Détection non supervisée d'évènements rares dans un flot vidéo : application à la surveillance d'espaces publics

Bertrand Luvison

N° d'ordre : D.U. 2089
EDSPIC : 487

UNIVERSITÉ BLAISE PASCAL - CLERMONT II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse

Présentée par :

Bertrand LUVISON

pour obtentir le grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : Vision pour la robotique

Détection non supervisée d'évènements rares dans un flot vidéo : Application à la surveillance d'espaces publics

---

Soutenue publiquement le 13/12/2010 devant le jury :

M. Serge MIGUET	Président
Mme. Catherine ACHARD	Rapporteur
Mme. Jenny BENOIS-PINEAU	Rapporteur
M. Jean-Marc ODOBEZ	Examinateur
M. Thierry CHATEAU	Encadrant
M. Quoc-Cuong PHAM	Encadrant
M. Patrick SAYD	Encadrant
M. Jean-Thierry LAPRESTÉ	Directeur de thèse

Remerciements

Ce manuscrit de thèse détaille les travaux de recherche qui ont été menés pour proposer un système de vidéoassistance pour la détection d'évènements anormaux dans une scène publique. Ces travaux ont été réalisés en co-tutelle entre le LASMEA à Aubière, dirigé par Michel DHOME et le LVIC du CEA, LIST à Gif-sur-Yvette, dirigé par François GASPARD que je remercie pour m'avoir donner l'opportunité de réaliser cette thèse au sein de leur laboratoire respectif.

Je remercie Serge MIGUET, Professeur et Directeur de Recherches au LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information à Lyon) pour avoir accepté la présidence de mon jury.

Je remercie Madame Catherine ACHARD, Maître de Conférences et Directeur de Recherches à l'ISIR (Institut des Systèmes Intelligents et de Robotique à Paris) ainsi que Jenny BENOIS-PINEAU, Professeur et Directeur de Recherches au LABRI (LAboratoire Bordelais de Recherche en Informatique) pour le temps qu'elles ont consacré et l'intérêt qu'elles ont porté à l'évaluation de mes travaux en tant que rapporteurs.

Je remercie également Monsieur Jean-Marc ODOBEZ, Chercheur Sénior à l'IDIAP en Suisse pour l'ensemble de ses remarques et conseils lors de nos différentes rencontres durant cette thèse ainsi que pour avoir accepté de participer au jugement de mes travaux.

J'adresse toute ma gratitude à Jean-Thierry LASPRESTÉ pour avoir accepter d'être mon directeur de thèse et pour avoir toujours été là lors des nombreuses difficultés auxquelles j'ai du faire face durant cette thèse. Merci pour toutes ces connaissances scientifiques que tu as su me transmettre.

Ma gratitude s'adresse aussi à l'ensemble de mes encadrants, Thierry CHATEAU, Maître de Conférences à l'université Blaise Pascal, Quoc-Cuong PHAM et Patrick SAYD, tous deux Ingénieurs Chercheurs au CEA. Merci pour votre soutien et votre encadrement tout au long de cette thèse, pour votre bonne humeur quotidienne et surtout pour avoir su me transmettre la passion de ce travail. Je suis aujourd'hui fier de pouvoir continuer à travailler avec vous.

Je remercie également l'ensemble des personnes du LASMEA et du LVIC que j'ai pu rencontrer qui m'ont aidé durant mes travaux et qui m'ont permis de réaliser cette thèse dans d'excellente condition.

Je remercie tout particulièrement l'ensemble des personnes du LASMEA et du LVIC que j'ai pu rencontrer et qui sont désormais de précieux amis. Les citer tous serait trop long mais merci à eux pour leur bonne humeurs, pour les fous rires et les merveillleux moments passer en leur compagnie aussi bien au laboratoire qu'en dehors.

Enfin je remercie ma famille. Merci de m'avoir toujours encouragé et soutenu tout au long de mes études. C'est grâce à votre dévotion que j'ai pu réaliser cette thèse.

Résumé

Cette thèse est une collaboration entre le LAboratoire des Sciences et Matériaux pour l'Électronique et d'Automatique (LASMEA) de Clermont-Ferrand et le Laboratoire Vision et Ingénierie des Contenus (LVIC) du CEA LIST à Saclay. La première moitié de la thèse a été accomplie au sein de l'équipe ComSee [1] du LASMEA et la deuxième au LVIC. L'objectif de ces travaux est de concevoir un système de vidéo-assistance temps réel pour la détection d'évènements dans des scènes possiblement denses.

La vidéosurveillance intelligente de scènes denses telles que des foules est particulièrement difficile, principalement à cause de leur complexité et de la grande quantité de données à traiter simultanément. Le but de cette thèse consiste à élaborer une méthode de détection d'évènements rares dans de telles scènes, observées depuis une caméra fixe. La méthode en question s'appuie sur l'analyse automatique de mouvement et ne nécessite aucune information *a priori*. Les mouvements nominaux sont déterminés grâce à un apprentissage statistique non supervisé. Les plus fréquemment observés sont considérés comme des évènements normaux. Une phase de classification permet ensuite de détecter les mouvements déviant trop du modèle statistique, pour les considérer comme anormaux. Cette approche est particulièrement adaptée aux lieux de déplacements structurés, tels que des scènes de couloirs ou de carrefours routiers. Aucune étape de calibration, de segmentation de l'image, de détection d'objets ou de suivi n'est nécessaire. Contrairement aux analyses de trajectoires d'objets suivis, le coût calculatoire de notre méthode est invariante au nombre de cibles présentes en même temps et fonctionne en temps réel.

Notre système s'appuie sur une classification locale du mouvement de la scène, sans calibration préalable. Dans un premier temps, une caractérisation du mouvement est réalisée, soit par des méthodes classiques de flot optique, soit par des descripteurs spatio-temporels. Ainsi, nous proposons un nouveau descripteur spatio-temporel fondé sur la recherche d'une relation linéaire entre les gradients spatiaux et les gradients temporels en des zones où le mouvement est supposé uniforme. Tout comme les algorithmes de flot optique, ce descripteur s'appuie sur la contrainte d'illumination constante. Cependant en prenant en compte un voisinage temporel plus important, il permet une caractérisation du mouvement plus lisse et plus robuste au bruit. De plus, sa faible complexité calculatoire est bien adaptée aux applications temps réel.

Nous proposons ensuite d'étudier différentes méthodes de classification :

- ▷ La première, statique, dans un traitement image par image, s'appuie sur une estimation bayésienne de la caractérisation du mouvement au travers d'une approche basée sur les fenêtres de Parzen. Cette nouvelle méthode est une variante parcimonieuse des fenêtres de Parzen. Nous montrons que cette approche est algorithmiquement efficace pour approximer de manière compacte et précise les densités de probabilité.
- ▷ La seconde méthode, basée sur les réseaux bayésiens, permet de modéliser la dynamique du

1. acronyme de *Computers that See*.

mouvement. Au lieu de considérer ce dernier image par image, des séquences de mouvements sont analysées au travers de chaînes de Markov Cachées. Ajouté à cela, une autre contribution de ce manuscrit est de prendre en compte la modélisation du voisinage d'un bloc afin d'ajouter une cohérence spatiale à la propagation du mouvement. Ceci est réalisé par le biais de couplages de chaînes de Markov cachées.

Ces différentes approches statistiques ont été évaluées sur des données synthétiques ainsi qu'en situations réelles, aussi bien pour la surveillance du trafic routier que pour la surveillance de foule. Cette phase d'évaluation permet de donner des premières conclusions encourageantes quant à la faisabilité de la vidéosurveillance intelligente d'espaces possiblement denses.

Mots clés : Flot optique, descripteurs spatio-temporels, machine d'apprentissage, fenêtre de Parzen, modèle de Markov cachés, classification du mouvement.

Abstract

The automatic analysis of crowded areas in video sequences is particularly difficult because of the large amount of information to be processed simultaneously and the complexity of the scenes. We propose in this thesis a method for detecting abnormal events in possibly dense scenes observed from a static camera. The approach is based on the automatic classification of motion requiring no prior information. Motion patterns are encoded in an unsupervised learning framework in order to generate a statistical model of frequently observed (aka. normal) events. Then at the detection stage, motion patterns that deviate from the model are classified as unexpected events. The method is particularly adapted to scenes with structured movement with directional flow of objects or people such as corridors, roads, intersections. No camera calibration is needed, nor image segmentation, object detection and tracking. In contrast to approaches that rely on trajectory analysis of tracked objects, our method is independent of the number of targets and runs in real-time.

Our system relies on a local classification of global scene movement. The local analysis is done on each blocks of a regular grid. We first introduce a new spatio-temporal local descriptor to characterize the movement efficiently. Assuming a locally uniform motion of space-time blocks of the image, our approach consists in determining whether there is a linear relationship between spatial gradients and temporal gradients. This spatio-temporal descriptor holds the Illumination constancy constraint like optical flow techniques, but it allows taking into account the spatial neighborhood and a temporal window by giving a smooth characterization of the motion, which makes it more robust to noise. In addition, its low computational complexity is suitable for real-time applications.

Secondly, we present two different classification frameworks :

- ▷ The first approach is a static (frame by frame) classification approach based on a Bayesian characterization of the motion by using an approximation of the Parzen windowing method or Kernel Density Estimation (KDE) to model the probability density function of motion patterns. This new method is the sparse variant of the KDE (SKDE). We show that the SKDE is a very efficient algorithm giving compact representations and good approximations of the density functions.
- ▷ The second approach, based on Bayesian Networks, models the dynamics of the movement. Instead of considering motion patterns in each block independently, temporal sequences of motion patterns are learned by using Hidden Markov Models (HMM). The second proposed improvement consists in modeling the movement in one block by taking into account the observed motion in adjacent blocks. This is performed by the coupled HMM method.

Evaluations were conducted to highlight the classification performance of the proposed methods, on both synthetic data and very challenging real video sequences captured by video surveillance cameras. These evaluations allow us to give first conclusions concerning automatic analyses of possibly crowded area.

Key-words : Optical Flow, spatio-temporal descriptors, Learning machine, *Kernel Density Estimation*, *Hidden Markov Model*, movement classification.

Table des matières

Table des figures

Liste des tableaux

Notations et conventions

Notations mathématiques

$\mathbf{x}$	le vecteur $\mathbf{x}$
$\mathcal{N}$	la fonction gaussienne
δ_j^i	le symbole de kronecker
$\|\mathbf{x}\|$	la norme euclidienne du vecteur $\mathbf{x}$ (ou norme L2)
$\|M\|_F$	la norme de Frobœnius de la matrice M
$\{a_1, a_2, \ldots, a_N\}$	l'ensemble des éléments $a_1, a_2, \ldots, a_N$
$(a_1, a_2, \ldots, a_N)$	le vecteur colonne contenant les coefficients $a_1, a_2, \ldots, a_N$
$a_1 a_2 \ldots a_N$	la séquence de variables $a_1, a_2, \ldots, a_N$

Représentation dense de champs de déplacement

Dans la suite du manuscrit, certains champs de vecteurs (en particulier des champs de déplacement) auront besoin d'être représentés graphiquement. Pour ce faire, l'orientation d'un vecteur sera caractérisée par une couleur issue d'une échelle cyclique décrite à la figure 1. La norme du vecteur est quant à elle représentée par la saturation de la couleur, la couleur est d'autant plus vive que la norme du déplacement est importante. Cette convention d'affichage permet une représentation dense des champs de déplacement.

FIGURE 1 – Représentation colorimétrique de l'orientation d'un vecteur.

Chapitre 1

Introduction à la vidéosurveillance d'espaces publics

Ce chapitre a pour but de faire un tour d'horizon sur ce qui est actuellement réalisé en vidéosurveillance de scènes publiques. Un bilan des différents objectifs visés, des différentes caractéristiques des vidéos étudiées ainsi que des méthodes d'apprentissage ou des traitements de ces caractéristiques est effectué. En réponse aux problèmes liés à cette thématique, les décisions prises et les grandes lignes du système en découlant sont présentées en fin de ce chapitre afin de fixer le cadre du reste de ce manuscrit.

1.1 Les enjeux et besoins

Les premières utilisations de caméras de vidéosurveillance remontent aux années 1950. Leur exploitation au sein de systèmes de télévision en circuit fermé (TVCF) sont apparues en 1970 mais c'est à partir des années 1990 que la vidéosurveillance a vraiment commencé à s'implanter. Les attentats de septembre 2001 aux États-Unis et de 2005 à Londres ont contribué à l'explosion du nombre de caméras (Gouaillier et Fleurant (2009)). D'après les sources de Norris *et al.* (2004), un rapport faisait état d'approximativement quatre millions de caméras (toutes caméras confondues) en Grande-Bretagne, soit une caméra pour dix-sept habitants.

Devant une telle quantité de caméras, les agents de sécurité ne peuvent pas toutes les surveiller en même temps. Ils sont contraints de ne regarder que ponctuellement chacune d'entre elles en permutant régulièrement (l'image 1.1 illustre la complexité de la tâche). Une étude de Dee et Velastin (2008) rapporte que à un instant donné et dans le meilleur des cas, une caméra sur quatre est sous l'observation d'un opérateur et dans le pire des cas ce ratio peut tomber à une caméra pour soixante-dix-huit. De plus, la répétitivité de la tâche, la faible fréquence des situations dangereuses ou anormales et la lassitude occasionnée, rendent le travail de surveillance particulièrement difficile. Certaines estimations de Gouaillier et Fleurant (2009) font état d'une chance sur mille de réagir en direct à un évènement anormal. Pour ces raisons, ce travail se doit d'être assisté, voire automatisé via des systèmes de vidéosurveillance ou de vidéo-assistance intelligents.

En outre, nombre de ces caméras sont utilisées pour la surveillance d'espaces publics très fréquentés (supermarchés, gares, axes routiers, rues ou places piétonnes, etc). Ce type de scène où les risques

FIGURE 1.1 – Poste de surveillance d'un système de télévision en circuit fermé.

sont d'autant plus grand que le nombre de véhicules ou de personnes est important, est aussi particulièrement difficile à surveiller à cause justement de cette forte densité. Ceci n'ira d'ailleurs pas en s'améliorant car la population ne cesse de grandir.

Indépendamment de la vision par ordinateur, l'étude des mouvements et des comportements de foule est une problématique à part entière qui nourrit de nombreuses études et de nombreux travaux, que ce soit pour en comprendre les mécanismes anthropologiques et sociologiques (Pan (2006)) ou pour des raisons plus pragmatiques tel que le contrôle de la foule. Ces études peuvent être intéressantes dans des supermarchés afin d'optimiser les profits, dans les musées afin d'éviter les encombrements, ou encore à des fins de modélisation dans le domaine du cinéma pour la réalisation de scène de mobilisation de grande armée par exemple (cf. logiciel *MASSIVE* avec Seigneur des Anneaux en 2002 ou plus récemment Avatar en 2009).

Conscient des travaux menés dans les autres domaines, ce qui nous intéresse en vision par ordinateur est de déterminer visuellement les indicateurs laissant annoncer des comportements particuliers. Replacés dans un contexte de vidéosurveillance, les indicateurs recherchés seront essentiellement ceux identifiant des comportements anormaux ou des états de crise. Par extension, nous ferons l'amalgame entre analyse de foule de personnes et de flot de véhicules. En effet, sous de nombreux aspects en particulier celui du mouvement, un flot de véhicule présente des caractères similaires à celui d'un mouvement de foule. Les déplacements sur les voies de circulation définissent des flux structurés de la même manière que peuvent le faire les foules sur des rues piétonnes, des passages piétons, dans des couloirs, etc.

Depuis une décennie maintenant, des projets ont vu le jour afin d'apporter une réponse quant à la faisabilité d'une vidéosurveillance intelligente. Longtemps cantonné à des scénarios très simples faisant intervenir un nombre limité de personnes dans des conditions bien contrôlés, les travaux réalisés ces dernières années commencent à s'attaquer à des situations plus complexes, plus proches de situations réelles. Selon les sources de Gouaillier et Fleurant (2009), certains projets européens tels

que CAVIAR [1] (motivé par la surveillance de centre ville et des études marketing), ISCAPS [2] (pour la surveillance d'espaces publics), ou des projets militaires ont été initiés. Citons aussi le projet *Combat Zone That See* lancé pas le DARPA [3], visant la surveillance de zones de conflits urbains en identifiant les véhicules (type, couleur, et plaque d'immatriculation). Cette tendance se poursuit encore aujourd'hui, comme par exemple en France avec le plan de vidéo protection des « 1000 caméras à Paris », etc.

1.2 Les contraintes applicatives

Ces systèmes de vidéosurveillance sont en pratique, particulièrement contraints. On peut citer :

- ▷ Des contraintes techniques, comme par exemples la calibration de caméra ou de réseaux de caméras, les éventuelles distortions provoquées par les capteurs, des taux de rafraîchissement faibles dans certaines situations, etc.
- ▷ Des contraintes environnementales, comme le manque de résolution dans certaines situations, les changements d'illumination (cf. illustrations 1.2 et 1.3), etc. Ce sont des problèmes récurrents en vision par ordinateur et il n'existe pas encore de solutions « miracle ». Ils se traitent en général au cas par cas selon la scène observée, ce qui rend la généralisation d'un système de vidéosurveillance d'autant plus difficile.
- ▷ Des contraintes sémantiques, les systèmes mis en œuvre sont généralement ciblés pour un type ou une plage d'évènements bien spécifiques. Par exemple, la détection de personnes trop proche du bord d'un quai ou la détection de présence de personnes sur la chaussée, etc. Une telle restriction des évènements à détecter est obligatoire car la notion de normalité est particulièrement difficile à modéliser en informatique car trop dépendante de la situation. En reprenant l'exemple de la surveillance de quai, une personne peut être considérée comme en danger si elle est trop proche du bord de la voie lorsque le train n'est pas présent. En revanche, lors de la présence du train, cela signifie tout simplement que cette personne est en train de monter ou de descendre comme illustré sur les images de la figure 1.4. Dans le cadre de la surveillance de personnes présentes sur la chaussée, il est normal de voir des personnes sur des passages piétons, mais pas en dehors, à moins que ces personnes ne soient des agents de l'ordre. L'ensemble de ces règles que l'homme apprend lors de son enfance sont particulièrement difficiles à mettre en place informatiquement. En pratique, le système sera défini soit par des règles bien précises correspondant à la situation, on parlera d'approches basées modèles, soit par des approches statistiques, on parlera dans ce cas d'approches basées apprentissage.

Plus spécifiquement, lors de l'analyse de scènes denses, telles que des foules, on rencontre de nouveaux problèmes, tels que :

- ▷ Des contraintes de traitement de volume d'information importantes. Une foule représente une quantité phénoménale de données à traiter, ce qui peut poser des problèmes de temps de calcul pour la vidéosurveillance temps-réel.
- ▷ Des contraintes de point de vue qui sont d'autant plus sensibles que sur des vidéos denses, certains points de vue peuvent souffrir d'occultations très importantes (cf. image 1.5) et très longues.

1. acronyme de *Context Aware Vision and Recognition*
2. acronyme de *Integrated Surveillance of Crowded Areas for Public Security*
3. acronyme de *Defense Advanced Research Projects Agency*

FIGURE 1.2 – Exemple de changement d'illumination. Les images ont été capturées à 1 seconde d'intervale.

FIGURE 1.3 – Exemple de scène souffrant d'un manque de résolution.

FIGURE 1.4 – Surveillance de quai de métro. Selon la présence de la rame ou non, la proximité des voies n'a pas la même signification.

FIGURE 1.5 – Dépendant de l'orientation de la caméra, les occultations des personnes au sein d'une foule peuvent être plus ou moins importantes.

- ▷ L'absence de contexte : contrairement à certaines études sociologiques, où une connaissance du contexte et des us et coutumes liés à l'environnement peut être connu, en vision par ordinateur, seul ce qui est vu peut-être exploité. On peut citer par exemple le comportement d'une foule en sortie de stade qui n'agira pas nécessairement de la même manière selon la défaite ou la victoire de son équipe, le tout démultiplié par l'enjeu du match. Le système idéal peut être vu comme un couplage de la vision avec une intelligence artificielle qui tient compte du contexte. Nous nous focalisons ici sur l'aspect vision.

Pour subir le moins possible ces contraintes, notre ligne de conduite est de penser un système sans calibrage, adaptatif et générique autant que possible tout en étant le moins possible supervisé. L'idée sous-jacente est d'avoir un système facilement déployable.

Le paragraphe suivant présente l'existant concernant l'analyse de foule en insistant sur les finalités visées, les technologies mises en œuvre pour y arriver ainsi que leurs avantages et leurs inconvénients.

1.3 Panorama de l'existant

1.3.1 Différents types d'application

Un système de vidéosurveillance est lié à un type de scène. En conséquence, il convient de se poser certaines questions, afin de restreindre le domaine de détection de l'application de vidéosurveillance. Parmi ces questions, la nature de la scène (couloir intérieur, place publique, hall d'entrée ...), l'évènement que l'on souhaite détecter (intrusion, personne couchée, personne à contresens, comptage de la foule), la nature et la densité de la foule ciblée (éparpillée, dense, marée humaine, passage régulier, intermittent ...), sont autant de questions qui guideront le choix des systèmes de détection.

L'objectif de cette section est de faire un tour d'horizon des traitements qui ont été réalisés pour des applications de vidéosurveillance de scènes plus ou moins denses de personnes ou de véhicules. Deux grandes familles d'approches peuvent être mises en avant. La première concerne les approches visant à individualiser les personnes ou les véhicules et à traiter les comportements de ces individualités séparément (méthode qui sera baptisée « microscopique » par la suite). La seconde vise à étudier les caractéristiques de la foule ou du trafic dans sa globalité sans se soucier de l'entité « personne » ou « véhicule » (méthode qui sera baptisée « macroscopique » par la suite). L'état de l'art fait aussi

référence à une approche intermédiaire bien plus rare, dite mésoscopique Zhan *et al.* (2008), qui dans le cas d'une foule, considère les forces d'interaction entre les individus au sein de groupes de personnes indifférenciées.

1.3.1.1 Les approches microscopiques

Pour ce qui est des approches dites microscopiques, parmi les différents types de traitement, on peut répertorier la segmentation de personne pour déterminer les trajectoires ou pour l'étude de la posture.

Étude de la trajectoire Ces approches visent d'une part à déterminer la trajectoire et d'autre part à l'étudier. On peut répertorier différents objectifs :

- ▷ Le suivi du maximum d'entités possibles que ce soit en mono-caméra (Zhao et Nevatia (2004), Park et Trivedi (2005), Bardet *et al.* (2009) et Yu et Medioni (2009)), par stéréoscopie (Tyagi *et al.* (2007), Kelly *et al.* (2009)) ou en multi-caméra à champs faiblement joints (Wang *et al.* (2010)) ou au contraire fortement joints pour robustifier le suivi lors de décrochages ou d'occultations (Zhou *et al.* (2009)).
- ▷ Les systèmes travaillant sur les trajectoires de chacun des acteurs de la scène (cf. figure 1.6). Ces trajectoires peuvent, dans une approche basée apprentissage, servir à définir les trajectoires fréquentes de la scène, pour ensuite détecter les trajectoires déviantes de celle de référence comme c'est le cas dans Dee et Hogg (2006), Singh *et al.* (2007), Junejo et Foroosh (2007), Hu *et al.* (2006) et Saleemi *et al.* (2008).
- ▷ Les mouvements relatifs entre ces différentes trajectoires afin d'en inférer des interactions entre les personnes (se séparent, se rapprochent, se battent ...) comme dans Blunsden *et al.* (2007) ou dans Oliver *et al.* (2000) ou pour caractériser des scénarios normaux comme dans Jiang *et al.* (2009).
- ▷ Les applications de comptage de foule par franchissement d'un portail virtuel par une personne suivie comme c'est le cas avec Liu *et al.* (2005), Rabaud et Belongie (2006) ou Sidla et Lypetskyy (2006).

Étude instantanée L'approche ici est de segmenter la personne pour en déterminer une caractéristique précise à un instant précis, comme sa posture ou sa position dans la scène.

- ▷ La détection de l'attitude de déplacement (marche, course ...) par l'étude des membres inférieurs Zhao et Nevatia (2004) ou de posture, en particulier la position couchée comme dans Pham *et al.* (2007) pour détecter des personnes au sol grâce à une caméra infrarouge.
- ▷ La détection d'intrusion en zone interdite, comme c'est la cas dans Monitzer (2006), où le but est de détecter le franchissement des lignes de sécurité sur les quais de métro.

1.3.1.2 Les approches macroscopiques

L'autre grande famille d'approches, dite macroscopique vise à étudier des aspects globaux de la foule (très souvent le mouvement global). Toujours à cause de la mauvaise définition de l'anormal, la frontière entre les domaines de détections des différents systèmes est parfois très fine. Deux modélisations conceptuellement très différentes peuvent au final servir à détecter des évènements communs. On peut citer par exemple :

FIGURE 1.6 – Exemples de trajectoires obtenues par suivi en vue d'une analyse ultérieure.

L'analyse de flux Comme par exemple la détection de mouvements à contresens ou d'absence suspecte de mouvement comme dans Bouchafa *et al.* (1997), Velastin *et al.* (2006) ou Monteiro *et al.* (2007). Kratz et Nishino (2009) considèrent des foules denses en vue de détecter des variations non cohérentes avec ce qui a été observé précédemment. Ces variations peuvent concerner des mouvements à contresens, des personnes flânant ou s'immobilisant. On peut aussi citer l'analyse de la congestion dans un flux étudié dans Andrade *et al.* (2006) ou Zhong *et al.* (2007). Enfin Saleemi *et al.* (2010) se focalisent sur la manière de modéliser des schémas de déplacement.

La détection d'évènements dans la foule Andrade et Blunsden (2006) visent à détecter des perturbations locales causées par la chute d'une personne par exemple. Saad et Shah (2007) détectent des perturbations locales au sein de foules de très grandes ampleurs (ex : vue aérienne de marathon). Mehran *et al.* (2009) cherchent à détecter des ruptures dans les interactions existantes entre les personnes d'une foule (mouvement d'éclatement de la foule), tout comme Wu *et al.* (2010). Adam *et al.* (2008) utilisent plusieurs caméras et différentes informations primaires pour effectuer une détection robuste d'évènements comme des mouvements irréguliers en termes d'orientation ou de vitesse. Mahadevan *et al.* (2010) cherchent à détecter l'anormal, en définissant une notion de saillance de scène permettant de détecter des mouvements irréguliers ou le passage d'entités inhabituelles (vélo sur des chemins piétons par exemple). Enfin, Kim et Grauman (2009) abordent le problème de la détection d'activité anormale dans des scènes de couloir de métro (contresens, personnes flânant, fraude, etc.) avec mise à jour du modèle de normalité.

La perception d'activité au sein d'une foule Ce genre d'applications assez proches de la famille précédente, vise à segmenter la scène en zones de comportements cohérents. Les travaux de Varadarajan et Odobez (2009) et Wang *et al.* (2009) parviennent ainsi à « étiqueter » certains comportements normaux en différentes zones d'une scène (cf. figure 1.7), comme par exemple les trottoirs sont

FIGURE 1.7 – Exemple de segmentation sémantique d'un carrefour issu de Varadarajan et Odobez (2009). En rouge la zone des blobs de petite taille (piétons), en jaune la zone des blobs de grande taille allant vers le bas et en cyan la zone des blobs de grande taille allant vers le haut.

« réservés » aux objets petits, la route aux objets plus gros, etc. Savoir définir de tels comportements peut aussi permettre d'en détecter des anormaux tels que des piétons traversant en dehors de passages piéton, l'arrêt de véhicule dans des zones piétonnes ou dans des zones non autorisées (milieu d'un carrefour), des mouvements à contresens, etc.

L'analyse de succession de comportements En relation avec la perception d'activité, certains travaux se focalisent sur la modélisation des successions et des interactions entre ces comportements. Li *et al.* (2008) mettent en place un système pour la corrélation de comportement routier ou piéton. Ce système permet de détecter des évènements tels que le changement de circulation lors de l'apparition d'un véhicule de secours, certains comportement dangereux de conduite tels que le non respect de distance de sécurité, etc. Küttel *et al.* (2010) proposent d'apprendre automatiquement les dépendances spatio-temporelles des véhicules en déplacement dans une scène complexe. Tout comme Hospedales *et al.* (2009) qui parviennent à déterminer des successions d'évènements que ce soit pour la surveillance de quai de métro (alternance d'attente sur le quai lors de l'absence du train et rapprochement du quai lors de l'arrivée de la rame) ou sur des séquences routières.

La détection d' « image anormale » Les travaux de Breitenstein *et al.* (2009) sont assez à part. Au-delà des approches macroscopiques, ils ne visent pas à étudier le contenu d'une scène mais à analyser l'image dans son intégralité. Toute image trop distante d'un corpus d'images appris représentera l'anormal. Ce corpus d'images est régulièrement mis à jour, il correspond à ce qui est « normal », c'est à dire, ce qui a déjà été vu (cf. images 1.8).

La reconnaissance de geste Ces méthodes cherchent à détecter des gestes de personnes (signe de la main, ...) ou plus généralement des mouvements types (jet d'eau d'une fontaine, plongeon d'un

FIGURE 1.8 – Détection d'image anormale par rapport à un corpus d'images cumulées issu de Breitenstein *et al.* (2009).

nageur, etc) dans une vidéo. Ces vidéos peuvent éventuellement comporter une foule ou un fond dynamique, comme c'est le cas dans Shechtman et Irani (2005) et Ke *et al.* (2007).

Le suivi de personne dans des foules très denses Initialement fondé sur une approche macroscopique pour déterminer un modèle global du mouvement, les travaux de Kratz et Nishimo (2010) s'appuient sur ce modèle global pour le suivi de personnes dans des foules très denses avec beaucoup d'occultations.

L'estimation de densité de foule Sans segmentation de la personne, Ma *et al.* (2004) ne se basent que sur une soustraction de fond et sur une estimation du nombre de personnes en fonction du nombre de pixels de premier plan.

Finalement concernant l'approche mésoscopique, on peut citer un autre système de comptage de personne dans une foule dans Dong *et al.* (2007) qui estiment par apprentissage le nombre de personnes contenues dans un blob en fonction de sa forme et ce grâce à une analyse de Fourrier. Ainsi la foule n'est plus comptée par individus mais par paquets.

1.3.2 Les technologies

La mise en œuvre de ces systèmes nécessite des technologies plus ou moins classiques qui ont chacune leurs avantages et leurs inconvénients. Le choix de ces technologies est d'autant plus critique que les vidéos de foule peuvent présenter des difficultés supplémentaires comme un nombre important d'occultations, une faible résolution des individus ou des contraintes sémantiques liées à l'environnement (comme énoncé dans le partie 1.1).

1.3.2.1 Approches microscopiques

L'idée prédominante est d'isoler les individus en vue de les suivre séparément. Un système de suivi, ou traqueur, repose essentiellement sur deux composantes. Une brique de représentation de la cible et de localisation de celle-ci (ou *Target Representation and Localization*) et une autre de filtrage

et d'association de données (ou *Filtering and Data Association*) se focalisant plus sur la dynamique de la cible. Dépendant du type de scène et du point de vue, un traqueur mettra plus ou moins l'accent sur l'une de ces deux briques. Un suivi de visage dans une foule nécessitera une description accrue de la représentation de la cible plutôt que de la dynamique de celle-ci, alors qu'à l'inverse un suivi pour une vidéosurveillance aérienne privilégiera la dynamique de la cible (Comaniciu *et al.* (2003)). De plus, lorsque l'on cherche à étudier des scènes possiblement denses, ce suivi doit nécessairement être multi-cible (cf. image 1.6).

Représentations des cibles Les différences apparaissent essentiellement aux niveaux du choix de modèles de personne et de la manière de faire correspondre la réalité à ces modèles. Les descripteurs rencontrés pour les segmentations de personnes sont les suivants :

- ▷ Zhao et Nevatia (2004) modélisent les individus par une ou plusieurs ellipses représentant la tête, le tronc et les jambes ou par la forme des épaules et de la tête (communément appelée forme oméga), ce qui est aussi le cas dans Sidla et Lypetskyy (2006). Pham *et al.* (2007) représentent une personne par une ellipse pour la tête, un rectangle pour le tronc et un pour les jambes. La position couchée sera déduite de ce modèle si la position de l'ellipse représentant la tête est sous un seuil donné. Kelly *et al.* (2009) utilisent à partir de mesures issues d'un système stéréoscopique, les proportions du corps humain pour segmenter les personnes
- ▷ Des blobs issus d'une soustraction du fond sont utilisés dans Stauffer et Grimson (2000) et Oliver *et al.* (2000), ou leur centroïde dans Hu *et al.* (2006). Des boites englobantes de blobs représentant à la fois les personnes et leur distance sociale [4], sont suivies par Park et Trivedi (2005). L'inconvénient majeur de ce genre d'approche est la fusion de blobs qui engendre des ambigüités qui demandent en général des informations complémentaires pour être levées. Ce type d'inconvénient, peu gênant dans le cas d'analyse de trafic routier l'est en revanche beaucoup plus dans le cas de foule.
- ▷ Bozzoli *et al.* (2007) proposent d'estimer le nombre de personnes d'une foule traversant une porte virtuelle en étalonnant la quantité moyenne de vecteur déplacement par personne. Cette quantité moyenne est calculée sur les contours irréguliers non jointifs censés s'apparenter le plus à une personne.
- ▷ Yu et Medioni (2009) consolident le suivi lors d'occultations ou de sur-segmentation de blobs, en se servant d'une part de l'information d'apparence avant occultation et d'autre part de l'hypothèse de vitesse continue du blob. Cette correction est réalisée en trouvant la meilleure association temporelle et spatiale des observations.
- ▷ Tyagi *et al.* (2007) utilisent une information combinée issue de N capteurs d'un système stéréoscopique. Cette information est représentée via un histogramme généralisé des couleurs des N caméras.
- ▷ Kratz et Nishimo (2010) discernent les personnes d'une foule dense en combinant à la fois un modèle d'apparence basé sur un histogramme de couleur et un modèle de mouvement déterminé de manière globale sur l'ensemble de la scène, tout comme Zhou *et al.* (2009).

Filtrages et Associations de données Les deux grandes familles de méthodes de filtrage pour le suivi sont le filtre de Kalman (Stauffer et Grimson (2000), Oliver *et al.* (2000), Zhao et Nevatia

4. distance que les personnes laissent naturellement entre elles dans une foule. Cette distance peut varier selon l'affinité entre les personnes (couple / parfaits inconnus, ...).

(2004) et Dee et Hogg (2006)) et plus récemment le filtre particulaire (Bardet *et al.* (2009), Yu et Medioni (2009)). La phase d'association de données est réalisée dans Kelly *et al.* (2009) grâce à un graphe pondéré biparti qui permet d'associer les trajectoires obtenues à l'instant t-1 et les personnes de l'instant t en maximisant la similarité d'un point de vue position courante, taille et couleur de la personne.

Suivi par regroupement de trajectoires De manière moins conventionnelle, certains suivis sont réalisés en regroupant des trajectoires de primitives basiques. L'hypothèse de base est de considérer que les personnes ou de manière plus générales les entités suivies sont assez peu déformables et donc que les traits caractéristiques de ces entités se déplacent selon une trajectoire similaire et à la même vitesse. En se basant sur ce postulat, le principe sera donc de suivre individuellement des primitives plus ou moins basiques d'une image à l'autre, pour ensuite regrouper les trajectoires très fortement similaires. C'est l'approche employée par Brostow et Cipolla (2006), Li et Ai (2007) et Sidla et Lypetskyy (2006). Ces approches ont la particularité de considérer le problème de regroupement comme un problème de classification (utilisation d'un cadre bayésien, avec une maximisation de la vraisemblance). Rabaud et Belongie (2006), quant à eux, répondent au problème du regroupement par une méthode d'estimation de paramètres basée sur l'algorithme RANSAC, Hu *et al.* (2006) utilisent la méthode des *K-means*, alors que Fradet *et al.* (2009) s'appuient sur une méthode très récente de clustering appellée J-Linkage qui semble donner des résultats intéressants. Singh *et al.* (2007) regroupent les trajectoires grâce à l'utilisation de règles répertoriant comment comparer des trajectoires dans des cas singuliers.

Ces méthodes ont l'avantage de définir la notion d'entité et surtout de fournir une information sémantique, à savoir « à l'instant t, une personne ou une voiture se situe à cet endroit précis ». Elles facilitent donc les traitements postérieurs en s'autorisant des *a priori* supplémentaires. Cependant cette segmentation à un coût de calcul proportionnel au nombre de personnes dans la scène, ce qui devient problématique pour des foules denses. A cela on peut ajouter que dans ce dernier cas, le nombre d'occultations augmente, ce qui pénalise fortement ces algorithmes.

1.3.2.2 Approches macroscopiques

Les approches macroscopiques partent d'hypothèses moins lourdes. Elles se basent sur une ou plusieurs informations globales qui peuvent être extraites de l'ensemble ou de parties de l'image.

Le mouvement Le mouvement est la caractéristique la plus commune étudiée sur une foule. Les champs de déplacement sont habituellement déterminés par des algorithmes de flot optique. Différents types d'algorithmes peuvent être utilisés comme le *Block Matching* (Bouchafa *et al.* (1997) et Velastin *et al.* (2006)) où l'orientation du déplacement de la foule est comparé avec une direction fixée selon la scène (type couloir de métro). Kim et Grauman (2009) utilisent aussi le *Block Matching* dans une version multi-échelle, fournissant différentes orientations de mouvement à différentes échelles ainsi qu'une information de vitesse. L'algorithme de Lucas & Kanade est aussi utilisé dans Adam *et al.* (2008), Varadarajan et Odobez (2009), Saleemi *et al.* (2010) et combiné à un filtre médian par bloc dans Monteiro *et al.* (2007). Varadarajan et Odobez (2009), Wang *et al.* (2009), Küttel *et al.* (2010) et Hospedales *et al.* (2009) discrétisent l'orientation du mouvement estimé afin de ne s'intéresser qu'à la tendance générale du mouvement et donc être moins sensibles à la variance de l'estimation. Enfin, la

méthode robuste et affine par morceaux de Black & Anandan est utilisée dans Andrade *et al.* (2006) et Andrade et Blunsden (2006).

