et la connexité des régions pourrait en souffrir.

Nous utiliserons toujours des régions dont les dimensions suivant les axes x et y sont égales, puisque nous n'avons aucune raison de supposer que les informations de mouvement seraient plus visibles sur un axe que sur l'autre. Nous noterons w cette largeur de voisinage spatial. En revanche, la dimension temporelle des régions analysées sera toujours égale à τ^*, la durée d'observation optimale telle que définie dans la section 4.4.2.

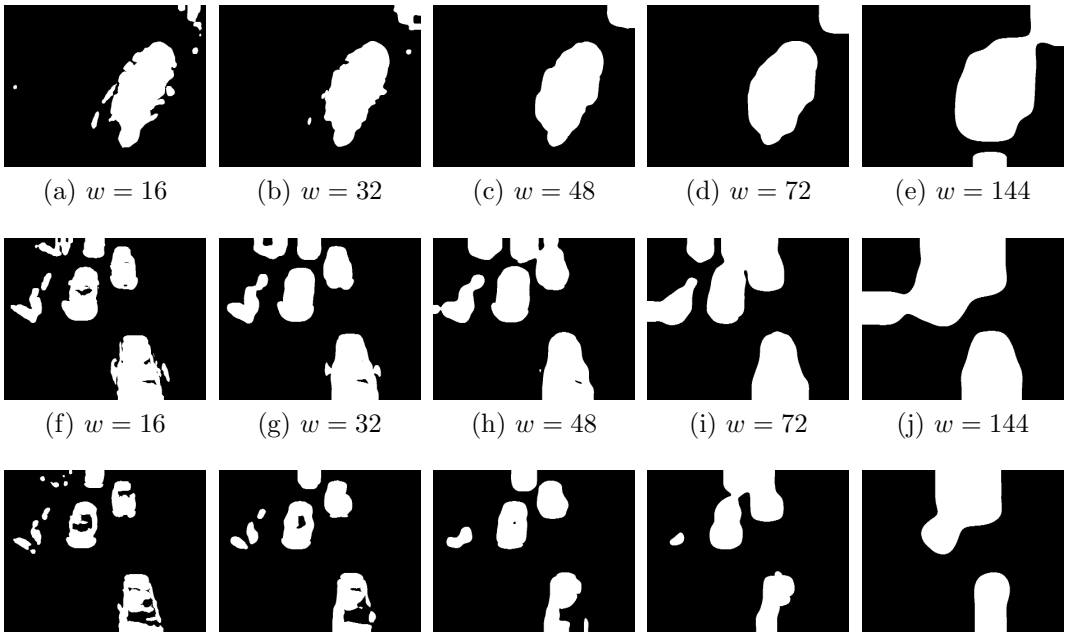
Tout comme dans la section précédente, nous étudions l'influence du paramètre w au travers d'une analyse qualitative puis d'une analyse quantitative.

Analyse qualitative

Les séquences utilisées pour mener les expérimentations sont les mêmes que dans la section précédente. La première séquence a valeur de référence puisque c'est celle qui nous a servi de support tout au long de ce chapitre pour exposer la méthode. De manière analogue à ce que l'on avait observé lors de l'étude de l'influence de la durée d'observation, la séquence 2 est également intéressante pour cette étude, puisqu'elle met en scène deux familles d'objets qui se distinguent non seulement par leur vitesse moyenne, mais aussi par leurs dimensions (des piétons et des véhicules).

La figure 4.25 présente les résultats pour ces deux séquences, avec des ré-

gions de taille $w \times w \times \tau^*$, où w vaut successivement 16, 32, 48, 72 et 144. Ces dimensions ont été choisies de manière à être des diviseurs communs de la largeur et de la hauteur des trames, soit 720 et 576 pixels. Pour la séquence 2, comme la valeur de τ^* est différente selon le scénario considéré, nous avons représenté les résulats obtenus dans les deux cas (figures 4.25f—j : scénario 1 ; figures 4.25k—o : scénario 2). On pourra retrouver la vérité terrain pour la séquence 1 sur la figure 4.22b, et sur la figure 4.23b pour la séquence 2.



(a) $w = 16$ (b) $w = 32$ (c) $w = 48$ (d) $w = 72$ (e) $w = 144$

(f) $w = 16$ (g) $w = 32$ (h) $w = 48$ (i) $w = 72$ (j) $w = 144$

(k) $w = 16$ (l) $w = 32$ (m) $w = 48$ (n) $w = 72$ (o) $w = 144$

FIG. 4.25 – Résultats obtenus à partir de deux séquences en faisant uniquement varier la taille des régions.

Conformément à nos attentes, les résultats obtenus pour la séquence 1 (figures 4.25a—e) indiquent que le nombre de composantes connexes de l'image segmentée augmente avec la taille des blocs. Pour cette séquence, on obtient une segmentation optimale avec $w = 48$. Pour la séquence 2, si l'on désire détecter les véhicules et les personnes, la F-mesure donne une largeur de blocs optimale de 16 pixels, mais visuellement, nous préférons le résultat obtenu pour $w = 32$ (dont le score est très proche) car la compacité des régions détectées est meilleure, ce qui facilitera la tâche du module de suivi d'objets. De manière analogue, dans le cas où l'on ne souhaite détecter que les véhicules, la F-mesure donne 32 comme valeur idéale de w, alors que nous aurions préféré conserver le résultat obtenu avec une valeur de w égale à 48, pour la même raison que précédemment. Cette expérience montre que la taille des blocs spatio-temporels doit être choisie en fonction de la taille moyenne des objets que l'on souhaite détecter, ce qui explique que l'on trouve des valeurs différentes de w^* pour la séquence 2 selon que l'on se place dans le cas du scénario 1 ou du scénario 2.

Analyse quantitative

Le tableau 4.3 fournit le rappel, la précision et la F-mesure obtenus dans chacun des cas. On indique également pour chaque séquence et chaque scénario, la largeur idéale des régions déduite de l'expérimentation, notée w^*.

Séquence	w	ρ	ν	F	w^*
Séquence 1	16	0,8287	0,6481	0,7273	48
	32	0,8425	0,6368	0,7253	
	48	0,9150	0,6341	**0,7491**	
	72	0,9673	0,5900	0,7330	
	144	1,0000	0,4087	0,5802	
Séquence 2 Scénario 1	16	0,9278	0,6942	**0,7941**	32
	32	0,9704	0,6434	**0,7737**	
	48	0,9855	0,5785	0,7290	
	72	0,9882	0,5021	0,6658	
	144	0,9435	0,3818	0,5436	
Séquence 2 Scénario 2	16	0,7348	0,7278	0,7313	48
	32	0,7748	0,7062	**0,7389**	
	48	0,7662	0,6690	**0,7143**	
	72	0,7505	0,5880	0,6593	
	144	0,8446	0,5153	0,6401	

Tab. 4.3 – Performances mesurées lorsque la taille des régions varie.

Les résultats présentés sur la figure 4.25 indiquent que la forme des régions détectées est de moins en moins précise lorsque la taille des blocs augmente. Ceci est confirmé par le fait que globalement, la précision est inversement proportionnelle à ce paramètre. Le rappel, quant à lui, s'améliore quand on utilise des blocs de taille plus importante. Le meilleur compromis entre rappel et précision (recherché par le biais de la F-mesure) est obtenu lorsque la taille des blocs est équilibrée par rapport à la taille des objets que l'on souhaite détecter.

Comme la durée des séquences élémentaires est différente dans chacun des cas étudiés, nous avons également mesuré les temps d'exécution pour chacune des configurations présentées ci-dessus. Ceux-ci sont représentés sur le graphe de la figure 4.26.

On constate que le temps d'exécution de l'algorithme décroît lorsque la taille des blocs augmente. Choisir des blocs de petite taille implique que le nombre de blocs sera important, et qu'il faudra donc calculer les paramètres des ellipses d'inertie plus souvent. L'outil de développement que nous avons utilisé pour faire nos expérimentations est Matlab®. Celui-ci est connu pour être très peformant pour les opérations d'algèbre linéaire (l'ACP dans notre cas), mais moins adapté que les langages de plus bas niveau pour exécuter des

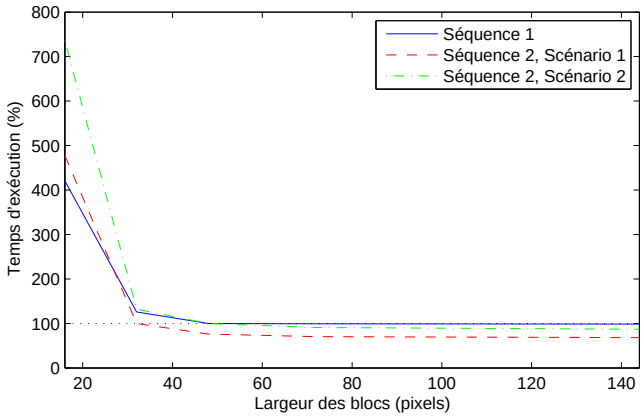


FIG. 4.26 – Temps d'exécution de l'algorithme en fonction de la taille des régions pour chacun des trois scénarios étudiés.

algorithmes itératifs (le calcul des statistiques par bloc) quand une optimisation n'est pas recherchée. Il est probable que si nous avions travaillé en langage C, par exemple, nous aurions obtenu des courbes un peu plus linéaires que celles de la figure 4.26.

Nous avons montré dans cette section que la taille des blocs spatio-temporels utilisés avait une grande influence sur la précision des résultats. Si l'on choisit des blocs de taille adaptée à celle des objets que l'on cherche à détecter, la segmentation obtenue peut être très satisfaisante du point de vue de la F-mesure et du nombre de régions d'avant-plan. Nous allons maintenant étudier l'influence des seuils utilisés pour obtenir la segmentation finale en sélectionnant les blocs associés à du mouvement.

4.4.4 Seuils pour la segmentation

Pour passer de l'ensemble des deux matrices de mesures statistiques (distance au centre du repère et surface des ellipses d'inertie qui modélisent les régions) à une image binaire représentant les deux classes « objets » et « fond », il est nécessaire de choisir deux seuils. Dans tous les exemples précédents, nous avons déterminé ces seuils automatiquement de manière à minimiser la variance intra-classe (ou maximiser la variance inter-classes), c'est-à-dire que nous avons appliqué à nos mesures statistiques la méthode de binarisation proposée pour des images en niveaux de gris dans [Otsu, 1979].

Bien que cette méthode fournisse des résultats tout à fait accpetables dans la plupart des cas, nous devons néanmoins étudier l'influence de la valeur de ces paramètres sur le résultat final afin de définir des règles pour les déterminer au mieux.

Comme les deux seuils ont une influence conjointe sur les résultats, nous proposons de déterminer le couple des valeurs optimales par l'analyse d'une courbe ROC (*Receiver Operating Characteristic*). Dans le cadre de l'optimisa-

tion d'un classifieur binaire à un paramètre, la courbe ROC est l'ensemble des points obtenus en faisant varier ce paramètre, dans le plan dont l'axe horizontal est le taux de faux positifs défini par

$$f_p = \frac{\#FP}{\#FP + \#VN}, \tag{4.42}$$

et l'axe vertical le taux de vrais positifs (autre nom du rappel ρ). Dans le cas où l'algorithme possède plusieurs paramètres, une distribution de points générés par un ensemble de combinaisons possibles des paramètres, est tracée dans ce même plan. La courbe ROC est alors la frontière supérieure de l'enveloppe convexe de la distribution.

Les paramètres α (seuil sur la distance à l'origine du repère du centre d'une ellipse) et β (seuil sur la surface d'une ellipse) trouvent leur valeur optimale dans un domaine *a priori* non borné ($\mathbb{R}_+$). Plutôt que de déterminer une plage de valeurs dans laquelle nous ferions varier uniformément α et β, nous avons choisi aléatoirement des valeurs de α et β en suivant deux distributions gaussiennes centrées autour des seuils déterminés automatiquement par l'algorithme de [Otsu, 1979].

La figure 4.27 représente les diagrammes ROC obtenus pour les trois scénarios de test avec 400 couples (α, β) tirés aléatoirement de cette manière. La durée d'observation et la taille des régions utilisées sont repsectivement τ^* et w^* telles qu'elles ont été définies dans les deux sections précédentes.

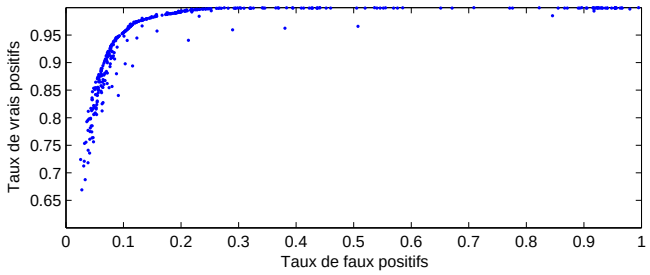
Pour chaque séquence et chaque scénario, nous avons sélectionné cinq couples de valeurs (α, β) tels que le point représentant les performances mesurées se trouve sur la courbe ROC présentée sur la figure 4.27. Nous ne nous sommes intéressé qu'aux couples de seuils tels que les valeurs de rappel et de précision soient toutes les deux supérieures à 0,5. Le tableau 4.4 rassemble le rappel, la précision et la F-mesure obtenus pour ces couples de valeurs. On indique également pour chaque séquence et chaque scénario, les valeurs idéales de α et β au sens de la F-mesure, notées α^* et β^*.

Les courbes ROC obtenues ont une forme caractéristique satisfaisante. En longeant l'enveloppe convexe supérieure de celles-ci, on trouve facilement un couple optimal (α^*, β^*) tel que la F-mesure soit maximisée.

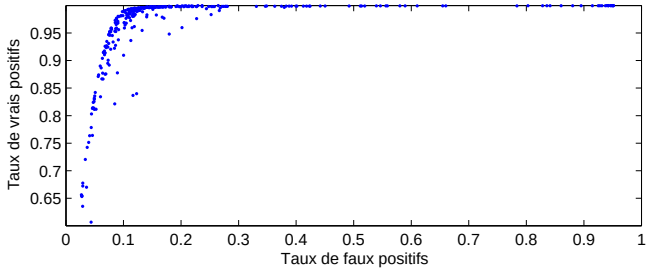
Comme les paramètres α et β ne sont que des seuils permettant de réaliser la segmentation finale, ils n'ont aucune influence sur les temps de calcul, c'est pourquoi nous ne réaliserons pas cette analyse.

La figure 4.28 montre les segmentations obtenues avec les paramètres (α^*, β^*) définis précédemment pour chacun des trois scénarios de test.

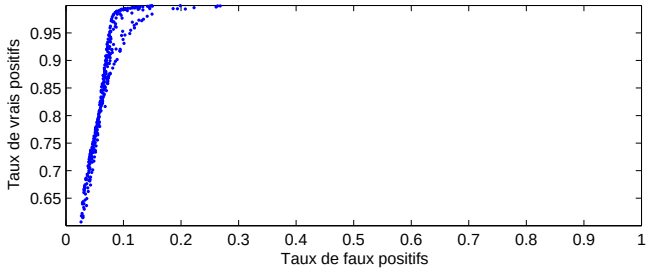
Maintenant que nous avons étudié l'influence de chacun des paramètres de l'algorithme sur la qualité des résultats et sur les temps de calcul, nous allons comparer les résultats fournis par celui-ci à ceux obtenus par d'autres algorithmes issus de la littérature.



(a) Séquence 1.



(b) Séquence 2, scénario 1.



(c) Séquence 2, scénario 2.

FIG. 4.27 – Courbes ROC obtenues en faisant varier les paramètres α et β pour les trois scénarios de test.

(a) Séquence 1.

(b) Séquence 2, scénario 1.

(c) Séquence 2, scénario 2.

FIG. 4.28 – Segmentations obtenues avec les valeurs optimales α et β au sens de la F-mesure, pour les trois scénarios de test.

4.4.5 Étude comparative

Pour évaluer les performances de notre algorithme, nous utilisons cinq séquences vidéo qui se différencient par la problématique demandée par l'appli-

Séquence	α	β	ρ	ν	F	α^*	β^*
Séquence 1	0,4131	0,0125	0,8115	0,7650	0,7876	0,4114	0,0076
	0,4114	0,0076	0,8535	0,7337	**0,7891**		
	0,3477	0,0069	0,8646	0,7225	0,7872		
	0,3573	0,0055	0,8894	0,6922	0,7785		
	0,3916	0,0044	0,9121	0,6705	0,7729		
Séquence 2 Scénario 1	0,2128	0,0163	0,8421	0,7339	0,7843	0,1699	0,0124
	0,1895	0,0141	0,8901	0,7136	0,7921		
	0,1699	0,0124	0,9259	0,6942	**0,7935**		
	0,1580	0,0111	0,9475	0,6793	0,7913		
	0,1553	0,0084	0,9730	0,6550	0,7829		
Séquence 2 Scénario 2	0,1114	0,0084	0,8526	0,6502	0,7378	0,1181	0,0063
	0,1320	0,0074	0,8998	0,6418	0,7492		
	0,1333	0,0067	0,9420	0,6347	0,7584		
	0,1181	0,0063	0,9753	0,6258	**0,7624**		
	0,1447	0,0061	0,9668	0,6236	0,7582		

TAB. 4.4 – Performances mesurées lorsque les seuils α et β varient.

cation et/ou les difficultés intrinsèques de la séquence.

- La première séquence (« Salon ») illustre un problème de comptage de personnes passant par le sas situé en bas de l'image. La difficulté est liée au fait que plusieurs personnes restent longtemps plus ou moins immobiles dans le champ de vision avant de franchir (ou non) le sas. Il faut donc que l'algorithme ne détecte pas les mouvements insignifiants.
- La seconde vidéo (« Passerelle ») représente également un problème de comptage de personnes, mais dans ce cas, les personnes ont tendance à se déplacer en groupes connexes. Il faut donc un algorithme suffisamment précis pour pouvoir discerner les différents membres de chaque groupe.
- La troisième séquence (« Route ») représente une application de surveillance de trafic routier. Elle met en scène des piétons et des véhicules circulant à grande vitesse. Nous nous plaçons dans le cas où l'on souhaite détecter indifféremment tous les objets mobiles. La difficulté provient du fait que les piétons et les véhicules se déplacent à des vitesses très différentes et sont représentés par des zones de l'image de dimensions variées. Le système doit être suffisamment générique pour gérer ces deux familles d'objets.
- Les deux dernières séquences (« Caviar1 » et « Caviar2 ») sont des séquences de test classiques utilisées dans de nombreux articles[2]. Elles sont utilisées dans le but de faciliter la comparaison des résultats présentés ici

[2]Les séquences présentées dans les deux dernières colonnes de la figure 4.29 proviennent du projet CAVIAR/IST 2001 37540 financé par la Commission Européenne, trouvé à l'URL : `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`.

avec ceux obtenus par d'autres méthodes.

Sur la figure 4.29 sont représentées les segmentations entre objets mobiles et arrière-plan obtenues sur ces cinq séquences vidéo, avec cinq algorithmes différents. Les algorithmes utilisés, ainsi que la manière dont ils ont été paramétrés pour cette étude comparative sont décrits ci-dessous.