Dans le cas où l'orientation du flot optique usuel n'est pas précisée par l'application, un apprentissage automatique est réalisé. Cet apprentissage est global dans Andrade *et al.* (2006) pour détecter des agglutinements de personnes, dans Varadarajan et Odobez (2009) et Wang *et al.* (2009) pour déterminer automatiquement les zones de comportements similaires au sein d'une scène ou dans Küttel *et al.* (2010) et Hospedales *et al.* (2009) pour établir un modèle de dépendance temporel entre les différents mouvements appris. Saad et Shah (2007) effectuent aussi un traitement global sur des vidéos de très grandes foules assimilables à un fluide (ex : vue aérienne de marathon). L'idée étant de détecter l'apparition de zones d'écoulement non homogènes grâce à l'advection [5] de particules dans un champ de mouvement déterminé par *Block Matching* dans le domaine spectral. L'apprentissage sera en revanche plus local dans Adam *et al.* (2008), Monteiro *et al.* (2007) et Andrade et Blunsden (2006), sur des blocs de taille fixe de l'image, pour tout ce qui est variation de trajectoire, mouvement à contresens.

Wu *et al.* (2010) qui traitent aussi la problématique de très grandes foules assimilables à un fluide, utilisent l'advection de particules dans un champ de déplacement obtenu par flot optique. L'objectif étant d'en déduire grâce à des équations de mécanique des fluides, des portions de trajectoires types qui serviront ensuite à la détection d'anomalie.

Structure spatio-temporelle Les structures spatio-temporelles présentent l'avantage d'inclure dans le descripteur une cohérence temporelle. Ces structures peuvent être définies de différentes manières, comme dans Ke *et al.* (2007) qui ré-exploitent les travaux de Shechtman et Irani (2005) en définissant une mesure de similarité entre deux patches. Un patch est une petite zone de l'image considérée sur T images. Cette mesure de similarité permet de mettre en évidence si les gradients du niveau de gris de chacun des patches retraduisent le même mouvement à travers le temps. Ce type de patch permet détecter des gestes particuliers comme un signe de la main, au sein d'une vidéo de foule. Pour un autre type d'application, les structures spatio-temporelles sont aussi utilisé dans Kratz et Nishino (2009) et (2010). L'évolution temporelle du mouvement local au sein d'une foule dense est codée à l'aide de patches spatio-temporels modélisés par une gaussienne 3D. Deux patches peuvent ensuite être comparés grâce à la divergence symétrique de Kullback-Liebler. Ces structures spatio-temporelles sont calculées le long d'une grille régulière en vue d'être apprises en chaque zone de cette grille.

Mahadevan *et al.* (2010) s'appuient sur les textures dynamiques proposées par Chan et Vasconcelos (2008). Ces textures peuvent représenter des changements répétitifs tels que l'aspect de la surface de l'eau ou du passage de personnes dans une foule régulière. Elles sont définies par un modèle génératif définissant une image à l'instant t comme le résultat d'un système dynamique linéaire. A l'image de certain procédés Markoviens, ce système permet de déterminer l'évolution d'un état caché ainsi que l'observation émise de cet état qui n'est autre que l'image à l'instant t, le tout en tenant compte d'éventuels processus perturbateurs modélisant les bruits.

Li *et al.* (2008) utilisent des structures spatio-temporelles d'un tout autre type, les caractéristiques étudiées sont en fait des boites englobantes de blobs obtenues par segmentation fond-forme. Une boite est décrite par un vecteur décrivant sa position, sa taille et son sens de déplacement moyen. Ces vecteurs cumulés dans le temps sont regroupés pour former des vidéos élémentaires dites atomiques. Les co-occurrences normales entre ces vidéos atomiques sont déterminées pendant une phase

5. transport de matière, découplé de la variation intrinsèque de la vitesse dû à des forces externes à un fluide

d'apprentissage pour ensuite déterminer les co-occurrences anormales.

Force Sociale Cette approche cherche à remplacer le champ de déplacement qui est habituellement utilisé pour l'analyse de foule. Les travaux de Mehran *et al.* (2009) s'inscrivent dans une approche physique où la caractéristique de la foule qui va être étudiée n'est pas un champ de déplacement mais un champ de force modélisant les forces intérieures au sein d'une foule. Cette force est calculée par la différence entre le flux moyen de la foule et les flux locaux déterminés par l'advection de particules. Pour illustrer ceci, Mehran *et al.* (2009) font l'analogie avec des feuilles dans un courant d'eau. Le déplacement des feuilles peut très bien ne pas correspondre à celui de l'eau si ces dernières sont ralenties ou bloquées par un obstacle.

Estimation énergétique Une autre analyse du champ de déplacement peut être effectuée à l'aide d'une approche énergétique, comme la présente Zhong *et al.* (2007). La notion d'énergie de foule est définie par analogie avec l'énergie cinétique des objets en physique (un corps en mouvement possède une énergie, donc en particulier la foule). Différentes fonctions d'énergie sont définies, la première approxime la vitesse par la variation de luminance, la seconde utilise l'estimation de la vitesse par le flot optique. Des fluctuations brutales mises en évidence grâce à une analyse en ondelettes, permettent de détecter des perturbations, telles que des congestions. Les résultats énoncés par Zhi montrent que la première définition de l'énergie, semble plus robuste au bruit mais aussi moins sensible à la détection contrairement à la seconde.

Caractéristiques combinées Certains travaux utilisent des critères de différentes natures afin de maximiser le panel des évènements détectés. Citons par exemple Varadarajan et Odobez (2009) qui en plus de considérer l'orientation du flot optique en un pixel (nord, sud, est, ouest), considèrent en même temps la position de ce pixel dans la scène ainsi que la taille du blob auquel il appartient. Cette variété dans la description permet de différencier, après apprentissage, les zones réservées aux objets de petite taille (piétons) des zones réservées aux plus gros objets (véhicules), les sens de circulation, etc.

Au delà de la foule Cette partie traite des travaux de Breitenstein *et al.* (2009). Cette approche un peu à part ne considère pas une caractéristique de la foule mais les images dans leur intégralité. C'est une pyramide d'histogramme de gradient qui va caractériser l'image. L'objectif ensuite sera de mémoriser de manière efficace tout ce qui à été vu afin de pouvoir mettre en évidence quelque chose apparaissant pour la première fois. Breitenstein *et al.* (2009) appliquent ces travaux sur des scènes pouvant accueillir beaucoup de trafic et ils mettent en évidence la possibilité de détecter des véhicules en panne, garés ou l'apparition de manifestations ponctuelles telles que des marchés, etc.

Du fait que le travail de segmentation des personnes n'a pas été réalisé, l'information traitée est « anonyme », si par exemple l'analyse s'effectue sur l'information de mouvement, cela peut très bien être à cause d'un piéton en mouvement, d'un arbre trop secoué par le vent, d'un reflet ou d'une ombre projetée, etc. C'est parce que ce genre d'approches ne repose pas sur des *a priori* forts que les systèmes sont conceptuellement très différents, contrairement aux approches microscopiques qui ont quasiment toutes en commun une phase de suivi. De plus, les évènements détectés peuvent être très différents, parfois inattendus ou dénués de sens.

1.3.2.3 Les mécanismes d'apprentissage

Nous avons vu dans la section précédente les différents traits qui pouvaient être utilisés pour traiter de l'analyse de foule, que ce soit des traits individualisant la personne ou non. Un traitement courant, une fois ces caractéristiques définies, est de déterminer celles qui correspondent à un comportement normal dans la scène ou non. Cette normalité peut parfois être déterminée soit par des règles a priori comme dans Bouchafa *et al.* (1997) et Velastin *et al.* (2006), soit par des règles métiers très liées aux caractéristiques étudiées et à la scène comme dans Junejo et Foroosh (2007) et Dee et Hogg (2006), ou par des mécanismes d'apprentissage s'appuyant sur des algorithmes spécialisés. Ces approches basées apprentissage cherchent en général à définir un modèle statistique d'une caractéristique donnée quelle qu'elle soit et donc de classifier le rare plutôt que l'anormal, ce qui dans de nombreux cas pourra suffire.

Les approches de partitionnement de données Ces approches intitulées *data clustering*, visent à diviser un ensemble de données en différents « paquets » homogènes. C'est le cas du *K-means* qui est utilisé dans Hu *et al.* (2006) pour déterminer des centroïdes de blob ou dans Wu *et al.* (2010) pour regrouper des trajectoires types. Pour déterminer le nombre optimal de clusters, Wu *et al.* (2010) initialisent volontairement un grand nombre de clusters pour ensuite en réduire le nombre itérativement, en fusionnant deux clusters si la divergence de Jensen-Shannon est en dessous d'un seuil.

Les approches bayésiennes Bien souvent les hypothèses émises pour la résolution d'un problème permettent de se placer dans un cadre bayésien. Dans ce genre d'approche une estimation de la vraisemblance *a posteriori* est généralement effectuée. Pour réaliser ces estimations, différents types d'algorithmes existent. On peut citer les algorithmes EM Park et Trivedi (2005), Liu *et al.* (2005), Saleemi *et al.* (2010) ou encore Mahadevan *et al.* (2010) qui l'utilisent pour définir le modèle des mélanges de textures dynamiques. Li *et al.* (2008) utilisent en plus le critère d'information bayésienne de Schwarz pour déterminer automatiquement le nombre de gaussiennes. On peut aussi citer les fenêtres de Parzen Saleemi *et al.* (2008), le *Sequential Kernel Density Approximation* et les chaînes de Markov Monte Carlo (MCMC) comme dans Pham *et al.* (2007), Saleemi *et al.* (2008) ou Yu et Medioni (2009).

De manière générale, les différentes méthodes d'apprentissage utilisées pour l'analyse de scènes possiblement denses sont en général sans mises-à-jour. Les apprentissages sont donc réalisés durant une phase hors ligne. Quelques exceptions méritent cependant d'être mises en avant comme Breitenstein *et al.* (2009) qui par une technique d'apprentissage ad hoc, mettent-à-jour le modèle issu de l'apprentissage une fois par jour durant une phase semi en ligne. Le système n'effectue plus d'analyse durant cette phase (relativement courte) et le modèle est enrichi des informations de la journée. Cette méthode ad hoc est assimilable à une approche bayésienne puisque la détection s'effectue par seuillage d'une distribution de scores de similarités.

La famille des HMMs (*Hidden Markov Model*) Zhao et Nevatia (2004) utilisent les chaînes de Markov cachées pour l'analyse des membres inférieurs des personnes segmentées. Kratz et Nishino (2009) ajoutent de la cohérence séquentielle sur les patches spatio-temporels calculés grâce à des HMMs définis en chaque zone de la grille fixe. Les états de ces HMMs sont déterminés à partir des patches les plus représentatifs qui ont été vus durant l'apprentissage. De plus Kratz et Nishino (2009) ajoutent à cette cohérence temporelle une cohérence spatiale entre un patch et ses voisins

directs, par le biais de *Couple-HMMs*. Oliver *et al.* (2000) utilisent aussi des *Coupled-HMMs* pour analyser les interactions entre deux trajectoires issues de suivis. Andrade *et al.* (2006) utilisent les *Multi Observation HMMs* pour apprendre les composantes principales du flot optique obtenues par une Analyse en Composantes Principales (ACP). Andrade et Blunsden (2006) ont aussi utilisé une variante des HMMs, les *Mixture Of Gaussian HMMs* utilisant des mélanges de gaussiennes en guise de lois d'émissions de probabilité, pour apprendre globalement ou localement (afin d'être plus sensible aux faibles variations) la distribution du flot optique. Jiang *et al.* (2009) quant à eux, adaptent les mécanismes d'apprentissage des HMMs afin de pouvoir utiliser des lois d'émissions qui leurs sont propres et ainsi modéliser les co-occurences d'évènements d'une scène. Hospedales *et al.* (2009) Küttel *et al.* (2010) combinent des variantes des HMMs avec des mécanismes issus du traitement du langage naturel qui seront décrits dans le paragraphe suivant. Dans les deux cas, ces approches permettent de créer des réseaux de dépendances entre les actions susceptibles d'apparaitre dans la scène.

Les champs de Markov Kim et Grauman (2009) utilisent un champ de Markov spatio-temporel pour la modélisation par bloc d'une scène complète. De plus, en le couplant avec une analyse en composantes principales pour déterminer le mouvement prépondérant par bloc, le champ de Markov modélisant la scène est mis à jour régulièrement.

Les approches issues de l'analyse de texte Ces approches permettent d'établir des relations entre un ensemble de documents et les termes, « les mots » qu'ils contiennent, en construisant des « sujets » liés aux documents et aux termes. Varadarajan et Odobez (2009) utilisent une variante du LSA (*Latent Semantic Analysis*), le *probabilistic LSA* pour apprendre les caractéristiques de position, de taille et de déplacement comme expliqué dans la partie 1.3.2.2. Mehran *et al.* (2009) utilisent l'algorithme du LDA (*Latent Dirichlet Allocation*) avec des mots extraits de la force sociale issue de leurs travaux. Les mots sont de petites imagettes extraites sur T images consécutives de la norme du champ de force sociale. Hospedales *et al.* (2009) utilisent aussi le LDA sur l'orientation de flot optique discrétisé selon les 4 points cardinaux. Wang *et al.* (2010) considèrent quant à eux des mots constitués d'une position et d'une direction pour une caméra donnée, extraites de trajectoires obtenues par un suivi sur cette même caméra. Küttel *et al.* (2010) s'appuient sur l'algorithme HDP (*Hierarchical Dirichlet Process*) pour déterminer les actions possibles au sein de la scène qui correspondront aux « sujets » déterminés par l'algorithme HDP. Cet algorithme présente l'avantage de ne pas faire d'*a priori* sur le nombre de « sujets » de la scène contrairement au LDA. Une extension de cet algorithme, le dual-HDP est comparé à sa version classique ainsi qu'au LDA dans Wang *et al.* (2009) montrant un gain en précision en contrepartie d'une complexité en phase d'apprentissage supérieure.

1.4 Notre approche

1.4.1 Nos choix

Pour répondre aux besoins énoncés au début de ce chapitre, différents choix ont été effectués. Dans l'optique d'un système facilement déployable, aucune calibration n'est effectuée, ce qui implique qu'aucun *a priori* sur la géométrie de la scène ne peut être exploité. Une calibration est en effet, une

étape délicate surtout quand il s'agit de déployer des réseaux de caméras particulièrement étendus comme dans le cas de la vidéosurveillance d'espaces publics.

Dans l'optique de s'adapter au mieux aux contraintes intrinsèques à ce genre de système, une approche globale a été choisie :

- ▷ Afin de ne pas être dépendant de la densité de la foule ou du trafic observé, en termes de puissance de calcul.
- ▷ Afin d'exploiter une caractéristique certes basique mais moins sujette à la propagation d'erreur. En effet, une caractéristique globale représente un état courant et ne repose pas sur un processus évolutif à long terme comme le suivi. Le flot optique s'il est erroné ne fait intervenir que deux images alors qu'une piste de suivi si elle se perd ne sera que très rarement récupérée.

Nous avons choisi d'analyser le mouvement. Cette caractéristique, nous a semblé la moins sensible au niveau de densité de la foule, contrairement à des blobs qui dans le cas d'une foule extrêmement dense n'apportent que très peu d'information, ou des caractéristiques telles que la force sociale ou des modélisations énergétiques qui nécessitent à l'inverse des foules particulièrement denses. Le mouvement présente aussi l'avantage de travailler plus ou moins directement avec les gradients de luminance. En conséquence, une telle caractéristique est particulièrement robuste aux changements de conditions climatiques et d'illuminations. De plus amples détails concernant son estimation, ses avantages et ses inconvénients seront détaillés dans le chapitre 2.

Pour ce qui concerne l'adaptativité et la généricité du système, une approche basée apprentissage semblait plus apropriée qu'une approche basée modèle. Ainsi aucun *a priori* sur l'apparence des entités traitées n'est utilisé. Notre système peut donc aussi bien fonctionner dans le cadre de la vidéosurveillance de foule que dans celui du trafic routier.

De plus, le cadre applicatif est la vidéosurveillance d'évènements anormaux, donc « rares ». L'exploitation des bases labellisées est impossible car trop fastidieuse à réaliser, trop dépendante de la scène observée et trop mal définie quand il s'agit de définir l'anormal. En conséquence, un mécanisme d'apprentissage non supervisé est requis. Différentes approches ont été évaluées et seront à l'étude dans les chapitres 3 et 4.

1.4.2 Notre système

Notre système consistera donc à subdiviser une scène de manière régulière selon une grille fixe. Chaque subdivision de la scène que nous appellerons par la suite « bloc », permettra une étude locale du mouvement. En chacun des blocs de cette grille sera caractérisé, à chaque instant, un mouvement. Dans une première phase hors ligne, un modèle du ou des mouvements susceptibles d'être observés dans chacun de ces blocs est déterminé par apprentissage, puis dans une seconde phase en ligne, les caractérisations courantes du mouvement sont comparées au modèle appris. Le mécanisme d'apprentissage étant non supervisé, un critère de choix d'appartenance au modèle sera défini selon la distribution issue de l'apprentissage. L'ensemble du système est schématisé à la figure 1.9

A titre illustratif, afin d'avoir une idée précise du contexte applicatif, des entrées et sorties de notre système, un exemple réel d'application qui respecte le schéma présenté à la figure 1.9 est fourni à la figure 1.10. Depuis une caméra assez distante de faible résolution, un carrefour routier complexe de région parisienne est analysé. Les conditions climatiques et d'illumination peuvent être très changeantes. Ici le soleil de fin d'après-midi engendre des ombres portées de grande taille. La circulation normale, peut être très faible ou au contraire extrêmement dense aux heures de pointe.

La phase d'apprentissage permet, par exemple, d'aboutir à un ensemble de directions prédomi-

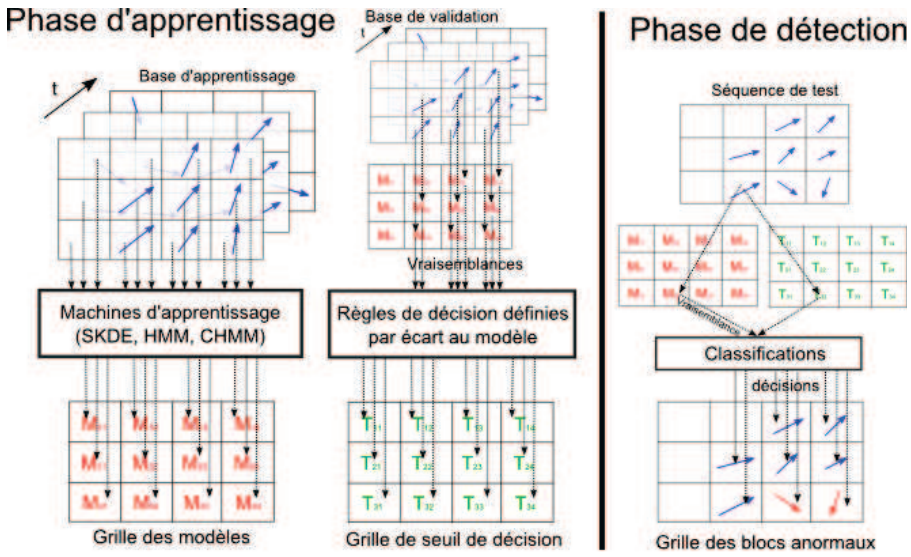


FIGURE 1.9 – Architecture du système proposé.

nantes représentées par des vecteurs dont la couleur dépend de l'orientation (cf. légende colorimétrique de la figure 1 présentée dans les notations page 3), en chaque bloc de la scène. Associé à chacun de ces modèles, le critère de décision est représenté sur la figure 1.10 par un bloc vert plus ou moins transparent dépendant du seuil de classification obtenu. Plus le seuil est élevé, plus la transparence est faible, ce qui correspond à des zones où le mouvement est très précisément défini et où l'incertitude de classification est faible.

En phase d'analyse, la classification par bloc permet par exemple de détecter le camion de pompier allant à contresens pour éviter les autres véhicules arrêtés au feu (cf. image 1.10). Cet évènement n'est pas anormal sémantiquement parlant puisque c'est un véhicule prioritaire mais cela reste néanmoins un évènement rare qu'il est légitime de mettre en exergue.

1.4.3 Les principaux verrous

Parmi les pré-requis nécessaires à notre système, on peut citer la nécessité d'avoir une caméra fixe ainsi que l'analyse d'un flux d'entité relativement structuré. En effet, une place publique dans laquelle les mouvements sont erratiques, où toutes les orientations de mouvement sont susceptibles d'être observées, ne sera pas réellement contrôlable par notre système. Pour ce genre de scène, la caractéristique de mouvement n'est pas celle susceptible de porter l'information de situations anormales.

Zhong *et al.* (2004) qualifient le problème de détection d'évènement anormaux de « dur à décrire mais facile à vérifier ». Facile à vérifier dans le sens où si l'on possède une grande quantité d'observations, il est relativement facile de vérifier si ces évènements sont inhabituels ou non. En revanche, le choix de la description est particulièrement délicat comme on a pu le voir au début de ce chapitre.

Son analyse met aussi en avant le fait que pour des détections de longues durées, à plus forte raison des systèmes de vidéosurveillance qui sont censés fonctionner continuellement, des primitives d'image robustes et surtout persistantes doivent être utilisées. Or dans des environnements faiblement contrôlés, les approches microscopiques auront plus facilement tendance à échouer, d'où notre choix de ne pas s'appuyer sur des techniques de suivi. A contrario, l'utilisation de primitives plus basiques, tel que le mouvement ou des calculs de blobs sont plus fiables pour ce genre d'application.

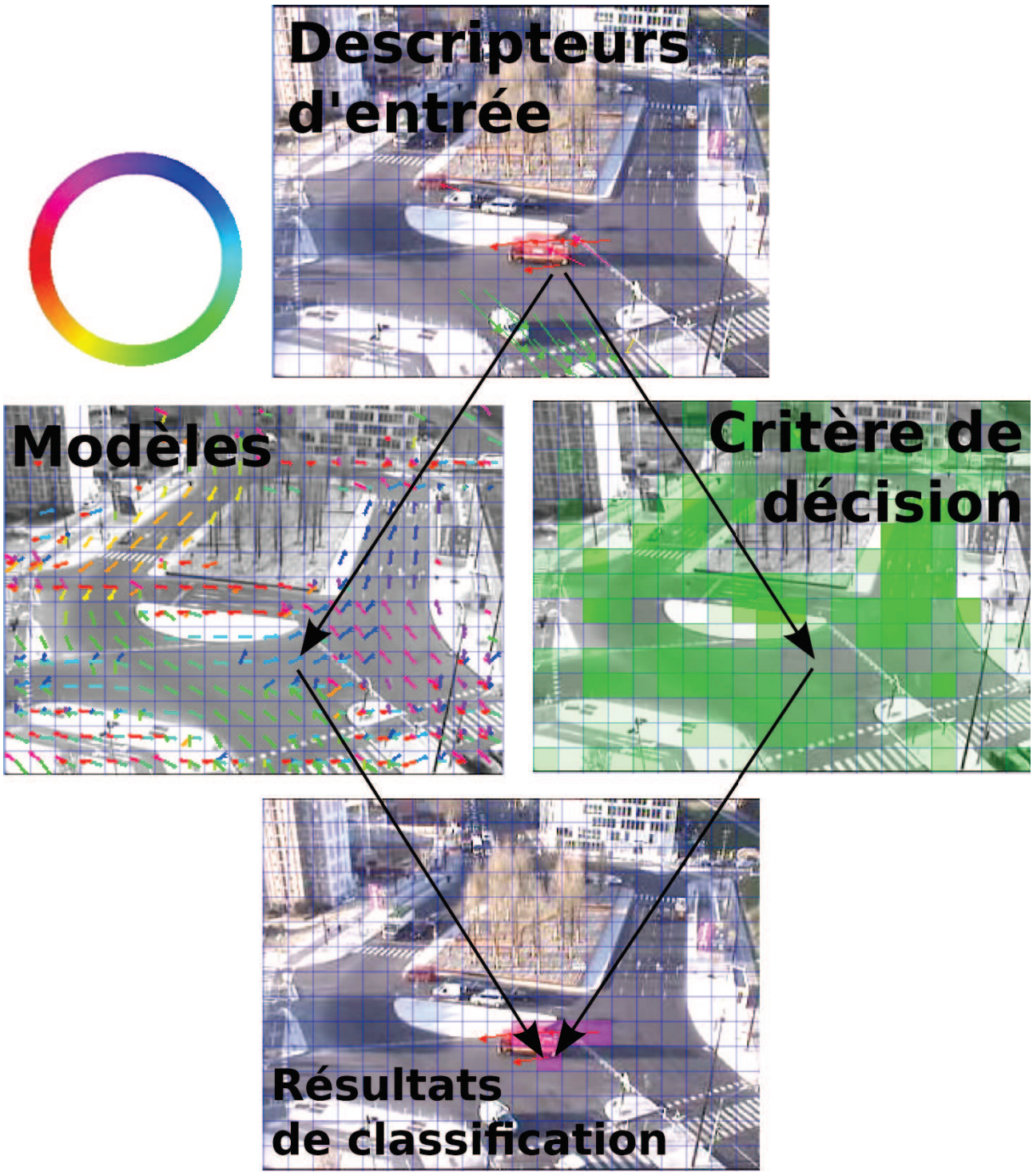


FIGURE 1.10 – Exemple d'utilisation type de l'application.

En revanche, ces dernières primitives souffrent essentiellement de manque de précision, dû au bruit constamment présent.

C'est précisément sur ce point qu'un important travail sera effectué. Ce sera un des objectifs principaux d'atténuer autant que possible ce bruit (ou ses effets), afin de pouvoir exploiter au mieux l'information importante de ces primitives.

Cependant, l'utilisation de descripteurs plus élaborés caractérisant le mouvement de manière plus robuste mais aussi plus complexe, implique d'utiliser des machines d'apprentissage compatibles avec ces descripteurs. En effet, certaines approches statistiques utilisent des distances spécifiques, telles que la distance de Mahalanobis par exemple, ce qui pour certains descripteurs peut ne pas avoir de sens. Cette contrainte entre descripteur et machine d'apprentissage ainsi que les solutions pour pallier ce problème feront aussi l'objet d'un intérêt particulier au travers de ce manuscrit.

1.4.4 Plan du mémoire

Ce manuscrit est articulé autour des différentes briques nécessaires à la construction de notre système. Le chapitre suivant a pour but d'introduire les descripteurs utilisés pour la caractérisation du mouvement. Le chapitre 3 détaille une première approche statistique pour l'apprentissage du mouvement via un procédé non supervisé et automatique. Associé à cette première méthode, un critère de décision pour la classification sera aussi présenté. Dans le chapitre 4, l'utilisation de réseaux bayésiens est à l'étude comme alternative à l'apprentissage statistique image par image réalisé au chapitre 3. Enfin, le chapitre 5 clôture ce manuscrit en comparant les différentes briques algorithmiques mises en œuvre via un protocole expérimental détaillé permettant une évaluation précise de notre système. Des conclusions générales et des perspectives sur le sujet de la détection d'évènements exceptionnels dans un flot terminent ce manuscrit.

Chapitre 2

Extraction de primitives de caractérisation du mouvement

Ce chapitre a pour but d'explorer les techniques permettant d'exploiter l'information de mouvement d'une scène. La première partie de ce chapitre est une revue des méthodes classiques d'estimation du mouvement au travers des algorithmes de flot optique. La deuxième présente une autre famille d'approches fondées sur des structures spatio-temporelles. Nous en ferons un tour d'horizon avant d'apporter notre contribution en proposant notre propre descripteur. Les travaux réalisés sur le descripteur proposé ont été publiés dans Luvison *et al.* (2011).

2.1 Introduction

Comme nous l'avons expliqué au chapitre précédant notre système doit détecter des évènements anormaux à partir de l'étude du mouvement d'une foule ou, de manière plus générale, d'un flot d'objets. Dans ce chapitre, nous nous attarderons sur les différentes manières d'estimer ce mouvement ou de le caractériser. L'utilisation finale de cette estimation étant la classification, l'idée sous-jacente est de pouvoir différencier au mieux deux mouvements différents. De plus, une caractérisation la plus robuste possible est recherchée afin d'éviter au maximum le bruit pouvant conduire à l'estimation de mouvements aberrants.

Nous présenterons tout d'abord une famille d'algorithmes répondant au problème de l'estimation de mouvement que sont les algorithmes de flot optique. Après avoir rappelé le principe fondamental sur lequel s'appuient ces algorithmes, un état de l'art général sur les différentes variantes existantes est effectué avant de terminer sur les principaux avantages et inconvénients de ces méthodes et le choix qui a été retenu pour notre application.

La seconde partie de ce chapitre abordera une autre approche pour la caractérisation du mouvement basée sur des structures spatio-temporelles. Une de nos contributions sera de présenter une nouvelle méthode s'appuyant sur des calculs de corrélation spatio-temporelle au sein d'une structure elle-même spatio-temporelle.

2.2 Une approche classique : Le flot optique

2.2.1 État de l'art

La problématique de l'estimation du mouvement dans une vidéo est un problème particulièrement difficile car il s'appuie sur des hypothèses faibles. En effet, aucune notion d'objet n'étant définie, tous les pixels sont traités de manière indépendante sans modéliser explicitement d'éventuelles discontinuités entre les objets.

Toutes ces méthodes s'appuient sur le même principe de base, à savoir, « la luminance d'un objet est constante d'une image à l'autre ». En d'autres termes, cela revient à chercher en une position $\mathbf{x}$ de l'image, le vecteur déplacement $\mathbf{v}$ tel que à l'image suivante, $\mathbf{x} + \mathbf{v}$ permette d'obtenir la même luminance, donc le même niveau de gris. De manière plus formelle, ce principe souvent appelé dans la littérature « contrainte d'illumination » ou « conservation des données » peut s'écrire :

$$I(\mathbf{x}, t) = I(\mathbf{x} + \mathbf{v}, t + 1) \tag{2.1}$$

avec $I(\mathbf{x}, t)$ la valeur du pixel à l'instant t, à la position $\mathbf{x}$.

Cependant, ce principe est en pratique très souvent violé comme par exemple au niveau des frontières des objets (apparition de pixels préalablement occultés) ou sur des surfaces réfléchissantes provoquant des spécularités. En conséquence, une seconde contrainte, est appliquée. Cette contrainte suppose que des pixels voisins ont de fortes chances d'appartenir au même objet et donc que le mouvement au voisinage d'un pixel évolue de manière continue. Cette hypothèse suppose une certaine rigidité des objets, du moins localement. Cette « cohérence spatiale », nécessaire pour pouvoir répondre au problème du flot optique, n'est pourtant pas systématiquement vérifiée, elle non plus (comme par exemple une nouvelle fois aux frontières des objets). Les algorithmes existants diffèrent par la formulation de ces deux contraintes afin de corriger au mieux ces discontinuités et autres problèmes du flot optique.

Cet état de l'art des techniques de flot optique, n'a pas vocation à détailler précisément toutes les techniques existantes. L'objectif est de réaliser un tour d'horizon de ce qui est communément utilisé en vision par ordinateur. Différentes approches existent pour répondre à ce problème. Parmi les principales, on peut citer, à l'instar de Barron *et al.* (1992), les méthodes différentielles, les méthodes par corrélation et les méthodes par régression.

2.2.1.1 Les méthodes différentielles

Ces méthodes visent à résoudre le problème (2.1) en l'exprimant de manière différentielle :

$$I(\mathbf{x}, t) = I(\mathbf{x} - \mathbf{v}t, 0) \tag{2.2}$$

En utilisant un développement de Taylor de $\dfrac{dI(\mathbf{x}, t)}{dt}$, la contrainte de gradient peut s'exprimer :

$$\boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x}, t).\mathbf{v} + I_t(\mathbf{x}, t) = 0 \tag{2.3}$$

où $I_t(\mathbf{x}, t)$ représente la dérivée partielle par rapport au temps et $\boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x}, t)$ les dérivées spatiales.

Différents algorithmes utilisant ce principe existent. Très anciennes mais pourtant toujours très utilisées, les méthodes de Lucas et Kanade (1981) et de Horn et Schunck (1981), se différencient par la manière de formuler la contrainte spatiale.

Lucas et Kanade (1981) Cet algorithme suppose que le vecteur déplacement $\mathbf{v}$ est constant sur un voisinage donné. La résolution du problème (2.3) revient donc à résoudre sur une fenêtre spatiale Ω le problème de la détermination de :

$$\underset{\mathbf{v}}{argmin} \sum_{x \in \Omega} \|W(\mathbf{x}) \left[\boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x},t).\mathbf{v} + I_t(\mathbf{x},t)\right]\|^2 \tag{2.4}$$

où W est une matrice de pondération de la même taille que Ω permettant de donner plus d'influence aux pixels proches du centre de la fenêtre. En l'exprimant de manière matricielle, cela revient en pratique à résoudre :

$$A^T W^T W A v = A^T W^T W b \tag{2.5}$$

Avec pour l'ensemble des N pixels $\mathbf{x_i}$ appartenant à Ω,

$$A = [\boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x_1}), ..., \boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x_N})]^T, \tag{2.6}$$

$$W = diag[W(\mathbf{x_1}), ..., W(\mathbf{x_N})], \tag{2.7}$$

$$\mathbf{b} = -(I_t(\mathbf{x_1}), ..., I_t(\mathbf{x_N}))^T \tag{2.8}$$

Certaines variantes comme celle de Simoncelli *et al.* (1991) analysent les valeurs propres associées à la matrice $A^T W^T W A$. Si les deux valeurs propres sont trop faibles, la région de l'image étudiée au travers de la fenêtre Ω ne représente qu'une zone de basse fréquence et donc le mouvement ne peut être déterminé.

Horn et Schunck (1981) L'approche ici consiste à combiner la contrainte d'illumination et la contrainte spatiale en définissant une fonction d'énergie qui sera à minimiser. On recherche :

$$\underset{\mathbf{v}}{argmin} \int_D (\boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x},t).\mathbf{v} + I_t(\mathbf{x},t))^2 + \alpha^2(\|\nabla u\|_2^2 + \|\nabla v\|_2^2) d\mathbf{x} \tag{2.9}$$

avec D une fenêtre d'analyse, u et v les composantes du vecteur $\mathbf{v}$ et α un paramètre de relaxation caractérisant l'influence de la contrainte spatiale qui suppose la continuité du champ de mouvement. En pratique, les composantes u et v sont déterminées de manière itérative :

$$\begin{aligned} u^{k+1} &= \bar{u}^k - \frac{I_x[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \\ v^{k+1} &= \bar{v}^k - \frac{I_y[I_x\bar{u}^k + I_y\bar{v}^k + I_t]}{\alpha^2 + I_x^2 + I_y^2} \end{aligned} \tag{2.10}$$

Avec k le nombre d'itérations et $\bar{u}^k$ et $\bar{v}^k$ la moyenne des voisinages de u^k et v^k.
Les vitesses initiales u^0 et v^0 sont fixées à zéro.

Faiblesses Cependant cette approche et de manière générale les méthodes différentielles souffrent du bruit et de la discrétisation pixelique qui rend l'estimation des dérivées très imprécises. Dans le cas de Horn et Schunck (1981), une phase de pré-filtrage est réalisée en appliquant un filtre gaussien spatialement mais aussi temporellement sur les images.

A noter qu'il est aussi possible d'exprimer la contrainte de gradient à l'aide de dérivées du second ordre. Cependant ces dérivés, de par leur sensibilité au bruit, sont encore plus difficiles à estimer

et ce genre de modélisation de la contrainte d'illumination est surtout utilisé pour l'estimation du flux potentiellement divergent ou rotatif (images satellitaires par exemple, Corpetti *et al.* (2002)). La nature des scènes étudiées dans le cadre de nos travaux met plutôt en avant des flots laminaires, c'est pourquoi ce type de méthode n'a pas été envisagé.

2.2.1.2 Les méthodes par corrélation

Ces méthodes font partie de la vaste famille d'algorithme appelée *Block Matching* qui consiste a mettre en corrélation une fenêtre de taille D centrée aux coordonnées (x_s, y_s), extraites d'une image que nous appellerons fenêtre source, avec une autre fenêtre centrée aux coordonnées (x_c, y_c) sur une autre image, que nous appellerons fenêtre cible. Une fois ce couple de fenêtres trouvé, le vecteur de déplacement $\mathbf{v}$ est le vecteur défini entre les deux centres. Essentiellement utilisé dans les algorithmes de compression vidéo pour leur adéquation avec la philosophie de traitement par bloc, l'information de mouvement est particulièrement utile pour éliminer les redondances temporelles entre les images d'une vidéo.

Cette mise en corrélation s'effectue en maximisant une mesure de similarité ou en minimisant une mesure de distance. A noter que la résolution de ce problème d'optimisation en tous les pixels de l'image est extrêmement coûteuse, rendant impossible tout traitement temps-réel. En conséquence, le *Block Matching* n'est calculé que de manière parcimonieuse, sur une image sous-échantillonnée. De plus, des stratégies de recherche sont mises en place pour déterminer plus rapidement l'association de fenêtre optimale. Les différentes variantes de ces méthodes se différencieront essentiellement à la fois sur le critère de comparaison utilisé ainsi que sur la stratégie de recherche.

Parmi les différents critères de comparaison on peut citer Brown *et al.* (2003) :

▷ Maximiser la corrélation croisée normalisée :

$$NCC_{1,2}(\mathbf{x},\mathbf{v}) = \frac{\sum_{\mathbf{d}\in D}(I_1(\mathbf{x}+\mathbf{d})-\bar{I_1})(I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})-\bar{I_2})}{\sqrt{\sum_{\mathbf{d}\in D}(I_1(\mathbf{x}+\mathbf{d})-\bar{I_1})^2}\sqrt{\sum_{\mathbf{d}\in D}(I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})-\bar{I_2})^2}} \tag{2.11}$$

▷ Minimiser la somme des carrés des différences :

$$SSD_{1,2}(\mathbf{x},\mathbf{v}) = \sum_{\mathbf{d}\in D}[I_1(\mathbf{x}+\mathbf{d}) - I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})]^2 \tag{2.12}$$

▷ Minimiser la somme des carrés des différences normalisées :

$$NSSD_{1,2}(\mathbf{x},\mathbf{v}) = \sum_{\mathbf{d}\in D}\left(\frac{(I_1(\mathbf{x}+\mathbf{d})-\bar{I_1})}{\sqrt{\sum_{\mathbf{d}\in D}(I_1(\mathbf{x}+\mathbf{d})-\bar{I_1})^2}} - \frac{(I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})-\bar{I_2})}{\sqrt{\sum_{\mathbf{d}\in D}(I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})-\bar{I_2})^2}}\right)^2 \tag{2.13}$$

▷ Minimiser la somme des différences absolues :

$$SAD_{1,2}(\mathbf{x},\mathbf{v}) = \sum_{\mathbf{d}\in D}|I_1(\mathbf{x}+\mathbf{d}) - I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})| \tag{2.14}$$

Avec $\bar{I_1}$ la moyenne de la fenêtre $I_1(\mathbf{x}+\mathbf{d})$ et $\bar{I_2}$ celle de $I_2(\mathbf{x}+\mathbf{v}+\mathbf{d})$.

Concernant la stratégie de recherche de la fenêtre cible permettant d'obtenir la corrélation optimale avec la fenêtre source, elle est généralement limitée à une fenêtre maximale. De plus, la plupart des algorithmes effectuent des recherches de proche en proche sur des voisinages particuliers, à l'image d'algorithme de descente de gradient. Ces différentes approches ont toutes le même déroulement général, à savoir :

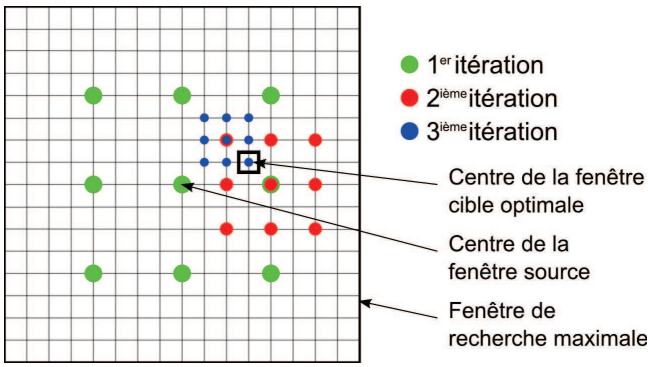


FIGURE 2.1 – Exemple de déroulement de l'algorithme de recherche *Three Step Search*.

- ▷ Considérer comme point central, le pixel de coordonnées (x_s, y_s) sur l'image cible.
- ▷ A partir de ce point central, déterminer un voisinage parcimonieux de pixels. Ce voisinage est échantillonné à une échelle p initialement grande.
- ▷ En chacun des points de ce voisinage ainsi que sur le point central, une comparaison entre la fenêtre source et une fenêtre centrée en ce point est effectuée.
- ▷ Le centre de la fenêtre donnant la similitude optimale est désigné comme nouveau point central et la phase 2 est répétée avec un voisinage échantillonné de manière plus précise, en diminuant p.
- ▷ L'algorithme s'arrête lorsque la précision de l'échantillonnage est de l'ordre du pixel ($p = 1$).

Plus concrètement, on peut citer [1] les méthodes suivantes :

Full Search Algorithm (FSA) L'algorithme de recherche « naïf » qui consiste à rechercher le centre de la fenêtre cible sur l'ensemble des pixels de la fenêtre maximale. De complexité importante, cet algorithme donne la réponse optimale dans la fenêtre maximale.

Three Step Search (TSS) Mise en place par Koga *et al.* (1981), cette méthode consiste à considérer comme voisinage des points disposés en 8-connexité [2] et éloignés d'un pas p (en général $p = 4$). A l'itération suivante le pas est divisé par 2. Un exemple de déroulement de l'algorithme est présenté figure 2.1. Les points issus de la première itération sont représentés en vert, ceux de la seconde en rouge et enfin ceux de la dernière itération en bleu et le point de similitude optimale est entouré d'un rectangle noir.

Two Dimensional Logarithmic Search (TDL) Similaire dans les grandes lignes au TSS, cette approche proposée par Jain et Jain (1981) utilise un voisinage disposé en 4-connexité [3] au lieu de 8. De plus le pas n'est divisé par deux que lorsque l'optimum est le point central courant.

Orthogonal Search Algorithm (OSA) Cette approche hybride entre le TSS et le TDL développée par Puri *et al.* (1987), possède les mêmes étapes que le TSS mais l'exploration du voisinage est réalisée en deux temps. Dans une première phase, un voisinage est composé de deux voisins alignés

1. http ://www.ece.cmu.edu/~ee899/project/deepak_mid.htm
2. points adjacents horizontalement, verticalement et diagonalement
3. points adjacents horizontalement et verticalement

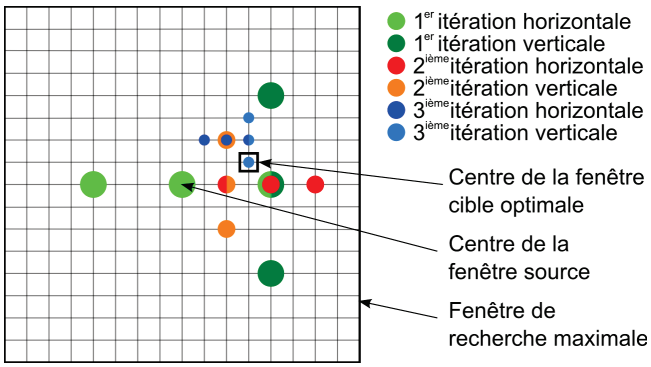


FIGURE 2.2 – Exemple de déroulement de l'algorithme de recherche *Orthogonal Search Algorithm*.

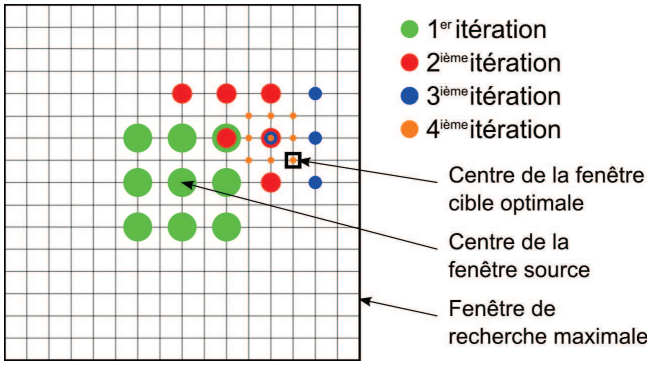


FIGURE 2.3 – Exemple de déroulement de l'algorithme de recherche *Four Step Search*.

horizontalement de part et d'autre du point central. Dans la deuxième phase, un nouveau voisinage composé de deux voisins alignés verticalement centrés sur le meilleur point du voisinage précédent, est considéré. Une fois les deux étapes réalisées, le pas est divisé par deux. Un exemple est fourni à la figure 2.2.

Four Step Search (FSS) Cette recherche itérative sur des sous voisinages a été présentée par Po et Ma (1996). Elle vise à échantillonner avec un pas de $p = 2$, le point central initial de manière isotrope avec un voisinage en 8-connexité, puis de manière anisotrope en tenant compte de la position de la recherche précédente par la suite et ce jusqu'à ce que l'optimum soit le point central courant. Une fois ce point fixe obtenu, le pas est divisé par 2 et une ultime recherche isotropique est réalisée. Deux cas peuvent se présenter pour la définition du voisinage anisotropique selon la situation de la cible optimale de l'étape précédente dans un coin ou sur un bord. La figure 2.3 représente la disposition de ces voisinages au travers des ensembles de points rouges et bleus.

Bien que prévue pour répondre rapidement au problème de déplacement de bloc, ce qui correspond totalement à l'architecture de notre application, la qualité du champ de déplacement est en général bien inférieure aux méthodes différentielles comme le montre la figure 2.4. Cette comparaison est effectuée entre l'algorithme de Lucas et Kanade (1981) et la version FSA du *Block Matching* calculé de manière dense pour les besoins de la comparaison. Ce manque de qualité est essentiellement dû à l'absence de contrainte spatiale sur la forme du flux de déplacement.

Une autre faiblesse du *Block Matching*, pouvant être particulièrement handicapante dans notre cas, est le mauvais comportement de l'algorithme dans le cas de textures répétitives. Ce problème

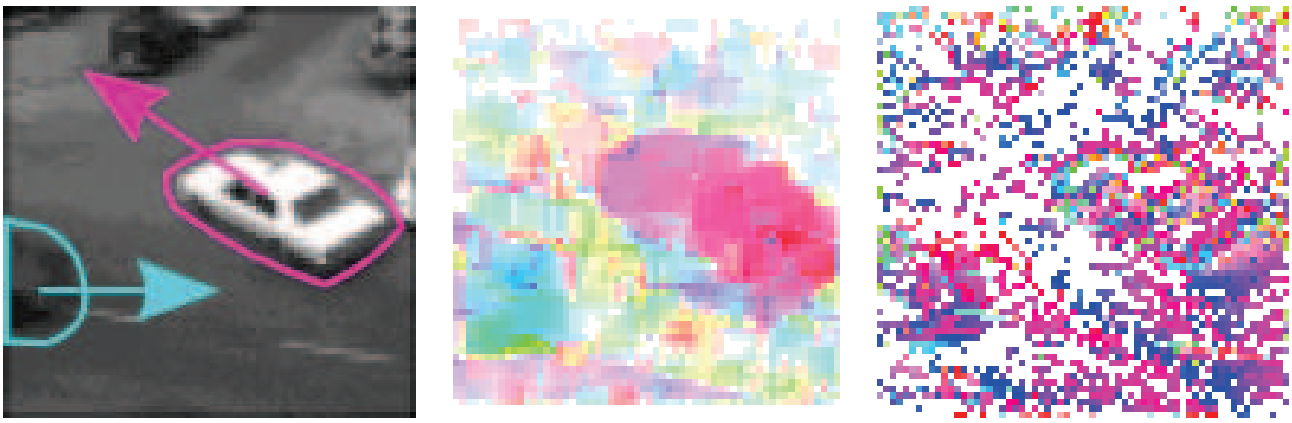


FIGURE 2.4 – Comparaison entre la méthode différentielle de Lucas et Kanade (1981) au centre et la méthode de *Block Matching* « naïve » à droite.

est assez intimement lié au précédent puisque là encore, c'est l'absence de cohérence spatiale qui est en faute. Les autres méthodes de calcul de flot optique peuvent aussi être mises à mal par ce genre de texture mais de manière moins importante en général. Ce problème est d'autant plus gênant dans notre cas que des vidéos de foule ou de trafic dense sont assez sujettes à ce genre de répétition de texture.

2.2.1.3 Les méthodes par régression

Ces méthodes supposent un modèle pour le champ de déplacement et cherchent à en déterminer les paramètres. C'est le cas par exemple pour la méthode de Mémin et Pérez (2002) qui s'appuie sur des approches énergétique pour estimer ou segmenter le mouvement d'une scène. Black et Anandan (1996) supposent, quant à eux, que le le champ de déplacement est affine par morceaux. Dans ce dernier cas, en supposant ce modèle, la contrainte de conservation des données est exprimée de manière différentielle. L'équation (2.3) peut être ici réécrite :

$$\nabla \mathbf{I}_{\mathbf{xy}}(\mathbf{x}, t).\mathbf{v}(\mathbf{a}) + I_t(\mathbf{x}, t) = 0 \tag{2.15}$$

avec

$$\mathbf{v}(x, y; \mathbf{a}) = \begin{bmatrix} u(x, y) \\ v(x, y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ a_3 + a_4 x + a_5 y \end{bmatrix} \tag{2.16}$$

Dans leur travaux Black et Anandan (1996) répondent aussi au problème des discontinuités en utilisant des M-estimateurs (Geman et McClure (1987), Lorentzien, ...) lors de la résolution du problème de minimisation (2.15). L'utilisation d'estimateurs robustes permet de considérer les points trop distants à cause d'une discontinuité, comme des *outliers*. De plus dans des scènes classiques, où les déplacements sont constitués de flux laminaires, l'hypothèse de champs de déplacements affines par morceaux est très bien adaptée. Cela permet d'obtenir des champs beaucoup moins bruités qu'avec des méthodes plus classiques.

En pratique les problèmes de minimisation sont résolus par la méthode itérative de surrelaxations successives. Néanmoins, cette méthode est réputée être très sensible à la convexité du problème. Dans des cas défavorables tels que des mouvements de grandes amplitudes, elle peut converger vers un extremum local et non global. Pour éviter ce phénomène, Black et Anandan (1996) rajoutent à leur

estimation une approche hiérarchique de type *coarse-to-fine*, qui permet en créant une pyramide de résolutions de l'image de limiter le phénomène d'extrema locaux. Ce dernier aspect n'est d'ailleurs pas propre à la méthode de Black et Anandan (1996), comme nous allons le voir au paragraphe suivant.

2.2.1.4 Les méthodes hiérarchiques

Ces approches ne sont pas des algorithmes de flot optique à proprement parlé. Ce sont plutôt des extensions pouvant être adaptées aux algorithmes vus précédemment. Elles s'appuient sur des pyramides de résolutions afin de faciliter la recherche de l'optimum. Ces pyramides de résolutions sont obtenues en sous-échantillonnant successivement les images après leur avoir appliqué un filtre passe-bas. Pour un algorithme de flot optique donné, résoudre le problème d'optimisation à une résolution grossière permet de guider la résolution de ce problème à une résolution plus précise et ainsi de suite. La plupart des algorithmes de flot optique ont une adaptation hiérarchique. C'est le cas de l'algorithme de Lucas et Kanade (1981) ou du *Block Matching* avec l'algorithme de recherche de TSS proposé par Nam *et al.* (1995). Ces approches fournissent en principe de meilleurs résultats en terme de précision que leurs homologues non hiérarchiques ; bien que plus coûteuses, elles restent viables pour des applications temps-réels.

2.2.2 Notre choix

Comme énoncé dans la partie 1.4.3 du chapitre 1, le choix d'opter pour l'analyse de primitive globale, telle que le mouvement, permet une analyse plus stable sur le long terme mais ces primitives sont particulièrement sensibles au bruit ou aux erreurs d'estimation affaiblissant très souvent la qualité de l'analyse. Parmi ces erreurs, la plus critique est celle liée au problème d'ouverture.

2.2.2.1 Le problème d'ouverture

Ce problème bien connu dans le cadre de l'estimation de mouvement provient de l'analyse locale. L'estimation du déplacement peut dans certains cas ne pas être complète. Une manière très simple de mettre en évidence ce phénomène, est de considérer un carré sombre se déplaçant sur un fond blanc (cf. figure 2.5). Prenons par exemple un mouvement réel vers le bas et la gauche, en ne regardant que localement le déplacement du carré au niveau du bord gauche, on ne peut qu'observer un mouvement vers la gauche. La composante tangentielle à l'orientation du bord n'est pas évaluable. Une solution naturelle pour pallier ce problème serait d'agrandir la zone d'analyse, afin d'obtenir un autre bord orienté différemment et qui permettrait de lever l'ambiguïté. En pratique, cette solution n'est pas viable car l'agrandissement de la zone d'analyse risque d'englober des mouvements différents issus d'objet différents. Ce paradoxe appelé dans la littérature, problème d'ouverture généralisé, est plus ou moins bien résolu selon la méthode utilisée.

L'un des traitements possibles pour diminuer cet effet, est de corréler le champ de déplacement avec un score de Harris pour ne conserver l'information de mouvement qu'au niveau des zones de hautes fréquences. Pour donner un exemple concret, ce problème intervient souvent au niveau des ombres projetées. Dans ce cas, le long d'un contour rectiligne de l'ombre, le mouvement estimé est orthogonal à ce contour, ce qui peut être complètement erroné comme on peut le voir à la figure 2.6 où l'ombre posant problème est entourée en jaune sur l'image représentant le champ de déplacement. Le véhicule se dirige vers la gauche (représenté par couleur rouge selon la légende de la figure 1 présenté

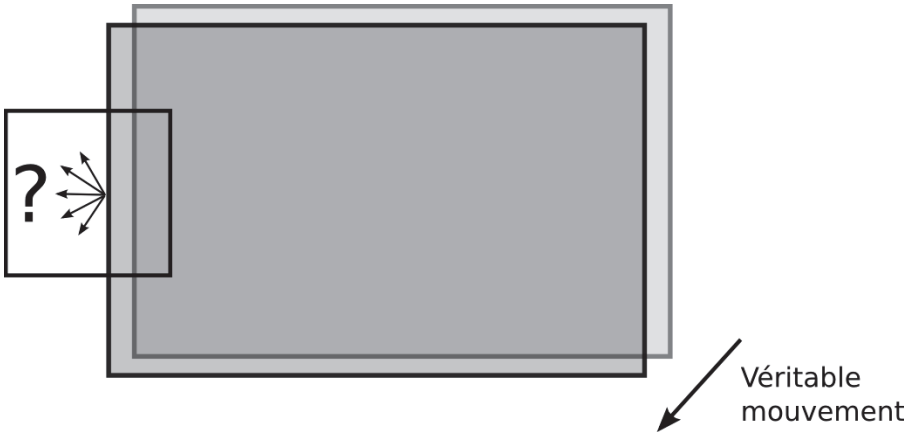


FIGURE 2.5 – Illustration du problème d'ouverture. En ne regardant que localement un contour rectiligne, seule la composante du mouvement normale à ce contour peut être déterminée.

dans les notations au début de ce manuscrit) cependant les contours de l'ombre projetée à l'arrière étant majoritairement horizontaux, le déplacement estimé sur cette ombre est vers le haut (couleur violette). Nous reviendrons sur ce traitement dans le chapitre 5.

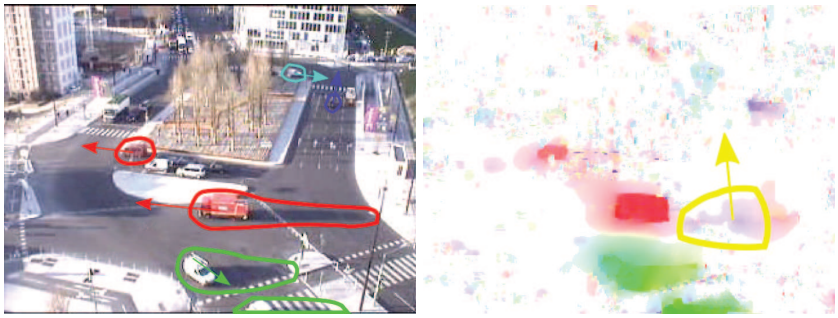


FIGURE 2.6 – Exemple concret du problème d'ouverture au niveau d'une ombre. A gauche, la scène réelle et à droite le problème d'ouverture mis en évidence sur l'ombre du camion.

2.2.2.2 Comparaison qualitative

La méthode de Black et Anandan (1996) a été retenue pour la qualité de son champ de déplacement. Trois autres méthodes lui ont été comparées : la méthode de Lucas et Kanade (1981) dans ses versions simple et hiérarchique ainsi que la méthode de Horn et Schunck (1981). La figure 2.7 présente les résultats des différentes méthodes sur la séquence de Yosemite (paysage avec des mouvements indépendants), une séquence de passage de personne et une de trafic routier. Comme on peut le constater, la méthode de Black et Anandan (1996) est moins bruitée que les autres méthodes. L'hypothèse d'affinité par morceaux permet d'obtenir des champs plus lisses et cohérents spatialement et ainsi de limiter en partie l'effet du problème d'ouverture. Enfin l'utilisation d'estimateurs robustes pour la résolution du problème d'optimisation favorise la précision angulaire.

En ce qui concerne les temps de calcul des différents algorithmes, aucune comparaison n'a été réalisée en raison de la différence de niveau d'optimisation des implémentations utilisées. Cependant, on peut noter que la complexité calculatoire de la méthode de Black et Anandan (1996) reste raisonnable et viable pour des applications temps-réels.

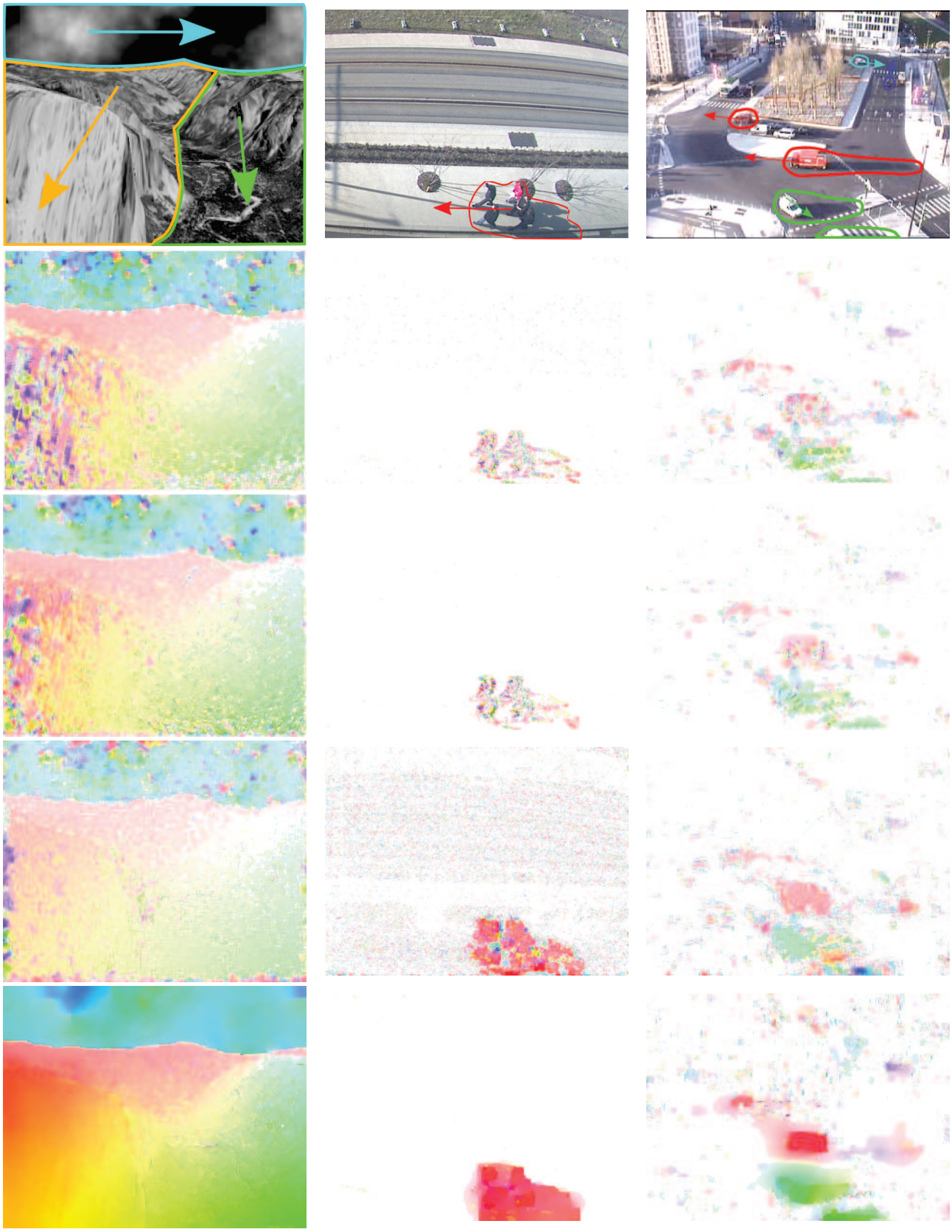


FIGURE 2.7 – Résultats obtenus avec différents algorithmes d'estimation de flot optique. La première ligne représente les images réelles ainsi que les vrais déplacements, la deuxième ligne montre les résultats obtenus avec l'algorithme de Lucas et Kanade (1981), la troisième ceux de Horn et Schunck (1981), la quatrième ceux de Lucas et Kanade (1981) version hiérarchique et enfin la dernière ligne représente les résultats obtenus avec la méthode de Black et Anandan (1996).

2.2.2.3 Propriétés en termes de descripteur

L'utilisation de la sortie d'un algorithme de flot optique comme descripteur de mouvement d'un objet est possible moyennant certaines précautions.

Il faut tout d'abord noter que contrairement à l'orientation, l'amplitude du mouvement est, dans bien des cas assez peu significative. D'une part, l'estimation précise de la vitesse via ces méthodes est difficile au vu de l'hypothèse d'illumination seule et d'autre part, l'amplitude du mouvement est très fortement influencée par la géométrie de la scène et l'angle de vue de la caméra. Les résultats des travaux sur les flots optiques traduisent d'ailleurs bien cette observation, puisque de manière générale la qualité des méthodes est évaluée en terme d'erreur angulaire seulement.

Sous forme de coordonnées cartésiennes les distances classiques de type distance euclidienne ou distance de Mahalanobis sont exploitables, cependant ces coordonnées sont à prendre avec précaution car elles codent aussi l'information de norme du déplacement qui comme mentionné ci-dessus, n'apporte pas systématiquement une information fiable.

La forme polaire de l'estimation du mouvement est donc préférable car elle représente directement l'orientation du mouvement, qui est l'information la plus utile. Il suffit simplement de s'assurer que l'orientation du déplacement d'un pixel correspond à un véritable mouvement et non à du bruit. Ceci est réalisé en seuillant la norme du vecteur déplacement. En revanche, la discontinuité angulaire empêche toute utilisation des distances classiques. Il est possible de définir une distance particulière :

$$d_\theta(\theta_1, \theta_2) = min(\ |\theta_2 - \theta_1|, |\theta_2 - (\theta_1 + 2\pi)|\ )\ avec\ \theta_1 < \theta_2 \tag{2.17}$$

Il est aussi possible de travailler avec une discrétisation du cercle trigonométrique en 4 ou 8 secteurs comme cela est fait dans de nombreux travaux Varadarajan et Odobez (2009) Wang *et al.* (2009) Küttel *et al.* (2010) Hospedales *et al.* (2009) présentés dans le chapitre 1. Ce type de représentation permet une modélisation, certes grossière, mais bien souvent suffisante en vue de l'utilisation de certains algorithmes d'apprentissage, tel que le LDA ou les HMMs comme nous le verrons au chapitre 4.

Enfin, pour son intégration dans l'architecture de l'application telle que nous l'avons conçue, il est nécessaire d'avoir un descripteur par bloc. Pour cela, le déplacement prépondérant d'un bloc est déterminé. Nous verrons dans le chapitre 3 la manière de faire cela.

2.3 Les descripteurs spatio-temporels

Au regard de notre application, il est important d'avoir une caractérisation du mouvement la plus continue possible au fil du temps. Le flot optique en estimant le mouvement sur seulement deux images est assez sujet aux perturbations ponctuelles. Pour filtrer ces perturbations, nous nous sommes intéressés à des caractérisations de mouvement prenant plus en compte son aspect temporel, à savoir les descripteurs spatio-temporels. Ces structures spatio-temporelles peuvent permettre, soit de filtrer le mouvement et ainsi d'être moins sujet au bruit soit de caractériser des mouvements plus complexes permettant ainsi une classification plus fine du mouvement.

2.3.1 Etat de l'art

Parmi les différentes utilisations de ces structures, on peut citer les travaux de Kratz et Nishino (2009) qui considèrent comme support spatio-temporel les gradients calculés sur des cuboïdes de

niveaux de gris, c'est-à-dire une sous-zone de l'image cumulée sur quelques images consécutives. Ce support n'étant pas utilisable en l'état car trop volumineux, il est modélisé par une gaussienne 3D. Soit $\mathbf{\nabla I_i}$ le vecteur des gradients en x,y et t pour le pixel i d'un cuboïde de N pixels :

$$\mathbf{\nabla I_i} = [I_{i,x} I_{i,y} I_{i,t}] = \left[\frac{\partial I(i)}{\partial x} \frac{\partial I(i)}{\partial y} \frac{\partial I(i)}{\partial t}\right] \tag{2.18}$$

Les paramètres $\boldsymbol{\mu}$ et Σ de la gaussienne 3D, $\mathcal{N}$, sont déterminés par :

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i}^{N} \mathbf{\nabla I_i} \tag{2.19}$$

$$\Sigma = \frac{1}{N} \sum_{i}^{N} (\mathbf{\nabla I_i} - \boldsymbol{\mu})^T (\mathbf{\nabla I_i} - \boldsymbol{\mu}) \tag{2.20}$$

L'écart utilisé avec ce descripteur, n'est pas une distance à proprement parler. Il s'agit de la divergence symétrisée de Kullback-Leibler qui permet traditionnellement de mesurer la dissimilarité entre deux distributions de probabilités. Cette mesure s'exprime par la formule :

$$d_{KL}(f,g) = \int_{-\infty}^{\infty} f \log \frac{f}{g} + \int_{-\infty}^{\infty} g \log \frac{g}{f} \tag{2.21}$$

Ce qui dans le cas de deux distributions gaussiennes 3D, $f = \mathcal{N}(\boldsymbol{\mu}_f, \Sigma_f)$ et $g = \mathcal{N}(\boldsymbol{\mu}_g, \Sigma_g)$, se ré-exprime sous la forme :

$$d_{KL}(f,g) = \frac{1}{2} tr \left\{ (\Sigma_f^{-1} + \Sigma_g^{-1})(\boldsymbol{\mu}_f - \boldsymbol{\mu}_g)^T (\boldsymbol{\mu}_f - \boldsymbol{\mu}_g) + \Sigma_f \Sigma_g^{-1} + \Sigma_g \Sigma_f^{-1} - 2I \right\} \tag{2.22}$$

avec I la matrice identité (Moreno *et al.* (2004)).

Une telle modélisation permet selon Kratz et Nishino (2009) de ne pas se limiter aux mouvements uniformes. Dans le cas de mouvement plus complexe, la gaussienne 3D permet de mesurer la variation de déplacement au travers de la matrice de covariance.

D'autres travaux notables concernant les descripteurs spatio-temporels sont ceux de Shechtman et Irani (2005). En partant du même support spatio-temporel que Kratz et Nishino (2009), à savoir les gradients $\mathbf{\nabla I_i}$ d'un cuboïde de niveau de gris, l'approche est de déterminer un critère de cohérence caractérisant si le mouvement apparent d'un cuboïde peut être assimilé à un mouvement uniforme. Pour déterminer ce critère, la contrainte de gradient définie à l'équation (2.3) doit être généralisée non plus entre deux images mais sur l'ensemble des T images du cuboïde. En supposant un mouvement uniforme sur l'ensemble du cuboïde, on peut voir ce cuboïde comme un ensemble de droites de niveau de gris constant, parallèle entre elles et parcourant le cuboïde. Il est donc possible de trouver un vecteur directeur $[u\ v\ w]^T$ de ces droites d'intensité qui sera orthogonal à l'ensemble des gradients $\mathbf{\nabla I_i}$. Le schéma de la figure 2.8 synthétise cette approche. En notant G la matrice regroupant les gradients $\mathbf{\nabla I_i}$ des N pixels du cuboïde :

$$G = \begin{bmatrix} \mathbf{\nabla I_1} \\ \dots \\ \mathbf{\nabla I_N} \end{bmatrix} \tag{2.23}$$

on obtient :

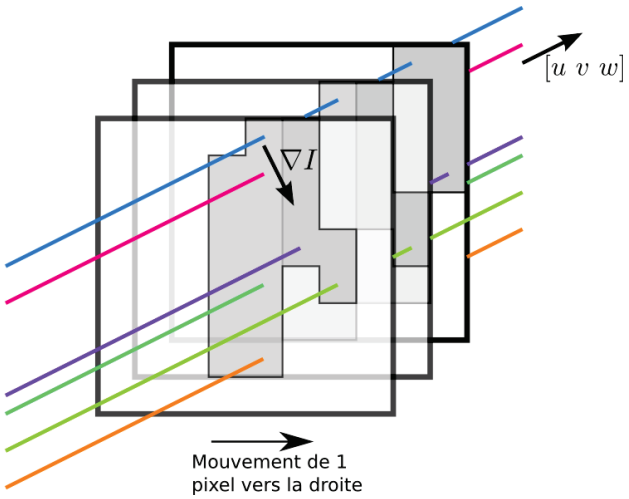


FIGURE 2.8 – Structures spatio-temporelles dans le cas d'un mouvement uniforme. Les niveaux de gris constants forment des droites parallèles entre elles.

$$G \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ \dots \\ 0 \end{bmatrix} \tag{2.24}$$

Le problème peut être ré-exprimé à l'aide des matrices de Gram, l'équation (2.24) peut alors se réécrire :

$$G^T G \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.25}$$

Notons M la matrice de Gram $G^T G$ associée :

$$M = G^T G = \begin{bmatrix} \sum_i I_{i,x}^2 & \sum_i I_{i,x} I_{i,y} & \sum_i I_{i,x} I_{i,t} \\ \sum_i I_{i,y} I_{i,x} & \sum_i I_{i,y}^2 & \sum_i I_{i,y} I_{i,t} \\ \sum_i I_{i,t} I_{i,x} & \sum_i I_{i,t} I_{i,y} & \sum_i I_{i,t}^2 \end{bmatrix} \tag{2.26}$$

Par analogie avec le détecteur de coin de Harris et Stephens (1988), on peut considérer la matrice M comme une généralisation de la matrice de Harris, dont l'expression est :

$$M^\diamond = \begin{bmatrix} \sum_i I_{i,x}^2 & \sum_i I_{i,x} I_{i,y} \\ \sum_i I_{i,y} I_{i,x} & \sum_i I_{i,y}^2 \end{bmatrix} \tag{2.27}$$

La matrice M contient donc toutes les informations nécessaires à la détection de « coins spatio-temporels ».

A noter que résoudre l'équation (2.24) ou (2.25) qui lui est équivalente, n'est possible que si la matrice G (ou M qui à le même rang) est de rang non plein $rg(G) = rg(M) \neq 3$. Dans le cas contraire, le mouvement n'est pas uniforme, ce qui correspond au cas d'un « coin spatio-temporel » des lignes d'intensité. C'est sur ce constat que les travaux de Shechtman et Irani (2005) se fondent, puisqu'une étude de l'augmentation du rang entre la matrice extraite $M^\diamond$ définie à l'équation (2.27) et la matrice M est effectuée. Si la dimension temporelle contribue à l'augmentation du rang de la matrice M, cela signifie que le cuboïde comporte plusieurs mouvements.

La comparaison de deux cuboïdes s'effectue en concaténant les deux cuboïdes selon la dimension temporelle et en étudiant l'uniformité du mouvement dans ce nouveau cuboïde concaténé grâce au même critère que précédemment. De plus, en notant :

$$G_{12} = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \tag{2.28}$$

on obtient alors $M_{12} = G_{12}^T G_{12} = G_1^T G_1 + G_2^T G_2 = M_1 + M_2$, ce qui permet de modéliser la concaténation de deux cuboïdes en sommant leur matrice de Gram.

Cependant, ce critère d'augmentation de rang est binaire et n'est donc pas utilisable en pratique en raison du bruit. Pour pallier ce problème, une mesure continue d'augmentation du rang Δr est proposée :

$$\Delta r = \frac{\lambda_2.\lambda_3}{\lambda_1^\diamond.\lambda_2^\diamond} = \frac{\det(M)}{\det(M^\diamond).\lambda_1} \tag{2.29}$$

avec $\lambda_1 \geq \lambda_2 \geq \lambda_3$ les 3 valeurs propres de la matrice M et $\lambda_1^\diamond \geq \lambda_2^\diamond$ les deux valeurs propres de la matrice $M^\diamond$. Dans un souci d'efficacité de calcul, cette mesure est approchée par la mesure :

$$\Delta\hat{r} = \frac{\det(M)}{\det(M^\diamond).\|M\|_F} \tag{2.30}$$

Enfin, dans le cas de la comparaison de deux cuboïdes, Shechtman et Irani (2005) propose une mesure d'inconsistance entre cuboïdes définie par :

$$m_{12} = \frac{\Delta\hat{r}_{12}}{min(\Delta\hat{r}_1, \Delta\hat{r}_2) + \epsilon} \tag{2.31}$$

Cette mesure permet d'avoir un comportement plus proche d'une distance que $\Delta\hat{r}$. En effet, la comparaison d'un descripteur avec lui-même donne $\Delta\hat{r_{ii}} = \Delta\hat{r_1}$ qui n'est pas nécessairement nul. En revanche, $m_{ii} = 0$, ce qui est un comportement plus conventionnel pour une mesure comparative.

Ces structures spatio-temporelles permettent une modélisation plus élaborée du mouvement, en encodant la variabilité du mouvement dans le cas du descripteur de Kratz et Nishino (2009) ou des mouvements plus lisses dans le cas du descripteur de Shechtman et Irani (2005). Cependant, afin d'obtenir les meilleures performances de classification possibles, nous avons développé notre propre descripteur s'appuyant initialement sur les mêmes hypothèses que le descripteur de Shechtman et Irani (2005). Nous proposons dans le paragraphe suivant de décrire ce descripteur et de le comparer aux deux descripteurs présentés ici.