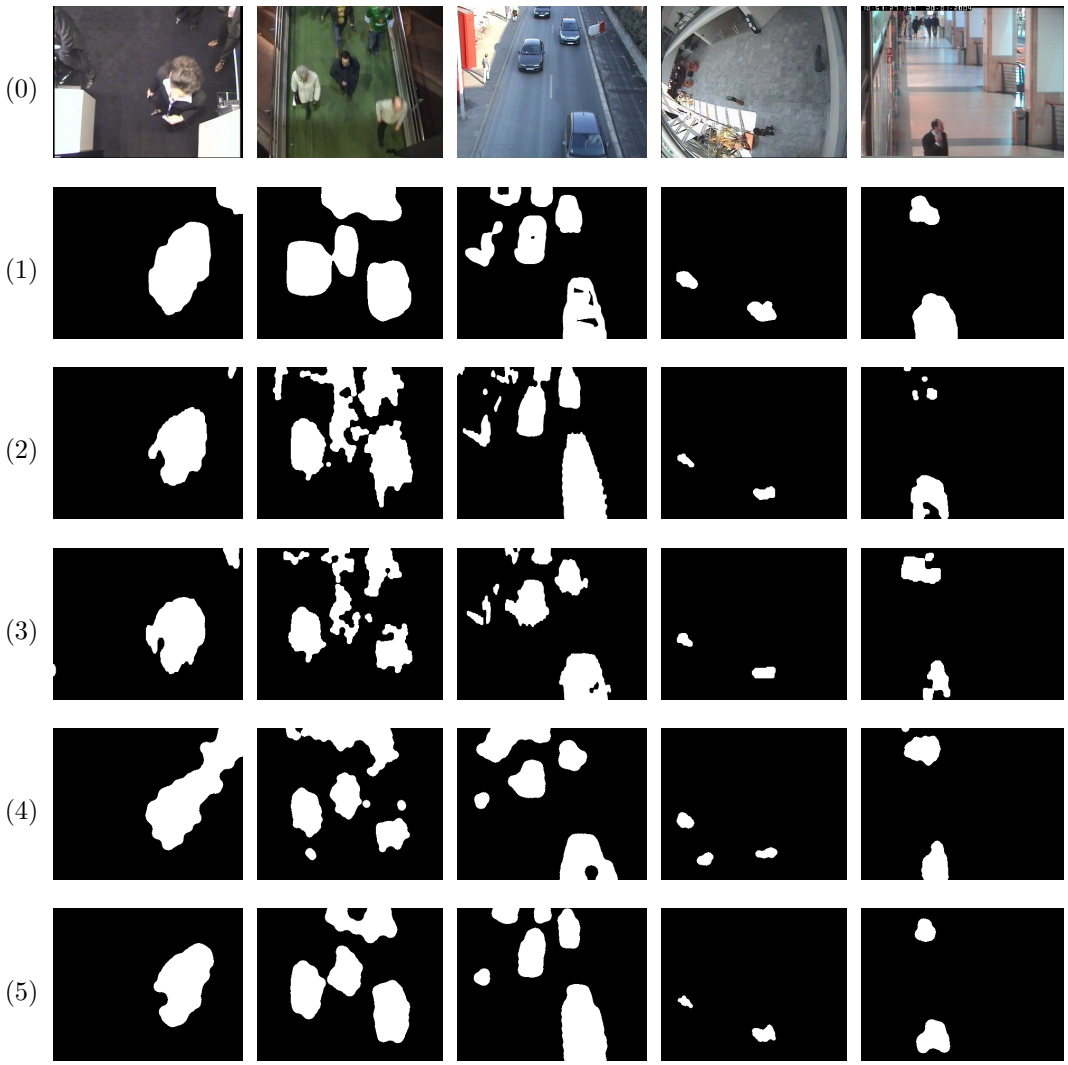


FIG. 4.29 – Résultats obtenus sur cinq séquences avec cinq algorithmes.

Notre méthode. Sur la ligne 1 de la figure 4.29 est représenté un échantillon de résultats obtenus avec l'algorithme présenté dans ce chapitre.

Comme les séquences vidéo ont des dimensions différentes, et les objets recherchés sont hétérogènes du point de vue de leur taille et de leur vitesse de déplacement, nous avons choisi la longueur des séquences élémentaires et la taille des blocs spatio-temporels en fonction de ces particularités. Les seuils

utilisés pour la segmentation ont été déterminés par la méthode présentée dans la section 4.4.4 lors de l'étude de l'influence de ces paramètres sur les résultats.

Dérivée temporelle. Dès les débuts de l'analyse vidéo [Jain et Nagel, 1979], la valeur absolue des différences entre images consécutives s'est imposée comme mesure de mouvement grâce à la vitesse à laquelle elle peut être calculée, et aux résultats satisfaisants qu'elle permet d'obtenir dans les cas les plus simples. Ceux-ci sont représentés sur la ligne 2 de la figure 4.29.

La plupart du temps, la mesure obtenue est lissée par un opérateur de moyenne mobile afin de ne pas détecter que le contour des objets mobiles. Nous avons fixé le coefficient de mise à jour à une valeur de 0,7 (cf. équation 2.2) qui fournit un bon compromis entre rappel et précision des résultats.

Mélange de gaussiennes. La ligne 3 représente un échantillon de résultats obtenus par soustraction de l'arrière-plan lorsque les fonctions de densité de probabilité des pixels sont modélisées par des mélanges de gaussiennes. Depuis la publication de l'article de [Stauffer et Grimson, 1999], cette méthode est l'une des plus fréquemment utilisées dans les application d'analyse de séquences vidéo. Pour réaliser nos expérimentations, nous avons utilisé l'algorithme publié et implémenté par [Zivkovic et van der Heijden, 2006]. Il s'agit d'une version améliorée de l'algorithme original, dans laquelle le nombre de gaussiennes est déterminé automatiquement.

Cette méthode possède deux paramètres : la vitesse de mise à jour des fonctions de densité et le seuil sur les densités de probabilité à partir duquel on considère qu'une gaussienne explique un niveau de gris observé. L'origine de ces paramètres est détaillé dans la section 2.2.3.2. Les meilleurs résultats avec cette méthode ont été obtenus en réglant la vitesse de mise à jour de manière à ce que la moyenne des gaussiennes soit calculée sur 10 à 20 images selon les séquences. C'est cette valeur que nous avons retenue. De même, nous avons choisi le seuil sur les densités de probabilité qui a fourni les meilleurs résultats, soit $2{,}5\sigma_{i,t}$, où $\sigma_{i,t}$ est l'écart-type de la i-ème gaussienne au temps t.

Eigenbackgrounds. La méthode des *eigenbackgrounds* (cf. section 3.2.2) constitue une application de l'ACP à la détection de mouvement qui est très différente de la nôtre, puisque dans ce cas, les images sont considérées comme des réalisations d'une variable aléatoire à n dimensions, où n est le nombre de pixels dans le plan de l'image. Nous avons considéré qu'il était intéressant de pouvoir comparer les résultats obtenus par les deux approches. Ceux fournis par cette méthode sont représentés sur la ligne 4 de la figure 4.29.

En raison du temps de calcul nécessité par la version originale publiée dans [Oliver *et al.*, 2000], cette méthode ne permet pas de respecter les contraintes de temps réel imposées par les applications de traitement vidéo. C'est pourquoi, dans le programme que nous avons implémenté, l'ACP standard n'est exécutée qu'une seule fois en se basant sur un échantillon d'images tirées aléatoirement dans la séquence à étudier, puis le modèle constitué des valeurs et

vecteurs propres fournis par l'ACP est mis à jour par la méthode de l'ACP incrémentale proposée dans [Hall *et al.*, 1998]. Comme théoriquement, le nombre de vecteurs propres peut croître indéfiniment, nous avons supprimé celui associé à la plus petite valeur propre (comme suggéré dans [Rymel *et al.*, 2004]) dès que le nombre de vecteurs propres excède une valeur optimale. Celle-ci a été déterminée expérimentalement afin d'optimiser les résultats.

Entropie spatio-temporelle. Notre méthode a en commun avec celle du calcul de l'entropie spatio-temporelle de la séquence [Ma et Zhang, 2001] le fait de ne pas chercher à bâtir un modèle de l'arrière-plan, mais plutôt à rechercher dans le volume 2D+T que constitue la séquence, des points où la « variabilité » de l'apparence des pixels est maximale.

Comme nous l'avons évoqué à la section 2.2.2.2, l'entropie des niveaux de gris est élevée non seulement aux endroits où un mouvement a lieu, mais également le long des contours des objets. Afin de nous concentrer sur le mouvement, nous avons implémenté la méthode de [Guo *et al.*, 2004] qui consiste à calculer l'entropie des niveaux de gris dérivés par rapport au temps au sein d'un voisinage spatio-temporel de chaque pixel. La taille de ce voisinage, ainsi que le seuil utilisé pour la segmentation, ont été, ici encore, choisis de manière à optimiser la qualité des résultats, représentés sur la ligne 5 de la figure 4.29.

Analyse qualitative

Dans la littérature, les méthodes concurrentes que nous avons testées sont toujours suivies d'une phase de post-traitement pour faciliter l'extraction des composantes connexes. Pour les lignes 2 à 5, nous avons donc appliqué aux résultats obtenus une suppression des composantes connexes dont l'aire est inférieure à un certain seuil, suivie d'une fermeture puis d'une ouverture morphologique par un disque dont le diamètre a été choisi en fonction de la dimension des objets attendus.

On constate que la dérivée temporelle lissée fournit un résultat tout à fait convenable lorsque la difficulté de la séquence est peu importante (pour la séquence « Salon », par exemple). En revanche, lorsque le nombre d'objets en mouvement est important (deuxième colonne), la sensibilité de cette méthode à tout type de bruit (comme les ombres), ainsi que l'effet fantôme, rendent l'étiquetage des composantes connexes de l'image segmentée peu cohérent avec la localisation des objets réels dans l'image. Une autre conséquence de l'effet fantôme, bien visible dans la troisième colonne (séquence « Route »), est la déformation des objets se déplaçant à grande vitesse.

L'algorithme utilisant une modélisation ponctuelle statistique de l'arrière-plan (mélange de gaussiennes) ne souffre pas de l'effet fantôme et fait apparaître les contours des objets mobiles de manière plus précise. En revanche, lorsque des objets ont un niveau de gris moyen qui est proche de celui du fond, l'affectation des pixels à une distribution gaussienne plutôt qu'à une autre devient fortuite, et on peut obtenir un nombre important de faux positifs, comme

dans des séquences « Passerelle » et « Caviar2 ».

Comme la méthode des *eigenbackgrounds* utilise plusieurs modèles de l'arrière-plan (les vecteurs propres révélés par l'ACP), elle est moins sensible aux bruits dus à un arrière-plan qui n'est pas parfaitement statique. Ainsi, la connexité des régions détectées correspond en général bien à la vérité terrain. Par contre, si parmi les vecteurs propres conservés, certains représentent le passage d'un objet, l'absence de cet objet peut être confondue avec un mouvement, provoquant ainsi des faux positifs importants, comme dans le cas de la séquence « Salon ».

Le seuillage de l'entropie spatio-temporelle de la séquence permet d'obtenir une très bonne compacité des régions d'avant-plan pour chacune des séquences étudiées. En revanche, tout comme la dérivée temporelle lissée, cette méthode est sujette à l'effet fantôme lorsque la vitesse des objets est élevée (séquence « Route »). Comme cette méthode ne consiste pas à comparer l'image courante à un modèle de l'arrière-plan, les contours des objets sont plus approximatifs qu'avec les deux méthodes précédentes. Par ailleurs, l'entropie étant calculée au sein d'un voisinage spatio-temporel sans tenir compte du reste de l'image, la segmentation obtenue peut souffrir du « problème d'ouverture » (*aperture problem*), bien connu dans le domaine de l'estimation du flot optique, qui se produit lorsque l'on observe une surface uniforme en mouvement à travers une fenêtre plus petite que la zone uniforme (séquence « Caviar2 »).

Notre méthode, tout comme celle de l'entropie spatio-temporelle, sacrifie la précision des contours au profit d'une plus grande robustesse. La compacité des régions d'avant-plan est également très satisfaisante, ce qui se révèlera capital lorsque nous chercherons à suivre les objets mobiles.

Analyse quantitative

En utilisant les métriques définies dans la section 4.4.1, nous avons mesuré les performances de chacun de ces algorithmes sur l'ensemble des séquences de test. Les chiffres présentés dans le tableau 4.5 ont été obtenus en utlisant 210 images de chacune des séquences vidéo. Pour chaque séquence et chaque critère (rappel ρ, précision ν et F-mesure F), nous avons indiqué en gras l'algorithme qui permet d'obtenir les meilleurs résultats.

On constate qu'à une exception près, notre algorithme est toujours celui qui permet d'obtenir le meilleur rappel. En effet, le fait de considérer des blocs spatio-temporels pour calculer les mesures statistiques produit une segmentation lisse qui optimise la compacité des régions détectées au détriment de la précision des contours. Cela est confirmé par la valeur relativement modeste de la précision obtenue par notre algorithme. Nous obtenons néanmoins la meilleure F-mesure moyenne lorsque l'on considère l'ensemble des séquences de test.

La méthode de segmentation basée sur la dérivée temporelle lissée de la séquence est bien trop sensible aux bruits pour fournir de bons résultats dans chaque cas. Les performances mesurées peuvent être très bonne (séquence

		Salon	Passerelle	Route	Caviar1	Caviar2	Moy.
Notre méthode	ρ	**0,828**	**0,739**	0,885	**0,779**	**0,848**	**0,816**
	ν	0,669	0,751	0,677	0,578	0,678	0,670
	F	0,730	0,733	0,764	**0,656**	**0,742**	**0,725**
Dérivée temporelle	ρ	0,626	0,720	**0,908**	0,421	0,374	0,610
	ν	**0,864**	0,737	0,681	0,797	0,661	0,748
	F	0,703	0,716	**0,775**	0,517	0,455	0,633
Mélange de gaussiennes	ρ	0,710	0,613	0,796	0,559	0,716	0,679
	ν	0,782	0,848	**0,769**	0,563	0,810	0,755
	F	0,722	0,705	0,771	0,514	0,736	0,690
Eigenback-grounds	ρ	0,675	0,699	0,890	0,563	0,640	0,693
	ν	0,747	**0,838**	0,686	0,737	**0,909**	0,783
	F	0,702	**0,758**	0,770	0,626	0,714	0,714
Entropie	ρ	0,568	0,670	0,839	0,396	0,549	0,604
	ν	0,750	0,819	0,702	**0,905**	0,874	**0,810**
	F	0,634	0,724	0,763	0,532	0,662	0,663

TAB. 4.5 – Performances de différents algorithmes de détection de mouvement.

« Route ») comme très mauvaise (séquence « Caviar2 »).

La modélisation de l'arrière-plan par un mélange de gaussiennes permet d'obtenir de très bons résultats en général, sauf dans le cas de la séquence « Caviar1 » qui est très bruitée. La principale qualité de cette méthode est la grande précision que permet d'obtenir le principe de la soustraction entre l'image courante et un modèle. En revanche, en raison des nombreux faux négatifs obtenus, les chiffres du rappel sont moins bons.

Dans la méthode des *eigenbackgrounds* également, la notion de voisinage spatial est absente, et la segmentation est obtenue en confrontant ponctuellement la valeur de niveau de gris courante d'un pixel au contenu du modèle à cet endroit. Par conséquent, ici aussi, on obtient en général une très bonne précision. Pour les mêmes raisons que dans le cas précédent, les chiffres du rappel sont moins bons.

Le calcul de l'entropie spatio-temporelle fait intervenir la notion de voisinage spatial, mais, quand celui-ci n'est pas trop large, la précision obtenue est tout de même très bonne. C'est d'ailleurs cette méthode qui obtient globalement les meilleurs chiffres de précision. En revanche, le rappel obtenu est en général assez faible, ce qui explique les résultats modestes de cette méthode au sens de la F-mesure.

Par cette étude comparative, nous avons pu vérifier que la méthode de détection de mouvement présentée dans ce chapitre permet d'obtenir des résultats de très bonne qualité au sens de la F-mesure. Les régions d'avant-plan obtenues présentent en général une compacité mieux adaptée à l'étape suivante du traitement (le suivi d'objets) que celles fournies par les autres méthodes testées, au prix d'une moindre précision des contours. La qualité des résultats est stable quelle que soit la séquence considérée, ce qui confirme que cette méthode

offre une bonne généricité.

4.5 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode de détection de mouvement cohérent dans une séquence vidéo. Contrairement à la plupart des méthodes présentes dans la littérature, nous ne cherchons pas à modéliser l'arrière-plan de la scène pour détecter les objets, mais plutôt à exprimer les données vidéo dans un espace de représentation de dimension réduite, dans lequel la classification entre zones en mouvement et zones statiques est aisée. Pour obtenir cet espace nous appliquons une analyse en composantes principales sur les données d'entrée, et nous ne conservons que les deux premiers facteurs principaux. La séquence est ensuite découpée en blocs spatio-temporels qui sont classifiés par rapport à la position et l'aire de l'ellipse d'inertie qui les représente dans le plan utilisé.

La segmentation obtenue est satisfaisante dans le sens où le nombre de composantes connexes correspond souvent au nombre attendu. De plus, aucun post-traitement n'est nécessaire à l'exploitation des images segmentées. En revanche, les contours des objets sont détectés moins précisément qu'avec les algorithmes de modélisation statistique de l'arrière-plan. Cela dit, dans le contexte d'une utilisation industrielle de la méthode, la précision des contours n'est pas d'une importance capitale ; il est bien plus important de connaître précisément le nombre d'objets présents dans la scène, ainsi que leurs positions et leurs surfaces approximatives. Dans le cas où une grande précision des contours est requise, on pourra utiliser un contour actif [Kass *et al.*, 1988] que l'on initialisera sur le contour fourni par notre méthode. Par ailleurs, la généricité de la méthode est confirmée par le fait que l'on n'observe aucune disparité importante des performances obtenues pour les différentes séquences utilisées lors de l'évaluation.

Chapitre 5

Modélisation des déplacements

Sommaire

Dans le chapitre précédent, nous avons présenté une méthode de détection de mouvement qui nécessite peu de paramétrage pour obtenir un ensemble de régions d'avant-plan cohérentes du point de vue du mouvement qu'elles représentent. Ce type de résultat peut constituer une fin en soi pour certaines applications, telles que la détection d'intrusions dans une zone protégée. Cependant, dans bien des cas, les objectifs des applications de traitement vidéo sont plus ambitieux et nécessitent de posséder des informations telles que le nombre d'objets présents dans la scène, leurs trajectoires, voire une interprétation sémantique de leurs mouvements. Dans ce cas, la sortie du module de détection de mouvement doit être analysée et interprétée de manière à en déduire les informations voulues. La méthode proposée au chapitre précédent est

bien adaptée à une telle analyse car, comme le montrent les résultats obtenus, le nombre de régions connexes de l'image segmentée a tendance à correspondre au nombre de régions présentes dans la vérité terrain.