2.3.2 Notre descripteur

Cette section a pour but de présenter un nouveau descripteur spatio-temporel rapide à calculer et permettant de caractériser le mouvement de manière à faciliter l'étape ultérieure de classification. Pour cela, nous proposons une caractérisation du mouvement en évaluant la linéarité de la relation qui peut exister entre les gradients spatiaux et les gradients temporels par le biais d'un calcul de corrélation.

2.3.2.1 Théorie initiale

En reprenant les hypothèses initiales de Shechtman et Irani (2005), nous étudions la corrélation qui existe entre les gradients spatiaux et les gradients temporels au lieu d'étudier le rang de la matrice M. Nous avons vu précédemment que dans le cas d'un mouvement uniforme au sein d'un cuboïde, il existe un vecteur non nul solution de l'équation (2.24). Une telle solution suppose que la matrice G est de rang non plein et plus particulièrement que la dimension temporelle n'intervient pas dans l'augmentation du rang.

Une autre manière de formuler ce constat est de dire que les gradients temporels sont des combinaisons linéaires des gradients en x et en y. La mesure de cette relation de linéarité entre les gradients qui sera utilisé ici, se ramène à un calcul de la corrélation. La corrélation normalisée de deux variables numériques $X = (x_1, ..., y_n)$ et $Y = (y_1, ..., y_n)$ est donnée par la formule de Pearson :

$$\begin{aligned} \rho_{XY} &= \frac{E[(X - E(X))(Y - E(Y))]}{\sigma_X \sigma_Y} \\ &= \frac{E[(X - E(X))(Y - E(Y))]}{\sqrt{E[(X - E(X))^2]}\sqrt{E[(Y - E(Y))^2]}} \\ \rho_{XY} &= \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \end{aligned} \tag{2.32}$$

Avec $E(.)$, l'espérance mathématique. Il existe d'autres techniques pour déterminer les dépendances entre deux variables numériques, telle que l'information mutuelle par exemple. Cependant ces techniques plus complexes visent à rechercher des dépendances quelconques non nécessairement monotones. Or d'après la nature du problème à résoudre, à savoir l'existence d'une dépendance linéaire entre les gradients spatiaux et les gradients temporels, l'utilisation de la corrélation de Pearson est la mesure de dépendance qui convient.

A noter aussi que pour que le score de corrélation soit valide, il est nécessaire que les variances des variables numériques ne soient pas nulles. Dans notre cadre d'utilisation, le bruit naturel de l'image suffit généralement à l'assurer. Deux cas peuvent poser problème, le cas des régions avec un gradient constant correspondant à des zones de dégradés linéaires parfaits, ce qui sur des images réelles, n'est pas très fréquent et le cas des régions uniformes donc de gradients nuls. Ce dernier type de régions n'apportant aucune information concernant le déplacement, elles peuvent être filtrées préalablement par seuillage sur l'amplitude des gradients.

En calculant la corrélation entre les gradients en x et t d'une part et y et t d'autre part, on obtient un vecteur $\mathbf{C} = [\rho_{xt}\ \rho_{yt}]^T$ caractéristique du mouvement. La comparaison de deux cuboïdes modélisés par notre descripteur, est réalisée grâce à la distance :

$$d_{corr}(\mathbf{C_1}, \mathbf{C_2}) = 1 - \frac{\mathbf{C_1}.\mathbf{C_2}}{\|\mathbf{C_1}\|\|\mathbf{C_2}\|} \tag{2.33}$$

qui est à valeur dans $[0, 2]$ puisque le terme $\frac{\mathbf{C_1}.\mathbf{C_2}}{\|\mathbf{C_1}\|\|\mathbf{C_2}\|}$ correspondant au cosinus entre le vecteur C_1 et C_2 est compris dans l'intervalle $[-1, 1]$.

2.3.2.2 Propriétés

Nous parlerons de caractérisation du mouvement plutôt que d'estimation car notre approche vise essentiellement à différencier un déplacement unitaire d'un autre.

En effet, l'amplitude du mouvement n'est pas déterminée et le vecteur $\mathbf{C}$ ne représente qu'une mesure de confiance sur l'existence d'une dépendance linéaire entre le gradient x et t d'une part, et y et t d'autre part. A titre illustratif, la figure 2.9 montre la relation de linéarité qui existe entre les gradients spatiaux et le gradient temporel lors d'un mouvement vers la droite. Pour cet exemple, une forme sombre sur fond blanc est déplacée vers la droite. Les gradients sont calculés par différences finies en convoluant les images par un filtre $[-1\ 0\ 1]$ selon x, y et t. On constate que c'est au niveau des bordures qu'il est possible de déterminer une dépendance entre les gradients spatiaux et temporels. Lors d'un mouvement à droite, la corrélation entre x et t est égale à -1 car lorsque le gradient $G_x < 0$, $G_t > 0$ et inversement. La corrélation entre y et t est quant à elle nulle, car lorsque $G_y > 0$ (en bas), G_t est à la fois positif et négatif tout comme lorsque $G_y < 0$ (en haut).

Cette mesure de dépendance ne tient pas compte de l'amplitude des gradients puisque le calcul du score de corrélation exploite des données normalisées. Ainsi les distinctions entre les mouvements diagonaux dans le même quartile ne peuvent, en théorie, plus être effectuées. En effet, si l'on calcule le descripteur associé à un mouvement de translation orienté selon le vecteur $(2, 1)$ et $(1, 2)$, dans les deux cas une corrélation entre x et t d'une part et y et t d'autre part existe, donc dans les deux cas le vecteur $\mathbf{C}$ associé serait $\mathbf{C} = [1\ 1]^T$. En pratique, sur des images réelles, les corrélations ne sont jamais parfaites. Plus un mouvement sera franc dans une direction plus la corrélation sera importante. De ce fait, les descripteurs de deux mouvements diagonaux différents issus du même quartile ne donneront pas les mêmes vecteurs $\mathbf{C}$.

En revanche, la norme du vecteur $\mathbf{C}$ donne une information sur la confiance à avoir dans cette caractérisation du mouvement. Nous avons vu que les composantes du vecteur $\mathbf{C}$ sont une manière de quantifier l'existence d'un mouvement selon x ou y. Dans le cas où le vecteur $\mathbf{C}$ possède une norme trop faible, on peut considérer qu'aucun mouvement dominant n'existe au sein du cuboïde. Cette information est comparable au critère défini par Shechtman et Irani (2005) pour vérifier si un mouvement est élémentaire ou non.

De plus, normaliser les données rend le descripteur invariant aux changements affines de luminosité de type $\widehat{I} = aI + b$ sur l'ensemble des niveaux de gris du cuboïde. Ce cas de figure revient à considérer la matrice $\widehat{G} = aG$ qui possède la même relation de linéarité que la matrice G, et pour lequel $C_{\widehat{G}} = C_G$. A noter que cette propriété importante est en fait générale à toutes les méthodes s'appuyant sur la contrainte d'illumination de l'équation (2.3) qui s'exprime dans le cas d'un changement affine sous la forme :

$$a\boldsymbol{\nabla}\mathbf{I}_{\mathbf{xy}}(\mathbf{x}, t).\mathbf{v} + aI_t(\mathbf{x}, t) = 0 \tag{2.34}$$

laissant la solution $\mathbf{v}$ de l'équation identique. C'est précisément le cas du flot optique de Black et Anandan (1996), du descripteur de Shechtman et Irani (2005) ainsi que de notre descripteur. En revanche, le descripteur de Kratz et Nishino (2009) n'est pas robuste à ce genre de changement. En effet, un changement affine pour ce descripteur remplace la gaussienne 3D $\mathcal{N}(\mu, \Sigma)$ par la gaussienne $\mathcal{N}(a\mu, a^2\Sigma)$ qui sont deux distributions bien différentes selon la divergence de Kullback-Leibler définie à l'équation (2.22).

Cette invariance permet de comparer des descripteurs issus d'image prise de jour et de nuit par exemple, ce qui est particulièrement intéressant dans le cadre de notre application pour des observations en environnement extérieur.

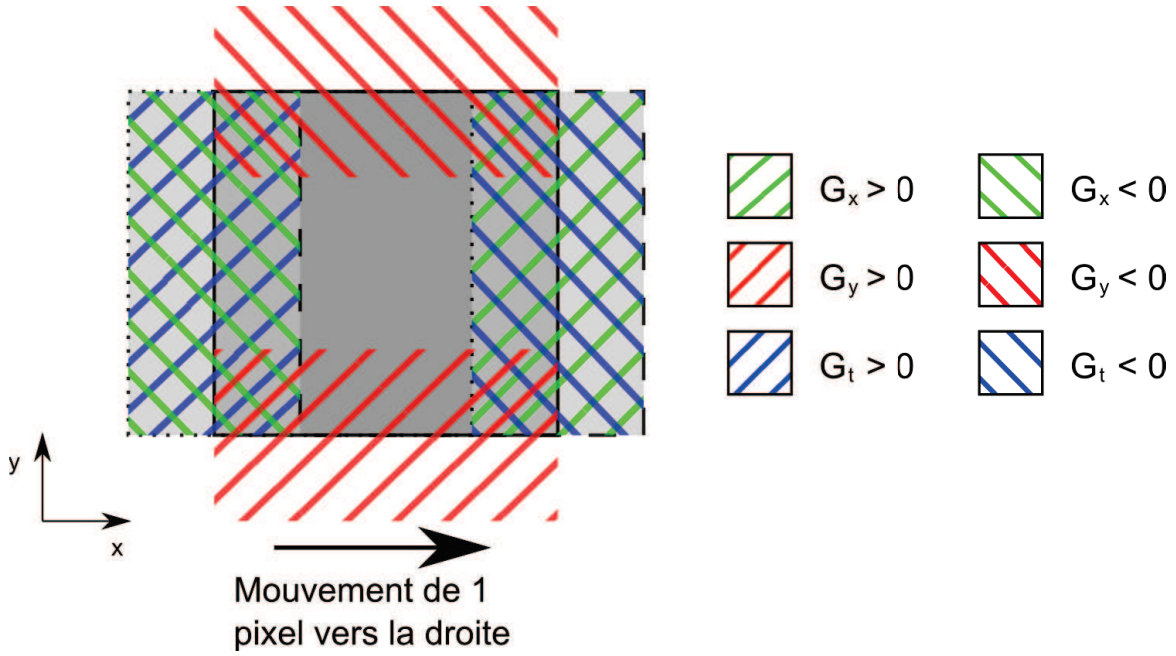


FIGURE 2.9 – Illustration de la notion de relation linéaire entre le gradient spatial en x et le gradient temporel t lors d'un mouvement vers la droite. Les zones de gradients temporels non nuls correspondent aux zones de gradients spatiaux en x non nuls. Les signes des gradients sont opposés.

2.3.2.3 Vers un descripteur robuste à la forme

Cependant, cette approche souffre comme les algorithmes de flot optique du problème d'ouverture. Selon la géométrie de la forme contenue dans le cuboïde, la caractérisation du mouvement peut être biaisée.

Comme vu à la section 2.2.2.1, le problème d'ouverture apparait au niveau des contours rectilignes. Pour déterminer complètement un mouvement, il faut dans l'idéal des contours orthogonaux au sein de la fenêtre d'analyse. C'est en pratique ce qui arrive au niveau d'un coin. Ce problème est théoriquement indépendant de l'orientation des contours.

Cependant en pratique, lorsque l'on calcule les gradients, une base est implicitement fixée, à savoir celle de l'image. Dans cette base, il est possible de préciser ce problème et d'en proposer une solution. Un gradient aligné selon l'axe x ne peut fournir d'information que sur la composante en x du mouvement, de même pour y. Pour les autres gradients, les gradients diagonaux, leur prise en compte peut biaiser notre caractérisation du mouvement. En effet, théoriquement la relation de « corrélation positive » n'est pas transitive. Cependant sous certaines conditions, elle peut le devenir. Cette notion de transitivité numérique a été discutée par Langford *et al.* (2001) qui montrent que pour trois variables aléatoires A, B et C tel que $\rho_{AB} > 0$ et $\rho_{BC} > 0$ alors la corrélation entre A et C est comprise dans l'intervalle :

$$\rho_{AB}\rho_{BC} - \sqrt{(1-\rho_{AB}^2)(1-\rho_{BC}^2)} \leq \rho_{AC} \leq \rho_{AB}\rho_{BC} + \sqrt{(1-\rho_{AB}^2)(1-\rho_{BC}^2)} \tag{2.35}$$

Dans le cas de bords diagonaux, une corrélation entre les composantes x et y du gradient existe. À cause de cette relation de linéarité, un déplacement fera apparaître une corrélation forte à la fois entre les gradients en x et t mais aussi entre les gradients en y et t. En conséquence, un bord diagonal prédominant au sein du cuboïde aura tendance à influencer la caractérisation du mouvement de manière à

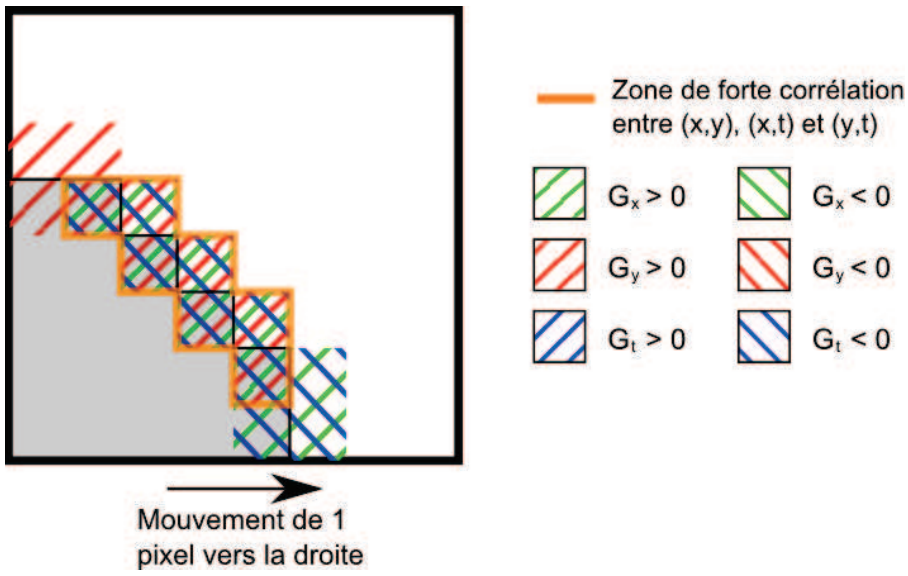


FIGURE 2.10 – Illustration de l'influence de la forme de l'objet sur l'estimation des relations linéaires. La zone en orange qui englobe

avoir $\mathbf{C} = [\alpha\ \beta]$ avec $|\alpha| \approx |\beta|$, les signes de α et β dépendent de l'orientation du bord diagonal, et ce quelque soit le véritable mouvement sous-jacent. Pour illustrer ce phénomène, la figure 2.10 montre que si une forme est majoritairement composée d'un contour diagonal, ce qui signifie que $|\rho_{xy}| \gg 0$, alors même si le mouvement n'est que selon x, une forte corrélation ρ_{yt} sera tout de même mesurée (zone entourée en orange sur la figure).

Pour éviter cela, il est possible de filtrer l'information contenue dans le cuboïde pour ne conserver qu'un sous ensemble permettant une caractérisation fiable. Pour cela, on considère un sous-ensemble de points du cuboïde pour lesquels le gradient spatial est orienté selon l'axe x. Notons respectivement S_x et S_{xt} les ensembles des gradients respectivement $I_{i,x}$ et $I_{i,t}$ en ces points.

$$\begin{aligned} S_x &= \{I_{i,x} | \arg(I_{i,x}, I_{i,y}) = 0[\pi]\} \\ S_{xt} &= \{I_{i,t} | \arg(I_{i,x}, I_{i,y}) = 0[\pi]\} \end{aligned} \tag{2.36}$$

avec $\arg(I_{i,x}, I_{i,y})$ l'argument du gradient défini par le vecteur $(I_{i,x}, I_{i,y})$.

La composante ρ_{xt} du vecteur $\mathbf{C}$ est ensuite déterminée en calculant le coefficient de corrélation entre les ensembles S_x et S_{xt}. De manière analogue, la composante ρ_{yt} du vecteur $\mathbf{C}$ est déterminée par $\rho_{yt} = \rho_{S_y S_{yt}}$, avec

$$\begin{aligned} S_y &= \{I_{i,y} | \arg(I_{i,x}, I_{i,y}) = \frac{\pi}{2}[\pi]\} \\ S_{yt} &= \{I_{i,t} | \arg(I_{i,x}, I_{i,y}) = \frac{\pi}{2}[\pi]\} \end{aligned} \tag{2.37}$$

Un tel prétraitement permet de fiabiliser l'estimation de la direction et donc la distinction des classes mais ne considérer qu'un sous-ensemble des gradients peut engendrer des cas dégénérés. Dans ce genre de cas, un nombre insuffisant de gradients selon l'un des deux axes, voire les deux, peut empêcher de calculer le descripteur. Pour limiter ce phénomène, la contrainte de l'alignement des gradients selon les axes pour conserver des sous ensembles de gradients fiables a été relâchée. Un

intervalle angulaire de $\frac{\pi}{4}$ autour des axes est autorisé pour la définition des sous-ensembles S_x, S_{xt}, S_y et S_{yt}. Ce descripteur amélioré sera dans la suite de ce manuscrit appelé « corrélation sélective disjointe » (CSD). L'algorithme 1 permet de synthétiser le processus final. Ce relâchement de contrainte permet d'augmenter la cardinalité des sous-ensembles S_x, S_{xt}, S_y et S_{yt}, cependant dans certains cas spécifiques le problème peut tout de même subsister. Dans ces cas précis, au lieu d'estimer une direction erronée, il nous a semblé préférable de ne rien évaluer et de rajouter l'information que le mouvement ne peut pas être défini correctement en le plaçant dans un état d'invalidation. Ces cas de figure sont typiques des cuboïdes ne contenant qu'une bordure régulière, qu'elle soit diagonale (S_x et S_y sont vides), verticale (S_y est vide) ou horizontale (S_x est vide).

ENTRÉES: Les gradients du cuboïde G
$S_x = \varnothing$, $S_{xt} = \varnothing$, $S_y = \varnothing$, $S_{yt} = \varnothing$
pour tout $\nabla \mathbf{I_i} \in G$
$n = \|(I_{i,x}, I_{i,y})\|$
$\theta = \arg(I_{i,x}, I_{i,y})$
si $n > \epsilon$
si $-\frac{\pi}{8} \leq \theta[\pi] < \frac{\pi}{8}$
$S_x = S_x \bigcup \{I_{i,x}\}$
$S_{xt} = S_{xt} \bigcup \{I_{i,t}\}$
sinon si $-\frac{\pi}{8} \leq (\theta + \frac{\pi}{2})[\pi] < \frac{\pi}{8}$
$S_y = S_y \bigcup \{I_{i,y}\}$
$S_{yt} = S_{yt} \bigcup \{I_{i,t}\}$
fin si
fin si
fin pour
$\mathbf{C} = \begin{bmatrix} \rho_{S_x S_{xt}} \\ \rho_{S_y S_{yt}} \end{bmatrix}$
retourne Le vecteur de corrélation $\mathbf{C}$

Algorithme 1 : Algorithme de génération du descripteur CSD

2.3.2.4 Validation expérimentale

Séparation des mouvements Afin de valider le descripteur proposé, une évaluation de la distinction entre classes a été réalisée. Des cuboïdes modélisant des mouvements selon 16 directions différentes ont été générés et comparés entre eux (cf. figure 2.11). Les descripteurs sont calculés sur $T = 3$ images. Les tests présentés ci-dessous sont obtenus à partir de données de synthèse ou de mouvement de translation artificiel sur des images réelles. Sur ce genre de séquence l'influence du paramètre T est assez faible. En revanche, en situation réelle, son influence consiste essentiellement à renforcer la caractérisation du mouvement, sous réserve que le mouvement en question reste représentable par le descripteur en question. En effet, dans le cas du descripteur de Shechtman et Irani (2005) ou du descripteur CSD proposé, le cuboïde doit contenir un mouvement unique et élémentaire. C'est précisément le risque de choisir un paramètre T trop grand et d'englober dans un même cuboïde des mouvements disjoints.

Les gradients spatio-temporels ont été calculés par la méthode de Canny permettant ainsi d'avoir un champ de gradient relativement lisse. Le lecteur est invité à se reporter en annexe B pour plus

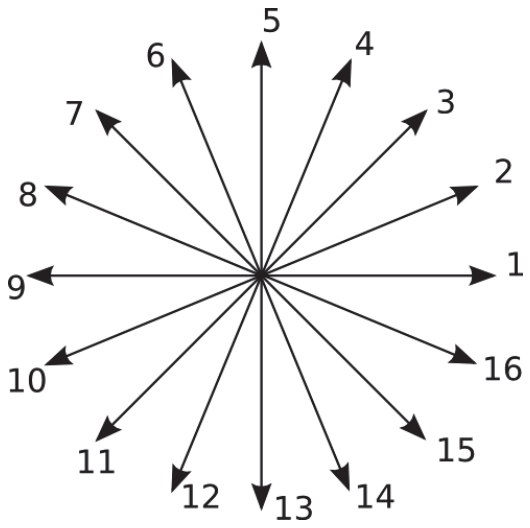


FIGURE 2.11 – Numérotation des mouvements échantillonnés selon 16 directions différentes.

d'information concernant cette méthode de calcul des gradients. Dans notre cas, nous avons choisi une variance égale à 1 pour les filtres gaussiens de Canny, ce qui donne des filtres de taille 5x5.

Les résultats sont représentés sous forme de matrice de distance de taille $(16, 16)$. Le terme général de cette matrice de distance est défini par :

$$M_{ij} = d_D(D_i, D_j) \tag{2.38}$$

avec D_i le descripteur considéré, calculé pour un mouvement dans la direction i et d_D la distance associée. Pour rappel, les distances utilisées pour les descripteurs de Shechtman et Irani (2005), Kratz et Nishino (2009) et celui proposé dans ce chapitre sont respectivement celles définies aux équations (2.31), (2.22) et (2.33).

Cette matrice est symétrique, supposée posséder une diagonale minimale, correspondant à la distance entre un mouvement et lui-même, ainsi qu'une sous-diagonale (et une sur-diagonale symétrique) maximales, correspondant à la distance entre un mouvement et son mouvement opposé. A titre d'exemple, notre descripteur dans sa version initiale sans filtrage des gradients diagonaux et sa version CSD, ont été comparés aux descripteurs de Shechtman et Irani (2005) et de Kratz et Nishino (2009). En déplaçant un carré blanc de taille $(4, 4)$ sur fond noir dans les 16 directions, on obtient les matrices de distance de ces différents descripteurs représentées à la figure 2.12. On constate que la version initiale du descripteur par corrélation donne pour ce cas d'école la matrice de distance optimale. La version CSD donne une matrice par bloc, ce qui illustre bien le phénomène énoncé plus haut sur la non distinction des mouvements diagonaux dans le cas de corrélation parfaite. En effet, les directions 2,3,4 qui correspondent a des mouvements diagonaux dans le même quartile ont des vecteurs de corrélation $\mathbf{C} = [-1 \quad -1]^T$. C'est aussi le cas pour les trois autres quartiles, à savoir les mouvements (6,7,8), (10,11,12) et (14,15,16) (cf. figure 2.11). A noter que l'on observe ce phénomène sur la version CSD et non pas sur la version initiale car pour cette dernière les gradients aux niveaux des coins du carrés permettent d'avoir une réponse graduée de la corrélation plus fidèle au mouvement.

Pour être plus représentatif d'une utilisation réelle, les cuboïdes ont été générés à partir d'images réelles, en différentes régions d'intérêt $r_i \in R$ représentées en rouge sur la figure 2.13. Les boites bleues correspondent aux zones englobantes susceptibles d'être aperçues durant l'ensemble des images nécessaires au calcul du gradient et du descripteur. En chaque région d'intérêt, l'image est

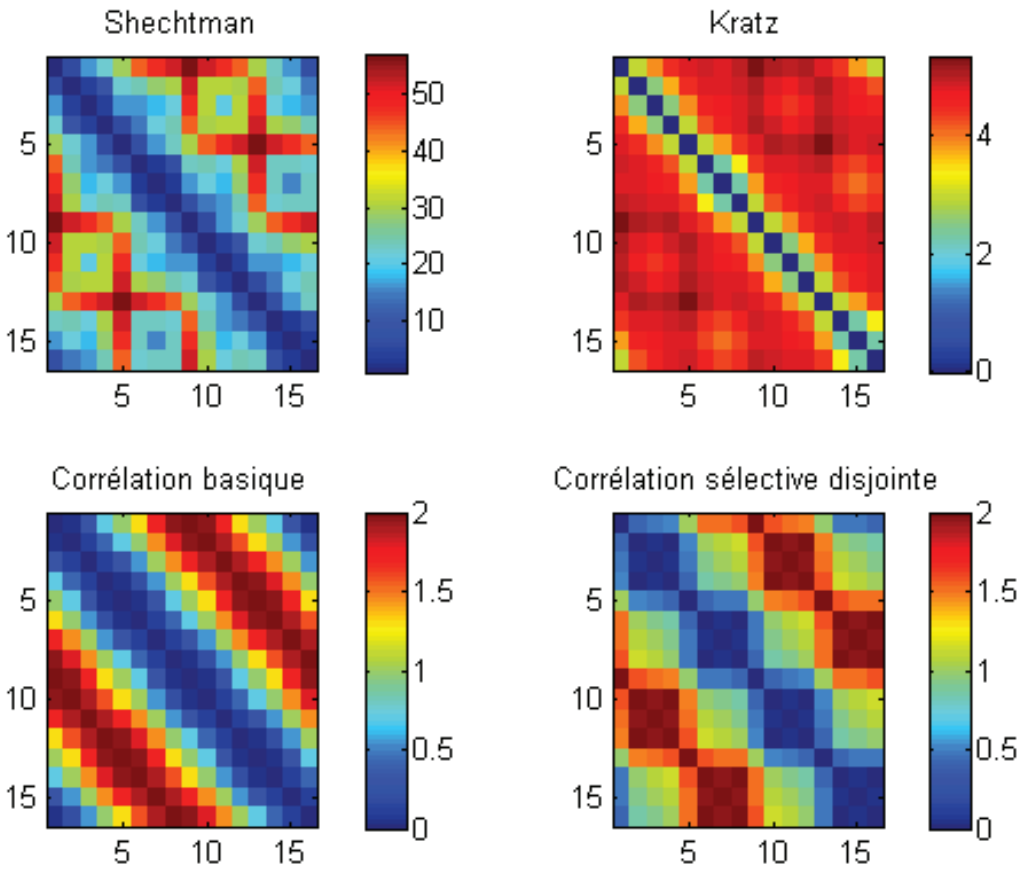


FIGURE 2.12 – Exemple de matrice de distance obtenue pour le déplacement d'un carré blanc sur fond noir selon 16 directions différentes. La matrice obtenue pour la corrélation basique représente la matrice de distance optimale.

FIGURE 2.13 – Régions d'intérêt utilisées pour la comparaison de descripteurs.

	Shechtman	Kratz	Corrélation simple	CSD
$\overline{i_{max}}$	6	8.5	7.625	7.875
$\sigma_{\overline{i_{max}}}$	2.7809	2.3094	0.8851	0.6191

TABLE 2.1 – Décalage moyen et écart type entre deux mouvements extremum.

soumise à un mouvement de translation constant selon une direction particulière.

La caractérisation du mouvement ne devant pas être sujette à la forme contenue dans le cuboïde, les distances entre deux directions i et j sont calculées pour tous les couples d'imagettes puis moyennées. Chaque coefficient des matrices de distance est donc égal à :

$$M_{ij} = \frac{1}{\|R\|^2} \sum_{r_k \in R} \sum_{r_l \in R} d_D(D_i^{(r_k)}, D_j^{(r_l)}) \tag{2.39}$$

avec $D_i^{(r_k)}$ un descripteur calculé à l'emplacement r_k de l'image lors d'un déplacement dans la direction i et $\|R\|$ le cardinal de l'ensemble R.

Sur un ensemble de 75 régions d'intérêt issues des trois images de la figure 2.13, les résultats sont présentés à la figure 2.14. On constate que les trois descripteurs avec leur distance respective permettent dans l'ensemble de distinguer les mouvements entre eux, cependant notre descripteur dans sa version CSD est plus constant et plus précis. On remarque d'une part, que les matrices sont bien toutes à diagonale minimale. Pour chaque direction i, la distance maximale est supposée être atteinte pour le mouvement opposé, à savoir le mouvement d'indice $i + 8$ [16], or en calculant la moyenne de la position ayant la distance maximale $\overline{i_{max}}$, on constate dans le tableau 2.1 que nos méthodes et à plus forte raison la version CSD de notre descripteur par corrélation est plus proche des 8 théoriques que les autres méthodes. De plus, les écarts type des positions de ces maxima est à l'avantage de nos méthodes, ce qui montre la stabilité de notre distinction de mouvement quelque soit la forme contenue dans le cuboïde.

On peut aussi remarquer que dans le cas de notre descripteur initial les différents mouvements se distinguent en deux classes avec pour chaque classe une faible distinction intraclasse. La raison de ce phénomène est la corrélation spatiale susceptible d'exister et qui vient biaiser l'estimation du vecteur $\mathbf{C}$ pour le rendre constant pour les mouvements d'une classe, quelque soit le véritable mouvement. La version CSD diminue cet effet de manière significative.

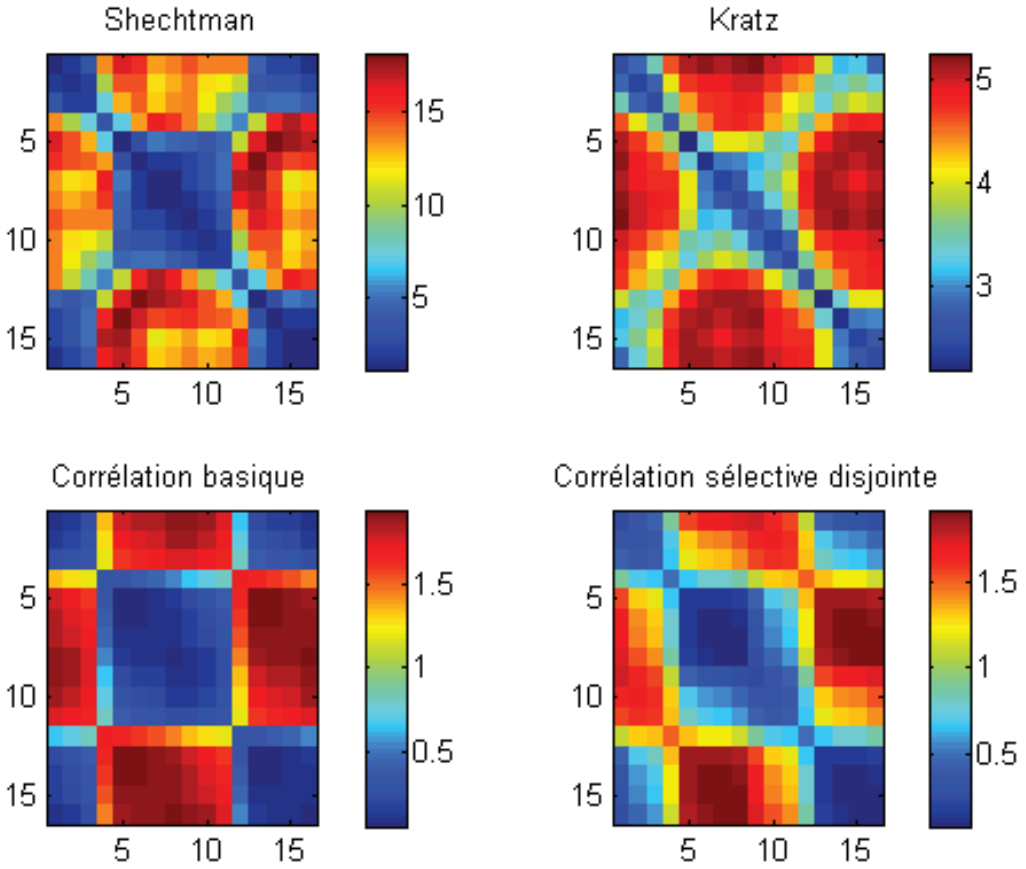


FIGURE 2.14 – Moyenne des matrices de distance entre 16 directions différentes pour tout couple tiré parmi 75 imagettes. Les diagonales minimales de chacune des matrices montrent que les quatre descripteurs identifient correctement deux mouvements dans la même direction. La distinction des autres mouvements est plus délicate pour certains descripteurs comme celui de Shechtman et Irani (2005) qui ne défini pas toujours le mouvement opposé comme le mouvement avec la distance maximale. Le descripteur CSD en revanche arrive assez bien à faire ressortir la sous-diagonale maximale.

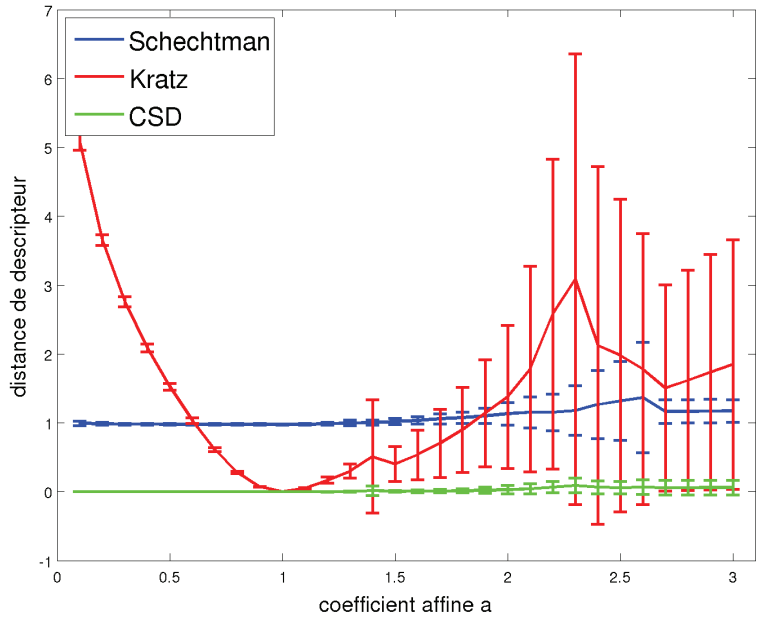


FIGURE 2.15 – Évolution de la distance entre descripteurs face une transformation affine de la luminosité du cuboïde de niveau de gris. La moyenne et la variance des distances calculées sur l'ensemble des 75 imagettes sont représentées pour chaque coefficient de changement affine a dans un intervalle de 0.1 à 3.

Invariance aux changements affines de luminosité Comme énoncé dans la partie 2.3.2.2, notre descripteur tout comme celui de Shechtman et Irani (2005) est invariant au changement affine d'illumination. Pour valider cela, les régions d'intérêt $r_i \in R$ représentées en rouge sur la figure 2.13 ont été réutilisées pour évaluer la distance entre des descripteurs avec et sans changement d'illumination. Ainsi, pour une direction particulière, un descripteur est calculé sur une région d'intérêt originale et un autre est calculé sur la même région d'intérêt ayant subi une transformation affine des niveaux de gris de type $\widehat{I} = aI + b$. Comme les descripteurs sont déterminés à partir de gradient, le terme b peut-être omis. Les différentes courbes de la figure 2.15 représentent l'évolution de la moyenne et de l'écart-type sur l'ensemble des régions d'intérêt de R, des distances entre deux descripteurs en fonction du coefficient a. Les descripteurs caractérisant la même direction, la distance entre eux devrait être minimale. Dans le cas du descripteur de Kratz et Nishino (2009) et du descripteur CSD cette distance minimale est égale à 0 alors que dans le cas de celle de Shechtman et Irani (2005), elle est de 1. La version basique du descripteur n'a pas été représentée car les courbes obtenues étaient quasiment identiques à celles du descripteur CSD.

On constate comme espéré, qu'à l'exception du descripteur de Kratz et Nishino (2009), les autres descripteurs ont une distance minimale et une variance associée très faible, indépendamment du coefficient a du moins jusqu'à certaines valeurs de celui-ci. En effet, pour des coefficients a trop élevés, les niveaux de gris saturent en blanc ce qui dénature les formes contenues dans le cuboïde et met les descripteurs en défaut. En revanche, le descripteur de Kratz et Nishino (2009) pour des coefficients a faible ne fournit pas une distance minimale, comme attendu.

Cette lacune nous a semblé importante par rapport aux autres approches du flot optique ou des autres descripteurs spatio-temporels qui possèdent cette propriété.

Efficacité de calcul Les performances en terme de temps de calcul ont été évaluées pour notre descripteur par corrélation dans sa version améliorée CSD et pour le calcul du flot optique de Black et Anandan (1996). Les implémentations ont été réalisées en C++ avec un code optimisé. Les temps mesurés ici concernent des descripteurs spatio-temporels de taille (16x16x5). Il est important de noter que la nature de l'information de déplacement est très différente entre les descripteurs spatio-temporels et le flot optique. En effet, ce dernier donne en sortie un champ de déplacement dense alors les descripteurs spatio-temporels donnent une information par bloc. Ces deux approches ne sont donc pas comparables entre elles directement en termes de performance. Le nombre de cycle machine du flot optique est plutôt donné ici à titre indicatif.