Dans ce chapitre, nous proposons une méthode de modélisation des déplacements d'objets dans une séquence vidéo, basée sur la sortie de la méthode de détection de mouvement présentée au chapitre 4. Nous allons dans un premier temps décrire le problème qui nous intéresse, ainsi que la manière dont celui-ci est traité dans la littérature. Nous exposerons ensuite le modèle théorique qui représentera les informations de déplacement des objets, à savoir un graphe[1] désigné sous le nom de « graphe d'association ». Dans la section suivante, nous proposerons une stratégie d'élagage visant à simplifier l'analyse du graphe obtenu. Finalement, nous expliquerons comment extraire du graphe des informations à valeur sémantique.

5.1 Introduction

L'interprétation sémantique des résultats de la détection de mouvement nécessite de modéliser davantage les informations dont nous disposons, c'est-à-dire une liste de régions d'avant-plan pour chaque sous-ensemble des données vidéo correspondant à une durée élémentaire. Dans cette section, nous commencerons par définir précisément le problème que nous cherchons à résoudre, puis nous passerons rapidement en revue les différentes approches proposées dans la littérature.

5.1.1 Positionnement du problème

Le problème qui nous intéresse est celui de l'association de données pour établir des correspondances entre mouvements détectés. En effet, les données dont nous disposons à ce niveau constituent une suite d'images segmentées en deux classes, que nous avons précédemment nommées « fond » et « objets ». Chaque image binaire peut être vue comme une collection de régions d'avant-plan qui peuvent être repérées par étiquetage des composantes connexes des images binaires.

Cependant, en sortie du module de détection de mouvement, tous ces ensembles de régions d'avant-plan sont dissociés les uns des autres. Rien ne permet de savoir par exemple que la i-ème région détectée au temps t représente le même objet que la j-ème région détectée au temps $t+1$. Le but de l'étape de modélisation des déplacements est d'établir des hypothèses quant aux origines communes que pourraient avoir des régions détectées à des instants différents.

Ces hypotèses permettront par la suite d'inférer des informations telles que le nombre d'objets présents dans la scène à un instant donné, la trajectoire

[1] Nous utiliserons dans ce chapitre du vocabulaire emprunté à la théorie des graphes. Pour une description des différentes notions utilisées, le lecteur pourra se référer à [Gondran et Minoux, 1995].

d'un objet particulier ou la nature éventuellement suspecte du comportement d'un objet. Les hypothèses d'association entre zones en mouvement peuvent être nombreuses et mutuellement contradictoires. De plus, à chaque hypothèse peut être associé un degré de certitude dépendant de nombreux facteurs. Par conséquent, l'ensemble des hypothèses que nous cherchons à établir constitue une structure logique complexe qui ne pourra être correctement analysée que si elle est modélisée de manière rigoureuse dans un agencement permettant de représenter les différentes caractéristiques des régions détectées, les mesures de similarité qui permettent de déduire des degrés de certitude, et les possibilités de coexistence des hypothèses.

Selon les époques et la nature des détections utilisées, les méthodes de modélisation et d'analyse ont évolué. Dans la section suivante, nous allons proposer un bref aperçu de ces méthodes.

5.1.2 Travaux antérieurs

Les premières méthodes d'association de données (association d'un mouvement détecté à un objet réel) ont été développées avant l'utilisation intensive des caméras vidéo que nous connaissons aujourd'hui. À l'époque, le domaine d'application de ces méthodes était essentiellement l'analyse des données fournies par des appareils de détection tels que les radars ou les sonars. C'est pourquoi, dans cette section, nous utiliserons le vocabulaire de ce domaine. Pour désigner les objets mobiles que l'on cherche à suivre, nous parlerons de « cibles », et pour désigner l'équivalent de nos régions d'avant-plan, c'est-à-dire les zones de mouvement détectées, nous parlerons de « mesures » ou de « détections ». Par ailleurs, dans les premières méthodes présentées, nous verrons que la seule caractéristique des détections qui est utilisée, est sa localisation. Cela s'explique par le fait que les radars et les sonars ne fournissent généralement pas d'autre information sur ce qu'ils ont détecté (ni surface, ni couleur, etc.)

La méthode d'association de données la plus simple est la méthode dite du « plus proche voisin » [Deriche et Faugeras, 1990]. Il s'agit, lorsque l'on suit un ensemble de cibles dont on possède une estimation de la position, de considérer que chaque détection a été générée par la cible dont la position est la plus proche de celle de la détection. Bien souvent, on estime la position des cibles, non pas par un lieu (x, y) précis, mais par une zone de confiance dans laquelle la probabilité d'observer la cible est élevée. Lorsqu'on considère que les densités de probabilité sont gaussiennes, la notion de proximité est généralement exprimée à l'aide de la distance de Mahalanobis, et les positions successives du centre des zones de confiance des cibles sont prédites à l'aide d'un filtre de Kalman [Kalman, 1960]. Dans ce cas, il suffit de connaître la position initiale des cibles pour que le suivi puisse être réalisé pendant toute la durée de la séquence.

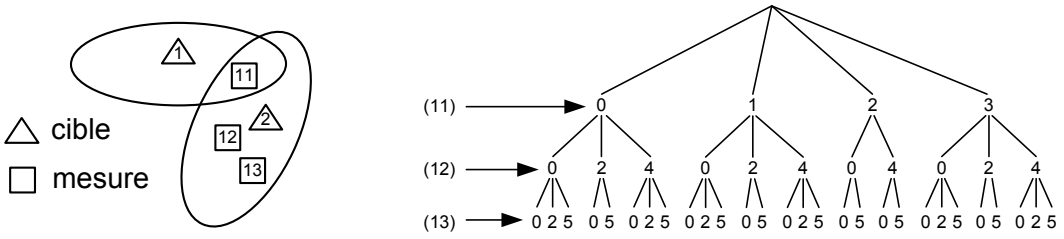
Cette méthode, bien que séduisante par sa simplicité théorique et sa rapidité d'exécution, présente certains inconvénients. En premier lieu, comme une

détection est associée à une seule cible, on court le risque de voir se propager la moindre erreur. L'algorithme se base uniquement sur l'image courante pour réaliser les affectations, alors que l'on pourrait utiliser l'information contenue dans les images suivantes : de meilleurs résultats pourraient être obtenus en retardant la décision d'affectation dans l'espoir que des mesures futures viennent clarifier les ambiguïtés. En outre, cette méthode nécessite que l'on connaisse le nombre de cibles à suivre, et que ce nombre n'évolue par au cours de la séquence. L'apparition ou la disparition d'une cible ne sont pas gérées.

L'algorithme MHT (*Multiple-Hypothesis Tracking*) présenté dans [Reid, 1979] est le premier à constituer une solution au problème de l'initiation et de la terminaison de nouvelles pistes. À chaque instant, cet algorithme consiste à maximiser la probabilité des enchaînements d'associations de toutes les détections présentes depuis leur apparition. Chaque itération se base sur l'ensemble des hypothèses établies à l'itération précédente. Chaque hypothèse fournit une interprétation de toutes les mesures effectuées depuis le début de l'algorithme. La structure logique mise en œuvre est représentée sous la forme d'un graphe (plus précisément, sous la forme d'un arbre).

L'arbre créé par le MHT est constitué à l'origine d'un sommet racine représentant l'hypothèse nulle — puisqu'aucune détection n'a été traitée. À chaque nouvelle détection, on crée un nouvel étage dans l'arbre en prolongeant chaque feuille par l'ensemble des possibilités d'affectation de la nouvelle détection. Les arcs ainsi créés sont pondérés par la probabilité que la nouvelle affectation soit vraie, compte tenu de l'ensemble des hypothèses représentées par le sommet parent. La figure 5.1 représente un exemple de graphe obtenu après que trois détections ont été traitées. On remarquera que les détections sont traitées séquentiellement, bien qu'elles aient été obtenues simultanément. La figure 5.1a indique que deux cibles sont connues. Les ellipses représentent les zones de confiance dans lesquelles on peut détecter chacune des cibles, ce qui signifie que la détection 11 peut être attribuée aux cibles 1 et 2, alors que les détections 12 et 13 ne peuvent être attribuées qu'à la cible 2. La figure 5.1b représente l'arbre obtenu lorsque l'on traite successivement les détections 11, 12 et 13. Chaque étage correspond à une détection, et chaque sommet est étiqueté avec le numéro de la cible à laquelle on attribue la détection concernée. On remarquera que lorsque les cibles sont numérotées de 1 à n, on pourra étiqueter un sommet par un nombre allant de 0 à $n + 1$. Le fait d'étiqueter un sommet avec l'identifiant 0 signifie que l'on considère que la détection correspondante est une fausse alarme, et le fait de lui attribuer l'identifiant $n + 1$ symbolise l'apparition d'une nouvelle cible. De cette manière, l'algorithme MHT permet de suivre un nombre de cibles qui évolue au cours du temps.

Les arcs sont pondérés par la probabilité qu'une détection corresponde à une cible. Les zones de confiances des cibles (les ellipses sur la figure 5.1a) modélisent des distributions gaussiennes. Celles-ci sont calculées grâce à un filtre de Kalman. Ainsi, les probabilités d'affectation sont simplement calculées comme des densités de probabilité d'une loi normale dont les paramètres sont représentés par l'ellipse correspondante.



(a) Disposition des cibles et des mesures.

(b) Ensemble des hypothèses représentées sous forme d'arbre.

FIG. 5.1 – Graphe généré par l'algorithme MHT (adapté de [Reid, 1979]).

Dans un tel graphe, le nombre de sommets croît de manière exponentielle. La recherche de la meilleure interprétation de l'ensemble des détections traitées à un instant donné revient à rechercher dans l'arbre un flot maximum entre la racine et l'une des feuilles. Ce problème de théorie des graphes est connu pour être NP-complet. De nombreuses méthodes d'élagage ont été proposées au fil des années pour réduire le nombre d'hypothèses afin de pouvoir exécuter cet algorithme en temps raisonnable. Le lecteur intéressé pourra se référer à [Cox, 1993] pour un aperçu de certaines d'entre elles.

Ainsi, le MHT a constitué la première proposition de modélisation des déplacements d'objets dans une scène par un graphe. Le modèle proposé est élégant d'un point de vue théorique, permet d'intégrer de nouvelles cibles aussi bien que de supprimer celles qui sortent du champ de vision, et offre la possibilité de revenir sur une décision d'affectation lorsque celle-ci s'avère erronée. Malheureusement, cette méthode est trop lente pour être appliquée en temps réel. Néanmoins, la représentation sous forme de graphe est une idée qui a perduré et des auteurs ont proposé d'autres schémas plus compatibles avec les contraintes de performances que l'on connaît en analyse de séquences vidéo.

Par la suite, de nombreux auteurs ont proposé des méthodes de suivi d'objets n'utilisant pas de graphes pour modéliser les déplacements. On pourra citer entre autres l'algorithme JPDA (*Joint-Probabilistic Data Association*) [Bar-Shalom et Fortmann, 1988], ou encore le suivi bayésien [Genovesio, 2005]. Cependant, le caractère multi-hypohèses du suivi à l'aide de graphes n'est pas présent dans ces méthodes.

Certains auteurs ont cherché à préserver ce caractère intéressant en proposant de nouvelles structures de graphes dans lesquelles la recherche des trajectoires optimales est moins coûteuse en temps de calcul. Par exemple, dans [Cohen et Medioni, 1999] le graphe créé possède un sommet par détection. Dans ce cas, une détection est une région d'avant-plan fournie par un algorithme de détection de mouvement, et possède donc plus de caractéristiques mesurables que les détections ponctuelles traitées précédemment. Les sommets du graphe sont disposés en étages, un étage correspondant à une image de la séquence vidéo (un instant t). La figure 5.2 représente un exemple de graphe qui aurait pu être généré par cet algorithme. Les arcs possèdent un poids qui

est la vraisemblance du fait que les deux régions connectées correspondent au même objet. Dans cet article, la fonction de vraisemblance fait intervenir la corrélation des valeurs de niveaux de gris des deux régions et leur distance spatiale.

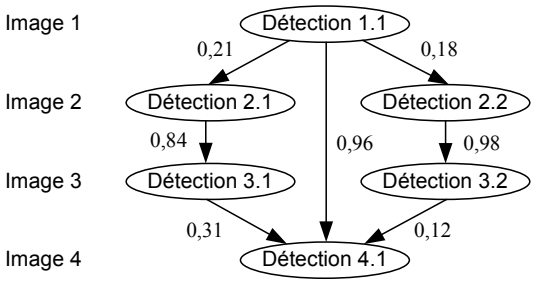


FIG. 5.2 – Exemple de graphe généré par l'algorithme de [Cohen et Medioni, 1999].

Les auteurs de [Cohen et Medioni, 1999], considèrent qu'une composante connexe de l'image binaire représente au plus un objet, par conséquent, la recherche des trajectoires optimales dans le graphe consiste à rechercher dans chaque composante connexe du graphe, le chemin maximisant à la fois le nombre de sommets qu'il contient et la somme des vraisemblances qui pondèrent ses arcs. Ainsi, cette méthode ne gère pas les cas où deux objets seraient temporairement agrégés dans une seule région de l'image binarisée.

Les travaux de [Chia *et al.*, 2006] utilisent la même structure de graphe, mais la mise en œuvre est différente. La connexité du graphe est réduite en supprimant dans un premier temps tous les arcs qui ne respectent pas des contraintes de similarité très strictes basées sur la couleur des régions considérées ainsi que leur position dans le plan-image et leur surface. Dans cette première phase, on interdit également qu'un sommet ait un demi-degré intérieur ou extérieur supérieur à 1, c'est-à-dire que les composantes connexes du graphe obtenu correspondent nécessairement à des chemins. Dans un deuxième temps, les chemins obtenus sont interconnectés si les arcs à réintégrer pour y parvenir respectent des contraintes de similarité plus souples que précédemment. L'interprétation des composantes connexes obtenues après cette phase se base sur l'hypothèse qu'un objet mobile est représenté par au plus une région d'avant-plan. Ceci permet de prendre en compte les fusions temporaires d'objets au sein d'une même région.

Notre méthode se base sur une structure de graphe similaire à celle utilisée dans [Cohen et Medioni, 1999] ou [Chia *et al.*, 2006] et sur une stratégie d'élagage originale. Dans la section suivante, nous allons présenter le graphe utilisé et nous allons décrire les informations que nous avons jugé nécessaire de retenir dans les sommets et les arcs pour gérer le plus grand nombre de cas.

5.2 Graphe d'association

Le graphe que nous utiliserons pour modéliser les déplacements des objets dans la scène est basé sur la même architecture que celui de [Cohen et Medioni, 1999] et [Chia *et al.*, 2006]. Nous l'appellerons « graphe d'association » car chacun de ses arcs modélise l'hypothèse que les deux sommets ainsi liés correspondent à des détections issues du même objet. Autrement dit, dans ce graphe, un arc *associe* deux détections entre elles. Nous allons maintenant proposer une définition théorique d'un tel graphe.

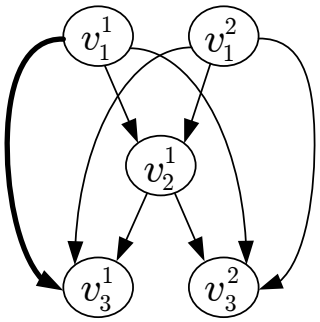
5.2.1 Définition

Notons $G = [V, E]$ un graphe orienté, avec V l'ensemble de ses sommets et E l'ensemble de ses arcs. Celui-ci est associé à un ensemble de N images binaires qui représentent la segmentation entre l'arrière-plan et les objets mobiles dans une séquence vidéo découpée en N séquences élémentaires. À chaque composante connexe de ces images binaires, on associe un sommet $v \in V$. Lorsque nous construirons une représentation graphique de G, les sommets seront disposés en étages correspondant à une séquence élémentaire (c'est-à-dire à un instant donné), et les arcs seront orientés du haut vers le bas, comme sur la figure 5.2. Si l'on note M_t le nombre de régions en mouvement détectées à l'instant t, les sommets du graphe seront notés v_t^i avec $1 \leq i \leq M_t$ et $1 \leq t \leq N$.

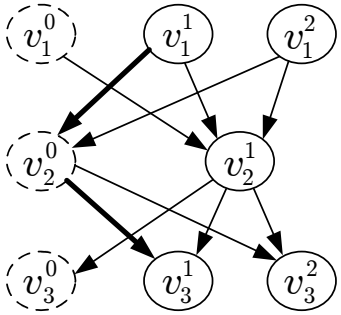
Nous avons vu que la présence d'un arc $e = (u, v)$ modélise l'hypothèse que les sommets u et v $(u, v \in V)$ sont associés à des détections d'un même objet. En dehors de toute contrainte, le nombre d'arcs dans le graphe peut donc être au maximum $|E| = (|V| - 1)|V|$. La figure 5.3a représente le graphe que l'on pourrait obtenir par cette méthode pour une séquence imaginaire de longueur 3. Dans le cadre d'une application temps réel, nous ne pouvons pas accepter que le nombre d'arcs du graphe que nous allons devoir analyser soit une fonction quadratique du nombre de détections. Pour cette raison, nous ne prendrons en compte que les associations entre régions détectées à des instants adjacents, c'est-à-dire que nous ne créerons des arcs qu'entre les paires de sommets situés sur des étages voisins du graphe.

Pour modéliser le cas des apparitions et des disparitions d'objets, nous proposons d'ajouter un sommet supplémentaire à chaque étage t du graphe, que l'on notera v_t^0. Ce nouveau sommet représente une abscence de détection au temps t. Ainsi, l'arc $e = (v_{t-1}^i, v_t^0)$, avec $1 \leq i \leq M_{t-1}$, modélisera le fait que l'objet représenté par la i-ème détection au temps $t - 1$ a disparu au temps t. Symétriquement, l'arc $e' = (v_t^0, v_{t+1}^j)$, avec $1 \leq j \leq M_{t+1}$ modélisera le fait que l'objet représenté par la j-ème détection au temps $t + 1$ vient d'apparaître.

Les sommets nuls ainsi créés seront reliés à tous les sommets non-nuls des étages directement inférieur et supérieur à celui sur lequel ils se trouvent. En effet, un sommet nul modélisant une abscence de détection, un arc entre deux sommets nuls n'aurait pas de sens. La figure 5.3b représente le graphe ainsi modifié associé à la même séquence imaginaire de longueur 3 que précédem-



(a) Graphe obtenu lorsqu'on crée toutes les associations entre détections, quel que soit l'intervalle de temps qui les sépare.



(b) Graphe obtenu après l'introduction des « sommets nuls » (en pointillés).

FIG. 5.3 – Graphes obtenus pour une même séquence selon deux architectures différentes.

ment.