L'essentiel du temps de calcul de notre descripteur est composé du calcul du gradient comme on peut le voir dans le tableau 2.2. Pour ce qui est de la phase de calcul de la corrélation, elle peut être optimisée en utilisant une autre formulation du coefficient de corrélation de Pearson. En effet, d'après la relation $E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y)$, la corrélation calculée via la formule (2.32) peut être réexprimée :

$$\rho_{XY} = \frac{E[(X - E(X))(Y - E(Y))]}{\sqrt{E[(X - E(X))^2]}\sqrt{E[(Y - E(Y))^2]}} \tag{2.40}$$

$$\rho_{XY} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2}\sqrt{E(Y^2) - E(Y)^2}} \tag{2.41}$$

Et donc finalement :

$$\rho_{XY} = \frac{N\sum_{i=1}^{N}(x_i y_i) - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{N\sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2}\sqrt{N\sum_{i=1}^{N} y_i^2 - (\sum_{i=1}^{N} y_i)^2}} \tag{2.42}$$

Cette dernière formulation permet un calcul de la corrélation en une seule passe améliorant ainsi les performances de calcul. Cependant cette formulation n'est pas stable numériquement. Il faut donc se prémunir de garde-fous pour s'assurer de calculer la corrélation convenablement. Le principal risque de cette formulation apparait lorsque l'une des variances de la variable numérique X ou Y est très faible. Dans ce cas, l'un des termes sous la racine risque d'être numériquement négatif. Nous avons vu que pour la formulation d'origine de la corrélation, les variances des variables numériques devaient être non nulles et donc qu'il fallait pré-filtrer les zones parfaitement uniformes qui engendraient des gradients nuls. Ici la condition est un peu renforcée puisque les zones quasi-uniformes peuvent aussi poser problème. Dans tous les cas le filtrage par seuillage sur l'amplitude des gradients suffit à éviter ces erreurs numériques.

On peut constater dans le tableau 2.2 que la version optimisée du CSD grâce à son unique passage, prend environ deux fois moins de cycles machine que sa version originale. De plus, le descripteur de Shechtman et Irani (2005) a des performances comparables à celle de notre descripteur CSD optimisé, en étant toutefois légèrement moins véloce. Ceci est dû au calcul de la somme du produit des gradients entre x et y qui n'est pas nécessaire dans le cas du CSD. De plus, l'étape de pré-filtrage diminue la taille des ensembles de gradient sur lesquels mesurer la corrélation.

On peut aussi remarquer que les méthodes spatio-temporelles ont un temps de calcul linéaire avec la taille de l'image, puisque le passage d'une résolution de 320x240 à 640x480 quadruple les temps de calcul, ce qui n'est pas le cas du flot optique.

A noter que sous cette forme l'utilisation d'image intégrale (Crow (1984)) peut être avantageuse pour calculer ces descripteurs surtout si la caractérisation du mouvement doit être effectuée sur des

Taille d'image	Calcul de gradients	CSD non optimisée	CSD optimisée	Shechtman	Black & Anandan
320x240	67.92	45.28	19.81	25.47	152.82
640x480	297.15	155.65	76.41	104.71	761.27

TABLE 2.2 – Nombre de cycles machine moyen (en million de cycles) pour le calcul de gradients, du descripteur CSD lorsqu'il est calculé de manière exacte et de manière efficace, le descripteur de Shechtman et Irani (2005) et enfin le flot optique de Black et Anandan (1996)

cuboïdes se chevauchant ou de manière dense sur l'ensemble des pixels de l'image, ou encore pour des approches multi-échelles.

Illustrations qualitatives Comme expliqué tout au long de ce chapitre, le descripteur CSD ne représente pas une estimation du mouvement mais une caractérisation. Cependant, il est possible de représenter le vecteur C pour chaque bloc (ce qui n'est pas possible de manière simple pour les descripteurs de Shechtman et Irani (2005) et Kratz et Nishino (2009)). L'orientation de ce vecteur est en pratique assimilable à l'orientation du mouvement. A titre illustratif, la caractérisation de ce descripteur a été représentée en chaque bloc par une flèche dont la couleur respecte le code couleur d'orientation introduit dans la partie notation de ce manuscrit. Les scènes étudiées pour l'analyse du flot optique ainsi que des scènes de foule ont été utilisées pour illustrer les résultats de notre caractérisation du mouvement, comme illustré à la figure 2.16. On constate que la caractérisation du déplacement obtenue est assez fidèle et cohérente au mouvement sous-jacent. A noter aussi qu'aucune caractérisation n'est obtenue pour certains blocs car l'amplitude des mouvements en ces zones est trop faible, comme c'est le cas au centre de l'image sur la séquence de Yosemite et pour les personnes et les véhicules en haut des images du marathon et du carrefour routier.

2.4 Conclusion

Dans le cadre de nos travaux et des choix d'architecture qui ont été réalisés, ces descripteurs basés sur des structures spatio-temporelles cuboïdes, s'adaptent parfaitement au découpage par bloc qui est réalisé contrairement au flot optique qui nécessite des traitements supplémentaires pour se ramener à une étude par bloc. De plus, le pré-filtrage sur les orientations des gradients spatiaux est équivalent à l'étape de filtrage du champ de déplacement obtenu par le flot optique avec un score de Harris pour diminuer les effets du problème d'ouverture. Les descripteurs spatio-temporels permettent une représentation plus lisse du mouvement qui correspond mieux aux flux laminaires qui sont à l'étude dans ces travaux. Une caractérisation du mouvement plus continue au fil du temps est aussi la garantie de moins de fausses alarmes dues à des *outliers*.

Concernant les différents descripteurs spatio-temporels, notre proposition de descripteur présente des performances surpassant les autres descripteurs en termes de séparation de mouvement. C'est sur cette particularité que nous avons insisté car elle est fondamentale en vue d'une utilisation avec des machines d'apprentissage. C'est d'ailleurs sur certaines de ces machines que les deux prochains chapitres vont porter. Nous avons aussi montré que le descripteur proposé est invariant aux changements affines de luminosité contrairement au descripteur de Kratz et Nishino (2009). Cette propriété est très importante pour notre système puisque cette invariance permet de dispenser aux machines d'apprentissage la tâche supplémentaire de modéliser les changements d'illumination.

FIGURE 2.16 – Exemples de caractérisation du descripteur CSD. Les images représentent dans l'ordre la vue aérienne d'un marathon, un couloir de métro dans une gare japonaise, la séquence de Yosemite, la vue aérienne d'un trottoir et un carrefour routier.

Les performances en terme de classification des deux grandes familles de modélisation du mouvement qui ont été vues dans ce chapitre, à savoir le flot optique et les structures spatio-temporelles seront présentées au chapitre 5, une fois que l'ensemble du cadre expérimental aura été mis en place au travers des chapitres 3 et 4.

Chapitre 3

Estimation de densité de probabilité pour la classification probabiliste

L'objectif de ce chapitre est de proposer un mécanisme d'estimation de densité de probabilité qui sera utilisé à des fins d'apprentissage non supervisé permettant d'exploiter les différents descripteurs présentés au chapitre 2. Cette méthode originale est une variante parcimonieuse des fenêtres de Parzen qui bénéficie de la souplesse de modélisation de ce modèle sans ses contraintes calculatoires. Dans un contexte de classification, une étude de l'écart à l'estimation d'une vraisemblance de mouvement est réalisée. Le critère de décision d'appartenance au modèle s'appuiera sur un calcul de seuil de confiance qui sera largement réutilisé dans la suite de ce manuscrit. Les travaux réalisés sur cette méthode d'estimation ainsi que son utilisation à des fins de classification ont été publiés dans (Luvison *et al.*, 2009c,a,b, 2010, 2011).

3.1 Introduction

Outre la manière de déterminer la caractéristique de l'image à analyser, à savoir le mouvement dans notre cas, l'autre partie importante d'un système de détection réside dans le choix du type de fonction de décision. Comme nous pouvons disposer de bases vidéos décrivant le fonctionnement nominal du système à étudier, nous nous sommes naturellement orientés vers des modèles générés par apprentissage. Néanmoins, ne disposant pas de bases décrivant les comportements anormaux, nous sommes dans un cadre d'apprentissage particulier. Il est en effet impossible de réaliser une base d'évènements anormaux à cause de la rareté des évènements et de leur mauvaise définition. De plus, nous utilisons des descripteurs qui s'appuient sur des distances particulières. Nous ferons donc particulièrement attention à l'adéquation des techniques d'apprentissage mises en œuvre avec ces descripteurs.

Dans le cadre qui nous intéresse, la problématique est donc la suivante : un ensemble de caractéristiques décrivant le fonctionnement « normal » ou « nominal » d'un système est fourni. L'objectif est de définir une fonction de décision capable de classer une nouvelle observation comme normale ou anormale.

Nous nous intéressons ici aux approches probabilistes. Ces approches sont largement utilisées en vision par ordinateur que ce soit pour des applications de reconnaissance, de détection ou de

suivi. Lorsqu'utilisées dans le cadre des apprentissages non supervisés, elles sont toutes basées sur l'estimation d'une vraisemblance suivie d'un seuillage de cette valeur pour décider de la classe de l'échantillon test. La notion de "vraisemblance" est un concept utile pour identifier la distribution réelle ayant engendré un échantillon donné. En maximisant cette vraisemblance, on tend vers une densité de probabilité (appelé densité *a posteriori*) qui approche au mieux la distribution réelle. Savoir modéliser cette vraisemblance et la maximiser est le verrou principal de ces approches bayésiennes. C'est un problème difficile pour différentes raisons : 1) de manière générale, aucun *a priori* sur la forme de la fonction de vraisemblance n'est disponible pour déterminer sa forme paramétrique et 2) les modèles doivent permettre de gérer des grandes dimensions et de nombreuses observations du modèle. L'objectif est donc d'estimer à partir de $\mathcal{Z} \doteq \{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_N}\}$ un ensemble de réalisations d'une variable aléatoire $\mathbf{z}$, la loi de probabilité $p_\mathbf{z}$ associée que nous noterons par la suite simplement p.

Concernant l'étape de seuillage, celle-ci peut être très simple (via un seuil empirique) ou plus subtile, comme avec une approche basée sur un critère de confiance (cf. section 3.3.4).

Après un tour d'horizon des méthodes d'estimation, nous présenterons une nouvelle approche qui s'appuie sur une estimation paramétrique bayésienne parcimonieuse de la vraisemblance, extraite des fenêtres de Parzen. Nous détaillerons aussi un mécanisme de décision associé à cette estimation permettant de déterminer automatiquement un seuil de classification selon la forme de la distribution de probabilité.

3.2 Etat de l'art

Ce problème d'estimation de distribution étant au cœur de très nombreux problèmes en vision par ordinateur et dans d'autres domaines d'ailleurs, beaucoup de travaux ont porté sur le sujet. Seuls les principaux sont présentés ici. Parmi eux, il est possible de distinguer deux catégories, la première utilise des modèles non paramétriques comme le Kernel Density Estimation (KDE ou modèle de fenêtre de Parzen Duda *et al.* (2001)) et les K Plus Proches Voisins (KPPV, Duda *et al.* (2001)) qui mettent en œuvre l'intégralité des observations de l'apprentissage pour estimer une valeur de probabilité. Ces méthodes convergent vers la densité réelle, *a priori* inconnue, lorsque le nombre d'observations tend vers l'infini mais aux dépends d'une évaluation du modèle extrêmement lourde et coûteuse. La complexité calculatoire est en effet linéairement croissante avec le nombre d'échantillons. Pour réduire la complexité de ce modèle, des approches visant à adapter les noyaux existent. Parmi ces approches, on peut citer les travaux de Bugeau et Pérez (2009) qui généralisent l'algorithme du *mean-shift* [1] en adaptant la largeur des noyaux. Cette adadptation est réalisée en recherchant la largeur permettant d'obtenir la distribution plus stable par ajout ou retrait d'observations. Les travaux de Taron *et al.* (2009) s'intéressent aussi à la problématique des noyaux variables en rappelant des techniques hybrides entre les fenêtres de Parzen et les KPPV qui visent à adapter la largeur des noyaux des fenêtres de Parzen selon un critère de « K ième plus proche voisin ».

La seconde catégorie concerne les méthodes paramétriques, qui nécessitent en général d'une part de faire quelques hypothèses, parfois abusives, sur la forme de la densité et d'autre part d'estimer les paramètres de ce modèle. Les modèles de mélange de gaussiennes (MMG) sont largement utilisés et les paramètres associés (moyennes et matrices de covariance) peuvent être estimées avec un algorithme « *Expectation Maximisation* » (EM) Dempster *et al.* (1977). Figueiredo et Jain (2002)

1. un méthode basée noyaux pour la localisation de modes dans une distribution

proposent une version de cet algorithme où le nombre de gaussiennes utilisées n'est plus un *a priori*. Récemment, Han *et al.* (2008) ont proposé un nouvel algorithme, appelé SKDA pour *Sequential Kernel Density Approximation*, afin d'approcher une densité à partir de MMG, en ajoutant séquentiellement une gaussienne lorsqu'une nouvelle donnée n'est pas expliquée par les gaussiennes précédentes. Dans les paragraphes qui suivent, une explication des grandes lignes de chaque algorithme sera réalisée.

Méthodes non paramétriques

Les fenêtres de Parzen Cette approche non paramétrique utilise l'ensemble des observations $\mathcal{Z}$, réalisations de la variable aléatoire $\mathbf{z}$, pour en déduire une approximation d'une distribution *a posteriori*. L'idée de la méthode est d'attribuer à chaque observation, un voisinage d'influence constant paramétrable. Cette influence étant cumulative, un groupe d'observations très proches les unes des autres, donc caractérisant un aspect du modèle, engendre un mode dans la distribution *a posteriori*. C'est une sorte de généralisation de l'histogramme. La probabilité $p(\mathbf{z})$ d'une observation d'après la distribution *a posteriori* s'écrit formellement :

$$p(\mathbf{z}) \approx N^{-1} \sum_{n=1}^{N} \phi_n(\mathbf{z}) \tag{3.1}$$

où $\phi_k(.) = k(., \mathbf{z_k})$ est une fonction noyau. Cette fonction noyau est paramétrée par une largeur de fenêtre h qui permet un lissage plus ou moins important de la distribution *a posteriori*. C'est généralement une densité gaussienne qui est choisie, la variance de la gaussienne faisant office de paramètre h. L'estimateur converge théoriquement vers la véritable distribution quelque soit le paramètre h choisi. De plus, cet estimateur est réputé pour converger plus rapidement que les méthodes paramétriques. En pratique, le nombre d'observations ne pouvant être infini, le réglage du paramètre h reste très important. Enfin, ce modèle qui utilise l'ensemble des N observations pour déterminer $p(\mathbf{z})$ n'est pas, en général, utilisé à cause de la complexité algorithmique prohibitive.

Les K plus proches voisins Cette autre approche non paramétrique définit la probabilité d'une observation en étudiant la proximité de cette observation aux autres observations qui sont issues du modèle étudié. En effet, si une observation ne fait pas partie du modèle, la distance aux K plus proches observations du modèle sera grande et inversement. Théoriquement, en définissant k_n le nombre d'observations les plus proches à déterminer pour n observations $\{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_n}\}$ de la variable aléatoire $\mathbf{z}$, on détermine $p_n(\mathbf{z})$ par :

$$p_n(\mathbf{z}) = \frac{\frac{k_n}{n}}{V_n} \tag{3.2}$$

avec V_n le volume permettant d'englober les k_n plus proches observations du modèle autour de $\mathbf{z}$. Le terme k_n qui dépend en principe de n, est le paramètre de lissage de la distribution *a posteriori*. Une valeur de k_n grande réduit l'effet du bruit mais affaiblit aussi la distinction entre les modes (ou classes). Tout comme les fenêtres de Parzen, les KPPV souffrent aussi d'une complexité algorithmique prohibitive à cause de la recherche des plus proches voisins et des nombreux calculs de distance associés. Des mécanismes sophistiqués de type arbre de recherche peuvent permettre d'aller plus vite.

Méthodes paramétriques

L'algorithme EM L'idée ici est de déterminer le maximum de vraisemblance des paramètres d'un modèle probabiliste. En pratique, la résolution direct de ce problème d'optimisation n'est pas simple et nécessite des approches plus fines. L'algorithme d'*Expectation-Maximisation* cherche à résoudre ce problème de manière itérative en améliorant successivement le jeu de paramètre $\boldsymbol{\theta}_i$ jusqu'à convergence.

Soit Y des données dites complétées mais inconnues. Soit $\mathcal{L}(\mathcal{Z}, Y; \boldsymbol{\theta}) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{z_n}; \boldsymbol{\theta}) p(\mathbf{z_n}; \boldsymbol{\theta})$ la vraisemblance de l'ensemble $\mathcal{Z}$ pour le modèle de mélange de gaussiennes paramétré par $\boldsymbol{\theta}$. On peut donc définir l'algorithme EM comme dans l'encart 2.

ENTRÉES: Un jeu de paramètre initial $\boldsymbol{\theta}_0$
$i = 0$
tant que non convergence
Étape E : Estimation de la log-vraisemblance des données

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}_i) = E_Y[\ln \mathcal{L}(\mathcal{Z}, Y; \boldsymbol{\theta})|\mathcal{Z}; \boldsymbol{\theta}_i]$$

Étape M : Recherche des paramètres $\boldsymbol{\theta}$ pour maximiser la log-vraisemblance

$$\boldsymbol{\theta}_{i+1} = \underset{\boldsymbol{\theta}}{argmax}\, Q(\boldsymbol{\theta}; \boldsymbol{\theta}_i)$$

$i = i + 1$
fin tant que
retourne Le modèle probabiliste paramétré par $\boldsymbol{\theta}_i$.

Algorithme 2 : Algorithme EM

Dans le cas où l'algorithme EM cherche à estimer les paramètres d'un mélange de gaussiennes, l'ensemble Y représente le fait que la $c^{ième}$ composante du mélange de gaussiennes soit à l'origine ($y_{n,c} = 1$) ou non ($y_{n,c} = 0$) de l'observation $\mathbf{z_n}$. On obtient ainsi :

$$\ln \mathcal{L}(\mathcal{Z}, Y; \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{c=1}^{C} y_{n,c} \ln(w_c\, \mathcal{N}(\mathbf{z_n}, \boldsymbol{\mu}_c, \Sigma_c)) \tag{3.3}$$

avec $\boldsymbol{\theta}$ les paramètres de mélange de gaussiennes. l'estimation de la log-vraisemblance des données s'écrit :

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}_i) = \sum_{n=1}^{N} \sum_{c=1}^{C} t_{n,c} \ln(w_c\, \mathcal{N}(\mathbf{z_n}, \boldsymbol{\mu}_c, \Sigma_c)) \tag{3.4}$$

avec $t_{n,c} = E[y_{n,c}|X, \boldsymbol{\theta}_i]$, la probabilité que l'élément $\mathbf{z_n}$ soit généré par la composante c du mélange de gaussiennes. Grâce à la loi de Bayes, on peut estimer cette probabilité, ce qui représente l'étape d'estimation :

$$t_{n,c} = \frac{w_c \mathcal{N}(\mathbf{z_n}, \boldsymbol{\mu}_c, \Sigma_c)}{\sum_{j=1}^{C} w_j \mathcal{N}(\mathbf{z_n}, \boldsymbol{\mu}_j, \Sigma_j)} \tag{3.5}$$

s'en suit l'étape de maximisation qui réévalue les paramètres du mélange de gaussiennes :

$$w_c^{i+1} = \frac{1}{N}\sum_{n=1}^{N} t_{n,c} \tag{3.6}$$

$$\boldsymbol{\mu}_c^{i+1} = \frac{\sum_{n=1}^{N} \mathbf{z_n} t_{n,c}}{\sum_{n=1}^{N} t_{n,c}} \tag{3.7}$$

$$\Sigma_c^{i+1} = \frac{\sum_{n=1}^{N} t_{n,c}(\mathbf{z_n} - \boldsymbol{\mu}_c^{i+1})(\mathbf{z_n} - \boldsymbol{\mu}_c^{i+1})^T}{\sum_{n=1}^{N} t_{n,c}} \tag{3.8}$$

Cependant cet algorithme présente le principal inconvénient de travailler à nombre de gaussiennes C fixe. Figueiredo et Jain (2002) proposent de pallier ce problème en réajustant ce nombre pendant la phase d'estimation. Si une composante n'est pas représentative des données, elle est supprimée. Par ce mécanisme, le mélange de gaussiennes initiale est choisie avec de nombreuses composantes. Pour plus de détail sur le nombre exact de composantes initiales, le lecteur est invité à se rapporter à l'étude de Figueiredo et Jain (2002) sur la question. De plus, l'algorithme EM dépend d'une initialisation du jeu de paramètre du modèle pouvant engendrer l'arrêt de la méthode dans un maximum local. Enfin, dans le cas où la variable aléatoire $\mathbf{z}$ n'est pas associée à une loi de probabilité de type MMG (une loi uniforme par exemple), l'algorithme EM peut être mal adapté et avoir des difficultés à converger.

Le *Sequential Kernel Density Approximation* Cette autre approche paramétrique proposée par Han *et al.* (2008) cherche aussi à déterminer les paramètres d'un mélange de gaussiennes (et exclusivement cette distribution), à partir d'un ensemble d'observations. La particularité de cette approche est de ne travailler qu'une seule fois sur chacune des observations en les exploitant séquentiellement. Ce genre d'approche très intéressante pour des apprentissages en ligne par exemple, offre des temps d'apprentissage très courts. L'idée de cet algorithme est d'ajouter à un mélange de gaussiennes déjà existant f_t issue de t observations $\{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_t}\}$, une nouvelle gaussienne centrée sur l'observation $\mathbf{z_{t+1}}$. Cette nouvelle gaussienne, si elle est isolée, crée un nouveau mode au mélange, sinon elle contribue à la ré-estimation de la moyenne et la variance du mode ou des modes auxquels elle vient se mélanger.

L'étape de fusion, si elle a lieu, a pour objectif de représenter M gaussiennes $\{\mathcal{N}_1, \ldots, \mathcal{N}_M\}$ de poids respectif $\{w_1, \ldots, w_M\}$ par une seule $\mathcal{N}_{new}$ de poids w_{new}. Pour cela, la fusion s'appuie entre autre sur l'algorithme du *mean-shift* qui sert à identifier la moyenne du mode (identifier comme le maximum local) le plus proche d'un point donné dans l'espace des observations. Le poids de la gaussienne finale est égal à la somme des poids des gaussiennes à fusionner ($w_{new} = \sum_{i=1}^{M} w_i$). La moyenne $\mathbf{c}_{new}$ est déterminée par le *mean-shift* sur le mélange de gaussiennes f_{t+1} en partant de l'observation $\mathbf{z_{t+1}}$. Enfin la ré-estimation de la matrice de covariance est réalisée grâce à la courbure de f_{t+1} au voisinage de $\mathbf{c}_{new}$. En supposant que la matrice hessienne $\hat{H}(\mathbf{c}_{new}) = \boldsymbol{\nabla} f_{t+1}(\mathbf{c}_{new})\boldsymbol{\nabla}^T$ est définie négative, alors la matrice de covariance de la gaussienne finale est égal à :

$$P(\mathbf{c}_{new}) = \frac{w_{new}^{\left(\frac{2}{d+2}\right)}}{|2\pi(-\hat{H}(\mathbf{c}_{new})^{-1})|^{\frac{1}{d+2}}}(-\hat{H}(\mathbf{c}_{new})^{-1}) \tag{3.9}$$

avec d la dimension de l'espace d'observation. Dans le cas où la matrice hessienne n'est pas définie négative, les M gaussiennes sont laissées telles quelles. La procédure est détaillée dans l'encart 3. A noter que la nouvelle gaussienne ajoutée est paramétrée par :

▷ α son poids qui peut être constant (apprentissage à mémoire) ou égal à $\frac{1}{t+1}$
▷ Σ une matrice de covariance fixe.

ENTRÉES: Le mélange de gaussiennes courant $f_t = \sum_{i=1}^{C} w_i \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$
$S = \mathbf{z_{t+1}}$
$f_{t+1} = (1-\alpha) f_t + \alpha \mathcal{N}(\mathbf{z_{t+1}}, \boldsymbol{\mu}_{t+1}, \Sigma_)$
$\mathbf{c}_{new} = MeanShift(f_{t+1}, \mathbf{z_{t+1}})$
$\hat{f} = f_t$
$isolated \leftarrow faux$
tant que $\overline{isolated}$
  $\boldsymbol{\mu}_i = MeanShift(\hat{f}, \mathbf{z_{t+1}})$
  $\mathbf{c} = MeanShift(f_{t+1}, \boldsymbol{\mu}_i)$
  si $\mathbf{c}_{new} \neq \mathbf{c}$
    $isolated \leftarrow vrai$
  sinon
    $S = S \bigcup \{\boldsymbol{\mu}_i\}$
    $\hat{f} = \hat{f} - w_i \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$
  fin si
fin tant que
retourne Fusionner les gaussiennes de l'ensemble S (cf. equation (3.9)).

Algorithme 3 : Algorithme SKDA

Le SKDA est donc une méthode d'estimation particulièrement rapide. Cependant, là encore le MMG est de rigueur et approcher certains type de densité très différent peut s'avérer très difficile. De plus, la convergence peut aussi s'arrêter dans un maximum local puisque pour un même ensemble $\mathcal{Z}$ l'ordre d'exploitation des observations durant l'estimation peut aboutir à des estimations de densité différentes.

Inconvénients majeurs de ces approches pour notre application Si les méthodes non paramétriques ont le principal inconvénient de fournir des modèles très lourds, inutilisables dans des applications temps réel possédant un nombre de données important, les méthodes paramétriques qui fournissent des modèles beaucoup plus légers, ont cependant un inconvénient majeur dans notre cas d'utilisation. En effet, à cause du modèle supposé, les algorithmes EM ainsi que le SKDA se servent implicitement de la distance de Mahalanobis pour comparer les primitives. Or cette distance nécessite de faire la différence entre deux descripteurs, ce qui n'a pas de sens pour les descripteurs que nous utilisons, que ce soit avec l'orientation d'un vecteur mouvement ou avec les descripteurs spatio-temporels. Imaginer des variantes de ces algorithmes qui utiliseraient ces distances particulières est loin d'être évident. Si l'on prend le cas du SKDA par exemple, un calcul du hessien du mélange de gaussiennes est nécessaire. Ce calcul peut devenir très compliqué voir irréalisable si la distance est elle-même trop complexe.

En conséquence, une méthode d'estimation paramétrique permettant de définir simplement sa propre distance entre primitives à l'instar du KDE et des KPPV, sans pour autant souffrir de la lourdeur du modèle associé, est proposée. Une description détaillée de cette méthode est faite dans la partie suivante.

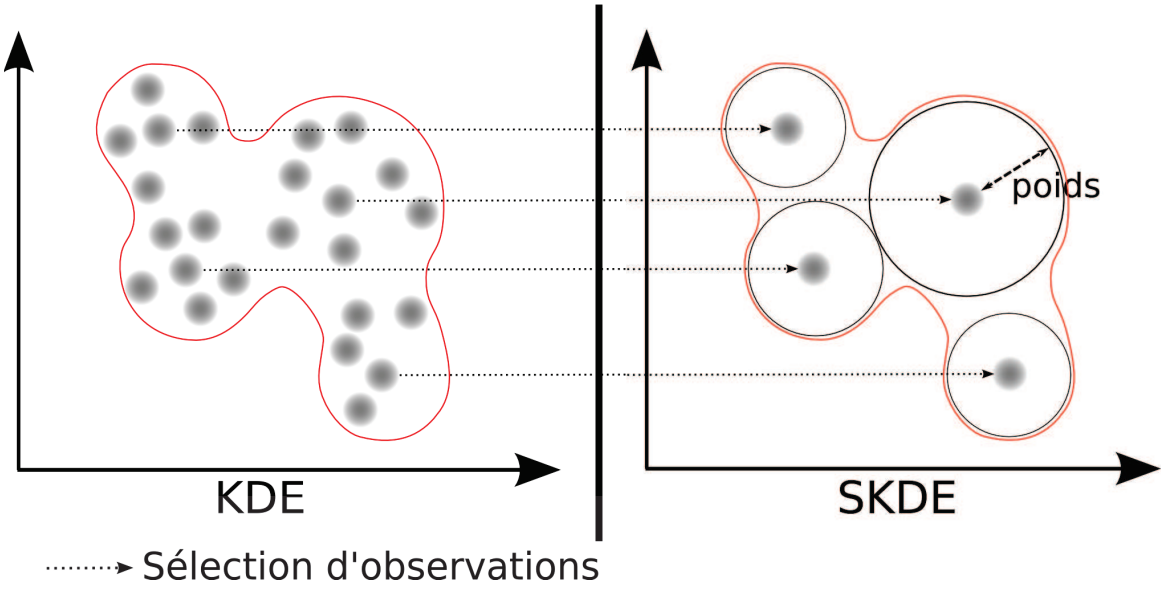


FIGURE 3.1 – Sélection parcimonieuse d'observations pour l'approximation.

3.3 Une méthode hybride : le SKDE

Cette méthode est une approximation parcimonieuse du KDE que nous appellerons *Sparse Kernel Density Estimation* (SKDE). L'approximation s'appuie sur un critère de minimisation au sens des moindres carrés.

3.3.1 Théorie

Nous proposons d'approcher le KDE avec un modèle parcimonieux composé d'une somme pondérée de fonctions noyaux afin d'éviter le problème de la complexité de calcul intrinsèque au KDE. Notre méthode est une méthode itérative basée sur la sélection d'observations représentatives d'un ensemble d'apprentissage, comme illustré figure 3.1.

L'idée ici est de déterminer un modèle vu comme un KDE pondéré où la majorité des poids serait mise à zéro :

$$p(\mathbf{z}) \approx \sum_{n=1}^{N} w_n \phi_n(\mathbf{z}) \approx \sum_{n=1}^{\tilde{N}} w_{s(n)} \phi_{s(n)}(\mathbf{z}) \tag{3.10}$$

où $s(n)$ est une fonction de sélection et $\mathbf{w} \doteq (w_1, w_2, \ldots, w_N)$ est un vecteur de poids. L'équation (3.10) est une approximation de l'équation (3.1) avec $\tilde{N} << N$. La section suivante présente l'algorithme itératif utilisé pour choisir à la fois les fonctions de base et estimer les poids associés.

3.3.2 Sparse Kernel Density Estimation

L'équation (3.1) peut aussi être exprimée sous la forme :

$$p(\mathbf{z}) \approx (N^{-1}) 1_{(1,N)} \left(\boldsymbol{\phi}[\mathbf{z}]\right) \tag{3.11}$$

où $\boldsymbol{\phi}$ est un vecteur de fonctions défini par $\boldsymbol{\phi}[\mathbf{z}] \doteq (\phi_1(\mathbf{z}), \phi_2(\mathbf{z}), \ldots, \phi_N(\mathbf{z}))$. $1_{(1,N)}$ est un vecteur ligne formé de N fois la valeur 1.

La méthode proposée, définie par l'équation (3.10), peut aussi être écrite sous la forme réduite :

$$p(\mathbf{z}) \approx \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\phi}}[\mathbf{z}] \tag{3.12}$$

où $\tilde{\mathbf{w}} \doteq (w_{s(1)}, \ldots, w_{s(\tilde{N})})$ est un vecteur contenant les poids non nuls et $\tilde{\boldsymbol{\phi}} \doteq (\phi_{s(1)}(\mathbf{z}), \ldots, \phi_{s(\tilde{N})}(\mathbf{z}))$ un vecteur de fonctions de base ϕ associées aux poids non nuls.

Soit Φ la matrice de design de taille $N \times N$ construite de telle manière que l'élément de la ligne i et de la colonne j soit donné par $\Phi_{i,j} = \phi_j(\mathbf{z_i})$. Φ est une matrice carrée symétrique, à partir de laquelle un estimateur KDE de la probabilité associée à l'observation $\mathbf{z_n}$ de l'ensemble d'observations est donné par la somme des éléments de la ligne k de Φ :

$$p(\mathbf{z_n}) \approx N^{-1} \sum_{n'=1}^{N} \phi_{n'}(\mathbf{z_n}) = N^{-1} \sum_{n'=1}^{N} \Phi_{n,n'} \tag{3.13}$$

Un vecteur de probabilités $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_N)$ en relation avec l'ensemble d'observations est construit :

$$\boldsymbol{\varphi} = (N^{-1})\Phi \mathbf{1}_{(N,1)} \approx (p(\mathbf{z_1}), p(\mathbf{z_2}), \ldots, p(\mathbf{z_N})) \tag{3.14}$$

Un algorithme itératif à deux étapes est proposé pour calculer à la fois la fonction de sélection $s(.)$ et le vecteur de poids $\tilde{\mathbf{w}}$. La première itération démarre avec $m = 1$ et nous définissons un vecteur d'erreurs résiduelles $\mathbf{r}$. Ce vecteur correspond à la différence entre l'approximation courante (contenant m noyaux) et l'estimation KDE de la distribution pour l'ensemble des observations $\mathcal{Z}$. Formellement, cela donne :

$$\mathbf{r} = \boldsymbol{\varphi} - \tilde{\Phi}\, \tilde{\mathbf{w}} \tag{3.15}$$

où $\tilde{\Phi}$ est une matrice de taille N par m telle que l'élément de la ligne i et de la colonne j est donné par $\tilde{\Phi}_{i,j} = \tilde{\phi}_{s(j)}(\mathbf{z_i})$.

A la première itération, l'approximation est égale à la distribution nulle. Le vecteur $\mathbf{r}$ est donc initialisé à $\mathbf{r} = \boldsymbol{\varphi}$. Les itérations sont ensuite :

1. $s(m)$ sélectionne le maximum de vraisemblance du vecteur d'erreurs résiduelles $\mathbf{r}$:

$$s(m) = \underset{i}{argmax}\, \mathbf{r}_i \tag{3.16}$$

2. Le vecteur de poids associé $\tilde{\mathbf{w}} = (\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_m)$ est estimé en utilisant un critère de minimisation au sens des moindres carrés sur l'ensemble des observations :

$$\tilde{\mathbf{w}} = \underset{\mathbf{w}}{argmin} \left(\|\boldsymbol{\varphi} - \tilde{\Phi}\, \mathbf{w}\| \right) \tag{3.17}$$

Le vecteur d'erreurs résiduelles est alors mis à jour par :

$$\mathbf{r} = \boldsymbol{\varphi} - \tilde{\Phi}\, \tilde{\mathbf{w}} \tag{3.18}$$

Le critère d'arrêt de cet algorithme sera présenté dans la section suivante. On peut résumer l'ensemble de la méthode SKDE avec l'algorithme 4.

Les étapes 1 et 2 sont répétées jusqu'à ce que $\max \mathbf{r} < h(\mathcal{Q}_l)$. Le paramètre $\mathcal{Q}_l$ représente la précision désirée et h est une fonction retournant un seuil correspondant à la précision désirée (section

ENTRÉES: Matrice de design $\mathfrak{\Phi}$, quantile du critère d'arrêt $\mathcal{Q}_l$
Calcul du vecteur de vraisemblance : $\boldsymbol{\varphi}$
Initialisation : $m = 1$ et $\mathbf{r} = \boldsymbol{\varphi}$
répéter
Extraction du maximum de vraisemblance :

$$s(m) = \underset{i}{argmax}\, \mathbf{r}_i$$

Calcul du vecteur de poids $\tilde{\mathbf{w}}$:

$$\tilde{\mathbf{w}} = \underset{\mathbf{w}}{argmin} \left(\| \boldsymbol{\varphi} - \tilde{\mathfrak{\Phi}}\, \mathbf{w} \| \right)$$

Mise à jour du vecteur d'erreurs résiduelles $\mathbf{r} = \boldsymbol{\varphi} - \tilde{\mathfrak{\Phi}}\, \tilde{\mathbf{w}}$
$m = m + 1$
jusqu'à $\max \mathbf{r} < h(\mathcal{Q}_l)$
retourne Vecteur de poids $\tilde{\mathbf{w}}$ et indices des observations sélectionnées :
$\mathbf{s} = (s(1), s(2), \ldots, s(m-1))^T$

Algorithme 4 : Algorithme SKDE

3.3.4). Pour une approximation plus grossière, $\mathcal{Q}_l$ peut être diminué. Dans ce cas le nombre d'observations sélectionnées (que nous appellerons par la suite points de contrôle) diminue. Une illustration des effets de ce paramètre est donnée dans la section 3.4. Cette approche permet de donner une bonne approximation de la vraisemblance avec $\tilde{N}$ points de contrôle, $\tilde{N} \ll N$. Par la suite nous noterons, $\tilde{\mathbf{z}}_\mathbf{i} = \mathbf{z}_{\mathbf{s(i)}}$ les observations conservées comme points de contrôle avec $\tilde{w}_i$ leur poids associés.

Lors de l'utilisation de noyaux gaussiens, la méthode proposée peut s'apparenter à un MMG avec des matrices de covariance fixes. De plus, le modèle est proche des *Support Vector Machine* (SVM) décrit par Burges (1998) ou des *Relevant Vector Machine* introduit par (RVM) Tipping (2001) car il utilise le même formalisme et c'est un modèle parcimonieux.

3.3.3 SKDE Local

Plutôt que d'estimer les poids à l'aide d'un critère prenant en compte l'ensemble des observations $\mathbf{z_n}$, il est possible de contraindre le problème uniquement au niveau des points de contrôle $\mathbf{z_{s(n)}}$. Ceci permet de diminuer légèrement la complexité algorithmique en résolvant un système plus simple. L'équation (3.17) peut alors être réécrite :

$$\tilde{\mathbf{w}} = \underset{\mathbf{w}}{argmin} \left(\| \tilde{\boldsymbol{\varphi}} - \check{\mathfrak{\Phi}}\, \mathbf{w} \| \right) \tag{3.19}$$

où $\tilde{\boldsymbol{\varphi}} \doteq (\varphi_{s(1)}, \varphi_{s(2)}, \ldots, \varphi_{s(\tilde{N})})^T$ et $\check{\mathfrak{\Phi}}$ est une matrice de taille $m \times m$ telle que l'élément de la ligne i et de la colonne j est donné par $\check{\mathfrak{\Phi}}_{i,j} = \phi_{s(j)}(\mathbf{z_{s(i)}})$. Cela revient à résoudre un système linéaire carré de faible dimension.