On constatera que cette représentation permet également de modéliser les disparitions temporaires d'objets. En effet, l'arc en gras sur la figure 5.3a correspond à un tel cas, et l'introduction des sommets nuls permet de modéliser la même hypothèse (représentée par le chemin en gras dans la figure 5.3b) sans nécessiter que le graphe ne comporte un nombre d'arcs trop important.

Nous avons donc défini une structure de graphe dans laquelle chaque sommet (hormis les sommets nuls) correspond à une région d'avant-plan fournie par le module de détection de mouvement. Entre chaque paire de sommets correspondant à des dates adjacentes est créé un arc modélisant l'hypothèse que les deux sommets qu'il relie correspondent au même objet. Les arcs sont orientés dans le sens des dates croissantes. Pour extraire du graphe les trajectoires optimales, nous allons devoir pondérer les arcs avec une mesure de vraisemblance des hypothèses qu'ils modélisent. Cette mesure de vraisemblance sera nécessairement fonction des caractéristiques des régions auxquelles elles se réfèrent. C'est pourquoi, dans la section suivante, nous allons présenter les différentes caractéristiques mesurées sur les régions d'avant-plan, et associées aux sommets correspondants.

5.2.2 Description des sommets

Nous avons pour l'instant deux informations qui permettent d'identifier un sommet $v_t^i \in V$: la date t à laquelle a été détectée la région associée, et l'indice i ($1 \leq i \leq M_t$) de cette région dans l'ensemble de celles qui ont été détectées à la même date. Afin de mesurer la vraisemblance de l'hypothèse que deux sommets représentent le même objet, il faut que nous relevions des caractéristiques supplémentaires.

Le problème consistant à établir une correspondance entre deux régions d'images a été beaucoup étudié dans des domaines divers tels que le calcul du

flot optique [Camus, 1995] déjà évoqué au chapitre 2, la vision stéréo [Sun, 2002] ou encore la recherche d'images similaires dans des bases de données [Ke *et al.*, 2004]. Dans ces domaines, les techniques les plus utilisées sont le calcul d'un coefficient de corrélation entre les imagettes à comparer et la recherche de vecteurs caractéristiques en des points saillants des deux images.

L'utilisation de ces techniques est problématique dans le cas de l'application qui nous intéresse. Tout d'abord, comme la segmentation des régions en mouvement est nécessairement imparfaite, une partie des zones dont on dispose représentera l'arrière-plan de la scène, ce qui met à mal les méthodes de calcul de corrélation. De plus, ces méthodes sont difficilement applicables lorsque l'on compare deux régions de forme différente, ce qui est fréquent lors du suivi d'un objet non rigide. D'autre part, la recherche de vecteurs caractéristiques tels que les vecteurs SIFT (*Scale-Invariant Feature Transforms*) [Lowe, 2004] est bien trop coûteuse en temps de calcul pour pouvoir envisager de l'appliquer entre toutes les paires de régions adjacentes dans le temps, dans le cadre d'une application temps réel. Cette dernière méthode a également la particularité d'échouer lorsque la taille des régions analysées est trop petite.

Nous préférons relever un ensemble de mesures relatives à l'apparence des régions ainsi qu'à leur position dans le plan-image, afin de pouvoir effectuer des comparaisons simples et rapides qui permettront, lorsqu'on les considérera toutes, d'obtenir une estimation robuste de la similarité des régions considérées.

Ainsi, pour chaque région d'avant-plan, nous relèverons les caractéristiques suivantes :

- les coordonnées, dans le repère de l'image, du centre de gravité des points qui la composent ;
- son aire, c'est-à-dire le nombre de points qu'elle contient ;
- l'ensemble des sommets de son enveloppe convexe (le plus petit polygone convexe qui peut la contenir entièrement), qui constitue un descripteur de forme compact ;
- une représentation compacte de la distribution des couleurs dans la région.

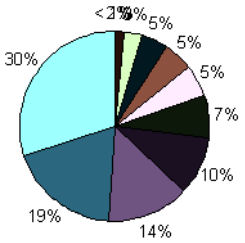
Concernant le dernier point, plusieurs options s'offrent à nous. En sortie d'une caméra vidéo, les couleurs des pixels sont représentées par le triplet des valeurs de rouge, de vert et de bleu (espace RGB). Chaque composante est quantifiée en 256 niveaux, ce qui implique que l'on peut rencontrer plus de 16 millions de couleurs différentes dans une image numérique. Pour représenter de manière compacte la distribution des couleurs dans une région, il est nécessaire de sous-échantillonner l'espace dans lequel les couleurs sont représentées. En général, cette opération n'est pas réalisée dans l'espace RGB car celui-ci n'est pas le plus cohérent avec la perception humaine des couleurs.

Il existe de nombreux espaces de représentation des couleurs plus adaptés, dont un panorama peut être trouvé dans [Trémeau *et al.*, 2004]. Quand le but du sous-échantillonnage est la comparaison d'imagettes, il a été observé dans [Lee *et al.*, 2005] que l'espace fournissant les meilleurs résultats était l'espace CIELab, composé d'une composante de luminance et de deux composantes de

chrominance. Dans ce même article, il est suggéré que dans l'espace CIELab, pour un nombre de classes de valeurs données, il est judicieux de segmenter davantage les composantes chromatiques que la composante achromatique. Par exemple, si l'on décide de sélectionner 512 couleurs élémentaires, il est préférable d'échantillonner la composante L (luminance) en deux classes et les composantes a et b (chrominance) en 16 classes chacune, plutôt que d'échantillonner les trois composantes en huit classes chacune. La figure 5.4 représente un exemple de région d'avant-plan représentée dans l'espace RGB d'origine (256 classes pour chaque composante), et la même image sous-échantillonnée dans l'espace CIELab avec 4 valeurs de luminance et 8×8 valeurs de chrominance.

(a) Espace RGB proposant 256^3 couleurs.

(b) Espace CIELab proposant $4 \times 8 \times 8$ couleurs.



(c) Distribution des couleurs dans l'espace CIELab.

FIG. 5.4 – Exemple d'image avant et après le sous-échantillonnage de ses couleurs.

En observant la distribution des couleurs élémentaires dans l'espace CIELab sous-échantillonné (figure 5.4c), on constate que certaines couleurs sont présentes en proportion importante après le traitement, alors qu'elles semblaient être absentes de l'image dans l'espace RGB. Cela s'explique par le fait que dans les zones de l'image où la luminosité et/ou la saturation en couleur sont faibles, les valeurs de chrominance n'ont pas de sens.

Pour cette raison, nous proposons de représenter les images dans un espace couleur quantifié dont une partie des classes sera réservée aux niveaux de gris, et le reste aux couleurs. L'apparence d'un pixel sera représentée par un niveau de gris si sa saturation ou sa luminance sont trop faibles. Il existe plusieurs définitions de la saturation et de la luminance dans la littérature. Dans [Lefèvre et Vincent, 2006], les auteurs conseillent d'utiliser le modèle de [Travis, 1991] car il fournit des résultats équivalents à ceux obtenus avec d'autres modèles plus coûteux en temps de calcul, comme celui de [Gonzalez et Woods, 1992] par exemple. Si l'on note R, G et B les composantes couleur d'un pixel, la luminance en ce point est définie par

$$L = \max(R, G, B), \tag{5.1}$$

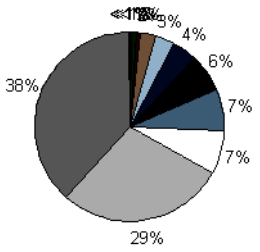
et la saturation par

$$S = \frac{L - \min(R, G, B)}{L}. \tag{5.2}$$

Ainsi, en fixant un seuil θ_L de luminance minimale et un seuil θ_S de saturation minimale, on pourra choisir pour chaque point de la région considérée s'il doit être représenté par une couleur ou par un niveau de gris. La figure 5.5 représente la même image que précédemment, dont les couleurs sont exprimées dans un espace à 260 valeurs d'apparence, soit 4 niveaux de gris et 256 couleurs. Les 256 couleurs ont été obtenues en sous-échantillonnant l'espace CIELab avec 4 valeurs de luminance et 8×8 valeurs de chrominance. Les seuils θ_L et θ_S ont été fixés à 0,1.

(a) Résultat du seuillage de la luminance et de la saturation. Les pixels achromatiques sont représentés en noir.

(b) Espace proposé, permettant de représenter $4 \times 8 \times 8 + 4$ couleurs.



(c) Distribution des couleurs dans l'espace proposé.

FIG. 5.5 – Sous-échantillonnage d'une image couleur dans l'espace proposé.

On peut constater sur la figure 5.5c que notre algorithme classe la majeure partie des pixels de la région considérée comme étant achromatiques, ce qui correspond bien à notre perception visuelle de l'image d'origine.

Nous utiliserons donc pour modéliser l'apparence des régions un histogramme des couleurs et des niveaux de gris comme nous venons de le définir.

Maintenant que nous avons présenté les différentes caractéristiques relevées sur les régions en mouvement détectées et associées aux sommets du graphe, nous allons proposer un ensemble de mesures de similarité entre régions qui permettront de pondérer les arcs.

5.2.3 Description des arcs

Dans notre graphe d'association, tous les sommets non-nuls correspondant à des détections obtenues à des dates adjacentes sont reliés par un arc orienté dans le sens des dates croissantes. Dans l'objectif de l'extraction des trajectoires optimales, nous devons associer à chacun de ces arcs une ou plusieurs mesures de similarité qui permettront d'estimer la vraisemblance de l'hypothèse selon laquelle les deux sommets connectés par un arc représentent le même objet.

Afin d'exploiter au mieux les caractéristiques mesurées sur les régions en mouvement, nous allons calculer une mesure de similarité pour chaque valeur dont nous disposons.

La première caractéristique que nous avons relevée est la localisation du centre de gravité des régions. En effet, lorsqu'un objet se déplace à une vitesse raisonnable, on peut supposer que son centre de gravité se déplacera peu entre deux prises de vue consécutives. Par conséquent, la première pondération w_1 de l'arc $e = (u, v)$ que nous calculerons est la distance euclidienne entre les centres de gravité des régions associées aux sommets u et v, soit

$$w_1(e) = \left\| \frac{1}{|\mathcal{P}_u|} \sum_{\mathbf{x} \in \mathcal{P}_u} \mathbf{x} - \frac{1}{|\mathcal{P}_v|} \sum_{\mathbf{x} \in \mathcal{P}_v} \mathbf{x} \right\|_2, \tag{5.3}$$

avec $\mathbf{x} = (x, y)$, et où $\mathcal{P}_u$ (respectivement $\mathcal{P}_v$) est l'ensemble des couples de coordonnées des pixels de la région associée au sommet u (respectivement v).

La seconde caractéristique associée à un sommet v est l'aire en pixels de la région correspondante, soit $|\mathcal{P}_v|$. Même en présence d'un effet de perspective modéré ou d'une légère déformation, la surface d'un objet ne devrait pas varier de manière trop significative entre deux détections successives. Ainsi, la seconde pondération w_2 associée à l'arc $e = (u, v)$ sera le rapport entre les surfaces des régions associées aux sommets u et v, défini par

$$w_2(e) = \frac{\min(|\mathcal{P}_u|, |\mathcal{P}_v|)}{\max(|\mathcal{P}_u|, |\mathcal{P}_v|)}. \tag{5.4}$$

Nous avons ensuite relevé l'enveloppe convexe $\mathcal{E}_v$ de la région associée à chaque sommet v. Dans le cas où l'on cherche à détecter des objets rigides, leur forme ne devrait pas varier de manière importante entre deux détections consécutives. Dans le cas d'objets non rigides, cette hypothèse risque d'être moins souvent vérifiée. Il faudra donc donner moins d'importance à ce critère lorsque l'on cherche à suivre des objets non rigides tels que des personnes par exemple.

Il existe plusieurs manières d'estimer la similarité de deux formes polygonales. En raison de son faible coût en temps de calcul, nous avons choisi la méthode présentée dans [Arkin *et al.*, 1991]. Celle-ci consiste à définir pour un polygone $\mathcal{E}$ une fonction angulaire θ qui associe à toute valeur d'abscisse curviligne s appartenant à l'intervalle $[0, 1]$ la direction angulaire de la tangente à la forme en ce point, là où elle est définie. Cette fonction est constante par morceaux.

Soient deux polygones $\mathcal{E}_u$ de fonction angulaire θ_u et $\mathcal{E}_v$ de fonction angulaire θ_v. La distance L_p entre $\mathcal{E}_u$ et $\mathcal{E}_v$ est définie par

$$\delta_p(\mathcal{E}_u, \mathcal{E}_v) = \sqrt[p]{\int_0^1 |\theta_u(s) - \theta_v(s)|^p \, \mathrm{d}s}. \tag{5.5}$$

La distance ainsi définie est sensible à la fois aux rotations des polygones et aux points de référence pris pour définir les abscisses curvilignes. Il est donc plus logique de prendre en compte la distance minimum parmi tous les choix

possibles, soit

$$d_p(\mathcal{E}_u, \mathcal{E}_v) = \sqrt[p]{\min_{\substack{s_0\in[0,1]\\ \theta_0\in[0,2\pi]}} \int_0^1 |\theta_u(s+s_0) - \theta_v(s+s_0) + \theta_0|^p \ \mathrm{d}s}. \tag{5.6}$$

Les auteurs ont montré que la distance euclidienne entre deux formes polygonales peut être définie en résolvant le problème de minimisation à une seule variable suivant :

$$\begin{aligned} d_2(\mathcal{E}_u, \mathcal{E}_v) = \Bigg[\min_{s_0\in[0,1]} \Bigg\{ & \int_0^1 (\theta_u(s+s_0) - \theta_v(s))^2 \ \mathrm{d}s \\ & - \left(\int_0^1 \theta_v(s)\,\mathrm{d}s - \int_0^1 \theta_u(s)\,\mathrm{d}s - 2\pi s_0 \right)^2 \Bigg\} \Bigg]^{\frac{1}{2}}. \end{aligned} \tag{5.7}$$

Nous venons de définir la troisième fonction de pondération qui à tout arc $e = (u, v)$ associe la valeur $w_3(e) = d_2(\mathcal{E}_u, \mathcal{E}_v)$.

La dernière caractéristique associée aux sommets du graphe est l'histogramme des couleurs de la région correspondante. Nous noterons $\mathcal{H}_v$ l'histogramme de la région associée au sommet v, et $h_v(i)$, avec $1 \leq i \leq \dim(\mathcal{H}_v)$, la i-ème composante de $\mathcal{H}_v$. Dans notre application, tous les histogrammes calculés auront le même nombre de classes N_H. Normalement, l'histogramme des couleurs d'une région ne devrait pas trop changer d'une détection à l'autre. Il existe de nombreuses métriques pour comparer des histogrammes couleur. L'intersection des histogrammes constitue une mesure robuste et fréquemment utilisée dans la littérature. Nous définirons donc une quatrième fonction de pondération w_4 qui associe au sommet $e = (u, v)$ la valeur

$$w_4(e) = \sum_{i=1}^{N_H} \frac{\min\left(h_u(i), h_v(i)\right)}{\min(|\mathcal{P}_u|, |\mathcal{P}_v|)}. \tag{5.8}$$

Nous venons donc de définir quatre mesures de similarité entre régions en mouvement qui participeront au calcul du critère évaluant la vraisemblance de l'hypothèse selon laquelle deux régions représentent le même objet.

Comme nous l'avons remarqué précédemment, la recherche d'un ensemble de trajectoires optimales dans un graphe est souvent un problème d'une grande complexité algorithmique. Afin de pouvoir réaliser cette tâche en un temps raisonnable, nous avons dû mettre au point une stratégie de suppression d'un maximum d'arêtes. Cette stratégie est présentée dans la section suivante.

5.3 Stratégie d'élagage

Nous disposons d'un graphe $G = [V, E]$ dont l'ensemble des sommets est arrangé en étages. Toutes les paires de sommets non-nuls situés sur des étages voisins sont reliées par un arc auquel sont associées quatre mesures de similarité. De ce graphe, nous souhaitons extraire une information à valeur sémantique consituée :

– du nombre d'objets en déplacement dans la scène modélisée,
– de l'ensemble des trajectoires de chacun des objets,
– de l'ensemble des interactions (fusion ou séparation) qui ont lieu entre les objets.

Comme chaque arc modélise l'hypothèse que les deux sommets qu'il relie sont associés à des régions qui représentent le même objet, il est nécessaire de décider pour chaque arc si l'hypothèse qu'il modélise est vraie ou fausse. On cherche donc le sous-ensemble $F \subset E$ des arcs modélisant les hypothèses vraies. Plutôt que de partir de l'ensemble E et d'en supprimer un à un tous les arcs qui modélisent une hypothèse que nous estimons fausse, nous allons partir de l'ensemble vide et y ajouter, dans un premier temps, les arcs dont la véracité des hypothèses qu'ils modélisent ne fait aucun doute. Ils consitueront l'ensemble $F_1 \subset F$. Puis, en se basant sur les arcs de F_1, nous déduirons quels arcs parmi ceux qui restent, peuvent être validés, constituant ainsi l'ensemble F_2 tel que $F_1 \cup F_2 = F$.

Il s'agit donc d'une stratégie en deux phases — une phase de validation des évidences, et une phase d'extrapolation de celles-ci. Dans la section suivante, nous allons présenter la permière phase de sélection des arcs qui consitueront l'ensemble F_1.

5.3.1 Première phase : associations évidentes

Nous cherchons à déterminer l'ensemble $F_1 \subset E$ des arcs de G qui représentent des hypothèses indiscutables. À chaque arc $e = (u, v)$, on peut associer quatre valeurs numériques $w_1(e)$ (équation 5.3), $w_2(e)$ (équation 5.4), $w_3(e)$ (équation 5.7) et $w_4(e)$ (équation 5.8), qui, selon des critères différents, mesurent la simlarité ou la dissimilarité des régions associées aux sommets u et v.

Plutôt que de rechercher les poids d'une combinaison linéaire de toutes ces mesures, nous proposons de définir de manière expérimentale quatre seuils t_1, t_2, t_3 et t_4 suffisamment stricts pour que l'ensemble F_1 défini par

$$F_1 = \{e \in E \,|\, w_1(e) < t_1 \wedge w_2(e) < t_2 \wedge w_3(e) < t_3 \wedge w_4(e) > t_4\} \qquad (5.9)$$

ne contienne aucune hypothèse fausse. Ces seuils peuvent être déterminés, par exemple, en utilisant une partie de la séquence à étudier comme un ensemble d'apprentissage. Nous noterons $G_1 = [V, F_1]$ le graphe partiel engendré par F_1.