3.3.4 Le Critère de Confiance

Nous présentons ici le critère d'arrêt basé sur une notion de confiance probabiliste. Le problème à résoudre ici, est de déterminer un seuil séparateur entre les régions où les probabilités sont fortes et celles où elles sont faibles, sous entendu négligeables. La difficulté du problème est de trouver un

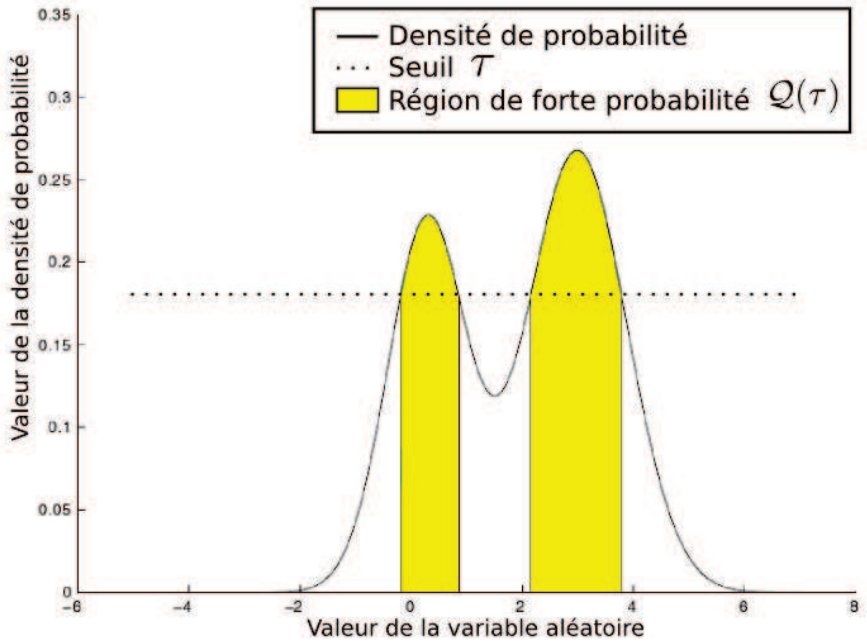


FIGURE 3.2 – Relation entre le seuil de la densité de probabilité, le quantile et la région de plus forte probabilité.

critère permettant de caractériser la notion de probabilité forte ou faible quelque soit la forme de la densité de probabilité.

Soit $\mathcal{Q}(\tau)$ le quantile pour une valeur de densité de probabilité donnée τ :

$$\mathcal{Q}(\tau) = \int_{p(\mathbf{x}) \geq \tau} p(\mathbf{x}) d\mathbf{x} \tag{3.20}$$

Ce quantile correspond aux zones de plus forte probabilité, celles pour lesquelles la densité est supérieure à τ (cf. figure 3.2). La fonction inverse $h(\mathcal{Q}) = \tau$ est définie telle que $\mathcal{Q} \in [0, 1]$. Ce quantile représente la portion de la densité de probabilité qui contient les fortes probabilités (0.9 pour 90% de la densité de probabilité par exemple). C'est donc cette fonction $h(.)$ qu'il est nécessaire de déterminer quelque soit la forme de la densité de probabilité. Ce problème est simple à résoudre pour des distributions simples telles que des densités gaussiennes. Mais pour des densités plus compliquées telles que des mélanges de gaussiennes, le problème peut ne pas avoir de solution exacte. Néanmoins, des méthodes approchées peuvent être utilisées.

La méthode approchée présenté ici pour résoudre ce problème s'appuie sur une méthode de Monte Carlo (Paalanen *et al.* (2006)). Soient $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ N points aléatoirement choisis selon la distribution p et $p_i = p(\mathbf{x}_i)\ \forall i \in [1, N]$. Les p_i sont triés par ordre croissant $Y = \{y_1, y_2, \ldots, y_N\}$). Cet ensemble est utilisé pour estimer $\mathcal{Q}(\tau)$ et $h(\mathcal{Q})$ en utilisant une interpolation linéaire :

$$i = \underset{i}{argmax} \{y_i | y_i \leq \tau\} \tag{3.21}$$

$$F(\tau) \approx \begin{cases} 1 & si\ \tau < y_1 \\ 0 & si\ \tau \geq y_N \\ 1 - \frac{i+l(i,\tau)}{N} & sinon \end{cases} \tag{3.22}$$

avec

$$l(i,\tau) = \begin{cases} 0.5 & si\ y_{i+1} - y_i = 0 \\ \frac{\tau - y_i}{y_{i+1} - y_i} & sinon \end{cases} \tag{3.23}$$

La transformée inverse $h(\mathcal{Q})$ peut être déduite par :

$$\tau = h(\mathcal{Q}) \approx \begin{cases} y_N & si\ i = N \\ y_i + (N(1-\mathcal{Q}) - i)(y_{i+1} - y_i) & sinon \end{cases} \tag{3.24}$$

avec $i = \lfloor (N-1) \times (1-\mathcal{Q}) + 1 \rfloor$.

Ce mécanisme permet donc de manière simple de distinguer à partir d'un tirage aléatoire d'observations issues d'une distribution, les zones de fortes probabilités, là où se concentrent des observations similaires, des autres. Dans notre cas, la distribution en question est la distribution réelle que l'on cherche à approcher. Pour cette distribution, les zones de forte probabilité sont celles porteuses d'information qu'il faut approcher de manière la plus précise possible, d'où le choix du critère d'arrêt de l'algorithme 4.

3.3.5 Dans le cadre applicatif de ces travaux

La fonction de décision L'utilisation finale de l'estimation de densité proposée étant la classification, une fonction de décision associée à cette estimation doit être déterminée. Plutôt que d'utiliser un seuil empirique sur la vraisemblance, le critère de confiance défini au paragraphe 3.3.4 est réutilisé.

En effet, une fois l'approximation réalisée, la transformation inverse de l'équation (3.24) peut aussi donner le seuil de classification lors d'une phase d'analyse. Selon notre approche, le modèle approximé se veut représentatif de la distribution réelle de mouvement sous-jacente. En considérant des observations $\mathcal{Z}$ d'une séquence quelconque comme tirage aléatoire X au modèle approximé, il est possible, toujours grâce à l'équation (3.24), d'en déduire un seuil de classification. Le réglage du quantile $\mathcal{Q}$ ici permet une détection plus ou moins « sévère » en considérant que $\mathcal{Q}\%$ des observations appartiennent au modèle.

En pratique, dans le cadre de notre application, le quantile $\mathcal{Q}$ permet d'avoir un paramètre avec une signification physique réelle, à savoir « les observations d'un bloc sont normales à $\mathcal{Q}\%$ ». Pour avoir un rapport *bonnes détections/fausses alarmes* correct, $\mathcal{Q}$ est choisi très proche de 1. Ceci est normal puisque, les évènements anormaux sont des évènements rares. C'est cette phase qui permet de déterminer les règles de décision pour chacun des blocs comme présenté sur la figure 1.9 du chapitre 1.

Cependant lorsque la cardinalité de l'ensemble X est insuffisante, l'estimation du seuil peut être biaisée. En effet, dans ce genre de cas (typiquement en des zones limitrophes au mouvement global de la foule), le seuil $h(\mathcal{Q})$ obtenu est en général égal à une interpolation très proche de y_1 qui peut très bien être une vraisemblance tout à fait normale vu que la statistique de X n'est pas suffisante. Une solution pourrait être de ne pas déterminer de seuil en ces zones, de les considérer comme singulières et de ne pas effectuer de classification. Le problème est de déterminer à partir de quelle cardinalité, il faut considérer la zone comme singulière. Une autre solution, est d'ajouter, une vraisemblance faible $y_{anormal}$, témoin de l'anormal, au vecteur Y. Statistiquement très minoritaire par rapport au reste des observations, elle permet d'assurer la mise à l'écart de l'anormalité. Cette vraisemblance devient le nouveau terme y_1 du vecteur Y et il est ainsi possible d'obtenir un seuil $h(\mathcal{Q})$ séparant une vraisemblance anormale $y_{anormal}$ d'une normale y_2 (anciennement y_1). Pour les ensembles de forte cardinalité, ce rajout est insignifiant puisque la plus grande vraisemblance considérée comme

anormale est désormais obtenue pour $i+1 \gg 1$, ce qui pour des ensembles fortement échantillonnés ne représente pas une grande variation du seuil.

Des évaluations en termes de classification sont effectuées au chapitre 5 pour valider l'apport de la mise en place de ce mécanisme de seuil de confiance à des fins de classification par rapport à un seuil de classification empirique.

Adéquation avec le flot optique Rappelons que le système présenté met en place une grille fixe sur une scène donnée et qu'en chaque bloc de cette grille, une fonction de vraisemblance est estimée en vue d'une classification temps réel. Lors de l'apprentissage, l'ensemble apprentissage $\mathcal{Z}$ du SKDE est constitué des observations au niveau de chaque bloc cumulées à travers le temps.

Le noyau utilisé pour le SKDE, est un noyau gaussien $\mathcal{N}(.,\mu,\sigma)$ de la forme :

$$\mathcal{N}(.,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d_D(.,\mu)}{2\sigma}\right) \tag{3.25}$$

avec d_D la distance associée au descripteur choisi.

A noter que dans le cas où les primitives d'analyse sont celles issues du flot optique (cf. chapitre 2), plusieurs observations sont obtenues à un instant t par bloc. Or, lors de la phase de classification, une seule de ces observations doit être analysée afin de maintenir une classification du mouvement par bloc rapide à réaliser et viable pour le temps réel. Pour déterminer l'observation la plus représentative du bloc, il est possible de réutiliser le procédé du SKDE. En posant $\mathcal{Z}^{(t)}$ les observations à un instant t pour un bloc donné comme ensemble d'apprentissage du SKDE et en arrêtant le procédé d'approximation à $\tilde{N}=1$, on détermine ainsi l'observation ayant la vraisemblance maximale du vecteur $\boldsymbol{\varphi}$ (cf. équation (3.14)), noté $\tilde{\mathbf{z}}^{(t)}$. De la même manière, en phase d'apprentissage, l'ensemble $\mathcal{Z}^{(t)}$ pourrait être rajouté à l'ensemble $\mathcal{Z}$. Cependant pour des problèmes de complexité due à un ensemble $\mathcal{Z}$ trop grand, seul l'observation la plus représentative à l'instant t, $\tilde{\mathbf{z}}^{(t)}$ est ajoutée.

Concernant le descripteur spatio-temporel, cette étape de réduction d'observation par bloc à un instant t n'est pas nécessaire car ce descripteur ne donne déjà qu'une seule observation par bloc à un instant donné.

3.4 Expérimentations

Cette section a pour but de valider la qualité d'approximation d'une distribution quelconque, via la méthode proposée du SKDE à des fins d'estimation de densité. Une étude des performances à des fins de classification qui l'utilisation finale qui nous intéresse, sera ensuite réalisée.

3.4.1 Influence des paramètres du SKDE

Pour illustrer l'équivalence des deux résolutions présentées aux sections 3.3.2 et 3.3.3, une illustration est montrée figure 3.3. La distribution KDE, qui représente la vérité terrain, est représentée par la courbe en tirets noirs. La variance σ du noyau gaussien utilisé sera choisie empiriquement égale à 1. Le modèle parcimonieux $\widetilde{\mathcal{Z}}$ du SKDE a été représenté pour la résolution globale de l'équation (3.17) ainsi que pour la résolution locale de l'équation (3.19), pour différentes valeurs de $\mathcal{Q}_l$. À $\mathcal{Q}_l$ égal, nous pouvons constater sur la figure 3.3 que notre approximation locale peut être comparée à la résolution globale en matière de précision quelque soit la distribution. La résolution globale converge

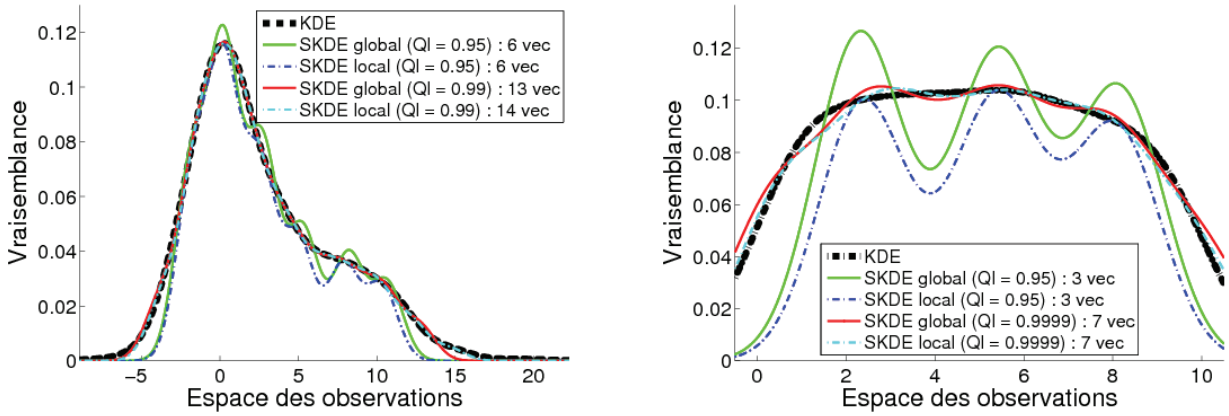


FIGURE 3.3 – Approximation du KDE avec le SKDE sur deux distributions différentes : la courbe en tirets noirs montre le KDE, les courbes en trait plein représentent le SKDE avec la résolution globale et celles en tirets et points alternés représentent le SKDE avec la résolution locale. Les courbes vertes et bleues sont des approximations grossières ($\mathcal{Q}_l = 0.95$) alors que les courbes rouges et cyans sont des approximations précises ($\mathcal{Q}_l = 0.99$).

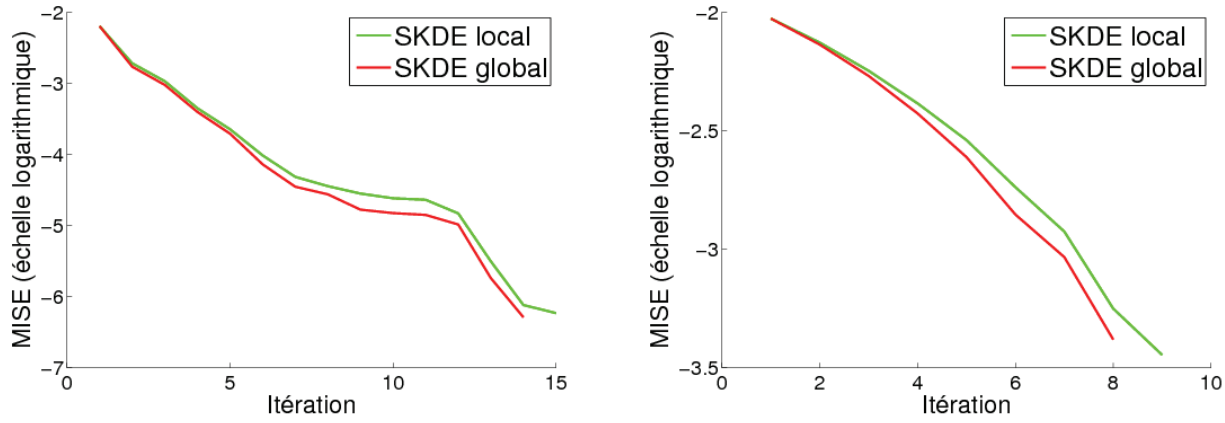


FIGURE 3.4 – Evolution du critère MISE à chaque itération pour la résolution globale et locale du problème du SKDE.

en oscillant vers la vérité terrain alors que la version locale s'en rapproche sans jamais la surestimer. Le nombre d'itération pour atteindre la convergence a aussi été représentée figure 3.4 grâce à la moyenne de l'intégrale des erreurs quadratiques (MISE pour *Mean Integrated Square Error*) entre chaque méthode d'estimation (locale ou globale) et la vérité terrain estimée par la distribution KDE. Pour une approximation p_{approx}, cette moyenne est calculée par rapport à une distribution de référence p_{ref} grâce à la formule :

$$MISE(p_{approx}, p_{ref}, \mathcal{Z}) = \frac{1}{N}\sum_{n=1}^{N}(p_{approx}(\mathbf{z_n}) - p_{ref}(\mathbf{z_n}))^2 \tag{3.26}$$

Nous pouvons constater sur ces courbes semilogs que les deux méthodes convergent de manière similaire.

Concernant l'influence du paramètre $\mathcal{Q}_l$, la distribution à deux modes de la figure 3.3 montre que plus $\mathcal{Q}_l$ est petit, plus l'approximation est grossière. Le nombre d'observations sélectionnées

(ou points de contrôle), c'est-à-dire le nombre de gaussiennes nécessaires à l'approximation de cette distribution, sont respectivement de $\tilde{N} = 6$ et $\tilde{N} = 14$ pour les courbes bleue et cyan.

La version locale de l'approximation sera utilisée dans le reste du manuscrit. La section suivante vise à comparer notre méthode SKDE locale avec les autres méthodes existantes.

3.4.2 Résultats comparatifs

3.4.2.1 Qualité d'approximation

L'objectif ici est de rendre compte de la qualité d'approximation de la méthode proposée indépendamment de la structure applicative que nous nous sommes imposée. Pour cela, une comparaison avec les autres méthodes classiquement utilisées pour l'estimation de densité de probabilité, est effectuée. Plusieurs algorithmes ont été testés : 1) le modèle KDE Duda *et al.* (2001), 2) le SKDA proposé par Han *et al.* (2008) et 3) l'algorithme EM de Paalanen *et al.* (2006) pour l'approximation de MMG.

Nous comparons les quatre algorithmes en matière de précision, de parcimonie et de temps de calcul. La précision est calculée grâce au critère MISE entre la véritable distribution, si connue, ou la distribution KDE dans le cas contraire. Plusieurs bases d'observations avec des données synthétiques (observations tirées d'une distribution connue ou inconnue) ont été utilisées pour la comparaison :

- ▷ B_1 : une base synthétique contenant 3000 observations d'un espace de dimension 1 tirées d'un mélange de gaussiennes à deux modes ($\mathcal{N}_1(0, 2^2)$ de poids 0.6 et $\mathcal{N}_2(7, 4^2)$ de poids 0.4)
- ▷ B_2 : une base synthétique contenant 3000 observations d'un espace de dimension 1 tirées d'une distribution uniforme entre 1 et 10.
- ▷ B_{ripley} [2] : une base synthétique de modèle inconnu contenant 2 classes différentes avec 125 observations de dimension 2 pour chacune des classes. Cette base de référence, est entre autre utilisée dans l'ouvrage sur la reconnaissance de forme et les réseaux de neurones de Ripley (1996). Les résultats ont été calculés sur chacune des classes séparément avant d'être moyennés.

La comparaison est résumée sur le tableau 3.1. Les jeux de paramètres pour chaque méthode sont les mêmes que ceux utilisés pour les bases B_1 et B_2 de la figure 3.5. Ils ont été choisis de manière à avoir des résultats les plus proches de la véritable distribution, c'est-à-dire le critère MISE par rapport à la véritable distribution (MISE / VD) minimum. Le paramètre de qualité $\mathcal{Q}_l$ de notre méthode a été choisi afin d'avoir le meilleur compromis entre le nombre de points de contrôle et l'écart MISE avec le KDE, sachant que l'écart MISE est décroissant lorsque le nombre de points de contrôle augmente. Comparé aux autres méthodes, le SKDE donne des résultats similaires en matière de précision. Comme attendu, le SKDE est très proche du KDE. Concernant le nombre de points de contrôle conservés, nous réduisons très largement le KDE, mais nous conservons généralement plus de points de contrôle que le nombre de gaussiennes utilisées dans les MMG type SKDA ou EM. Ceci est dû à la largeur de noyau fixe de notre méthode. Sur des distributions telles que les distributions uniformes qui ne se prêtent pas bien à l'approximation par un mélange de gaussiennes, nous pouvons voir que le SKDE se comporte de manière satisfaisante. Les temps de calcul sont donnés en nombre de cycles machine. Tous les algorithmes ont été exécutés sous Matlab. Les temps présentés sont donnés pour information seulement car les différents codes ne sont pas nécessairement optimisés et la complexité de l'algorithme EM est inconnue. Le SKDA qui a une complexité linéaire est clairement la méthode la plus rapide mais aussi la moins précise, ce qui est l'exact opposé de l'algorithme EM.

2. http ://www.stats.ox.ac.uk/pub/PRNN/

Bases	Méthodes	MISE / VD	MISE / KDE	Points de contrôle	Temps d'estimation
B_1	KDE	$7.02e^{-5}$		3000	
	SKDE	$6.93e^{-5}$	$\mathbf{6.71e^{-7}}$	14	4.23
	SKDA	$4.31e^{-4}$	$2.76e^{-4}$	**2**	**0.13**
	EM	$\mathbf{8.13e^{-6}}$	$3.1e^{-5}$	**2**	163.06
B_2	KDE	$2.64e^{-4}$		3000	
	SKDE	$2.8e^{-4}$	$\mathbf{6.6e^{-6}}$	7	4.3
	SKDA	$1.76e^{-3}$	$1.22e^{-3}$	**1**	**0.13**
	EM	$\mathbf{1.8e^{-4}}$	$3.65e^{-4}$	7	180.12
B_{ripley}	KDE	inconnu		150	
	SKDE	inconnu	$\mathbf{2.2e^{-3}}$	2	**0.005**
	SKDA	inconnu	$3.8e^{-3}$	**1**	**0.0005**
	EM	inconnu	1.75	3	0.237

TABLE 3.1 – Résultats d'estimation de distributions selon différents critères. Chacune des méthodes est évaluée en matière de précision par rapport à la véritable distribution ou à l'estimation KDE (grâce au critère MISE), en matière de parcimonie (le nombre de points de contrôle) et de temps de calcul (exprimé en milliard de cycles machine). Les meilleurs performances dans chacunes des catégories sont mises en gras.

Le SKDE est un bon compromis entre les deux. La majorité de notre temps de calcul est consacrée au calcul de la matrice Φ qui est en $O(N^2)$ (N le nombre d'observations du modèle).

Concernant la base B_{ripley}, la densité de probabilité sous-jacente n'est pas connue. En conséquence, la comparaison est effectuée uniquement avec une estimation de cette dernière par KDE. Les paramètres pour chaque méthode ont été estimés par validation croisée. L'apprentissage a été réalisé sur 2500 exemples et testé sur les 500 autres. La très grande erreur moyenne de l'algorithme EM n'est pas due à une erreur sur les moyennes des gaussiennes mais à une surestimation des modes par rapport au KDE. De plus, l'algorithme SKDE avec une approximation grossière (seulement deux points de contrôle conservés) donne des résultats comparables à ceux des autres méthodes.

Une représentation graphique de l'approximation des bases B_1 et B_2 est donnée figure 3.5. La véritable distribution est représentée par la courbe en tirets noirs. Théoriquement le KDE converge vers la véritable distribution pour une infinité d'observations, quelque soit la largeur de bande. Ici l'ensemble d'observations ne comporte que 3000 éléments. En conséquence, la largeur de bande devient un paramètre sensible. Elle est déterminée de manière expérimentale. Elle doit être suffisamment grande pour éviter à la distribution KDE de ressembler à un ensemble de pics de Dirac, chaque pseudo Dirac étant une gaussienne associée à une observation. Elle ne doit néanmoins ne pas être trop grande pour ne pas fusionner les différents modes réels en un seul. Nous pouvons voir sur la distribution à deux modes qu'à part le SKDA, les autres font ressortir les deux modes originaux dans l'approximation effectuée. Pour la distribution uniforme, l'algorithme EM donne une approximation oscillante alors que le SKDA modélise le créneau par une gaussienne très large. Le SKDE s'approche de la distribution KDE qui est l'approximation la plus vraisemblable du créneau (mis à part des effets de bord).

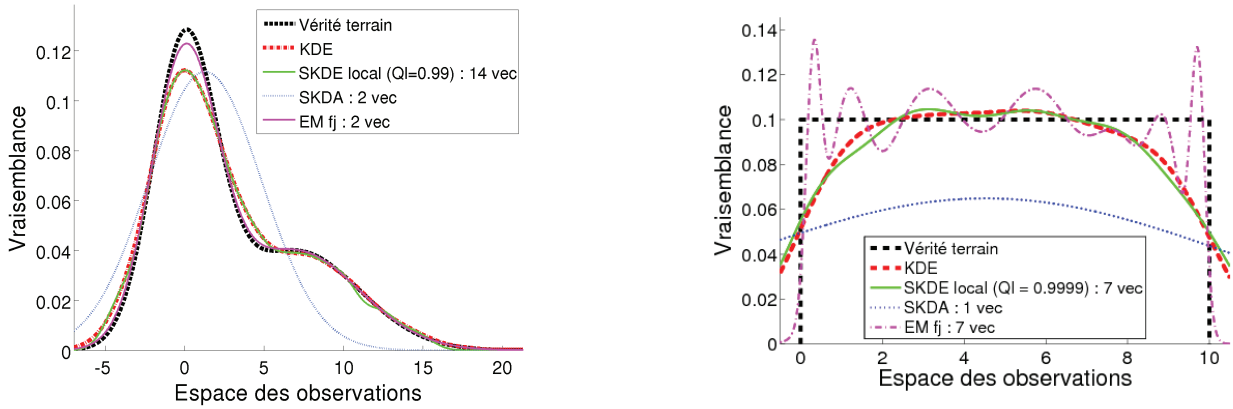


FIGURE 3.5 – Comparaison de l'approximation de densité par différentes méthodes : la courbe en tirets noirs montre la véritable distribution et la rouge, l'estimation du KDE. Les courbes en trait plein verte, en pointillé bleue et en tirets et points alternés violet représentent respectivement le SKDE, le SKDA et l'EM.

3.4.2.2 Qualité de classification

L'objectif ici est de montrer les performances de l'approche probabiliste proposée lorsqu'elle est utilisée en termes de classification. La fonction de décision utilisée est donc celle basée sur le critère de confiance comme expliqué au paragraphe 3.3.5.

L'approche proposée fait partie des approches probabilistes, il existe néanmoins d'autres approches pour répondre à la problématique de classification qui est la nôtre. Ces approches cherchent aussi à déterminer une fonction de décision capable de classer une nouvelle observation comme normale ou anormale à partir uniquement d'un ensemble de caractéristiques décrivant le fonctionnement « normal » ou « nominal » d'un système. Parmi elles, la plus connue est l'algorithme *One Class SVM* qui est l'adaptation des machines à vecteurs de support ou SVM au cas non supervisé.

Le *One Class SVM* Les Séparateurs à Vaste Marge ou *Support Vector Machine en anglais* (SVM) sont des techniques d'apprentissage supervisé initialement proposées par Vapnik (1998). A partir de deux ensembles d'apprentissage, un ensemble d'observations appartenant aux modèles et un ensemble d'observations n'y appartenant pas, les SVM sont utilisés pour répondre à des problèmes de classifications binaires ou de régression. C'est la méthode la plus connue parmi les machines à noyau. Elle utilise une fonction noyau pour transposer les observations dans un espace de dimension supérieure et ainsi déterminer une séparation linéaire, un hyperplan, entre les deux classes d'objets comme illustré à la figure 3.6. Le principe des SVM est ensuite de maximiser la marge, qui est la distance séparant les exemples les plus proches de la frontière entre les deux classes d'objet. Maximiser cette marge permet d'être plus discriminant entre les deux classes. La sortie d'un apprentissage SVM est un ensemble d'observations et leur poids associé permettant de définir l'hyperplan séparateur sous forme d'une combinaison linéaire de fonctions noyaux centrées en ces observations. Ces observations sélectionnées sont appelées, les vecteurs supports. Ce sont des points de l'espace des primitives qui entourent les points du modèle le long de la frontière.

Ici, ne connaître que les observations appartenant au modèle impose des adaptations de la méthodologie des SVM. Une variante, baptisée *One Class SVM* a été proposée par Schölkopf et Smola (2001). Un détail précis de cet algorithme nécessiterait de ré-expliquer précisément les principes des SVM

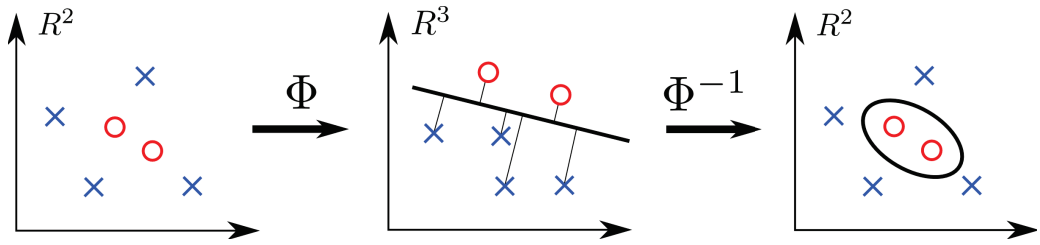


FIGURE 3.6 – Théorie des SVM. A partir de deux ensembles d'observations d'un espace donné, l'utilisation d'une fonction noyau Φ permet le passage dans un espace de dimension supérieur dans lequel les deux ensembles sont séparables linéairement. La séparation choisie est celle maximisant la marge entre les deux ensembles de points et la frontière.

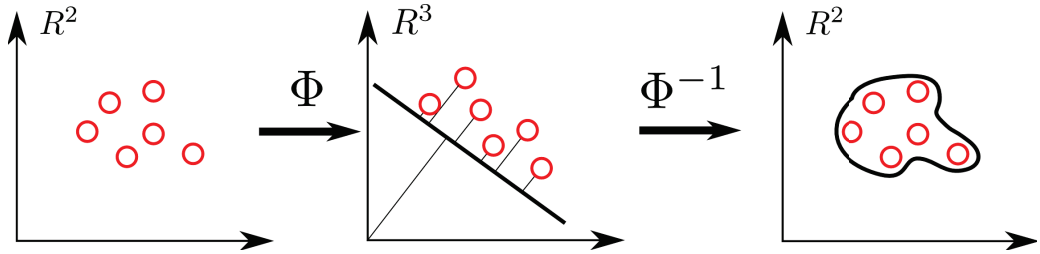


FIGURE 3.7 – Théorie des *One Class SVM*. A partir d'un ensemble d'observations d'un espace donné, la séparation linéaire est choisi de manière à maximiser la marge entre les observations dans de l'espace de dimension supérieuret l'origine de cet espace.

pour ensuite expliquer les variantes. Cet algorithme n'ayant pas fait l'objet d'une étude très poussée dans le cadre de nos travaux, nous nous contenterons de décrire les grandes lignes et invitons le lecteur à se référer au chapitre 8 du livre de Schölkopf et Smola (2001) pour de plus amples détails.

L'idée générale des *One Class SVM* est de trouver un fonction f valant $+1$ dans une région limitées qui capture la plupart des observations et -1 ailleurs. Pour y parvenir, les observations sont à nouveaux transposées via une fonction noyau dans un espace de dimension supérieure $\mathcal{H}$ puis la marge de l'hyperplan séparateur est maximisée, non pas entre deux classes mais entre la classe des observations et l'origine de l'espace $\mathcal{H}$ (cf. schéma de la figure 3.7). La fonction f permet ainsi de déterminer de quel côté de l'hyperplan se trouve une nouvelle observation à tester.

Un des avantages des machines à vecteurs support est de pouvoir travailler sur des primitives quelconques, du moment qu'une fonction noyau associée à cette primitive est définie. Cette fonction noyau doit respecter le théorème de Mercer, à savoir être symétrique et semi-définie positive. En pratique, ce sont souvent des noyaux gaussiens qui sont utilisés, de la forme :

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{d(\mathbf{x}, \mathbf{y})^2}{2\sigma^2}\right) \tag{3.27}$$

avec d, soit une distance propre aux primitives traitées, soit la distance euclidienne classique.

Ces vecteurs supports sont l'équivalent des points de contrôle du SKDE. Cependant contrairement à ces derniers, ces vecteurs support sont des points « sentinelles » autour des points appartenant au modèle et sont beaucoup plus durs à exploiter en dehors du cadre des SVM. Les points de contrôle des approches probabilistes représentent des observations caractéristiques du modèle, donc plus faciles à réexploiter individuellement car porteuses d'un sens physique, elles représentent une « facette » du

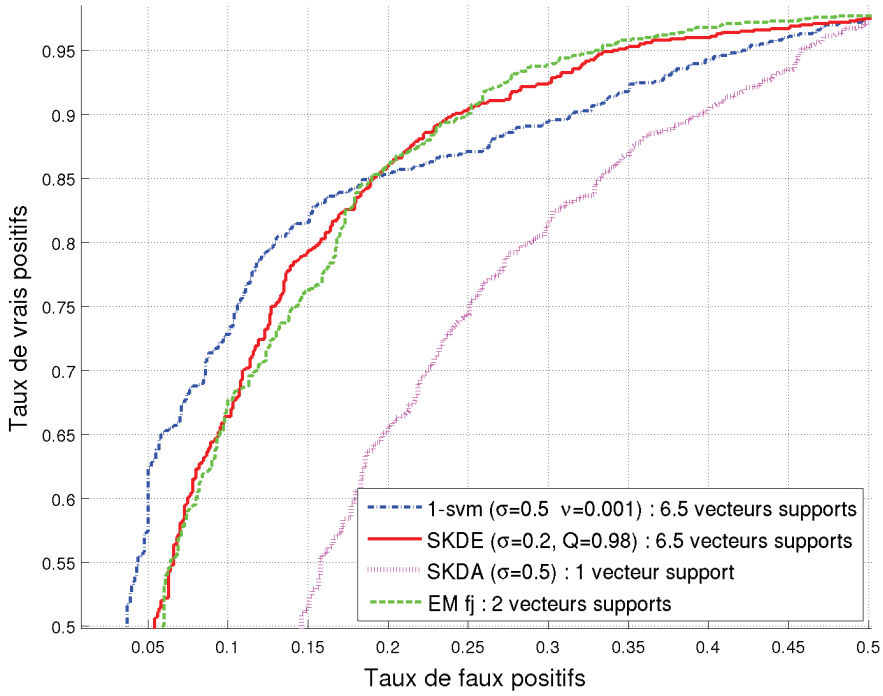


FIGURE 3.8 – Courbes ROC sur la base d'apprentissage B_{ripley}. Le nombre moyen de vecteurs conservés pour les *One Class SVM*est de 6.5, 6.5 pour notre algorithme, 1 pour le SKDA et 2 pour l'algorithme EM.

modèle. Nous verrons au chapitre 4, une ré-exploitation de ces observations caractéristiques dans le cadre très différent des réseaux bayésiens.

Comparatif La base B_{ripley} présentée dans la section précédente est réutilisée en effectuant l'apprentissage sur chacune des deux classes puis en les testant sur 1000 autres observations, une moitié de la première classe et l'autre de la seconde. L'implémentation utilisée pour l'algorithme du *One Class SVM* est celle de Rakotomamonjy [3].

La figure 3.8 représente les courbes obtenues avec les meilleurs jeux de paramètres pour chacune des méthodes. On constate, tout d'abord, que notre modèle obtient des résultats comparables à ceux obtenus grâce à l'algorithme EM, qui est l'algorithme le plus précis en termes d'estimation de densité. En revanche, notre méthode est légèrement inférieure au *One Class SVM* en termes de classification pour un nombre vecteurs support moyen égal. Le SKDA en revanche montre des résultats bien inférieurs.