Il est évident que si l'on choisit ces seuils de manière à n'avoir aucun « faux positif » dans F_1, un grand nombre d'hypothèses justes seront écartées par le test de l'équation 5.9. Comme les hpothèses que l'on traite correspondent à des déplacements d'objets, on peut s'attendre à observer une certaine cohérence entre les différentes hypothèses qui concernent un même objet. F_1 traduit des mouvements de courte durée sur lesquels on va baser un apprentissage permettant de les prolonger. Cela suppose qu'il existe un *modèle de mouvement* que les objets respectent. Afin de compléter le graphe partiel G_1 avec les arcs de $(E \setminus F_1)$ qui sont associés à des hypothèses cohérentes avec celles qui sont

représentées par F_1, nous allons chercher les hypothèses qui respectent un modèle de mouvement observable à partir de G_1. Cette phase est décrite dans la section suivante.

5.3.2 Deuxième phase : extrapolation

À l'abord de cette phase, nous disposons de fragments de déplacements modélisés par les chemins du graphe $G_1 = [V, F_1]$. Nous avons supposé que nous pouvions extrapoler ces déplacements incomplets en nous basant sur le modèle de mouvement qui leur est inhérent. La difficulté à laquelle nous sommes confronté vient du fait que nous ne savons *a priori* pas à quel type de modèle de mouvement nous attendre. Il en existe différents types, comme le modèle à vitesse constante, le modèle à accélération constante, ou encore le modèle brownien. Afin de faire un choix, nous devons nous placer dans le contexte d'application de la méthode, qui est celui de la vidéosurveillance. De ce fait, nous pouvons exclure le modèle de mouvement brownien que l'on rencontre essentiellement lorsque l'on analyse le déplacement de particules dans un fluide.

Avant de choisir entre un modèle de mouvement à vitesse constante et un modèle à accélération constante, nous devons prendre en compte la manière dont nous allons estimer le mouvement.

Nous avons affaire à des objets dont l'état est constitué de caractristiques d'apparence et de position dont nous avons obtenu une mesure (observation bruitée) dans la section 5.2.2, mais aussi de caractéristiques de mouvement qui nous sont inconnues (vitesse et accélération). Nous cherchons à estimer l'ensemble de ces caractéristiques (vecteur d'état) en nous basant uniquement sur les observations bruitées de certaines d'entre elles (vecteur de mesure). C'est exactement pour résoudre ce type de problèmes qu'a été conçu le filtrage de Kalman ([Kalman, 1960]).

Le problème du filtrage de Kalman consiste à considérer que l'on observe un processus régi par l'équation stochastique linéaire suivante :

$$\boldsymbol{\theta}_t = \mathbf{A} \cdot \boldsymbol{\theta}_{t-1} + \boldsymbol{\nu}_{t-1}, \tag{5.10}$$

où :

- $\boldsymbol{\theta}_t$ est le vecteur d'état de notre processus au temps t, c'est-à-dire l'ensemble des caractéristiques cachées de l'objet mobile qui participent à définir son déplacement ;
- $\mathbf{A}$ est une matrice décrivant la relation linéaire entre l'état du processus à un instant donné, et son état à l'instant suivant (appelée « matrice de transition ») ;
- $\boldsymbol{\nu}_t$ est un vecteur aléatoire de bruit supposé blanc et gaussien ($\boldsymbol{\nu}_t \sim N(\mathbf{0}, \mathbf{Q})$), qui modélise l'inexactitude de la relation linéaire décrite par $\mathbf{A}$ (on parle de « bruit de système »).

On constate donc que la méthode du filtrage de Kalman suppose deux hypothèses fortes : (1) la fonction qui lie l'état actuel du système à son état précédent est linéaire, et (2) on autorise le système à dévier de ce modèle à

condition que l'écart observé suive une loi gaussienne centrée en zéro et dont la matrice de variance-covariance $\mathbf{Q}$ est connue. Nous verrons par la suite que ces hypothèses ne sont pas en contradiction avec la définition de notre problème.

Comme nous l'avons dit précédemment, notre système (objet mobile) possède une partie observable qui est liée à son état. Nous avons associé ces mesures bruitées aux sommets non-nuls du graphe G. La relation qui lie l'état du système à l'observation qui en est faite est définie par

$$\mathbf{z}_t = \mathbf{H}\boldsymbol{\theta}_t + \boldsymbol{\mu}_t, \tag{5.11}$$

où :

- $\mathbf{z}_t$ est le vecteur des mesures relevées au temps t ;
- $\mathbf{H}$ est une matrice décrivant la relation linéaire qui existe entre l'état du système et la mesure qui en est faite (appelée « matrice de mesure ») ;
- $\boldsymbol{\mu}_t$ est un vecteur aléatoire de bruit supposé blanc et gaussien ($\boldsymbol{\mu}_t \sim N(\mathbf{0}, \mathbf{R})$), qui modélise les erreurs de mesure, c'est-à-dire l'inexactitude de la relation linéaire décrite par $\mathbf{H}$ (on parle de « bruit de mesure »).

L'équation 5.11 révèle elle aussi de fortes hypothèses : la relation qui lie l'état du système à un instant donné à la mesure qui en est faite au même instant, est linéaire à un bruit additif blanc et gaussien près, et la matrice de variance-covariance $\mathbf{R}$ du bruit est connue.

Que le modèle de mouvement que l'on cherche à estimer soit à vitesse constante ou à accélération constante, la relation qui lie l'état d'un objet à un instant donné à l'état du même objet à l'instant précédent, est toujours linéaire. En effet, si l'on note (x_t, y_t) la position de l'objet suivi au temps t, $(\dot{x}_t, \dot{y}_t)$ sa vitesse, et $(\ddot{x}_t, \ddot{y}_t)$ son accélération, la transition entre états consécutifs est régie par le système d'équations

$$\begin{cases} x_t = x_{t-1} + \dot{x}_{t-1} + \nu_1 \\ y_t = y_{t-1} + \dot{y}_{t-1} + \nu_2 \\ \dot{x}_t = \dot{x}_{t-1} + \nu_3 \\ \dot{y}_t = \dot{y}_{t-1} + \nu_4 \end{cases}, \tag{5.12}$$

dans le cas d'un modèle à vitesse constante où le vecteur d'état est $\boldsymbol{\theta}_t = [x_t\ y_t\ \dot{x}_t\ \dot{y}_t]^T$, et par le système d'équations

$$\begin{cases} x_t = x_{t-1} + \dot{x}_{t-1} + \ddot{x}_{t-1} + \nu_1 \\ y_t = y_{t-1} + \dot{y}_{t-1} + \ddot{y}_{t-1} + \nu_2 \\ \dot{x}_t = \dot{x}_{t-1} + \ddot{x}_{t-1} + \nu_3 \\ \dot{y}_t = \dot{y}_{t-1} + \ddot{y}_{t-1} + \nu_4 \\ \ddot{x}_t = \ddot{x}_{t-1} + \nu_5 \\ \ddot{y}_t = \ddot{y}_{t-1} + \nu_6 \end{cases}, \tag{5.13}$$

dans le cas d'un modèle à accélaration constante où le vecteur d'état est $\boldsymbol{\theta}_t = [x_t\ y_t\ \dot{x}_t\ \dot{y}_t\ \ddot{x}_t\ \ddot{y}_t]^T$. Le bruit de système $\boldsymbol{\nu}_t = [\nu_1\ \cdots\ \nu_{\dim(\boldsymbol{\theta}_t)}]^T$ est dû au fait qu'un objet peut modifier sa trajectoire en cours de suivi. Comme on ne peut pas préjuger de la direction que prendra l'objet, il est normal de

supposer que $\boldsymbol{\nu}_t$ a une moyenne nulle. Sa covariance $\mathbf{Q}$ peut être estimée de manière expérimentale sur une séquence d'apprentissage. Nous pouvons donc nous placer dans un cadre où les hypothèses sous-jacentes à l'équation 5.10 sont respectées.

En ce qui concerne les valeurs mesurées sur les objets, nous disposons de la position du centre de gravité, de l'aire, de l'enveloppe convexe, et de l'histogramme des couleurs. Parmi toutes ces observations, seule la position du centre de gravité intervient dans le modèle de mouvement d'un objet. L'observation bruitée de l'état de l'objet suivi est donc constituée des coordonnées de son centre de gravité. Le bruit de mesure est dû à l'imprécision de la segmentation en objets mobiles. Ici encore, on peut écrire un système linéaire d'équations qui décrit la relation entre l'état du système et la mesure, soit

$$\begin{cases} \tilde{x}_t = x_t + \mu_1 \\ \tilde{y}_t = y_t + \mu_2 \end{cases}, \tag{5.14}$$

où $\mathbf{z}_t = (\tilde{x}_t, \tilde{y}_t)$ est la position mesurée du centre de gravité de l'objet suivi. On remarquera que le système 5.14 est valable quel que soit le modèle de mouvement considéré. L'hypothèse selon laquelle le bruit de mesure $\boldsymbol{\mu}_t = [\mu_1 \ \mu_2]^T$ est blanc et gaussien semble, ici aussi, justifiée.

Nous pouvons maintenant choisir entre un modèle à vitesse constante et un modèle à accélération constante. Nous savons que les séquences vidéo que nous allons étudier sont des scènes de vidéosurveillance. Nous pouvons donc nous attendre à observer des êtres humains et des véhicules. Rien n'empêche ce type d'objets d'avoir une accélération non nulle. On peut même imaginer le cas d'un objet se déplaçant à vitesse constante, mais dont la projection dans le plan-image possède une accélération due à un effet de perspective. Cependant, l'accélération que l'on observerait devrait rester modérée, et il est peu probable qu'elle reste constante. Comme le filtrage de Kalman permet de prendre en compte l'erreur de modélisation, nous allons opter pour un modèle de mouvement à vitesse constante, en admettant qu'une éventuelle accélération observée serait intégrée au bruit de mesure $\boldsymbol{\nu}_t$.

Ainsi, le processus observé est régi par le système d'équations 5.12, ce qui revient à définir la matrice de transition de l'équation 5.10 par

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{5.15}$$

La relation entre le vecteur d'état et les mesures observées est décrite par le système d'équations 5.14, ce qui revient à définir la matrice de mesure de l'équation 5.11 par

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{5.16}$$

Un filtre de Kalman permet d'estimer l'état caché du processus observé compte tenu d'un ensemble d'observations en calculant itérativement à chaque

instant t une estimation *a priori* de l'état $\hat{\boldsymbol{\theta}}_t^-$ (étape de prédiction), puis, lorsque la mesure $\mathbf{z}_t$ est connue, une estimation *a posteriori* de l'état $\hat{\boldsymbol{\theta}}_t^+$ (étape de correction). Les erreurs d'estimation *a priori* et *a posteriori* sont gaussiennes, soit

$$p(\boldsymbol{\theta}_t|\mathbf{z}_1,\ldots,\mathbf{z}_{t-1}) \sim N(\hat{\boldsymbol{\theta}}_t^-,\mathbf{P}_t^-), \tag{5.17}$$

où $\mathbf{P}_t^-$ est la matrice de variance-covariance de l'erreur d'estimation *a priori*, et

$$p(\boldsymbol{\theta}_t|\mathbf{z}_1,\ldots,\mathbf{z}_t) \sim N(\hat{\boldsymbol{\theta}}_t^+,\mathbf{P}_t^+), \tag{5.18}$$

où $\mathbf{P}_t^+$ est la matrice de variance-covariance de l'erreur d'estimation *a posteriori*.

Sans entrer dans le détail des justifications probabilistes et calculatoires de l'algorithme (cf. [Welch et Bishop, 1995]), nous dirons simplement que l'étape de prédiction consiste à bâtir l'estimation *a priori* et à mettre à jour l'erreur d'estimation correspondante de la manière suivante :

$$\hat{\boldsymbol{\theta}}_t^- = \mathbf{A}\cdot\hat{\boldsymbol{\theta}}_{t-1}^+ \tag{5.19}$$

$$\mathbf{P}_t^- = \mathbf{A}\cdot\mathbf{P}_{t-1}^+\cdot\mathbf{A}^T + \mathbf{Q}. \tag{5.20}$$

De manière analogue, l'étape de correction constiste à bâtir l'estimation *a posteriori* et à mettre à jour l'erreur d'estimation correspondante de la manière suivante :

$$\hat{\boldsymbol{\theta}}_t^+ = \hat{\boldsymbol{\theta}}_t^- + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\cdot\hat{\boldsymbol{\theta}}_t^-) \tag{5.21}$$

$$\mathbf{P}_t^+ = \mathbf{P}_t^- - \mathbf{K}_t\cdot\mathbf{H}\cdot\mathbf{P}_t^-, \tag{5.22}$$

où $\mathbf{K}_t$ est appelée « matrice de gain de Kalman », et définie par

$$\mathbf{K}_t = \mathbf{P}_t^-\cdot\mathbf{H}^T\cdot(\mathbf{H}\cdot\mathbf{P}_t^-\cdot\mathbf{H}^T + \mathbf{R})^{-1}. \tag{5.23}$$

Nous pouvons maintenant utiliser le filtre que nous avons défini pour tenter d'extrapoler les déplacements partiels modélisés par les chemins de G_1. Cette opération permettra de considérer comme vraies de nouvelles hypothèses modélisées par des arcs de $(E \setminus F_1)$. Les arcs ainsi validés constituent l'ensemble F_2.

Plus précisément, pour chaque chemin $P^{(k)} = [V^{(k)}, E^{(k)}]$ de G_1, nous allons considérer que l'état initial du processus est constitué par la position du centre de gravité de la région modélisée par le premier sommet de $P^{(k)}$, et par le déplacement moyen de cette région le long de $P^{(k)}$. Nous allons ensuite exécuter une étape de prédiction et une étape de correction pour chaque sommet restant de $P^{(k)}$ de manière à apprendre le modèle de mouvement inhérent. Nous effectuerons ensuite une étape de prédiction pour obtenir la première position extrapolée $(\hat{x}, \hat{y})$ du centre de l'objet modélisé. S'il existe une région dont l'enveloppe convexe contient ce point (soit v' le sommet correspondant), on ajoute à F_2 l'arc reliant le puits actuel v de $P^{(k)}$ à v'. Le sommet v' se trouve forcément dans l'ensemble des successeurs de v dans G, noté $\Gamma(v)$. La

Algorithme 5.1 : Extrapolation des déplacements partiels vers l'avant.

```
   /* Initialiser l'ensemble des arcs validés dans cette
      phase :                                                    */
1  F_2 ← ∅;
2  pour chaque chemin P^(k) = [V^(k), E^(k)] de G_1 faire
       /* Notons N = |V^(k)|.                                   */
       /* Notons V^(k) = {v_1^(k), ..., v_N^(k)}.               */
       /* Notons {(x_1, y_1); ...; (x_N, y_N)} la suite des coordonnées
          des centres de gravité des régions associées aux
          éléments de V^(k).                                    */
       /* Créer un état initial.                                */
3      θ̂_1^+ ← [x_1  y_1  (x_N − x_1)/N  (y_N − y_1)/N]^T;
4      pour t ← 2 à N faire
           /* Effectuer une étape de prédiction :              */
5          Calculer θ̂_t^− par l'équation 5.19;
6          Calculer P_t^− par l'équation 5.20;
           /* Effectuer une étape de correction :              */
7          Calculer K_t par l'équation 5.23;
8          Calculer θ̂_t^+ par l'équation 5.21 en remplaçant z_t par [x_t y_t]^T;
9          Calculer P_t^+ par l'équation 5.22;
10     Soit v = v_N^(k);
       /* Effectuer une étape de prédiction :                  */
11     Calculer θ̂_{N+1}^− par l'équation 5.19;
12     Calculer P_{N+1}^− par l'équation 5.20;
       /* Notons θ̂_{N+1}^− = [x̂  ŷ  x̂̇  ŷ̇]^T.                */
13     si ∃v' ∈ Γ(v) tel que (x̂, ŷ) est dans E_v alors
           /* Notons (x', y') les coordonnées du centre de gravité
              de la région associée à v'.                       */
           /* Effectuer une étape de correction :              */
14         Calculer K_{N+1} par l'équation 5.23;
15         Calculer θ̂_{N+1}^+ par l'équation 5.21 en remplaçant z_{N+1} par
           [x' y']^T;
16         Calculer P_{N+1}^+ par l'équation 5.22;
           /* Ajouter (v, v') à l'ensemble des arcs validés dans
              cette phase :                                     */
17         F_2 ← F_2 ∪ {(v, v')};
18         N ← N + 1;
19         Soit v = v';
20         Aller à la ligne 11;
```

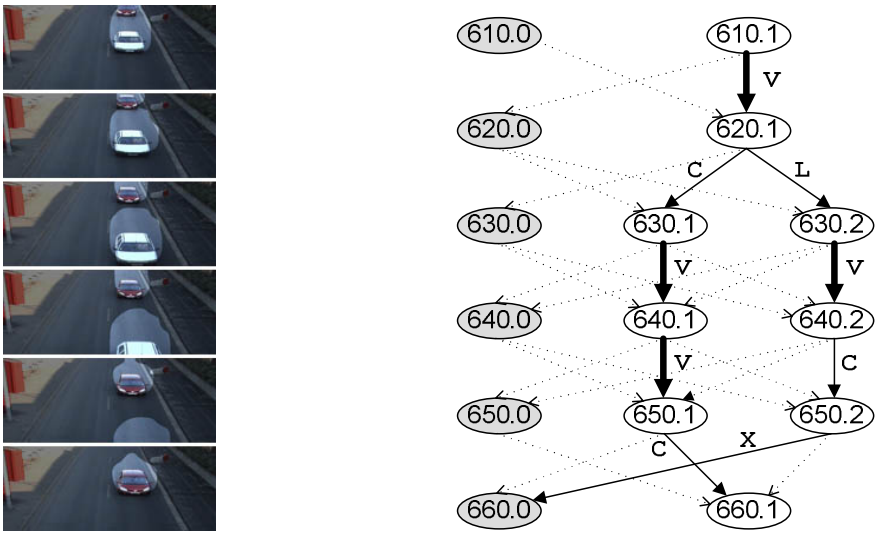
procédure d'extrapolation est réitérée tant que l'on trouve des sommets pour

corroborer les positions prédites. L'algorithme 5.1 résume cette opération.

Symétriquement, nous allons extrapoler les chemins de G_1 en considérant ses arcs dans leur orientation inverse, de manière à récupérer le passé des déplacements partiels contenus dans G_1. Les arcs ainsi validés sont eux aussi ajoutés à F_2.

Après cette étape, nous pouvons considérer que nous avons exploité au maximum les informations fiables de G en sélectionnant l'ensemble $F = F_1 \cup F_2$ des arcs modélisant les hypothèses les plus plausibles. Appelons G' le graphe possédant les mêmes sommets que G, mais uniquement les arcs de F : $G' = [V, F]$.

La figure 5.6 représente un exemple de graphe ainsi créé pour une séquence simple. Les sommets sont étiquetés avec la date à laquelle la région correspondante a été détectée et le numéro d'ordre de celle-ci parmi toutes les détections réalisées à cette date (format : `Date.Numero`). On remarquera que des sommets portent un numéro d'ordre égal à 0 : il s'agit des « sommets nuls » présentés à la section 5.2.1. Les arcs en pointillés correspondent aux hypothèses considérées comme fausses, autrement dit, ce sont les éléments de $(E \setminus F)$. Les arcs en gras correspondent aux hypothèses validées dès la première phase du traitement car elles respectent les critères décrits par l'équation 5.9 : ce sont les éléments de F_1. Les arcs pleins sont associés à des hypothèses considérées comme vraies car, bien qu'elles ne respectent pas les critères de l'équation 5.9, elles ont été déduites par propagation du mouvement : ce sont les éléments de F_2.



(a) Séquence d'entrée. Les zones surexposées sont les régions en mouvement détectées.

(b) Graphe généré à partir de cette séquence.

FIG. 5.6 – Exemple de simplification d'un graphe d'association pour une séquence simple.

On remarquera également sur la figure 5.6 que certains arcs sont étiquetés

par une lettre. Il s'agit de l'interprétation du graphe en termes de trajectoires d'objet. Celle-ci est détaillée dans la section suivante.

5.4 Interprétation

Dans les sections précédentes, nous avons créé un graphe d'association à partir d'une séquence vidéo, puis nous l'avons simplifié en supprimant tous les arcs correspondant à des hypothèses considérées comme fausses. Ce graphe est porteur d'information à plusieurs niveaux. Dans un premier temps, nous allons exploiter les événements notés pendant la phase d'extrapolation des chemins élémentaires en termes de départs, d'arrêts, d'entrées et de sorties du champ de vision des objets suivis. Dans un second temps, nous présenterons comment la structure même du graphe nous renseigne sur le nombre d'objets présents dans la scène. Enfin, nous détaillerons la procédure d'attribution d'identifiants uniques aux objets et leur localisation dans le plan de l'image.

5.4.1 Débuts et fins de déplacement

Dans la section 5.3.2, nous avons présenté la procédure d'extrapolation des chemins élémentaires dans le sens des arcs (vers le futur) et dans le sens inverse (vers le passé). Un chemin élémentaire est prolongé par extrapolation tant que l'on trouve une détection qui coïncide avec l'estimation fournie par le filtre de Kalman. Dans le cas contraire, la procédure prend fin et cet événement est porteur de sens. Lors d'une extrapolation vers le futur, plusieurs cas peuvent se présenter :

- Le filtre de Kalman prédit que le centre de gravité de l'objet suivi va se trouver en un lieu du domaine de l'image, et à l'instant considéré, il n'existe aucune détection dont l'enveloppe convexe contient ce point. Dans ce cas, nous interpréterons cet événement comme le fait que l'objet suivi s'est arrêté (ou en tout cas, que son mouvement n'est plus détecté).
- Le filtre de Kalman prédit que le centre de gravité de l'objet suivi va sortir du domaine de l'image. On ne pourra donc pas trouver de détection contenant le point inféré, et on interprétera cet événement comme le fait que l'objet suivi a quitté le champ de vision de la caméra. Il existe un tel cas dans le graphe de la figure 5.6b : l'arc allant du sommet étiqueté « 650.2 » au sommet « 660.0 » correspond au fait que la voiture blanche de la figure 5.6a est sortie du champ de vision.

Symétriquement, lors d'une extrapolation vers le passé, des cas similaires se produisent :

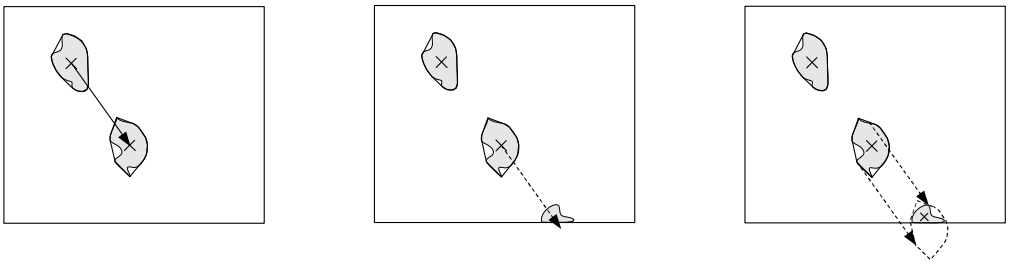
- Le filtre de Kalman prédit que le centre de gravité de l'objet suivi va se trouver en un lieu du domaine de l'image, et à l'instant considéré, il n'existe aucune détection dont l'enveloppe convexe contient ce point. Dans ce cas, nous interpréterons cet événement comme le fait que l'objet suivi a démarré, c'est-à-dire qu'il était déjà présent dans l'image avant

d'être détecté.

– Le filtre de Kalman prédit que le centre de gravité de l'objet suivi va sortir du domaine de l'image. On ne pourra donc pas trouver de détection contenant le point inféré, et on interprétera cet événement comme le fait que l'objet suivi vient d'entrer dans le champ de vision de la caméra.

Ces événements sont les premières informations à valeur sémantique que nous pouvons obtenir à partir du graphe d'association. Leur intérêt est double : elles pourront être transmises au module de niveau supérieur et elles peuvent nous permettre de corriger d'éventuelles erreurs de suivi.

En effet, considérons le cas d'un objet qui s'apprête à sortir du champ de vision comme sur la figure 5.7a. Bien souvent, lorsque cet objet est détecté pour la dernière fois, la région correspondante ne le représente qu'en partie (figure 5.7b), car il se trouve partiellement hors de la portée de la caméra. Comme l'algorithme d'extrapolation cherche à prédire la position du centre de gravité des objets, il est possible que celui-ci détecte une sortie d'objet, alors que l'objet en question est encore partiellement visible.



(a) Deux dernières détections d'un objet mobile et enveloppes convexes correspondantes.

(b) L'algorithme de la section 5.3.2 détecte une sortie d'objet.

(c) L'algorithme proposé ici permet d'associer à l'objet la dernière détection.

FIG. 5.7 – Correction du suivi basée sur l'interprétation du graphe.

Nous avons donc implémenté une mesure corrective à ce comportement non voulu. Dès que la procédure d'extrapolation génère un événement du type « sortie d'un objet », nous construisons une estimation $\hat{\mathcal{E}}$ de l'enveloppe convexe de la détection que l'on *aurait pu* associer à l'objet suivi à l'aide de l'état estimé du filtre de Kalman, soit

$$\hat{\mathcal{E}} = \{(x + \hat{\dot{x}}, y + \hat{\dot{y}}), (x, y) \in \mathcal{E}_v\}, \tag{5.24}$$

où les notations sont celles de l'algorithme 5.1, avec v le dernier sommet associé à l'objet suivi. Nous recherchons ensuite si à l'instant considéré, il existe une détection dont le centre de gravité est contenu dans $\hat{\mathcal{E}}$; auquel cas, l'arc pointant sur le sommet correspondant est validé. Cette opération corrective est illustrée par la figure 5.7c.

Un algorithme analogue existe pour la face d'extrapolation inverse (vers le passé). Celui-ci permet de s'assurer que la première détection générée par un

objet lui a bien été associée. Il est exécuté dès que l'algorithme d'extrapolation révèle un événement de type « entrée d'un objet ».

5.4.2 Nombre d'objets mobiles

Afin de déterminer le nombre d'objets que représente chaque sommet, un choix doit être fait dans l'interprétation du graphe obtenu :

- Soit nous considérons, comme dans [Cohen et Medioni, 1999], qu'une détection ne peut correspondre qu'à un objet, mais qu'un objet peut générer plusieurs détections : dans ce cas, deux objets dont les détections sont temporairement fusionnées seront considérés comme une seule entité.
- Soit nous considérons, comme dans [Chia *et al.*, 2006], qu'un objet peut générer au plus une détection, mais qu'une détection peut représenter plusieurs objets : dans ce cas, si un objet est temporairement fractionné en plusieurs détections, chaque partie sera identifiée comme une entité à part entière.

Nous considérons que la deuxième option est moins restrictive que la première. En effet, comme nous cherchons à définir un module de suivi d'objets qui soit adaptable à différents types d'objets et à différentes prises de vue, nous ne disposons pas d'information *a priori* sur la forme ou l'apparence des objets suivis. Cependant, le module de niveau supérieur — qui exploite les données générées par celui-ci — possède ces informations et pourra, si nécessaire, opérer des fusions si des entités suivies sont, par exemple, trop petites pour constituer un objet à part entière.

Notre but est donc de définir une fonction ψ qui à chaque sommet v du graphe $G' = [V, F]$ associe un nombre d'objets représentés, qui est un entier positif. Ce nombre peut être déduit directement de la structure du graphe et de l'hypothèse posée dans le paragraphe précédent.

Commençons par observer que les composantes connexes de G' peuvent être traitées indépendamment pour déterminer les valeurs prises par ψ. Le type de composantes connexes le plus simple est celui constitué d'un seul sommet et d'aucun arc. Il s'agit des détections qui n'ont été associées à aucun objet et qui peuvent donc être considérées comme du bruit. On peut en déduire une première règle :

$$\forall v \in V \quad d(v) = 0 \;\Rightarrow\; \psi(v) = 0, \tag{5.25}$$

où $d(\cdot)$ est la fonction qui associe à un sommet son degré dans G'.

Dans le cas où, dans le graphe, la composante connexe considérée est un chemin, d'après l'hypothèse posée plus haut, la fonction ψ prendra la valeur 1 pour chacun de ses sommets.

Les autres composantes connexes présentent des « fourches », c'est-à-dire que certains de leurs sommets ont plusieurs successeurs et/ou plusieurs prédécesseurs. Dans ce cas, il faut respecter une contrainte de cohérence, analogue aux lois de Kirchhoff dans le domaine de l'électricité, que l'on peut exprimer

ainsi :

$$\forall v \in V \quad \sum_{v' \in \Gamma(v)} \psi(v') = \sum_{v' \in \Gamma^{-1}(v)} \psi(v'), \tag{5.26}$$

où $\Gamma(\cdot)$ est l'application qui associe à un sommet l'ensemble de ses successeurs, et $\Gamma^{-1}(\cdot)$ est l'application réciproque qui lui associe l'ensemble de ses prédécesseurs.

Pour modéliser cette contrainte, nous allons utiliser la notion de *vecteur de flot*, bien connue en théorie des graphes. Un flot dans un graphe est un vecteur ϕ de nombres réels dont le nombre de composantes est égal au nombre d'arcs du graphe, et tel que, pour tout sommet, la première loi de Kirchhoff soit vérifiée, c'est-à-dire :

$$\forall v \in V \quad \sum_{e \in \omega^+(v)} \phi_e = \sum_{e \in \omega^-(v)} \phi_e, \tag{5.27}$$

où $\omega^+(v)$ désigne l'ensemble des arcs ayant v pour extrémité initiale, et $\omega^-(v)$ désigne l'ensemble des arcs ayant v pour extrémité terminale.

Comme les composantes connexes possèdent toutes au moins un sommet sans prédécesseur (apparition du premier objet concerné), et au moins un sommet sans successeur (disparition du dernier objet concerné), il n'existe pas de flot non-nul compatible avec leur topologie. Pour résoudre ce problème, nous ajoutons à chaque composante connexe $C^{(k)} = [V^{(k)}, E^{(k)}]$ de G' un sommet s (source) que l'on relie à chaque sommet de $C^{(k)}$ dépourvu de prédecesseur, et un sommet t (puits) que l'on relie à chaque sommet de $C^{(k)}$ dépourvu de successeur. On peut ensuite « fermer le circuit », pour reprendre l'analogie avec les circuits électriques, en reliant t à s. On a donc modifié chaque composante connexe $C^{(k)}$ pour obtenir un graphe $\tilde{C}^{(k)} = [\tilde{V}^{(k)}, \tilde{E}^{(k)}]$ tel que

$$\tilde{V}^{(k)} = V^{(k)} \cup \{s, t\}, \tag{5.28}$$

et

$$\tilde{E}^{(k)} = E^{(k)} \cup \{(s, v) | d^-(v) = 0\} \cup \{(v, t) | d^+(v) = 0\} \cup \{(t, s)\}, \tag{5.29}$$

où $d^-(v)$ désigne le demi-degré intérieur de v dans $C^{(k)}$, et $d^+(v)$ désigne le demi-degré extérieur de v dans $C^{(k)}$.

Ainsi, pour définir les valeurs prises par ψ pour les sommets d'une composante connexe $C^{(k)} = [V^{(k)}, E^{(k)}]$ de G', nous pouvons commencer par déterminer le flot minimum ϕ dans $\tilde{C}^{(k)}$ compatible avec la contrainte suivante :

$$\forall e \in \tilde{E}^{(k)} \quad \phi_e \geq 1, \tag{5.30}$$

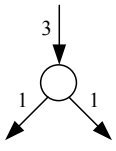
autrement dit, chaque arc représente le déplacement d'au moins un objet.

Le flot ϕ peut être déterminé de manière itérative par l'algorithme 5.2. Il consiste à initialiser toutes les composantes du flot à 1, puis, pour chaque sommet où la contrainte de l'équation 5.27 n'est pas respectée, à augmenter la quantité de flot (ou flux) entrant ou sortant selon la nature du déséquilibre observé.

Algorithme 5.2 : Définition du flot d'objets pour une composante connexe modifiée $\tilde{C}^{(k)} = [\tilde{E}^{(k)}, \tilde{V}^{(k)}]$ de G'.

```
   /* Initialiser tous les flux à 1 :                                  */
 1 pour tous les e ∈ Ẽ^(k) faire φ_e ← 1;
 2 tant que ¬Stop faire
 3     Stop ← Vrai;
 4     pour tous les v ∈ Ṽ^(k) faire
           /* Calculer la différence entre les flux entrant et
              sortant :                                                 */
 5         Soit Δφ = Σ_{e∈ω⁻(v)} φ_e − Σ_{e∈ω⁺(v)} φ_e;
 6         tant que Δφ ≠ 0 faire
 7             si Δφ < 0 alors
                   /* Il y a un déficit en flux entrant.               */
                   /* Augmenter le flux venant du prédecesseur
                      ayant le plus grand rapport surface/nombre
                      d'objets :                                        */
 8                 Soit v* = arg max_{v'∈Γ⁻¹(v)} |P_{v'}| / φ_(v',v);
 9                 φ_(v*,v) ← φ_(v*,v) + 1;
10                 Δφ ← Δφ + 1;
11             sinon si Δφ > 0 alors
                   /* Il y a un déficit en flux sortant.               */
                   /* Augmenter le flux vers le successeur ayant le
                      plus grand rapport surface/nombre d'objets :
                      */
12                 Soit v* = arg max_{v'∈Γ(v)} |P_{v'}| / φ_(v,v');
13                 φ_(v,v*) ← φ_(v,v*) + 1;
14                 Δφ ← Δφ − 1;
               /* Le flot a été modifié : il faut vérifier à
                  nouveau.                                              */
15             Stop ← Faux;
```

On remarquera qu'on a parfois le choix entre plusieurs arcs dont on peut augmenter le flux, comme illustré par la figure ci-contre. Ici, nous avons un flux entrant égal à 3, et un flux sortant égal à 2. Il faut donc augmenter le flux sortant d'une unité. Pour ce faire, nous avons le choix entre deux arcs. Dans un tel cas, si l'on note v le nœud central, on choisira d'incrémenter le flux de l'arc (v, v^*), où v^* est défini par



$$v^* = \arg \max_{v' \in \Gamma(v)} \frac{|\mathcal{P}_{v'}|}{\phi_{(v,v')}}, \tag{5.31}$$

c'est-à-dire que l'on considère que l'arc dont le flux a le plus besoin d'être augmenté est celui qui mène au sommet dont l'aire moyenne est la plus grande (en tenant compte du flux actuel).

Une fois ce flot déterminé, le nombre d'objets représentés par un sommet se déduit simplement de la quantité de flot incidente en chaque sommet. En effet, après l'exécution de l'algorithme 5.2, le flot ϕ dans la composante connexe modifiée $\tilde{G}^{(k)}$ est défini et non-nul, donc nous pouvons définir pour tout sommet v de $\tilde{G}^{(k)}$ le nombre d'objets $\psi(v)$ qu'il représente par

$$\psi(v) = \sum_{e \in \omega^-(v)} \phi_e = \sum_{e \in \omega^+(v)} \phi_e. \tag{5.32}$$

À l'issue de cette opération, nous avons une fonction ψ qui associe à chaque sommet du graphe G' le nombre d'objets qu'il représente. Afin de finaliser l'opération de suivi des objets mobiles, il reste à attribuer un identifiant unique à chaque objet et à associer une liste d'identifiants à chaque sommet. Dans le cas où plusieurs objets seraient représentés par une même région, il serait intéressant de localiser chacun d'entre eux au sein de la région. Cette dernière opération est présentée dans la section suivante.

5.4.3 Identification des objets

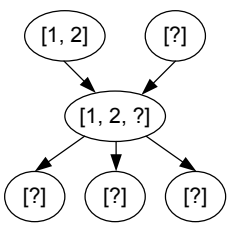
Nous allons ici attribuer à chaque sommet v une liste d'identifiants correspondant aux $\psi(v)$ objets représentés par la région associée à v. Nous allons procéder par propagation dans le sens des arcs du graphe G'. Simultanément, pour les régions représentant plusieurs objets, nous voulons localiser chaque objet identifié au sein de la région dans laquelle il se trouve. La localisation des objets est réalisée par la détermination de leurs rectangles circonscrits.

Comme dans la section précédente, nous allons traiter les composantes connexes de G' individuellement, puisqu'un même objet ne peut pas se trouver dans deux composantes connexes différentes.

L'algorithme proposé consiste à considérer chaque sommet sans prédécesseur de la composante connexe courante $C^{(k)}$, c'est-à-dire ceux qui représentent l'apparition d'un objet. À chacun de ces sommets nous attribuons autant de nouveaux identifiants que le sommet représente d'objets.

Pour propager les identifiants aux successeurs de ces sommets, il faut considérer toutes les situations qui peuvent se présenter, la plus simple étant celle d'un sommet à un seul successeur. Dans ce cas, les identifiants sont simplement recopiés dans le successeur.

Le parcours du graphe ne peut pas se faire « en profondeur d'abord », comme pour un étiquetage des composantes connexes par exemple. En effet, avant de propager les identifiants d'un sommet, il faut que cette opération ait été réalisée pour l'ensemble de ses prédécesseurs, comme illustré sur la figure ci-contre, où les sommets sont étiquetés avec la



liste des identifiants des objets qu'ils contiennent. Dans cet exemple, on constate que tant que l'on n'aura pas exécuté les opérations de propagation pour tous les prédécesseurs du sommet central, il est inutile de vouloir propager les identifiants de celui-ci.

Le cas le plus délicat est celui des sommets possédant plusieurs successeurs. Il existe de nombreuses situations de difficulté variable. La figure 5.8 en représente deux.

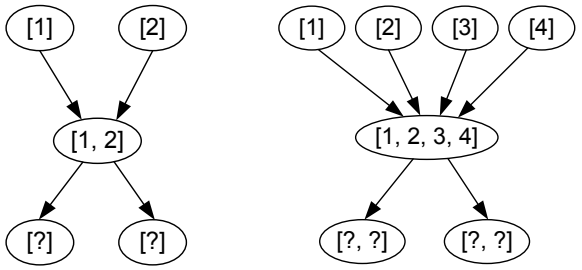


FIG. 5.8 – Deux situations dans lesquelles le recours aux caractéristiques d'apparence est nécessaire pour identifier les objets.