Une représentation 2D de la base B_{ripley} est fournie à la figure 3.9. Les frontières de classification pour chacune des méthodes (à l'exception du SKDA qui n'a pas été représenté en raison de la mauvaise qualité de ses résultats) sont aussi représentées en trait plein. Ces frontières sont celles obtenues pour les seuils de détection de chacune des méthodes donnant le meilleur rapport « taux de bonnes détections / taux de fausses alarmes » (point le plus proche du point optimal $(0, 1)$ sur la courbe ROC). On constate que les trois méthodes englobent plutôt bien les observations d'apprentissage représentées par une étoile « * ». Mais surtout, le point important à remarquer ici, est la position des vecteurs supports du *One Class SVM* et des points de contrôle (ou moyenne de gaussienne) de la méthode

3. http ://asi.insa-rouen.fr/enseignants/∼arakotom/toolbox/index.html

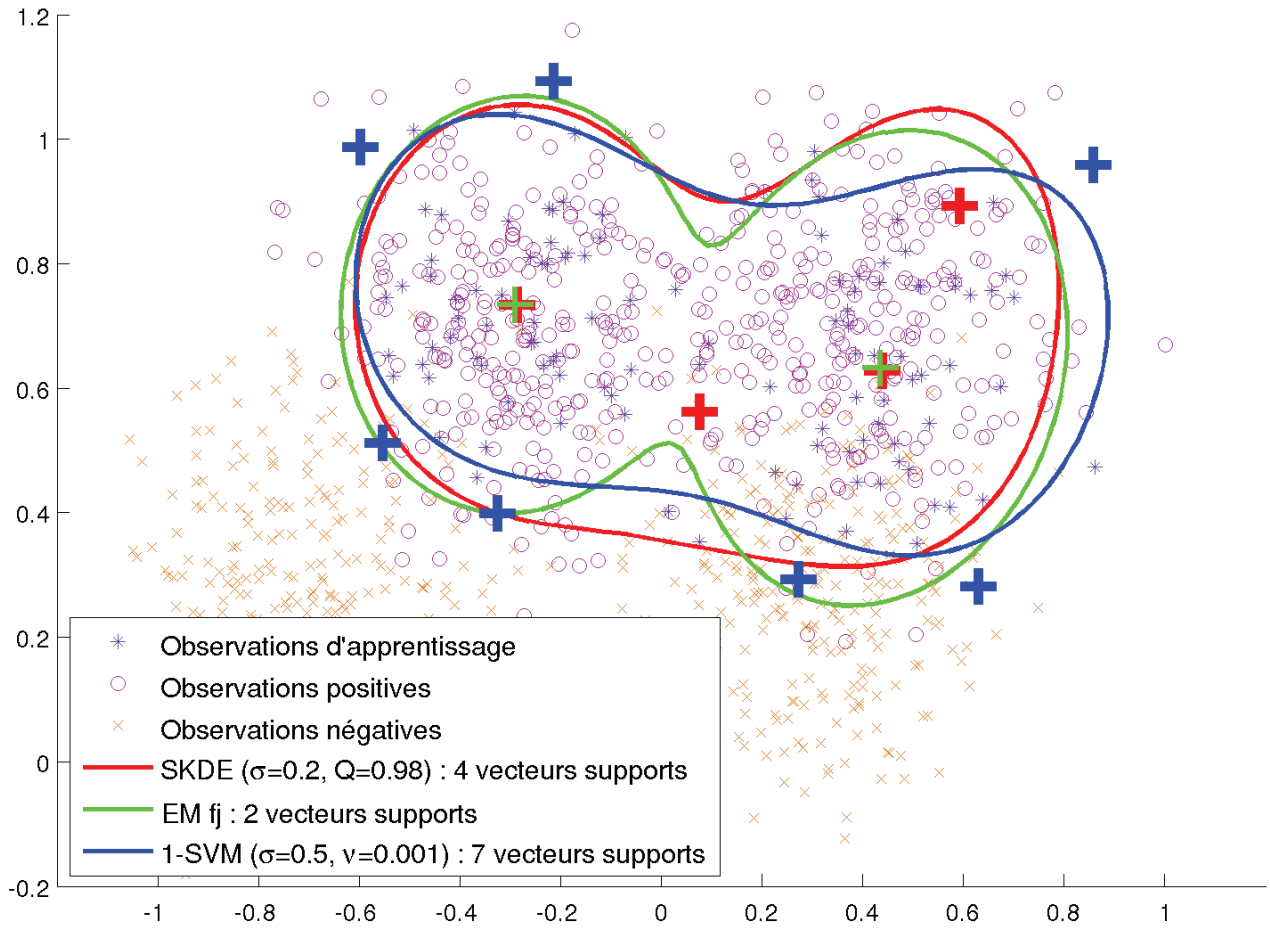


FIGURE 3.9 – Représentation 2D de la base B_{ripley}. Les frontières de classification des différentes méthodes sont représentées en trait plein. Le nombre de vecteurs conservés pour l'apprentissage d'une des deux classes est de 7 vecteurs pour les *One Class SVM*, 4 pour le SKDE et 2 pour l'algorithme EM.

SKDE ainsi que de l'algorithme EM, représentés par des « + » de couleurs correspondant à chacune des méthodes. Dans le cas du SKDE et de l'algorithme EM, les points de contrôle sont placés au milieu des observations d'apprentissage alors que les vecteurs supports du *One Class SVM* sont disposés tout autour de la frontière de classification.

Bien que légèrement moins performant en terme de classification que le *One Class SVM*, la représentation des estimations bayésiennes est plus facile à ré-exploiter. En effet, chacun des points de contrôle représente un état possible du modèle et est exploitable indépendamment des autres, ce qui n'est pas le cas des vecteurs supports des *One Class SVM* où l'ensemble des vecteurs sont nécessaires de manière indissociables pour déterminer la frontière de décision.

Cette propriété n'est pas anodine dans notre cas, aux vues de la mise en œuvre du classifieur décrite au chapitre 4.

Concernant le temps d'apprentissages de chacune des méthodes, un graphe semilog montre l'évolution du temps d'approximation selon la taille de l'ensemble d'observations sur figure 3.10. Comme établi à la section précédente, le SKDA est la méthode la plus rapide avec sa complexité linéaire. Le SKDE, que ce soit dans sa version initiale ou locale, est la seconde machine d'apprentissage la plus rapide. La différence entre les deux versions est infime car comme dit auparavant, la majorité

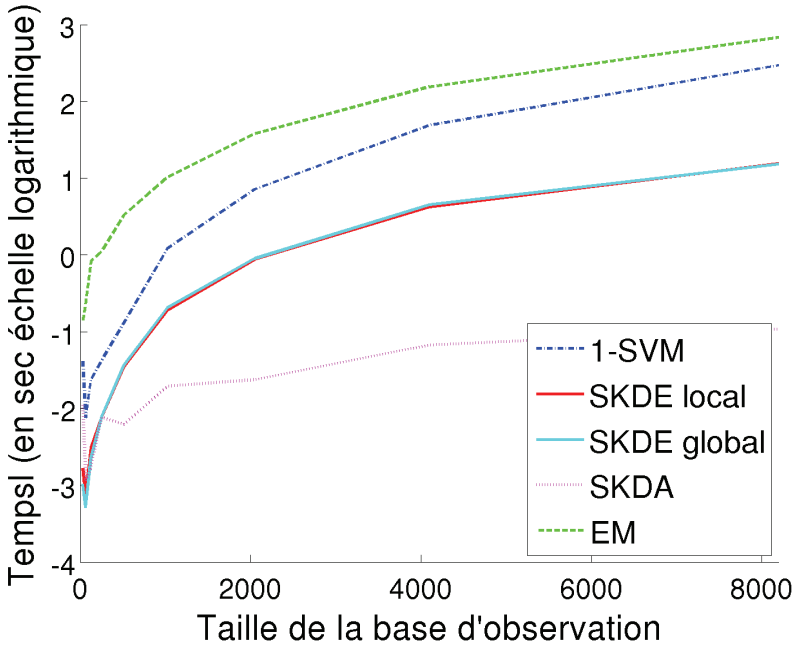


FIGURE 3.10 – Evolution du temps d'apprentissage selon la taille de l'ensemble d'observations, échelle semilog.

du temps de calcul est consacrée au calcul de la matrice Φ. L'algorithme EM est le plus lent mais possède une évolution similaire à celui du SKDE contrairement au *One Class SVM* qui possède une complexité légèrement supérieure puisque l'on s'aperçoit qu'avec l'augmentation de la taille de la base d'apprentissage, la croissance du temps d'apprentissage est plus rapide que pour les autres algorithmes.

Les temps d'exécution sont quant à eux à peu près équivalents puisque dans tous les cas ils correspondent au calcul d'une somme pondérée de gaussiennes. C'est donc le nombre de gaussiennes qui influe sur le temps de calcul mais au vu du faible nombre de gaussiennes conservées pour chaque méthode (vecteurs support dans le cas de *One Class SVM* ou point de contrôle pour l'approche proposée), les temps de calcul sont largement viables pour une application temps réel comme nous le verrons dans le chapitre 5.

3.5 Conclusion

Au travers de ce chapitre, une étude des approches pour l'estimation de densité a été réalisée. Parmi les différentes méthodes passées en revue, seules les estimations bayésiennes non paramétriques du KDE et des KPPV utilisent des distances particulières entre deux descripteurs. Cependant, ces approches n'étant pas des modèles viables pour une utilisation temps réel, un nouvel algorithme baptisé SKDE a été proposé. Cet algorithme qui est une estimation parcimonieuse du KDE permet une exploitation du modèle en temps réel qui est impérative dans notre cadre applicatif.

Le SKDE étant une méthode d'estimation de densité, nous l'avons comparée aux autres méthodes classiques d'approximation de densité. La méthode proposée a montré une qualité d'approximation très largement comparable aux autres méthodes dans des temps de calcul raisonnables et avec une représentation du modèle concise.

La méthode proposée dans ce chapitre ne se limite pas à la classification. Elle peut être utilisée dans de nombreux contextes (soustraction de fond, suivi de personne, etc), puisqu'elle ne nécessite que des observations d'une distribution réelle *a priori* inconnue et d'une distance entre deux de ces observations (une distance euclidienne ou de Mahalanobis à défaut d'une distance ad hoc) et ne fait aucun *a priori* sur la forme de la distribution réelle. Elle est donc plus souple à utiliser que des algorithmes traditionnels tels que l'algorithme EM.

En terme de classification cependant, puisque c'est ce dont il est question dans ce manuscrit, le SKDE couplé avec une fonction de décision basée sur un critère de confiance, présente par rapport aux approches probabilistes des résultats très convaincants. Une comparaison avec la méthode référence pour ce genre de problème, à savoir le *One Class SVM*, montre que les résultats restent acceptables.

Néanmoins, le SKDE présente une propriété très intéressante par rapport au *One Class SVM*. Cette propriété est de pouvoir exploiter les « points de contrôle » de manière individuelle car ceux-ci sont significatifs d'une observation représentative d'un état du modèle. Les vecteurs supports du *One Class SVM* sont, en revanche, indissociables les uns des autres puisqu'ils définissent à eux tous une frontière de classification unique. Cette propriété va être utilisée dans le chapitre 4 pour la mise en place de réseaux bayésiens de type chaîne de Markov cachée comme alternative à la classification du mouvement image par image.

Chapitre 4

Classification par réseaux bayésiens

Dans ce chapitre, le processus de classification n'est plus réalisé image par image mais sur des séquences d'observations permettant de modéliser la séquentialité de celles-ci. Réalisées par le biais de réseaux bayésiens et plus particulièrement la famille des chaînes de Markov Cachées, la première partie de ce chapitre rappelle le principe de ces chaînes et leur utilisation dans le cadre de notre application afin d'ajouter une cohérence séquentielle au mouvement. La seconde partie aborde l'utilisation d'une extension de ces chaînes de Markov permettant d'en coupler plusieurs. Cette approche ayant pour but d'ajouter de la cohérence spatiale à la classification du mouvement. Les travaux réalisés sur ces réseaux bayésiens ont été publiés dans Luvison *et al.* (2011).

4.1 Introduction

Les réseaux bayésiens sont des modèles graphiques probabilistes qui représentent des variables aléatoires et leur dépendance, grâce à des graphes. Cette famille de méthode est très vaste mais dans le cas où ces réseaux modélisent des séquences de variables telles que des séries temporelles, on parle alors de réseaux bayésiens dynamiques.

Ici, nous nous intéressons à l'une des spécifications des réseaux bayésiens dynamiques que sont les chaînes de Markov cachées. De telles machines d'apprentissage permettent de modéliser la séquentialité des observations. Dans le cas de flux laminaires, dont il est question dans ces travaux, la continuité du mouvement dans le temps est une caractéristique essentielle qu'il semble naturelle de vouloir modéliser. Cette modélisation de la continuité temporelle vient compléter la modélisation des structures spatio-temporelles. Bien qu'en partie redondante, cette modélisation et en particulier la théorie qu'il y a derrière, est étudiée en vue de l'ajout d'une cohérence spatiale du mouvement avec le voisinage direct. En effet, une telle cohérence spatiale est réalisée grâce à une extension des chaînes de Markov cachées qui permet de les coupler.

L'idée est donc d'avoir en chaque bloc de la grille d'analyse de la scène, une chaîne de Markov pour modéliser la séquentialité normale des observations en ce bloc. Par la suite, une modélisation plus complexe de la séquentialité des mouvements non plus d'un seul bloc mais d'une clique de

blocs [1] sera mise en place via un mécanisme de couplage de chaîne de Markov que nous étudierons en deuxième partie de ce chapitre après avoir rappelé le fonctionnement des chaînes de Markov cachées classiques.

4.2 Les chaînes de Markov cachées

Les modèles de Markov sont des automates probabilistes à états finis. Tous ces modèles se basent sur l'hypothèse de Markov d'ordre 1, qui signifie que l'état futur à l'instant $t+1$ ne dépend que de l'état à l'instant t, autrement dit, l'état courant se suffit à lui-même pour déterminer l'état suivant.

Cette hypothèse implique une description très fine du modèle pour permettre une prédiction parfaite du comportement du système. Dans la majorité des situations réelles, une telle description n'est pas réalisable en raison du trop grand nombre de paramètres. Cependant, en pratique en restreignant les hypothèses et les contraintes applicables au modèle, on peut se ramener à un cadre où la propriété Markovienne est presque respectée.

Le modèle Markovien de base correspond à un simple graphe d'états, doté d'une fonction de transition probabiliste. A chaque pas de temps, le modèle subit une transition qui va potentiellement modifier son état. Cette transition permet donc au système modélisé d'évoluer, selon une loi connue par avance. Néanmoins, cette loi de transition est probabiliste. En effet, l'évolution du système peut être incertaine, ou simplement mal connue. Cette fonction probabiliste permet donc d'exprimer simplement la loi d'évolution du modèle, sous la forme d'une matrice de probabilités.

4.2.1 Théorie

Dans le cas des modèles de Markov cachés (HMM), le processus stochastique définissant les états successifs du système n'est plus directement observable, il est caché par un autre processus stochastique, un processus d'observation. Ces deux processus sont liés entre eux encore une fois de manière probabiliste (Rabiner (1990)).

Plus formellement, soit $S = \{S_1, S_2, \ldots, S_N\}$ un ensemble d'états cachés, V un ensemble de M symboles discrets $v_1, v_2, \ldots, v_M$ représentant les états observables du système et x_k une variable aléatoire associée à l'état S_k qui prend ses valeurs dans V. On notera q_t l'état et O_t l'observation à l'instant t. Un modèle de Markov Caché peut être représenté par $\lambda = (\boldsymbol{\pi}, A, B)$, avec :

- $\boldsymbol{\pi}$ la distribution de probabilité des états initiaux : $\pi_i = P(q_1 = S_i)$, avec $1 \leq i \leq N$
- A la matrice de transition contenant la distribution de probabilité des transitions entre états : $A = \{a_{ij}\}$ avec

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), \quad 1 \leq i, j \leq N \tag{4.1}$$

- B le vecteur des distributions de probabilité d'émission des observations pour chaque état $B = b_j(.), \quad 1 \leq j \leq N$:
  - dans le cas discret,

$$b_j(v_k) = P(O_t = v_k | q_t = S_j), \quad 1 \leq k \leq M \tag{4.2}$$

1. ensemble de blocs voisins

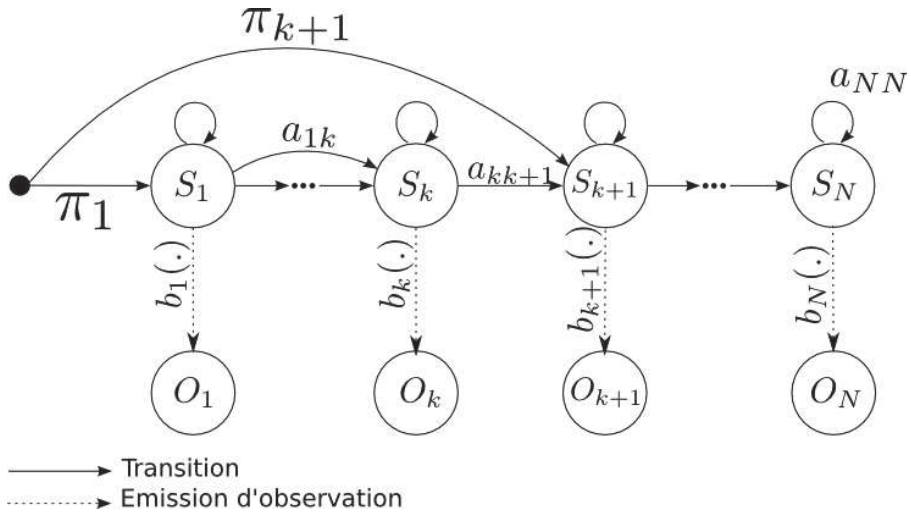


FIGURE 4.1 – Schéma déployé d'un HMM.

- dans le cas continu,

$$b_j(\mathbf{O_t}) = P(\mathbf{O_t}|q_t = S_j) = \sum_{m=1}^{M} w_{jm}\mathcal{N}(\mathbf{O_t}, \boldsymbol{\mu}_{jm}, \Sigma_{jm}), \tag{4.3}$$

avec $\mathcal{N}$ une densité de type gaussienne de moyenne $\boldsymbol{\mu}_{jm}$ et de covariance Σ_{jm}. Ces distributions doivent conserver la contrainte stochastique

$$\int_{-\infty}^{+\infty} b_j(x)dx = 1.$$

Les trois grands problèmes identifiés par Rabiner (1990) et qui constituent la théorie des HMMs sont :

1. Évaluer une séquence d'observations, $P(O|\lambda)$
2. Déterminer les états sous-jacents à partir d'une séquence d'observations. La solution de ce problème ne possède pas de solution exacte. Deux critères d'optimisation sont possibles. Le premier cherche à maximiser individuellement la vraisemblance de chaque état. Le second critère cherche à maximiser la vraisemblance de la séquence d'état dans son ensemble.
3. Mettre à jour les paramètres du modèle pour maximiser la vraisemblance d'une séquence d'observations. Maximiser $P(O|\lambda)$ en ajustant λ. Lorsque l'on veut mettre à jour la distribution d'émission d'un état ou les transitions sortant de cet état, il est nécessaire de savoir si l'on passe souvent par cet état ou ces transitions. Répondre à ce problème revient en fait à répondre au problème 2.

La résolution de ces trois problèmes s'effectue essentiellement grâce à deux algorithmes de parcours de la chaîne, l'algorithme *Forward-Backward* et l'algorithme de Viterbi, dont nous allons voir les grandes lignes.

4.2.1.1 Les algorithmes de parcours

Forward-Backward Baum (1970), en développant cet algorithme, s'est focalisé sur la résolution de deux problèmes très proches l'un de l'autre, à savoir :

▷ Quelle est la probabilité d'être dans l'état S_i à l'instant t en ayant observé $O_1O_2 \dots O_t$.
▷ Quelle est la probabilité d'être dans l'état S_i à l'instant t sachant que les observations à venir sont $O_{t+1}O_{t+2} \dots O_T$.

Pour répondre au premier problème, soit $\alpha_t(i)$ la probabilité d'être à l'état S_i à l'instant t sachant que la séquence $O_1O_2 \dots O_t$ a déjà été vue jusqu'à présent :

$$\alpha_t(i) = P(O_1O_2 \dots O_t, q_t = S_i|\lambda) \tag{4.4}$$

Cette probabilité est calculée itérativement en l'initialisant à l'aide des probabilités d'émission de la première observation $b_i(O_1)$ ainsi que des probabilités des états initiaux :

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \tag{4.5}$$

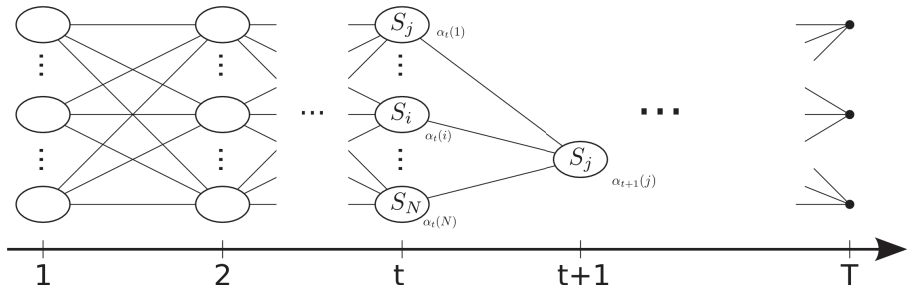


FIGURE 4.2 – Treillis de parcours pour l'algorithme *Forward-Backward*.

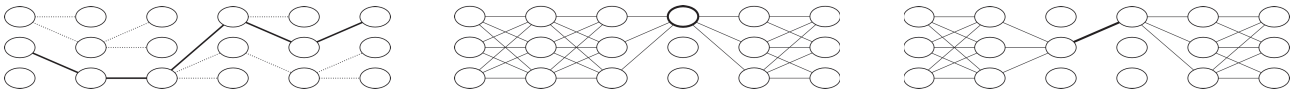
L'étape récursive du calcul de α repose sur la programmation dynamique. Connaissant les valeurs des variables α à l'instant t, il est possible d'en déduire celles à l'instant $t+1$ grâce au treillis représenté à la figure 4.2. Les colonnes du treillis représentent les états et l'axe des abscisses l'évolution temporelle. Ce treillis permet de collecter l'information combinée de tous les chemins partageant le même état au même instant dans la variable α. Cette information combinée est un « condensé » de l'information contenue dans l'ensemble des N^T chemins qui est obtenue en ne parcourant que N têtes de chemin. Il suffit simplement de passer à chaque instant, une fois par chaque état et chaque transition, d'où une complexité algorithmique de $O(TN^2)$. Cette relation d'inférence représentée sur la figure 4.2, peut-être explicitée comme suit : pour un état S_j à un instant $t+1$, ayant une probabilité d'émission $b_j(O_{t+1})$, la probabilité d'être dans l'état S_j en ayant observé $O_1O_2 \dots O_{t+1}$ est obtenue en sommant sur tous les états S_i, la probabilité d'être dans l'état S_i ayant observé $O_1O_2 \dots O_t$ puis en passant par la transition a_{ij} :

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1 \; et \; 1 \leq j \leq N \tag{4.6}$$

Pour ce qui est du second problème, le même procédé est utilisé mais en partant de la dernière observation. Soit $\beta_t(i)$ la probabilité d'être à l'état S_i à l'instant t avec $O_{t+1}O_{t+2} \dots O_T$ les observations à voir pour la fin de la séquence. La valeur initiale de cette probabilité est égale à :

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{4.7}$$

Pour l'étape d'induction, le parcours est inversé en sommant sur tous les états S_j, la probabilité d'être dans l'état S_j sachant que les observations à venir sont $O_{t+1}, O_{t+2}, \dots, O_T$, en passant par la transition a_{ij} sous la probabilité d'émission $b_j(O_{t+1})$:



(a) Viterbi (b) $P(q_t = S_i|O, \lambda)$ (c) $P(q_t = S_i, q_{t+1} = S_j|O, \lambda)$

FIGURE 4.3 – Parcours possibles aux travers du treillis. (a) Chemin le plus vraisemblable ; (b) Probabilité d'un état ; (c) Probabilité d'une transition

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \;\; 1 \leq t \leq T-1 \; et \; 1 \leq i \leq N \tag{4.8}$$

Viterbi Cet algorithme (Fornay (1973)), cherche quant à lui à trouver la meilleure séquence d'état $Q = q_1 \; q_2 \; \ldots \; q_T$ pour une séquence d'observations donnée $O = O_1 O_2 \ldots O_T$ comme illustré sur la figure 4.3(a). Pour cela, la variable δ est définie par :

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} P(q_1, q_2, \ldots, q_t = i, O_1 O_2 \ldots O_t|\lambda) \tag{4.9}$$

Cette variable définit ainsi le chemin le plus probable au temps t finissant à l'état i avec comme observation $O_1 O_2 \ldots O_t$. Pour la déterminer, à l'instar de l'algorithme *Forward-Backward*, celle-ci est calculée récursivement. L'initialisation est donnée par :

$$\begin{aligned} \delta_1(i) &= \pi_i b_i(O_1), \quad 1 \leq i \leq N \\ \psi_1(i) &= 0 \end{aligned} \tag{4.10}$$

avec ψ la variable permettant de mémoriser le meilleur chemin choisi. L'étape récursive est quant à elle définie par :

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t), \;\; 2 \leq t \leq T-1 \; et \; 1 \leq j \leq N \\ \psi_t(j) &= \underset{1 \leq i \leq N}{argmax} \left[\delta_{t-1}(i) a_{ij}\right], \;\; 2 \leq t \leq T-1 \; et \; 1 \leq j \leq N \end{aligned} \tag{4.11}$$

Ce qui permet d'aboutir à la séquence optimale se terminant par l'état $q_T^* = \underset{1 \leq i \leq N}{argmax} \left[\delta_T(i)\right]$ et ainsi remonter l'ensemble du chemin optimal par la relation $q_t^* = \psi_{t+1}(q_{t+1}^*)$ pour $t = T-1, \; T-2, \; \ldots, \; 1$

4.2.1.2 Résolution des trois problèmes

Problème numéro 1 Évaluer une séquence d'observations. L'algorithme *Forward-Backward* permet entre autre de répondre au premier problème énoncé par Rabiner (1990), à savoir évaluer $P(O|\lambda)$. En effet, si l'on suppose la séquence des états cachés connue $Q = q_1 \; q_2 \; \ldots \; q_T$ alors $P(O|Q, \lambda)$ est égale à :

$$P(O|Q, \lambda) = \prod_{t=1}^{T} P(O|q_t, \lambda) = \prod_{t=1}^{T} b_{q_t}(O_t) \tag{4.12}$$

Comme la probabilité d'obtenir la séquence d'observations Q est égale à :

$$P(Q|\lambda) = \pi_{q_1} \prod_{t=1}^{T-1} a_{q_t q_{t+1}} \tag{4.13}$$

La probabilité d'observer une séquence d'observations O sachant simplement le modèle λ est donc la somme, sur toutes les séquences d'états Q possibles, de la loi jointe $P(O, Q|\lambda) = P(O|Q, \lambda)P(Q|\lambda)$:

$$\begin{aligned} P(O|\lambda) &= \sum_{Q} P(O|Q, \lambda)P(Q|\lambda) \\ &= \sum_{Q=q_1\ q_2\ \dots\ q_T} \left[\prod_{t=1}^{T-1} a_{q_t q_{t+1}} b_{q_t}(O_t)\ b_{q_T}(O_T) \right] \\ P(O|\lambda) &= \sum_{i=1}^{N} \alpha_T(i) \end{aligned} \tag{4.14}$$

Cette probabilité s'exprime très simplement grâce à la variable α. De plus la programmation dynamique permet de calculer cette probabilité de manière efficace sans un parcourt exhaustif de tous les chemins qui est en pratique irréalisable.

A noter que l'évaluation de cette probabilité sert lors de la phase d'entraînement de la chaîne comme nous le verrons dans les deux prochains problèmes. Lors de l'utilisation de la chaîne, la vraisemblance de la séquence d'observations est traditionnellement calculée avec l'algorithme de Viterbi qui est plus rapide. Cette vraisemblance, que nous noterons P^*, est donc égale à :

$$P^* = \max_{1 \leq i \leq N} \delta_T(i) \tag{4.15}$$

Problème numéro 2 Déterminer les états sous-jacents à partir d'une séquence d'observations. Ce problème, comme énoncé, ne possède pas de solution unique. Tout dépend du critère d'optimisation choisi.

Le critère maximisant individuellement la vraisemblance de chaque état, permet de déterminer, pour une séquence donnée, les états sous-jacents les plus vraisemblables à chaque instant. Ces états sont déterminés independamment les uns des autres, sans réellement tenir compte du chemin parcouru. Pour répondre à cette optimisation, soit $\gamma_t(i) = P(q_t = S_i|O, \lambda)$ la probabilité d'être dans l'état S_i à l'instant t avec la séquence d'observations O. Cette probabilité s'exprime très facilement grâce aux variables *Forward-Backward* comme on peut s'en apercevoir sur la figure 4.3(b) :

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} \tag{4.16}$$

Le dénominateur $P(O|\lambda)$ est aussi égal à $\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$. Ce terme permet de normaliser et d'assurer que $\sum_{i=1}^{N} \gamma_t(i) = 1$.

L'état le plus vraisemblable à l'instant t est donc :

$$q_t = \underset{1 \leq i \leq N}{argmax} \left[\gamma_t(i)\right], \quad 1 \leq t \leq T \tag{4.17}$$

Un autre critère consiste à trouver le chemin complet permettant de maximiser la vraisemblance de l'ensemble de la séquence O. C'est précisément à ce critère d'optimisation que l'algorithme de Viterbi répond.

Problème numéro 3 Mettre à jour les paramètres du modèle pour maximiser la vraisemblance d'une séquence d'observations. Ce problème vise à optimiser le jeu de paramètres λ afin de maximiser $P(O|\lambda)$. Il repose sur la connaissance des sous-états les plus représentatifs de la séquence O afin d'adapter λ en conséquence. Connaître ces sous-états revient à résoudre le problème 2. Il existe donc plusieurs manières de mettre à jour le modèle. La plus classique, celle de Baum-Welch s'appuie sur l'optimisation par état. Intuitivement, ceci consisterait à comptabiliser le nombre de passage par chaque transition puis de normaliser les états en s'assurant que la somme des transitions sortantes soit égale à 1.

En pratique, toutes les transitions sont parcourables selon une probabilité donnée. En conséquence, pour une transition particulière, c'est la somme des probabilités de franchissement de cette transition à chaque instant de la séquence d'observations qui est calculée puis normalisée. Soit $\xi_t(i,j)$ la probabilité d'être à l'instant t dans l'état S_i et à l'instant $t+1$ dans l'état S_j :

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j|O,\lambda) \tag{4.18}$$

De la même manière que pour la variable γ, il est possible d'exprimer la probabilité ξ à partir des variables α et β comme illustré sur la figure 4.3(c). Plus formellement, on obtient :

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)} \tag{4.19}$$

Cette formulation traduit le fait d'être dans l'état S_i à l'instant t sous les observations $O_1O_2\dots O_t$ (ce qui est égal à la probabilité $\alpha_t(i)$), de franchir la transition a_{ij}, d'observer à l'instant suivant O_{t+1} depuis l'état S_j sachant qu'il reste les observations $O_{t+1}O_{t+2}\dots O_T$ (ce qui est égal à $\beta_{t+1}(j)$). Le dénominateur est encore une fois un terme de normalisation pour assurer ξ d'être une probabilité. A noter qu'il existe une relation entre ξ et γ, la probabilité d'être dans l'état S_i à l'instant t, en sommant sur l'ensemble des transitions sortantes de S_i les probabilités ξ :

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j) \tag{4.20}$$

Il est ainsi possible de déterminer une nouvelle estimation de la matrice de de transition A dont le terme $\overline{a_{ij}}$ est défini par :

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{4.21}$$

Les probabilités des états initiaux sont quant elles, mises-à-jour grâce à la probabilité d'être dans chaque état S_i à l'instant initial ($t=1$), à savoir $\gamma_1(i)$. Concernant la mise-à-jour des densités d'émission, selon l'utilisation de densités discrètes ou continues, la réestimation est légèrement différente. A l'image de la mise-à-jour des probabilités, l'idée est de comptabiliser le nombre de fois où le symbole k à été observé dans l'état S_j. En pratique, c'est la probabilité de passer dans cet état, à savoir $\gamma_t(i)$, lorsque l'observation courante représente le symbole k qui est comptabilisé.

Dans le cas discret, $b_j(v_k)$ est donc égal à :

$$\overline{b_j}(v_k) = \frac{\sum_{t=1}^{T-1}\delta_{v_k}^{O_t}\gamma_t(i)}{\sum_{t=1}^{T-1}\gamma_t(i)} \tag{4.22}$$

Avec $\delta_{v_k}^{O_t}$ le symbole de kronecker qui vaut 1 lorsque la $t^{ième}$ observation est le symbole v_k. Le dénominateur est à nouveau ici un terme de normalisation.

Dans le cas de distributions continues, la phase de mise-à-jour est assez similaire, à la différence que chaque composante du mélange de gaussiennes est réévaluée séparément en fonction de l'importance de cette gaussienne dans la vraisemblance de l'observation O_t. La probabilité $\gamma_t(i)$ laisse donc place à :

$$\gamma_t(j,k) = \gamma_t(i)\frac{w_{jk}\mathcal{N}(O_t, \boldsymbol{\mu}_{jk}, \Sigma_{jk})}{\sum_{m=1}^{M} w_{jm}\mathcal{N}(O_t, \boldsymbol{\mu}_{jm}, \Sigma_{jm})} \tag{4.23}$$

La réestimation des paramètres de la densité est ensuite :

$$\overline{w_{jk}} = \frac{\sum_{t=1}^{T-1}\gamma_t(j,k)}{\sum_{t=1}^{T-1}\sum_{k=1}^{M}\gamma_t(j,k)} \tag{4.24}$$

$$\overline{\boldsymbol{\mu}_{jk}} = \frac{\sum_{t=1}^{T-1}\gamma_t(j,k).\mathbf{O_t}}{\sum_{t=1}^{T-1}\sum_{k=1}^{M}\gamma_t(j,k)} \tag{4.25}$$

$$\overline{\Sigma_{jk}} = \frac{\sum_{t=1}^{T-1}\gamma_t(j,k).(\mathbf{O_t}-\boldsymbol{\mu}_{jk})(\mathbf{O_t}-\boldsymbol{\mu}_{jk})^T}{\sum_{t=1}^{T-1}\sum_{k=1}^{M}\gamma_t(j,k)} \tag{4.26}$$

L'autre manière de mettre à jour le modèle, est de favoriser uniquement les transitions faisant partie du chemin optimal. Pour cela, le chemin optimal est obtenu par l'algorithme de Viterbi comme énoncé pour répondre au problème numéro 2. Chaque passage par l'une des transitions ou l'un des états le long de ce chemin sera ensuite comptabilisé puis normalisé pour assurer la contrainte de probabilité. Cette optimisation n'assure pas le parcours et la mise à jour de l'ensemble de l'automate contrairement à l'approche de Baum-Welch, ce qui peut être dangereux en particulier si l'ensemble d'apprentissage est restreint. Cependant elle permet de réaliser un apprentissage rapide en se servant de l'automate comme d'une « mémoire » des chemins déjà vus.

Ce type d'apprentissage, assez peu fréquent, est néanmoins rencontré dans la littérature sous le nom de *Viterbi Path Counting* Liu et Lovell (2003). De plus certaines librairies, notamment celle que nous utilisons, à savoir la librairie Torch [2], mettent à disposition ce type d'apprentissage.

Adaptation de la théorie La théorie des HMMs telle que présentée par Rabiner (1990) n'est en pratique pas appliquée telle quelle. Ce sont des *Multi Observation HMMs* qui sont utilisées en pratique (Rabiner (1990)), car une seule séquence d'observations ne suffit pas en général pour représenter l'ensemble des séquentialités possibles et donc de mettre à jour correctement l'ensemble des probabilités d'émission et de transition. L'utilisation de plusieurs séquences d'observations implique de normaliser par séquence les fréquences d'occurrence qui servait à la réesstimation des paramètres lors de la phase d'apprentissage. Pour la matrice de transition A par exemple, le numérateur de la formule (4.21) caractérise la fréquence d'occurrence de passer au travers de la transition a_{ij} pour la séquence O alors que le dénominateur est un terme de normalisation. Lors d'un apprentissage à partir de K séquences $\mathbf{O} = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}$ avec $O^{(k)} = O_1^{(k)}O_2^{(k)}\ldots O_{T_k}^{(k)}$, les fréquences d'occurrence doivent être sommées sur chacune des séquences.

$$\overline{a_{ij}} = \frac{\sum_{k=1}^{K}\sum_{t=1}^{T-1}\xi_{t^{(k)}}(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T-1}\gamma_{t^{(k)}}(i)(i)} \tag{4.27}$$

2. http ://www.torch.ch/

avec

$$\xi_{t^{(k)}}(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O^{(k)}_{t+1})\beta_{t+1}(j)}{P(O^{(k)}|\lambda)} \tag{4.28}$$

et

$$\gamma_{t^{(k)}}(i,j) = \frac{\alpha_t(i)\beta_t(i)}{P(O^{(k)}|\lambda)} \tag{4.29}$$

en notant $P_k = P(O^{(k)}|\lambda)$, on obtient la réestimation de la matrice de transition :

$$\overline{a_{ij}} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T-1} \alpha_t(i)a_{ij}b_j(O^{(k)}_{t+1})\beta_{t+1}(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \tag{4.30}$$

La réestimation des autres paramètres s'effectue de la même manière en normalisant par P_k les fréquences d'occurrence utilisées avant de les sommer entre elles.

A noter aussi qu'en pratique les différents algorithmes mentionnés ci-dessus sont habituellement implémentés en échelle logarithmique. Les probabilités manipulées par les HMMs sont en effet généralement très faibles à cause des multiples produits de probabilité intervenant dans ces algorithmes. Dans le cadre ces travaux, l'implémentation des HMMs utilisée est celle de la librairie Torch. Cette librairie a été choisie d'une part car elle est couramment utilisée dans notre laboratoire et d'autre part car son implémentation relativement efficace était suffisante pour tenir les contraintes temps réelles que nous nous sommes fixés.

4.2.2 Vers l'implémentation pour la vidéosurveillance

Après avoir brièvement présenté la théorie des HMMs ainsi que les différents outils disponibles pour les manipuler, nous allons maintenant voir l'utilisation faite dans le cadre de notre système.

Toujours centrée sur l'architecture énoncée au chapitre 1, notre analyse via les HMMs s'effectue par bloc. En chaque bloc de la scène, une HMM est entrainée à partir de séquences de descripteurs caractérisant le mouvement. Ces séquences sont constituées des descripteurs calculés consécutivement sur chacune des images de la vidéo.

Différents problèmes sont cependant à résoudre pour l'intégration dans le cadre que nous nous sommes fixé. Nous allons détailler chacun de ces problèmes avant de proposer différentes solutions ayant chacunes leurs avantages et leurs inconvénients.

4.2.2.1 Les problèmes d'implémentation

Traditionnellement, le nombre d'état d'une HMM est déterminé par l'utilisateur. Ici une chaîne est utilisée par bloc, le nombre de bloc par scène étant généralement très important (à titre indicatif, une vidéo en 320x240 pixels contient 300 blocs de taille 16x16), il est donc nécessaire de déterminer de manière automatique le nombre d'état N par chaîne. Ce paramètre n'est normalement pas corrélé au nombre d'états observables M. Il serait plutôt influencé par la longueur des chaînes d'observations étudiées.