Ce type de problème ne peut pas être résolu en utilisant exclusivement la topologie du graphe. En effet, dans le cas illustré par la figure 5.8a, il est nécessaire d'avoir recours aux caractéristiques d'apparence des régions concernées pour décider des affectations à réaliser. Plusieurs options s'offrent à nous pour y parvenir. Nous disposons des quatre mesures de similarité présentées à la section 5.2.3 ; nous pouvons donc envisager d'utiliser une combinaison linéaire de celles-ci pour faire un choix.

Cependant, la situation décrite par la figure 5.8b nous incite à exclure cette possibilité. Cet exemple illustre le fait qu'il faut faire la distinction entre l'apparence d'une *région* et l'apparence d'un *objet*. En effet, le calcul des mesures de similarité définies à la section 5.2.3 entre un sommet ne représentant qu'un objet et un sommet représentant plusieurs objets ne permet pas de clarifier la situation de la figure 5.8b. Il nous faut une méthode permettant de rechercher une imagette (l'une des régions associées aux quatre sommets ne représentant qu'un seul objet) dans une image plus grande. Durant la phase de validation des associations évidentes (section 5.3.1), nous nous étions refusé à calculer un ensemble de vecteurs caractérisant des points saillants de l'image car cette approche, coûteuse en temps de calcul, ne semblait pas compatible avec les contraintes de temps réel que l'on connaît en analyse vidéo.

Ici, la situation est différente. Un grand nombre d'arcs ont été supprimés lors de phase d'élagage, et le type de cas décrit par la figure 5.8 devrait être plutôt rare puisqu'il s'agit de la séparation d'objets qui avaient préalablement été fusionnés. Dans ces conditions, nous pouvons nous permettre d'utiliser une méthode plus coûteuse en temps de calcul. Nous avons choisi de rechercher des appariements affines de vecteurs SIFT [Lowe, 2004] entre les régions concernées. Cette procédure peut être décrite par les étapes suivantes :

- rechercher des points saillants (extrema locaux) dans les deux images ;
- calculer un descripteur local (SIFT) en chacun de ces points ;
- considérer tous les couples de points dont la distance euclidienne des descripteurs locaux est inférieure à un certain seuil ;
- rechercher la transformation affine qui utilise le maximum de couples de points sélectionnés précédemment.

Pour une description complète de cette méthode, le lecteur pourra se référer à [Auclair *et al.*, 2007].

Ainsi, chaque association potentielle se verra attribuer un score correspondant au nombre de couples de points saillants qui appuient la matrice de transformation affine calculée. Nous privilégierons les affectations qui obtiennent les scores les plus élevés.

L'algorithme que nous venons de décrire permet de suivre les objets mobiles dans des séquences vidéo de difficulté variable, même lorsque des fusions/séparations d'objets ont lieu. Dans la section suivante, nous allons tester notre algorithme sur différentes séquences de test afin de mesurer ses performances et de déterminer ses limites.

5.5 Résultats

L'évaluation des performances d'un système de suivi d'objets mobiles dans une séquence vidéo est une tâche complexe qui nécessite la définition de métriques mettant en jeu des notions spécifiques à l'analyse vidéo, telles que la persistance temporelle par exemple. Nous présenterons dans un premier temps les métriques utilisées, puis nous évaluerons notre algorithme sur un ensemble de séquences.

5.5.1 Métriques

Les premières méthodologies pour l'évaluation des systèmes de suivi visuel automatique n'ont été publiées que très récemment mais semblent néanmoins converger vers un ensemble restreint de métriques. Nous utiliserons la méthodologie proposée dans [Smith *et al.*, 2005] et enrichie dans [Spangenberg et Döring, 2006].

Il est important de noter que les métriques utilisées interviennent à deux niveaux d'analyse : au niveau statique et au niveau dynamique. Les métriques utilisées au niveau statique consistent à comparer image par image, le résultat de l'algorithme de suivi et la vérité terrain. Ces mesures sont assez similaires à celles que nous avions utilisées à la section 4.4 pour évaluer notre méthode de détection de mouvement, à la différence près qu'ici, les quantités comparées sont exprimées en nombre d'objets plutôt qu'en nombre de pixels.

Dans toutes les méthodologies proposées, les objets sont repérés dans le plan de l'image par leur rectangle englobant, et non par l'ensemble des pixels qui les représentent. Cela est probablement dû à la formidable quantité de

travail que représente l'annotation manuelle de séquences vidéo nécessaire à l'obtention d'une vérité terrain.

Pour reprendre la notation de [Smith *et al.*, 2005], nous noterons $\mathcal{GT}_j$ le j-ème objet de la vérité terrain (GT pour *ground truth*), et $\mathcal{E}_i$ le i-ème objet détecté par le système (E pour *estimate*). Leurs représentations (rectangle englobant) à l'instant t seront notées respectivement $\mathcal{GT}_j^t$ et $\mathcal{E}_i^t$.

Comme pour l'évaluation de la détection de mouvement, les deux métriques de base utilisées pour l'évaluation statique du suivi d'objets sont le rappel et la précision. On définit donc le rappel $\rho_{i,j}^t$ comme la proportion de l'objet réel $\mathcal{GT}_j$ qui est couverte par l'objet détecté $\mathcal{E}_i$ à l'instant t, soit

$$\rho_{i,j}^t = \frac{|\mathcal{E}_i^t \cap \mathcal{GT}_j^t|}{|\mathcal{GT}_j^t|}, \tag{5.33}$$

et la précision $\nu_{i,j}$ comme la proportion de l'objet détecté $\mathcal{E}_i$ qui couvre l'objet réel $\mathcal{GT}_j$ à l'instant t, soit

$$\nu_{i,j}^t = \frac{|\mathcal{E}_i^t \cap \mathcal{GT}_j^t|}{|\mathcal{E}_j^t|}. \tag{5.34}$$

Cela nous permet de définir le *test de couverture* qui détermine conjointement si un objet réel est détecté et si un objet détecté correspond à un objet réel, autrement dit, s'il y a ou non association entre un objet réel et un objet détecté. La réponse est affirmative si la F-mesure dépasse un certain seuil t_C. La F-mesure est définie par

$$F_{i,j}^t = \frac{2\nu_{i,j}^t \rho_{i,j}^t}{\nu_{i,j}^t + \rho_{i,j}^t}. \tag{5.35}$$

On notera τ_j^t le booléen indiquant si l'objet réel $\mathcal{GT}_j$ a été détecté au temps t ($F_{i,j}^t > t_C$).

Grâce au test de couverture, nous disposons de trois données pour chaque image :

- une liste d'objets réels représentés par $\{\mathcal{GT}_j^t\}$ avec $1 \leq j \leq \#\mathcal{GT}^t$, où $\#\mathcal{GT}^t$ est le nombre d'objets réels à l'instant t ;
- une liste d'objets détectés représentés par $\{\mathcal{E}_i^t\}$ avec $1 \leq i \leq \#\mathcal{E}^t$, où $\#\mathcal{E}^t$ est le nombre d'objets détectés à l'instant t ;
- une matrice de booléens $(\alpha_{i,j}^t)$ de taille $\#\mathcal{E}^t \times \#\mathcal{GT}^t$ où $\alpha_{i,j}^t$ vaut 1 si le test de couverture est positif entre l'objet réel j et l'objet détecté i, et 0 sinon.

Ceci nous permet d'établir deux tableaux de contingence, du point de vue de la réalité terrain, et du point de vue du système. Le tableau 5.1a fournit pour chaque objet détecté la liste des objets réels qui lui sont associés. Idéalement, cette liste doit être de longueur 1. Une liste vide révèle un premier type d'erreurs : les *faux positifs* (FP), comme dans le cas de l'objet détecté identifié « 4 ». À l'opposé, une liste de longueur supérieure à 1 révèle une erreur de type *objets multiples* (OM), comme dans le cas de l'objet détecté identifié « 5 ».

Le tableau 5.1b fournit pour chaque objet réel la liste des objets détectés qui lui sont associés. Cette liste doit également être de longueur 1. Une liste vide révèle un *faux négatif* (FN), comme dans le cas de l'objet réel identifié « c ». À l'opposé, une liste de longueur supérieure à 1 révèle une erreur de type *détections multiples* (DM), comme dans le cas de l'objet réel identifié « b ».

$\mathcal{E}$	1	2	3	4	5
$\mathcal{GT}$	a	b	b	-	d,e

(a) Du point de vue de la vérité terrain.

$\mathcal{GT}$	a	b	c	d	e
$\mathcal{E}$	1	2,3	-	5	5

(b) Du point de vue du système de suivi.

TAB. 5.1 – Tableaux de contingence entre objets réels et objets détectés.

Le cas des occlusions entre objets nécessite un traitement spécifique. En effet, lorsqu'un objet en cache un autre temporairement, le comportement attendu d'un système de suivi est de détecter deux objets au même endroit. De ce fait, le système d'évaluation décrit ci-dessus génére deux erreurs OM et deux erreurs DM. Afin d'éviter cela, on définit en chaque instant t et pour tout objet réel $\mathcal{GT}_j$ un booléen β_j^t qui vaut 1 si l'objet j est caché, et 0 sinon. On dira que deux objets réels sont en situation d'occlusion si l'aire de leur intersection dépasse un seuil t_O.

Ainsi, dans [Smith *et al.*, 2005] sont définis quatre types d'erreurs qui, lorsqu'elles sont acculumées le long d'une séquence, permettent de comparer les performances de différents systèmes de suivi. Cependant, si l'on souhaite comparer les résultats obtenus sur différentes séquences, la variabilité du nombre d'images par séquence, et du nombre d'objets qui entrent en scène fait que ces mesures ne sont pas comparables d'une séquence à l'autre. Les auteurs proposent une méthode de normalisation qui prend en compte les particularités de chaque séquence. Pour une séquence de n images, les valeurs normalisées de FP, FN, OM et DM sont notées $\overline{FP}$, $\overline{FN}$, $\overline{OM}$ et $\overline{DM}$, et définies par

$$\overline{FP} = \frac{1}{n}\sum_{t=1}^{n}\frac{FP^t}{\max(1,\#\mathcal{GT}^t)}, \tag{5.36}$$

$$\overline{FN} = \frac{1}{n}\sum_{t=1}^{n}\frac{FN^t}{\max(1,\#\mathcal{GT}^t)}, \tag{5.37}$$

$$\overline{OM} = \frac{1}{n}\sum_{t=1}^{n}\frac{OM^t}{\max(1,\#\mathcal{GT}^t)} \text{ et} \tag{5.38}$$

$$\overline{DM} = \frac{1}{n}\sum_{t=1}^{n}\frac{DM^t}{\max(1,\#\mathcal{GT}^t)}. \tag{5.39}$$

Dans [Spangenberg et Döring, 2006] est introduite la notion d'erreur de localisation EL_j^t de l'objet réel j au temps t, définie comme la distance euclidienne entre le centre de gravité de l'objet réel j et le centre de gravité de

l'objet détecté qui lui est associé. Cette grandeur n'est définie que pour les objets qui ont effectivement été détectés au temps t (tels que $\tau_j^t = 1$). Cela nécessite d'annoter la vérité terrain avec une valeur supplémentaire : le centre de gravité des objets.

En ce qui concerne les métriques permettant d'évaluer le système de suivi d'un point de vue dynamique, elles mesurent la persistance de l'association entre un objet réel et un objet détecté. En effet, il est possible que le long d'une séquence, un objet réel $\mathcal{GT}_j$ soit associé pendant un certain temps à l'objet détecté $\mathcal{E}_{i_1}$, puis à l'objet détecté $\mathcal{E}_{i_2}$ pendant le reste de la séquence. Aussi, on définira l'indice $\hat{i}_j$ comme l'indice de l'objet détecté qui est le plus souvent associé au j-ème objet réel. Symétriquement, on définira l'indice $\hat{j}_i$ comme l'indice de l'objet réel le plus souvent associé au i-ème objet détecté.

Ce formalisme permet de définir deux nouveaux types d'erreurs :

- On comptera une erreur de *détection mal affectée* (DMA) à chaque instant où un objet détecté $\mathcal{E}_i$ est associé à un autre objet réel que $\mathcal{GT}_{\hat{j}_i}$ (pour chaque $\alpha_{i,j}^t = 1$ avec $j \neq \hat{j}_i$).
- On comptera une erreur d'*objet mal identifié* (OMI) à chaque instant où un objet réel $\mathcal{GT}_j$ est associé à un autre objet détecté que $\mathcal{E}_{\hat{i}_j}$ (pour chaque $\alpha_{i,j}^t = 1$ avec $i \neq \hat{i}_j$).

Comme pour les métriques d'évaluation statique, ces valeurs doivent être normalisées sur l'ensemble de la séquence pour pouvoir comparer les résultats obtenus avec des séquences différentes. Pour une séquence de longueur n, les valeurs normalisées de DMA et OMI sont notées $\overline{DMA}$ et $\overline{OMI}$, et définies par

$$\overline{DMA} = \frac{1}{n} \sum_{t=1}^{n} \frac{DMA^t}{\max(1, \#\mathcal{GT}^t)} \text{ et} \tag{5.40}$$

$$\overline{OMI} = \frac{1}{n} \sum_{t=1}^{n} \frac{OMI^t}{\max(1, \#\mathcal{GT}^t)}. \tag{5.41}$$

Les auteurs de [Smith *et al.*, 2005] définissent enfin la notion de *pureté* pour les objets réels comme pour les objets détectés, qui mesure le degré de cohérence dans la manière dont un objet réel est identifié, ou la manière dont un objet détecté est affécté. Plus précisément :

- La pureté d'un objet détecté $\mathcal{E}_i$ est définie comme le rapport entre le nombre de fois où il a été associé à l'objet réel $\mathcal{GT}_{\hat{j}_i}$ et sa durée de vie. On la notera $\pi(\mathcal{E}_i)$.
- La pureté d'un objet réel $\mathcal{GT}_j$ est définie comme le rapport entre le nombre de fois où il a été associé à l'objet détecté $\mathcal{E}_{\hat{i}_j}$ et sa durée de vie. On la notera $\pi(\mathcal{GT}_j)$.

Les métriques d'évaluation étant définies, nous allons maintenant tester notre système sur un ensemble de séquences de test afin d'évaluer ses performances et d'en déterminer les limites.

5.5.2 Évaluation

L'évaluation est en cours.

5.6 Conclusion

Dans ce chapitre, nous avons présenté une méthode de suivi d'objets basée sur l'étiquetage en composantes connexes d'une séquence d'images binarisées par la méthode de détection de mouvement proposée au chapitre précédent. L'architecture logique utilisée ici pour modéliser les déplacements est un graphe d'association. Nous avons proposé quatre descripteurs de régions d'avant-plan, ainsi que les mesures de similarité associées, afin d'évaluer la vraisemblance des associations entre régions détectées à des instants différents.

Dans la littérature, les méthodes de suivi d'objets utilisant de tels graphes ont la particluarité de présenter une grande complexité combinatoire. Afin de minimiser cet inconvénient, nous proposons une méthode en deux temps. Dans un premier temps, nous ne conservons dans le graphe que les arcs représentant des associations évidentes, c'est-à-dire celles qui respectent un critère sévère basé sur les mesures de similarité proposées. Dans un second temps, les portions de déplacement ainsi validées sont extrapolées à l'aide d'un ensemble de filtres de Kalman, dans les deux sens (vers le passé et vers le futur). Le graphe obtenu est composé de plusieurs composantes connexes dont les sommets correspondent aux détections d'objets qui ont été fusionnés au moins une fois. Nous proposons un algorithme utilisant la topologie de ces sous-graphes pour en déduire le nombre d'objets représentés dans chaque sommet. Enfin, nous décrivons une procédure permettant d'identifier les différents objets de manière unique dans chaque sous-graphe, afin de reconstruire leurs trajectoires.

Les expérimentations réalisées ont permis de vérifier que notre méthode permet de suivre correctement les objets dans différentes situations, pourvu que le nombre de fusions entre objets reste modéré. Dans nos futurs travaux, nous essaierons de pallier cette limitation en insérant un module de fragmentation des régions fusionnées, basé sur le flot optique et/ou sur des mesures liées au domaine d'application.

Chapitre 6

Conclusion générale

Ce travail de thèse traite du suivi d'objets en mouvement dans une séquence vidéo. L'objectif est de définir un ensemble d'opérations génériques qui peuvent être utilisées quel que soit le domaine d'application du système de suivi. L'étude de plusieurs exemples nous a permis d'identifier deux étapes fondamentales : la détection de mouvement et la modélisation des déplacements.

Une étude approfondie de la littérature sur la détection de mouvement nous a permis de définir une taxinomie des méthodes proposées, et de remarquer que la plupart des solutions consistent à considérer les séquences vidéo comme des successions d'images, et à opposer l'image courante à l'ensemble du passé. L'approche la plus fréquente consiste à résumer tout le passé dans un modèle le plus souvent statistique, et à confronter l'image courante à ce modèle afin de décider en tout point, si celui-ci représente l'arrière-plan ou un objet mobile. Afin de conserver une vision moins synthétique du passé, nous avons souhaité prendre mieux en considération la nature tridimensionnelle des données vidéo que nous voulons traiter.

Cette approche pose le problème de la dimensionnalité des données vidéo. C'est pourquoi nous avons également passé en revue les différentes méthodes de réduction de dimension qui ont déjà été utilisées pour l'analyse d'images et de séquences vidéo.

Nous avons ensuite proposé une nouvelle méthode de détection de mouvement dans laquelle l'unité élémentaire n'est plus l'image courante dans la séquence, mais une sous-séquence de quelques images. Afin de pouvoir réduire la dimension des données ainsi obtenues par le biais d'une méthode d'analyse de données, nous avons proposé un espace de représentation des données vidéo adapté au contenu de la scène. Ainsi, nous avons pu appliquer à la séquence élémentaire traitée une analyse en composantes principales qui permet de représenter les données dans un espace de dimension réduite dans lequel les zones de l'image en mouvement se distinguent des zones statiques par leurs statistiques du premier ordre. Ces statistiques sont calculées sur des blocs spatio-temporels partiellement recouvrants dans le domaine spatial, ce qui permet d'obtenir une estimation du mouvement lisse et robuste au bruit. En comparant la segmentation obtenue par cette méthode avec celle fournie par des algorithmes issus

de la littérature, nous avons pu vérifier que les régions détectées étaient de bonne qualité et bien adaptées au suivi d'objets.

En se basant sur la segmentation ainsi obtenue, nous avons proposé dans un deuxième temps une méthode de modélisation des déplacements utilisant un graphe d'association. Afin de réduire la combinatoire du problème de recherche des trajectoires dans le graphe, nous proposons de valider dans un premier temps les associations les plus évidentes, puis d'extrapoler les primitives de mouvement ainsi obtenues à l'aide d'un filtre de Kalman. Nous avons également défini un ensemble de mesures permettant de caractériser l'apparence des régions d'avant-plan, et d'estimer la similarité entre régions détectées à des instants différents. La procédure d'interprétation du graphe que nous proposons permet de déterminer pour chaque région d'avant-plan combien d'objets mobiles elle représente, ainsi que de leur attribuer la liste des identifiants des objets qui la composent.

Les contributions apportées par ce travail de thèse sont multiples. Tout d'abord, nous avons placé le problème de la généricité du suivi d'objet au centre de nos préoccupations, ce qui est rare dans la littérature. Nous avons ensuite choisi une approche originale pour réaliser la détection de mouvement, en proposant d'utiliser des séquences élémentaires plutôt que des images fixes à l'aide de l'analyse en composantes principales. Contrairement aux autres méthodes présentes dans la littérature, la nôtre ne nécessite pas de post-traitement pour que la segmentation fournie soit utilisable. Celle-ci a la particularité de faciliter l'opération suivante (suivi d'objets) car les régions obtenues sont compactes et leur nombre correspond généralement à la réalité terrain. Pour valider ces résultats, nous avons enfin proposé une méthode de suivi basée sur un graphe d'association qui permet de prendre en considération les incertitudes tout en offrant une faible complexité combinatoire. L'association de ces deux modules de traitement permet de répondre de manière satisfaisante au besoin qui a motivé cette thèse : celui de définir une méthode de suivi d'objets qui soit le plus générique possible.

Afin d'améliorer la continuité du mouvement détecté, il serait intéressant de remplacer l'ACP classique utilisée dans le chapitre 4 par une ACP incrémentale telle que décrite dans [Hall *et al.*, 1998]. Cette modification pourra s'accompagner de l'utilisation d'un M-estimateur pour rendre l'analyse plus robuste aux observations aberrantes. Comme nous l'avons constaté lors de l'évaluation de la méthode de détection de mouvement, les contours des objets détectés pâtissent du découpage de la séquence élémentaire en blocs spatio-temporels. Ce découpage est néanmoins nécessaire pour assurer la compacité des régions d'avant-plan. Pour remédier à cet inconvénient, nous pouvons envisager de définir des voisinages spatio-temporels non uniformes, dont les contours seraient localisés sur les points de l'image présentant la plus grande amplitude de gradient. En ce qui concerne la méthode de modélisation des déplacements, nous pensons qu'il est possible d'améliorer l'estimation de la similarité entre régions d'avant-plan en utilisant le flot optique comme mesure de référence. Par ailleurs, ces deux composants ne peuvent suffire à bâtir une application de

vidéo surveillance. Il faudrait pour cela proposer des composants supplémentaires dédiés à un type d'objet en particulier. On pourra penser en particulier à un module qui fusionnerait les trajectoires des objets considérés comme trop petits pour être intéressant. Il serait également intéressant de posséder un module capable de fractionner les régions d'avant-plan détectées en se basant sur le résultat d'une segmentation par le mouvement (flot optique) ou d'un détecteur dédié (détecteur de visages par exemple).

Bibliographie

Arkin, E., Chew, P., Huttenlocher, D. P., Kedem, K. et Mitchell, J. S. B. (1991). An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3): 209–216.

Artač, M., Jogan, M. et Leonardis, A. (2002). Incremental PCA for on-line visual learning and recognition. *In Proc. 16th Int. Conf. on Pattern Recognition (ICPR 2002)*, volume III, pages 781–784, Québec, Canada.

Auclair, A., Cohen, L. D. et Vincent, N. (2007). How to use SIFT vectors to analyze an image with database templates. *In 5th Int. Workshop on Adaptive Multimedia Retrieval*, Paris, France.

Bar-Shalom, Y. et Fortmann, T. E. (1988). *Tracking and Data Association*. Academic Press.

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

Bartlett, M. S., Lades, H. M. et Sejnowski, T. (1998). Independent component representation for face recognition. *In Proc. IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*, pages 528–539, San Jose (CA), USA.

Bashir, F. et Porikli, F. (2006). Performance evaluation of object detection and tracking systems. *In Proc. 9th Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)*, pages 7–14, New York (NY), USA.

Belhumeur, P. N., Hespanha, J. P. et Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition using class-specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

Besse, P. (1992). PCA stability and choice of dimensionality. *Statistics & Probability Letters*, 13:405–410.

Boult, T. E., Michaels, R. J., Gao, X., Lewis, P., Power, C., Yin, W. et Erkan, A. (1999). Frame-rate omnidirectional surveillance and tracking

of camouflaged and occluded targets. *In Proc. 2nd IEEE Int. Workshop on Visual Surveillance*, pages 48–55, Fort Collins (CO), USA.

Camus, T. (1995). Real-time quantized optical flow. *In Proc. IEEE Int. Workshop on Computer Architectures for Machine Perception (CAMP'95)*, pages 126–131, Villa Olmo, Italie.

Cheung, S.-C. S. et Kamath, C. (2005). Robust techniques for background subtraction in urban traffic video. *EURASIP Journal on Applied Signal Processing*, 2005(14):2330–2340.

Chia, A. Y. S., Huang, W. et Li, L. (2006). Multiple objects tracking with multiple hypotheses graph representation. *In Proc. 18th Int. Conf. on Pattern Recognition (ICPR 2006)*, pages 638–641, Hong-Kong, Chine.

Cohen, I. et Medioni, G. (1999). Detecting and tracking moving objects for video surveillance. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, volume 2, pages 319–325, Fort Collins (CO), USA.

Cox, I. J. (1993). A review of statistical data association techniques for motion correspondence. *Int. Journal of Computer Vision*, 10(1):53–66.

Cucchiara, R., Grana, C., Piccardi, M. et Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. on Pattern Analysis and Machine Application*, 25(10):1337–1342.

Cutler, R. et Davis, L. S. (1998). View-based detection and analysis of periodic motion. *In Proc. 14th Int. Conf. on Pattern Recognition (ICPR'98)*, volume 1, pages 495–500, Brisbane, Australie.

De la Torre, F. et Black, M. J. (2001). Robust principal component analysis for computer vision. *In Proc. 8th IEEE Int. Conf. on Computer Vision (ICCV 2001)*, volume 1, pages 362–369, Vancouver, Canada.

Deriche, R. et Faugeras, O. (1990). Tracking line segments. *In Proc. 1st European Conf. on Computer Vision (ECCV'90)*, pages 259–268, Antibes, France.

Elgammal, A. M., Harwood, D. et Davis, L. S. (2000). Non-parametric model for background subtraction. *In Proc. 6th European Conf. on Computer Vision (ECCV 2000)*, volume II, pages 751–767, Dublin, Irlande.

Fodor, I. K. (2002). A survey of dimension reduction techniques. Report UCRL-ID-148494, Lawrence Livermore National Laboratory, Livermore (CA), USA.

Genovesio, A. (2005). *Une méthode de poursuite de taches multiples : Application à l'étude de la dynamique d'objets biologiques en microscopie 3D+T.* Thèse de doctorat, Université Paris Descartes.

Gondran, M. et Minoux, M. (1995). *Graphes et algorithmes*. Collection de la Direction des Études et Recherches d'Électricité de France. Eyrolles.

Gonzalez, R. C. et Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley.

Guo, J., Chng, E. S. et Rajan, D. (2004). Foreground motion detection by difference-based spatial temporal entropy image. *In Proc. IEEE Region 10 Conf. (TenCon 2004)*, pages 379–282, Chiang Mai, Thaïlande.

Hall, P., Marshall, D. et Martin, R. (1998). Incremental eigenanalysis for classification. *In Proc. British Machine Vision Conf. (BMVC'98)*, volume I, pages 286–295, Southampton, Royaume-Uni.

Han, B., Comaniciu, D. et Davis, L. (2004). Sequential kernel density approximation through mode propagation: Applications to background modeling. *In Proc. 6th Asian Conf. on Computer Vision (ACCV 2004)*, Jeju Island, République de Corée.

Hayes, M. H. (1996). *Statistical Digital Signal Processing and Modeling*, chapitre 7. John Wiley & Sons.

Heikkilä, M. et Pietikäinen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):657–662.

Holland, P. W. et Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics: Theory and Methods*, A6:813–827.

Horn, B. K. P. et Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

Hu, W., Tan, T., Wang, L. et Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man, and Cybernetics*, 34(3):334–352.

Huber, P. J. (1981). *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Jain, R. et Nagel, H. (1979). On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1(2):206–214.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer, 2ème édition.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20:141–151.

KALMAN, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. of the American Society of Mechanical Engineers – Journal of Basic Engineering*, 82:35–45.

KARMANN, K.-P. et von BRANDT, A. (1990). Moving object recognition using an adaptive background memory. *In* CAPPELLINI, V., éditeur : *Time-Varying Image Processing and Moving Object Recognition, 2*, pages 289–307. Elsevier Science.

KASS, M., WITKIN, A. et TERZOPOULOS, D. (1988). Snakes : Ative contour models. *International Journal of Computer Vision*, 1(4):321–331.

KE, Y., SUKTHANKAR, R. et HUSTON, L. (2004). An efficient parts-based near-duplicate and sub-image retrieval system. *In Proc. 12th Annual ACM Conf. on Multimedia*, pages 869–876, New York (NY), USA.

KINCAID, D. R. et CHENEY, E. W. (2001). *Numerical Analysis: Mathematics of Scientific Computing, 3rd edition*, chapitre 6. Brooks Cole.

KOLLER, D., WEBER, J. et MALIK, J. (1993). Robust multiple car tracking with occlusion reasoning. Technical Report UCB/CSD-93-780, University of California at Berkeley, EECS Department, Berkeley (CA), USA.

LAZAREVIC-MCMANUS, N., RENNO, J., MAKRIS, D. et JONES, G. A. (2006). Designing Evaluation Methodologies: The Case of Motion Detection. *In Proc. 9th Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2006)*, pages 23–30, New York (NY), USA.

LEE, S. M., XIN, J. H. et WESTLAND, S. (2005). Evaluation of image similarity by histogram intersection. *Color Research and Application*, 30(4):265–274.

LEFÈVRE, S. et VINCENT, N. (2006). Apport de l'espace Teinte-Saturation-Luminance pour la segmentation spatiale et temporelle. *Traitement du Signal*, 23(1):59–77.

LI, Y., XU, L.-Q., MORPHETT, J. et JACOBS, R. (2003). An integrated algorithm of incremental and robust PCA. *In Proc. IEEE Int. Conf. on Image Processing (ICIP 2003)*, volume I, pages 245–248, Barcelone, Espagne.

LIU, H., HONG, T.-H., HERMAN, M. et CHELLAPPA, R. (1998). Accuracy vs. efficiency trade-offs in optical flow algorithms. *Computer Vision and Image Understanding*, 7(3):271–286.

LOWE, D. (2004). Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110.

LUCAS, B. D. et KANADE, T. (1981). An iterative image registration technique with an application to stereo vision. *In Proc. 7th Int. Joint Conf. on Artificial Intelligence*, pages 674–679, Vancouver, Canada.

Ma, Y. F. et Zhang, H. J. (2001). Detecting motion object by spatio-temporal entropy. *In Proc. IEEE Int. Conf. on Multimedia and Expo (ICME 2001)*, pages 265–268, Tokyo, Japon.

McKenna, S. J., Jabri, S., Duric, Z., Wechsler, H. et Rosenfeld, A. (2000). Tracking goups of people. *Computer Vision and Image Understanding*, 80(1):42–56.

Moeslund, T. B., Hilton, A. et Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2-3):90–126.

Moghaddam, B. et Pentland, A. (1995). Probabilistic visual learning for object detection. *In Proc. 5th IEEE Int. Conf. on Computer Vision (ICCV'95)*, pages 786–793, Cambridge (MA), USA.

Monnet, A., Mittal, A., Paragios, N. et Ramesh, V. (2003). Background modeling and subtraction of dynamic scenes. *In Proc. 9th IEEE Int. Conf. on Computer Vision (ICCV 2003)*, volume II, pages 1305–1312, Nice, France.

Nascimento, J. et Marques, J. S. (2004). New performance evaluation metrics for object detection algorithms. *In Proc. 6th Int. Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2004)*, pages 7–14, Prague, République Tchèque.

Nguyen, H. T., Ji, Q. et Smeulders, A. W. M. (2007). Spatio-temporal context for robust multitarget tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(1):52–64.

Odobez, J.-M. et Bouthemy, P. (1998). Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 6(2):143–155.

Oliver, N., Rosario, B. et Pentland, A. (2000). A Bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831–843.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62–66.

Pham, D. L., Xu, C. et Prince, J. L. (2000). Current methods in medical image segmentation. *Annual Review of Biomedical Engineering*, 2:315–337.

Piccardi, M. (2004). Background subtraction techniques: A review. *In Proc. IEEE Conf. on Systems, Man, and Cybernetics*, volume IV, pages 3099–3104, La Hague, Pays-Bas.

Pless, R. (2005). Spatio-temporal background models for outdoor surveillance. *EURASIP Journal on Applied Signal Processing*, 2005(14):2281–2291.

Rabiner, L. B. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286.

Radke, R. J., Andra, S., Al-Kofahi, O. et Roysam, B. (2005). Image change detection algorithms: A systematic survey. *IEEE Trans. on Image Processing*, 14(3):294–307.

Reid, D. B. (1979). An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854.

Ridder, C., Munkelt, O. et Kirchner, H. (1995). Adaptive background estimation and foreground detection using Kalman-filtering. *In* Kaynak, O., Özkan, M., Bekiroglu, N. et Tunay, I., éditeurs : *Proc. Int. Conf. on Recent Advances in Mechatronics (ICRAM'95)*, pages 193–199, Istanbul, Turquie.

Rittscher, J., Kato, J., Joga, S. et Blake, A. (2000). A probabilistic background model for tracking. *In Proc. 6th European Conf. on Computer Vision (ECCV 2000)*, volume II, pages 336–350, Dublin, Irlande.

Rymel, J. D., Renno, J.-P., Greenhill, D., Orwell, J. et Jones, G. A. (2004). Adaptive eigen-backgrounds for object detection. *In Proc. IEEE Int. Conf. on Image Processing (ICIP 2004)*, volume 3, pages 1847–1850, Singapour.

Sirovich, L. et Kirby, M. (1987). A low-dimensional procedure for the characterization of human faces. *The Journal of the Optical Society of America*, 4:519–524.

Smith, K., Gatica-Perez, D., Odobez, J.-M. et Ba, S. (2005). Evaluating multi-object tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshop on Empirical Evaluation Methods in Computer Vision (CVPR-EEMCV)*, San Diego (CA), USA.

Spangenberg, R. et Döring, T. (2006). Evaluation of object tracking in traffic scenes. *In Proc. ISPRS Commission V Symposium : 'Image Engineering and Vision Metrology'*, Dresde, Allemagne.

Stauffer, C. et Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'99)*, pages 2246–2252, Fort Collins (CO), USA.

Stenger, B., Ramesh, V., Paragios, N., Coetzee, F. et Buhmann, J. M. (2001). Topology free hidden Markov models: Application to background modeling. *In Proc. 8th IEEE Int. Conf. on Computer Vision (ICCV 2001)*, volume I, pages 294–301, Vancouver, Canada.

Sun, C. (2002). Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *Int. Journal of Computer Vision*, 47(1/2/3): 99–117.

Tian, Y. L. et Hampapur, A. (2005). Robust salient motion detection with complex background for real-time video surveillance. *In Proc. IEEE Workshop on Motion and Video Computing (WMVC 2005)*, volume II, pages 30–35, Breckenridge (CO), USA.

Toyama, K., Krumm, J., Brummit, B. et Meyers, B. (1999). Wallflower: Principles and practice of background maintenance. *In Proc. 7th IEEE Int. Conf. on Computer Vision (ICCV'99)*, volume I, pages 255–261, Kerkyra (Corfou), Grèce.

Travis, D. (1991). *Effective Color Displays: Theory and Practice*. Academic Press.

Trémeau, A., Fernandez-Maloigne, C. et Bonton, P. (2004). *Image numérique couleur : De l'acquisition au traitement*. Sciences Sup. Dunod.

Vieira Neto, H. et Nehmzow, U. (2005). Incremental PCA: An alternative approach for novelty detection. *In Proc. Towards Autonomous Robotic Systems (TAROS 2005)*, Londres, Royaume-Uni.

Wang, L., Hu, W. et Tan, T. (2003). Recent developments in human motion analysis. *Pattern Recognition*, 36:585–601.

Weiss, Y. et Adelson, E. H. (1996). A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'96)*, pages 321–326, San Francisco (CA), USA.

Welch, G. et Bishop, G. (1995). An introduction to the Kalman filter. Technical Report TR95-041, University of North Carolina at Chapel Hill, Department of Computer Science, Chapel Hill (NC), USA.

Wren, C., Azarbayejani, A., Darrell, T. et Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

Xu, L. et Yuille, A. L. (1995). Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. on Neural Networks*, 6(1):131–143.

Yambor, W. S., Draper, B. A. et Beveridge, J. R. (2002). Analyzing PCA-based face recognition algorithms: Eigenvectors selection and distance measures. *In* Christensen, H. I. et Philips, P. J., éditeurs : *Empirical Evaluation Methods in Computer Vision*, Series in Machine Perception and Artificial Intelligence, Volume 50, chapitre 3. World Scientific.

Yang, T., Li, S. Z., Pan, Q. et Li, J. (2004). Real-time and accurate segmentation of moving objects in dynamic scene. *In Proc. ACM 2nd Int. Workshop on Video Surveillance & Sensor Networks (VSSN 2004)*, pages 136–143, New York (NY), USA.

Yilmaz, A., Javed, O. et Shah, M. (2006). Object tracking: A survey. *ACM Journal of Computing Surveys*, 38(4). Article 13.

Zhang, Z. (1995). Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. Rapport de Recherche RR-2676, INRIA, Sophia Antipolis, France.

Zhong, J. et Sclaroff, S. (2003). Segmenting foreground objects from a dynamic textured background via a robust Kalman filter. *In Proc. 9th IEEE Int. Conf. on Computer Vision (ICCV 2003)*, volume I, pages 44–50, Nice, France.

Zivkovic, Z. et van der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780.

Publications

Nicolas Verbeke et Nicole Vincent : Réduction de dimension pour l'analyse de données vidéo. 7èmes Journées Francophones d'Extraction et de Gestion des Connaissances (EGC'2007), Namur, Belgique, janvier 2007 ; *Revue des Nouvelles Technologies de l'Information*, Vol. II, pages 397–408, Cépaduès.

Nicolas Verbeke et Nicole Vincent : Détection de mouvements cohérents dans une séquence vidéo. *In Proceedings of the 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2007)*, Hammamet, Tunisie, mars 2007.

Nicolas Verbeke et Nicole Vincent : A PCA-based Technique to Detect Moving Objects. *In Proceedings of the 15th Scandinavian Conference on Image Analysis (SCIA 2007)*, pages 641–650, Aalborg, Danemark, juin 2007.