Ces longueurs de séquence sont par ailleurs un autre problème en soi puisque dans notre cas elles ne sont pas fixes. Selon la densité de la foule, la présence de mouvement en un bloc peut être ininter-

rompue ou au contraire régulièrement interrompue par l'absence de mouvement, sorte d'équivalent au blanc en reconnaissance de la parole.

Un autre problème pour la mise en place des HMMs, concerne les densités de probabilité des émissions à choisir. Utiliser une densité discrète sous-entend être capable de discrétiser le descripteur caractérisant le mouvement, ce qui n'est pas systématiquement le cas si l'on considère des descripteurs tels que ceux de Shechtman et Irani (2005) ou de Kratz et Nishino (2009). A l'inverse, l'utilisation d'une densité continue avec les différents descripteurs vu au chapitre 2 quitte le cadre classique des mélanges de gaussiennes qui utilise la distance de Mahalanobis. L'utilisation de distances particulières pour chaque descripteur oblige à exploiter des mélanges de gaussiennes adaptées, comme vues au chapitre 3. Cependant avec ces distributions, la phase de mise-à-jour de la densité d'émission théorique telle que présentée (dans le paragraphe 4.2.1) ne s'applique plus.

Le dernier problème à résoudre pour l'utilisation des HMMs dans notre cadre applicatif, concerne l'exploitation de la vraisemblance $P(O|\lambda)$ en phase de test. Habituellement, une HMM est entrainée pour reconnaitre un modèle. Dans le cas de la reconnaissance de la parole par exemple, chaque mot aura une HMM qui lui sera associé et qui sera entrainée à partir de différentes prononciations possibles de ce mot. Lors de la phase de reconnaissance, une séquence d'observations sera évaluée au travers de toutes les HMM existantes et c'est la chaîne avec la vraisemblance la plus grande qui sera représentative du mot reconnu. Dans notre cas, pour un bloc donné seul deux modèles sont présents : le modèle normal et le modèle anormal. Cependant comme vu au chapitre 3, le modèle de l'anormal n'est ni définissable ni susceptible d'entraînement. Il faut donc être capable d'exploiter directement la vraisemblance obtenue en sortie de l'unique HMM entrainé sur un bloc et uniquement à partir d'observations du normal.

4.2.2.2 Les solutions et leurs conséquences

Les problèmes énoncés dans le paragraphe précédent sont assez liés entre eux. Essentiellement deux solutions ont été étudiées, avec des conséquences bien distinctes. L'une utilisant des densités d'émission continues issues de distributions basées noyaux et l'autre qui s'appuie sur des distributions discrètes.

Densités continues Cette approche utilisée par Kratz et Nishino (2009) consiste à utiliser des densités d'émission continues avec les distances particulières de chaque descripteur. Cette apprentissage nécessite une statistique préalable, semblable à notre méthode présentée au chapitre 3. A partir de cet statistique qui permet de déterminer les descripteurs les plus représentatifs, un état est associé à chacun de ces descripteurs. Habituellement le nombre d'état est lié à la longueur moyenne des observations. A titre d'exemple, en reconnaissance de la parole, une chaîne représentant le mot « quatre » possèdera moins d'état qu'une chaîne représentant le mot « cinquante » car le nombre de syllabe et le temps de prononciation de « quatre » est plus court que celui de « cinquante ». De plus la chaîne modélisant un mot vise essentiellement à apprendre les variations de prononciation de ce mot. Dans notre cas, le modèle de normalité que l'on cherche à modéliser peut lui-même être très varié comme dans des zones de croisement par exemple. Dans ce genre de cas deux séquences d'observations complètement différentes peuvent très bien toutes les deux représenter des mouvements normaux. Aussi la chaîne de Markov doit être susceptible de différencier toutes ces séquences de mouvement différentes, c'est pourquoi à défaut de trouver le nombre d'état optimal pour chaque bloc qui est un problème particulièrement difficile à résoudre, le nombre d'état est déterminé par cet apprentissage

statistique afin qu'il y ait au moins un état pour chaque point de contrôle (descripteur représentatif).

Rappelons que notre apprentissage statistique tel que présenté au chapitre 3, nous permet d'obtenir :

▷ N points de contrôle $\widetilde{\mathcal{Z}} = \{\widetilde{\mathbf{z_1}}, \widetilde{\mathbf{z_2}}, \ldots, \widetilde{\mathbf{z_N}}\}$.

▷ N noyaux gaussiens de variance fixe σ centrés sur chacun de ces points de contrôle.

▷ N poids associés à chacun de ces descripteurs $\tilde{\mathbf{w}} = \{\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_K\}$.

De la même manière que Kratz et Nishino (2009), à chaque point de contrôle $\widetilde{\mathbf{z_k}}$ un état S_k est associé ainsi qu'une probabilité d'émission égale au noyaux centré sur $\widetilde{\mathbf{z_k}}$:

$$b_k(O_t) = \sum_{m=1}^{K} \delta_m^k \mathcal{N}(O_t, \widetilde{\mathbf{z_m}}, \sigma) \tag{4.31}$$

avec δ_m^k le symbole de kronecker.

La probabilité d'émission pour un état est directement initialisée avec le noyau centré sur un descripteur attribuant à cet état sous-jacent le rôle de représentant de ce descripteur. Cette initialisation est particulièrement importante car elle ne sera pas mise à jour durant la phase d'apprentissage à cause de l'utilisation d'une distance particulière. Seule la matrice de transition et les probabilités initiales seront réestimées durant la phase d'apprentissage. De plus, à l'aide de cette représentation les poids $\tilde{\mathbf{w}}$ ne sont pas utilisés, ils apparaissent néanmoins dans la matrice de transition par l'importance du terme de fréquence de passage d'un descripteur par rapport à un autre.

Densités discrètes L'autre approche consiste à discrétiser le descripteur lorsque cela est possible. Dans le cas d'un vecteur déplacement issu du flot optique ou de notre descripteur CSD, une discrétisation angulaire est possible. En discrétisant en M symboles le descripteur, on peut alors procéder à l'apprentissage complet (probabilités d'émission comprises) de la chaîne. Pour le nombre d'états dans la chaîne, différentes solutions sont envisageables :

▷ Toujours grâce à un apprentissage statistique préalable, associer à chaque descripteur $\widetilde{\mathbf{z_k}}$, un état.

▷ Construire des chaînes de $N = M$ états. Cette solution permet de se dispenser ainsi de l'apprentissage préalable.

La première proposition n'est en pratique pas exploitée car combiner la restriction du nombre d'état et la rigidité d'un descripteur discret ne permet pas d'avoir des modèles représentant suffisamment la variabilité des mouvements. En effet, dans certains cas, d'autres descripteurs assez proches de $\widetilde{\mathbf{z_k}}$ s'expriment pourtant par le symbole d'une classe discrète adjacente, ce qui conduit à une non reconnaissance par la chaîne. En condition réelle, ce phénomène arrive très fréquemment, c'est pourquoi cette approche ne fera pas l'objet d'étude supplémentaire dans la suite de ce mémoire.

A noter que pour la version continue ainsi que pour la version discrète, il faut ajouter un état pour modéliser l'état de non-mouvement. Dans le cas de notre descripteur CSD, il faut aussi ajouter un état supplémentaire correspondant à l'état d'invalidité de notre descripteur. En effet, comme énoncé au chapitre 2, cet état d'indétermination apporte une information qui dans certains cas ne doit pas être négligée. De plus, cet état d'invalidation nous permet d'assurer des séquences d'observations de taille constante et donc d'avoir des vraisemblances $P(O|\lambda)$ en sortie de HMM possédant la même dynamique, c'est-à-dire le même nombre de transitions franchies.

Les autres problèmes La longueur des observations peut être extrêmement variable dans l'absolu, puisque dépendante de la densité de la foule. Pour pallier cette extrême variabilité qui rend le mécanisme d'apprentissage de HMM d'autant plus difficile, les séquences sont générées au travers d'une

fenêtre temporelle glissante de taille fixe. Dans le cas de non-mouvements, les séquences sont complétées par un descripteur caractérisant cet état. Nous avons mis en place un mécanisme de fenêtre glissante afin d'avoir une classification en chaque bloc, à chaque image, en vue d'une comparaison avec la classification mise en place au chapitre 3.

En revanche les séquences ne caractérisant que du non-mouvement ne sont pas considérées, ni pendant la phase d'apprentissage, ni pendant celle d'analyse. Celles-ci sont bien trop prédominantes sur de longues séquences vidéos, car en général les lieux de fort trafic, que ce soit routier ou de personnes, ne le sont que ponctuellement. Par exemple dans les couloirs de métro aux heures de pointe, la densité de personnes est oscillante au rythme de l'arrivée des rames, alternant foule dense et foule plus clairsemée. Ainsi en mettant de côté ces séquences de non-mouvement, l'analyse devient insensible au changement de densité du flux de personnes ou de véhicules.

Ces séquences pourraient être intéressantes dans le cas de l'analyse du non-mouvement comme avec des véhicules arrêtés en zone dangereuse par exemple. Cependant, il faut être capable de différencier un non-mouvement passager et donc éventuellement inhabituel, d'un non-mouvement permanent correspondant à un élément du décor. Cette distinction est d'autant plus importante pour des foules qui ne sont pas constamment denses, ce qui est en pratique le plus courant. Ce genre de distinction ferait appel à des algorithmes de soustraction de fond qui n'ont pas été intégrés à ces travaux.

Enfin, le dernier problème qui consiste à exploiter correctement la vraisemblance obtenue par la chaîne de Markov lors de la phase d'analyse a été résolu via la méthode de calcul de seuil automatique mis en place au chapitre 3. La chaîne de Markov, telle que nous l'utilisons, a pour but de modéliser la distribution réelle du déplacement. Une fois la chaîne entrainée, toute séquence d'observations d'un comportement normal peut être considérée comme un tirage aléatoire de la distribution réelle (inconnu) et donc par abus, de la modélisation représentée par la HMM. Ainsi, avec une base de validation suffisamment importante, l'ensemble des séquences de descripteurs calculées à partir de cette base peut être assimilé au tirage aléatoire X de l'algorithme du calcul de confiance vu au chapitre 3. Les vraisemblances associées Y ne sont autres que les vraisemblances $P(X|\lambda)$. Il est donc possible de déterminer de manière automatique un seuil de décision permettant d'exploiter la vraisemblance de sortie d'une HMM.

4.3 Prise en compte du voisinage : les chaînes couplées

Les analyses réalisées jusqu'ici ne mettaient en œuvre que des classifications par blocs, chaque bloc étant traité indépendamment des autres. Au vu du cadre applicatif, une telle hypothèse est faible. En effet, notre approche consiste à rechercher les anomalies au sein du mouvement global de la foule. Or localement le déplacement de la foule est très corrélé au déplacement de son voisinage, pour deux raisons :

- ▷ Un objet en déplacement traverse généralement différents blocs. Vu du bloc, cela se caractérise par une propagation du mouvement de proche en proche.
- ▷ Au sein d'une foule le mouvement local est très fortement contraint par le mouvement du voisinage.

C'est pour ces deux raisons qu'une modélisation de la propagation du mouvement est nécessaire. Pour cela, une extension de la théorie des chaînes de Markov peut être utilisée, le couplage de chaîne de Markov cachées.

4.3.1 État de l'art des couplages

Originellement, le couplage de chaîne a pour but de répondre au problème de modélisation lorsque plusieurs processus avec chacun leur chaîne respective, interagissent entre eux, comme représenté à la figure 4.4. Dans ce cas, l'hypothèse de Markov qui suppose que les paramètres de l'état courant suffisent à déterminer l'état complet du système, n'est plus vérifiée. C'est le cas par exemple, des systèmes interagissant entre eux comme deux tennismen jouant ensemble, un système ne modélisant qu'un seul joueur ne se suffit pas à lui-même.

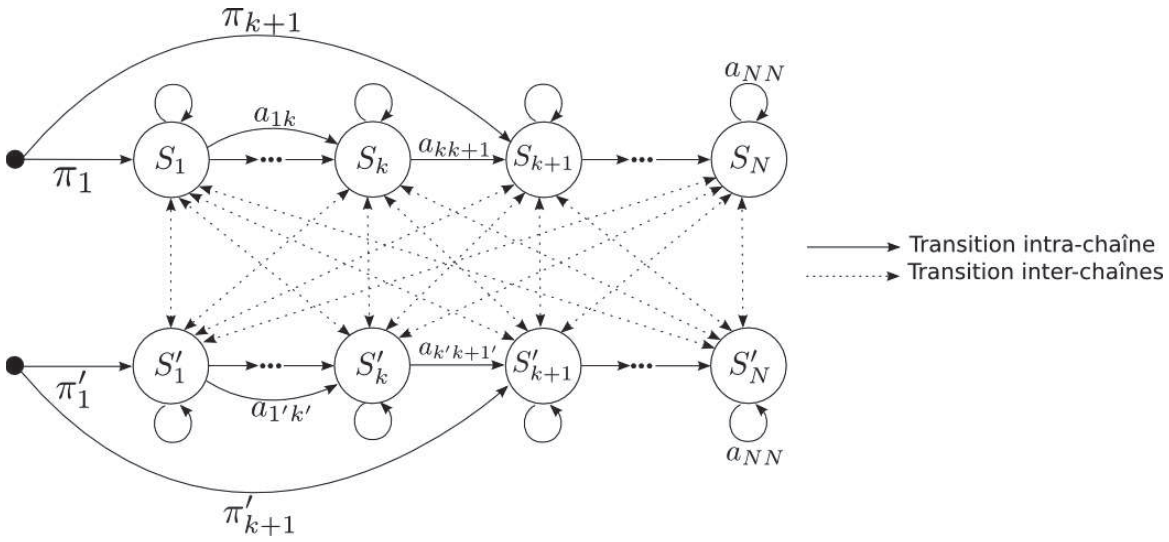


FIGURE 4.4 – Schéma déployé de deux HMMs intégralement couplées entre elles. La complexité des interactions entre les chaînes rendent la résolution exacte de ces problèmes extrêmement coûteuse.

Ce genre de problème peut être résolu en fusionnant les différentes chaînes en une seule chaîne. Les nouveaux états devenant ainsi les produits cartésiens entre les différentes chaînes et les probabilités d'émission devenant des densités de probabilité multivaluées. Une telle solution, nommée dans le littérature *Cartesian Product HMM*, n'est cependant pas viable en pratique à cause de la complexité algorithmique équivalente à un problème NP-complet.

D'autres approches mettent en place d'autres mécanismes plus subtils de couplage entre chaînes. Il répondent généralement à des couplages bien précis comme par exemple les *Factorial HMM* (Ghahramani et Jordan (1997)) qui visent plutôt à répondre à des problèmes comme différencier deux procédés indépendant sous-jacents à partir d'un seul flux d'observation. Les problèmes mettant en avant des procédés dépendant les uns des autres, comme lors d'une reconnaissance audio-visuelle de la parole, se résolvent dans la littérature par des algorithmes tels que les *Linked HMMs* (Saul et Jordan (1995)), les arbres de décision de Markov cachés (Jordan *et al.* (1996)) et les *Coupled HMMs* (Brand (1997)) pour lesquels nous porterons un intérêt tout partculier.

Les *Coupled HMMs* dans leur concept de base visent à ajouter au modèle des transitions entre les deux chaînes à chaque instant. Une telle représentation possède une complexité algorithmique équivalente à celle des *Cartesian Product HMMs*, c'est pourquoi Brand (1997) propose une approximation en limitant les transitions inter-chaînes possibles. Ceci permet d'obtenir un algorithme en $O(TCN^2)$ (avec C le nombre de chaîne) pour répondre au deuxième problème de Rabiner (1990) généralisé au cas de couplage de chaîne, à savoir déterminer les sous états optimaux de chacune des chaînes pour des séquences d'observations associées à chacune d'elles. Brand (1997) propose de rajouter en-

tre deux HMMs, H1 et H2, deux matrices de transition, l'une de H1 vers H2 et l'autre de H2 vers H1 permettant de modéliser l'influence à l'instant t de l'état le plus vraisemblable d'une chaîne sur l'autre, qui sera appelé état « ami » (*sideback* en anglais). Cette variante permet d'étudier l'évolution concurrente de différentes chaînes et ainsi de modéliser une notion de causalité entre les observations associées à deux chaînes différentes.

4.3.2 Théorie des *Coupled HMMs*

4.3.2.1 Principe du couplage

Un couplage total entre C chaînes est équivalent à un *Cartesian Product HMM* de N^C états, ce qui implique une complexité algorithmique pour l'algorithme *forward* de $O(TN^{2C})$. En ne considérant qu'un sous ensemble $S^{\{Q\}}$ de tous les chemins possibles $S^{\{\}}$, il est possible de faire diminuer cette complexité à $O(TCN^2)$. Reste à définir quel sous-ensemble de chemins $S^{\{Q\}}$ choisir ?

Pour répondre à ce problème, Brand (1997) s'impose deux *a priori*. Le premier, baptisé « contrainte de temps et d'espace », est de parcourir au plus $O(N)$ têtes de chemin afin d'obtenir la complexité espérée. La seconde, la « contrainte de visite », est de s'assurer que toutes les transitions considérées soient parcourues afin d'être ré-estimées correctement. Le critère choisi par Brand (1997) pour déterminer le sous-ensemble $S^{\{Q\}}$, est de minimiser l'entropie croisée $D(S^{\{\}}, S^{\{Q\}})$ entre la probabilité *a posteriori* du modèle exact $P(M_{S^{\{\}}}|\widetilde{\mathbf{O}})$ prenant en compte toutes les transitions et celle du modèle approché $P(M_{S^{\{Q\}}}|\widetilde{\mathbf{O}})$, pour un ensemble de séquences d'observations d'entrainement $\widetilde{\mathbf{O}} = \{\widetilde{O}^{(1)}\widetilde{O}^{(2)}\dots\widetilde{O}^{(K)}\}$ avec $\widetilde{O}^{(k)} = \{O^{[1](k)}, \dots, O^{[C](k)}\}$ et $O^{[c](k)} = O_1^{[c](k)}O_2^{[c](k)}\dots O_T^{[c](k)}$. La probabilité *a posteriori* d'un modèle étant égale à la somme des probabilités *a posteriori* de tous les chemins de ce modèle, on peut noter $\langle P(M_S|\widetilde{O}^{(k)})\rangle_S$, l'espérance de la probabilité *a posteriori* de toutes les séquences d'état $\widetilde{Q}_S$ de l'ensemble S connaissant l'observation $\widetilde{O}^{(k)}$. Les séquences d'états $\widetilde{Q}_S$ sont définies par un ensemble de C-tuples, $\widetilde{Q}_S = \left\{\{Q_s^{[1]}, \dots, Q_s^{[C]}\}_s\right\}$ et $Q_s^{[c]} = q_{s_1}^{[c]}q_{s_2}^{[c]}\dots q_{s_T}^{[c]}$. Par abus de notation mais dans un soucis de clarté, une séquence d'états pour une chaîne sera notée $Q_s^{[c]} = s_1^{[c]}s_2^{[c]}\dots s_T^{[c]}$. Toujours pour faciliter la lecture des formules, les probabilités de transition ainsi que les probabilités d'émission seront ici, exprimées à l'aide de notations probabilistes. La probabilité *a posteriori* est donc égale à :

$$P(\widetilde{Q}_s|\widetilde{O}) = \frac{\prod_c^C \left(\pi_{s_1^{[c]}} P(O_1^{[c]}|s_1^{[c]}) \prod_{t=2}^T \left(P(O_t^{[c]}|s_t^{[c]})P(s_t^{[c]}|s_{t-1}^{[c]}) \prod_d^{C\setminus(c)} P(s_t^{[c]}|s_{t-1}^{[d]})\right)\right)}{P(\widetilde{O})} \tag{4.32}$$

avec $P(\binom{[d]}{j}|\binom{[c]}{i})$ la transition inter-chaîne entre l'état S_i de la chaîne c et l'état S_j de la chaîne d.

Toujours dans un souci de clarté, nous nous contenterons d'étudier le cas de deux chaînes couplées entre elles. De plus, leur rôle étant totalement symétrique, nous nous dispenserons de l'indice (c). Nous nous contenterons de représenter les éléments provenant de l'autre chaîne par une apostrophe. L'équation (4.32) peut donc se réécrire :

$$P(\widetilde{Q}_s|\widetilde{O}) = \frac{\pi_{s_1}P(O_1|s_1)\pi_{s'_1}P(O'_1|s'_1)}{P(\widetilde{O})} \prod_{t=2}^T P(s_t|s_{t-1})P(s'_t|s_{t-1})P(O_t|s_t)P(s'_t|s'_{t-1})P(s_t|s'_{t-1})P(O'_t|s'_t) \tag{4.33}$$

L'entropie croisée est quant à elle définie par :

$$D(S^{\{\}}, S^{\{Q\}}) = \sum_{k=1}^{K} \langle P(M_{S^{\{\}}}|\widetilde{O}^{(k)})\rangle_{S^{\{\}}} \log \frac{\langle P(M_{S^{\{\}}}|\widetilde{O}^{(k)})\rangle_{S^{\{\}}}}{\langle P(M_{S^{\{\}}}|\widetilde{O}^{(k)})\rangle_{S^{\{\}}}} \tag{4.34}$$

Minimiser l'entropie croisée de l'équation (4.34) revient à maximiser le terme $\sum_{k=1}^{K} \log\langle P(M_{S^{\{\}}}|O_t)\rangle_{S^{\{\}}}$. Donc l'ensemble des chemins Q recherchés est donné par :

$$\begin{aligned}
Q &= \underset{s\in S^{\{\}}}{argmax} \sum_{k=1}^{K} \log\langle P(M_s|\widetilde{O}^{(k)})\rangle_s \\
&= \underset{s\in S^{\{\}}}{argmax} \sum_{k=1}^{K} \left\langle \log \begin{pmatrix} \pi_{s_1} P(O_1^{(k)}|s_1)\pi_{s'_1} P(O_1^{\prime(k)}|s'_1) \\ \prod_{t=2}^{T} P(s_t|s_{t-1})P(s'_t|s_{t-1})P(O_t^{(k)}|s_t)P(s'_t|s'_{t-1})P(s_t|s'_{t-1})P(O_t^{\prime(k)}|s'_t) \end{pmatrix} \right\rangle_s \\
Q &= \underset{s\in S^{\{\}}}{argmax} \sum_{k=1}^{K} \left\langle \begin{pmatrix} \log \pi_{s_1} + & \log \pi_{s'_1} + \\ \log P(O_1^{(k)}|s_1) + & \log P(O_1^{\prime(k)}|s'_1) \end{pmatrix} + \sum_{t=2}^{T} \begin{pmatrix} \log P(s_t|s_{t-1}) + & \log P(s'_t|s'_{t-1}) + \\ \log P(s'_t|s_{t-1}) + & \log P(s_t|s'_{t-1}) + \\ \log P(O_t^{(k)}|s_t) + & \log P(O_t^{\prime(k)}|s'_t) \end{pmatrix} \right\rangle_s
\end{aligned} \tag{4.35}$$

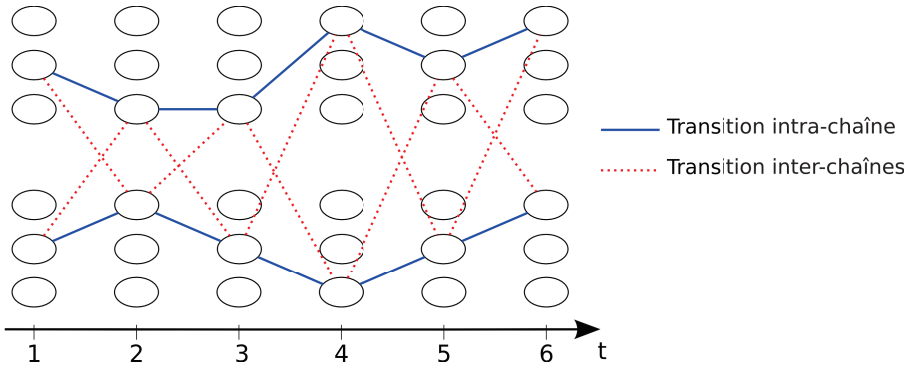


FIGURE 4.5 – Treillis couplé entre deux chaînes. Les probabilités de transition entre chaînes sont représentées en pointillés.

Répondre au problème (4.35) avec la « contrainte de temps et d'espace » oblige à définir pour chaque tête dans une chaîne un état « ami » dans l'autre chaîne, comme on peut le voir sur le schéma 4.5 qui représente la juxtaposition de deux treillis associés à deux chaînes de Markov couplées par les couples (tête, ami) représentés en pointillés. Ce couple unique (tête, ami) permet de ne pas faire exploser la complexité en ne conservant que $O(N)$ têtes à explorer. La deuxième contrainte, celle de « visite », impose un parcours de tous les états. Il faut donc s'assurer qu'à chaque instant t, tous les états d'une chaîne seront considérés comme tête. Ceci permet de simplifier l'équation (4.35) puisqu'en sommant sur tous les états d'une chaîne, les termes $\log \pi_{s_1}$, $\log P(O_1^{(k)}|s_1)$, $\log P(s_t|s_{t-1})$, $\log P(s'_t|s_{t-1})$ et $\log P(O_t^{(k)}|s_t)$ deviennent constants, on peut donc retirer ces termes de la moyenne sur s. Résoudre le problème (4.35) revient donc à résoudre :

$$Q = \underset{s\in S^{\{\}}}{argmax} \sum_{k=1}^{K} \left\langle \begin{pmatrix} \log \pi_{s'_1} + \\ \log P(O_1^{\prime(k)}|s'_1) \end{pmatrix} + \sum_{t=2}^{T} \begin{pmatrix} \log P(s'_t|s'_{t-1}) + \\ \log P(s_t|s'_{t-1}) + \\ \log P(O_t^{\prime(k)}|s'_t) \end{pmatrix} \right\rangle_s \tag{4.36}$$

Ce critère calculé dans une chaîne permet de définir l'état « ami » de la chaîne opposée et inversement. Il est ainsi possible de développer des algorithmes approchés *Forward-Backward* et de Viterbi qui serviront à la ré-estimation et à l'évaluation des chaînes couplées de manière analogue à celle d'une chaîne simple.

4.3.2.2 Adaptation des algorithmes de parcours

Soit $h_t(i)$ la ième tête à l'instant t et $k_t(i)$ l'état « ami » associé à l'état S_i, donc un état de l'autre chaîne ($k_t(i) = S'_j$).

Forward-Backward En maximisant la probabilité *a posteriori* partielle à un instant t, il est possible de déterminer l'état « ami » de la chaîne opposée. Cette probabilité partielle que nous noterons $\breve{q}^*_t(i)$, ne tient pas compte de l'influence des états « amis » courants (puisqu'il sont encore inconnus). Elle est déterminée de manière récursive par la relation :

$$\breve{q}^*_t(i) = P(O_t|i) \sum_j P(i|h_{t-1}(j))P(i|k_{t-1}(j))\alpha^*_{t-1}(j) \tag{4.37}$$

Avec $\alpha^*_{t-1}(j)$, la probabilité *a posteriori* complète à l'instant $t-1$. Concernant les initialisations, la probabilité complète $\alpha^*_1(i)$ ainsi que la probabilité partielle $\breve{q}^*_1(i)$ sont initialisées à :

$$\alpha^*_1(i) = \breve{q}^*_1(i) = \pi_i P(O_1|i) \tag{4.38}$$

L'état « ami » associé à une chaîne est donc l'état ayant la probabilité *a posteriori* partielle maximale dans l'autre chaîne $\breve{q}^{*\prime}_t(.)$:

$$k_t(i) = \underset{j}{argmax}\ \breve{q}^{*\prime}_t(j) \tag{4.39}$$

A noter dans l'equation (4.39) que $k_t(i)$ ne dépend pas de i mais seulement de t. De même, d'après la contrainte de visite, on a $h_t(i) = i$. Pour les équations (4.40) à (4.43), nous présentons à la fois les équations avec les notations complètes ainsi que les notations simplifiées. Finalement, il est possible de déterminer la probabilité *a posteriori* complète à l'instant t :

$$\begin{aligned}
\alpha^*_t(i) &= P(O_t|i)P(O'_t|k_t(i)) \sum_j P(i|h_{t-1}(j))P(i|k_{t-1}(j))P(k_t(i)|h_{t-1}(j))P(k_t(i)|k_{t-1}(j))\alpha^*_{t-1}(j) \\
\alpha^*_t(i) &= P(O_t|i)P(O'_t|k_t) \sum_j P(i|j)P(i|k_{t-1})P(k_t|j)P(k_t|k_{t-1})\alpha^*_{t-1}(j)
\end{aligned} \tag{4.40}$$

Cette probabilité complète, permet de propager les probabilités à l'instar de la variable α classique. Cependant elle ne permet pas de ré-estimer les paramètres correctement à cause du sous-échantillonnage sur les chemins parcourus. En conséquence, une étape de marginalisation sur l'ensemble des états « amis » possibles est nécessaire. Elle permet d'obtenir une variable prenant en compte de manière approchée, l'influence de tous les chemins antérieurs sur le treillis et ainsi assurer la « contrainte de visite ». Cette étape qui est aussi récursive, se déduit de α^* :

$$\begin{aligned}\alpha_t^{\#}(i) &= P(O_t|i)\sum_j P(i|h_{t-1}(j))P(i|k_{t-1}(j))\sum_g P(O'_t|k_t(g))P(k_t(g)|h_{t-1}(j))P(k_t(g)|k_{t-1}(j))\alpha^*_{t-1}(j)\\ \alpha_t^{\#}(i) &= P(O_t|i)\sum_j P(i|j)P(i|k_{t-1})\sum_{g'} P(O'_t|g')P(g'|j)P(g'|k_{t-1})\alpha^*_{t-1}(j)\end{aligned} \tag{4.41}$$

De la même manière la variable *backward* β^* est initialisée à 1 et la relation de récursivité est définie par :

$$\begin{aligned}\beta_t^{*}(i) &= P(O_{t+1}|i)P(O'_{t+1}|k_{t+1}(i))\sum_j P(h_{t+1}(j)|i)P(k_{t+1}(j)|i)P(h_{t+1}(j)|k_t(i))P(k_{t+1}(j)|k_t(i))\beta^*_{t+1}(j)\\ \beta_t^{*}(i) &= P(O_{t+1}|i)P(O'_{t+1}|k_{t+1})\sum_j P(j|i)P(k_{t+1}|i)P(j|k_t)P(k_{t+1}|k_t)\beta^*_{t+1}(j)\end{aligned} \tag{4.42}$$

La marginalisation est quant à elle égale à :

$$\begin{aligned}\beta_t^{\#}(i) = P(O_{t+1}|i)P(O'_{t+1}|k_{t+1}(i))&\sum_j P(h_{t+1}(j)|i)P(k_{t+1}(j)|i)\\ &\sum_g P(h_{t+1}(j)|k_t(g))P(k_{t+1}(j)|k_t(g))\beta^*_{t+1}(j)\\ \beta_t^{\#}(i) = P(O_{t+1}|i)P(O'_{t+1}|k_{t+1})&\sum_j P(j|i)P(k_{t+1}|i)\sum_{g'} P(j|g')P(k_{t+1}|g')\beta^*_{t+1}(j)\end{aligned} \tag{4.43}$$

Viterbi Très similaire à celui du *Forward-Backward*, l'algorithme de Viterbi consiste a reproduire les mêmes étapes en remplaçant les sommes sur les états par le maximum des états lors du calcul de la probabilité *a posteriori* partielle et complète. L'étape de marginalisation en revanche n'est pas nécessaire car cette étape sert uniquement à la ré-estimation des paramètres.

4.3.2.3 Résolution des trois problèmes

Tous comme pour les HMMs classiques, la résolution des trois problèmes de Rabiner (1990) généralisé au cas des chaînes couplées, s'effectue par l'intermédiaire des algorithmes *Forward-Backward* et Viterbi adaptés aux chaînes couplées comme vu précédemment.

Pour ce qui est de la vraisemblance finale du couplage $P(\mathbf{O}|M)$, elle est égale à la somme des vraisemblances $P(O|\lambda)$ et $P(O'|\lambda')$. La log-vraisemblance d'une chaîne est déterminée par la probabilité *a posteriori* complète à l'instant T :

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha^*_T(i) \tag{4.44}$$

Là encore, cette vraisemblance finale est utilisée lors de la phase d'entraînement de la chaîne. En phase d'utilisation de celle-ci, c'est l'algorithme de Viterbi adapté aux chaînes couplées qui est utilisé.

Le deuxième problème, selon le critère choisi, est directement résolu soit par l'algorithme *Forward-Backward* pour maximiser individuellement la vraisemblance de chaque état, soit par l'algorithme

Viterbi pour maximiser un chemin complet. Pour la maximisation individuelle, la variable γ est dans le cas des chaînes couplées, définie comme dans le cas des HMMs simples en utilisant les probabilités marginalisées $\alpha^{\#}$ et $\beta^{\#}$:

Cette variable γ est utilisée par ailleurs pour la résolution du dernier problème de Rabiner (1990). Toujours par analogie avec les HMMs classiques, les paramètres de chaque chaîne sont ré-estimés en comptabilisant la probabilité cumulée de chacun des paramètres le long d'une ou de plusieurs séquences. Pour les transitions par exemple, toujours en utilisant la variable ξ définie à l'équation (4.19) avec les probabilités marginalisées $\alpha^{\#}$ et $\beta^{\#}$, il est possible de ré-estimer les transitions d'une chaîne comme dans l'équation (4.21), ainsi que les densités d'émission. Pour les transitions croisées, on définit $\xi_t(i, j')$ la probabilité d'être dans l'état S_i de la chaîne considérée à l'instant t et dans l'état S_j de l'autre chaîne à l'instant $t + 1$:

$$\xi_t(i, j') = \frac{\alpha_t^{\#}(i) P(j'|i) P(O'_{t+1}|j') \beta_{t+1}^{\#}(j')}{P(\widetilde{O}|\lambda)} \tag{4.45}$$

La réestimation de la transition croisée est enfin égale à :

$$\overline{c_{ij'}} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j')}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{4.46}$$

4.3.2.4 Adaptation de la théorie

Il est assez fréquent d'être confronté à des problèmes de stabilité numérique lors de l'utilisation des chaînes couplées et ce malgré une implémentation en échelle logarithmique. Ce problème vient essentiellement des multiples produits de probabilités réalisés lors de l'algorithme *Forward-Backward*. Ce problème peut aussi être rencontré dans le cas de HMM simples lors de l'analyse de très longues séquences (T > 100). Rabiner (1990) propose un conditionnement que nous n'avons pas abordé dans la section 4.2 car au vu des longueurs de séquences utilisées, le problème ne se posait pas.

Pour pallier ce problème au niveau des chaînes couplées, une étape de mise à l'échelle est obligatoire lors des calculs des variables α^* et β^*. L'idée est de conditionner les variables α^* et β^* à chaque instant afin de les maintenir dans des valeurs raisonnables. Notons $\widetilde{\alpha^*}$ et $\widetilde{\beta^*}$ ces variables reconditionnées. Soit $\nu_t = \frac{1}{\sum_i \widetilde{\alpha_t^*(i)}}$, à chaque instant on réajuste les variables $\widetilde{\alpha^*}$ et $\widetilde{\beta^*}$ de la manière suivante :

$$\begin{aligned} \widehat{\alpha_t^*(i)} &= \nu_t \widetilde{\alpha_t^*(i)} \\ \widehat{\beta_t^*(i)} &= \nu_t \widetilde{\beta_t^*(i)} \end{aligned} \tag{4.47}$$

Les variables $\widehat{\alpha_t^*(i)}$ et de $\widehat{\beta_t^*(i)}$ sont ensuite utilisées en lieu et place de $\widetilde{\alpha_t^*(i)}$ et de $\widetilde{\beta_t^*(i)}$ dans la récursion à $t + 1$. Ce conditionnement ne modifie pas la ré-estimation des paramètres comme démontré dans le papier de Rabiner (1990). En revanche, l'estimation de la vraisemblance d'une séquence d'observations $P(O|\lambda)$ est modifiée. En effet, il est assez aisé d'établir la relation entre la variable $\widehat{\alpha_t^*(i)}$ et la variable $\alpha_t^*(i)$ sans mise à l'échelle comme définie à l'équation (4.40) :

$$\widehat{\alpha_t^*(i)} = \prod_{s=1}^{t} \nu_s \alpha_t^*(i) \tag{4.48}$$

Or, par définition du conditionnement $\sum_i \widehat{\alpha_t^*(i)} = \sum_i \nu_t \widetilde{\alpha_t^*(i)} = 1$. On peut donc établir la relation :

$$\begin{aligned}\sum_i \widehat{\alpha_T^*(i)} &= \sum_i \prod_{s=1}^{T} \nu_s \alpha_T^*(i) \\ &= \prod_{s=1}^{T} \nu_s \sum_i \alpha_T^*(i) \\ &= \prod_{s=1}^{T} \nu_s P(O|\lambda) \\ &= 1\end{aligned} \tag{4.49}$$

Et ainsi en déduire que :

$$P(O|\lambda) = \frac{1}{\prod_{s=1}^{T} \nu_s} \tag{4.50}$$

Dans le cadre des chaînes couplées un deuxième conditionnement est nécessaire. Ce conditionnement vise à assurer l'invariant $\sum_{i=1}^{N} \gamma_t(i) = 1$ pour les chaîne couplées. En effet, alors que cette invariant est assuré dans le cas des HMMs comme présenté à l'équation (4.16), à cause du sous échantillonnage sur les chemins parcourus, les chaînes couplées ne bénéficient que de $\sum_{i=1}^{N} \gamma_t(i) < 1$. Ce second conditionnement consiste donc à forcer $\beta_t^{\#}(i)$ pour assurer l'invariant, de la manière suivante :

$$\beta_t^{\#}(i) \leftarrow \frac{\beta_t^{\#}(i)}{\sum_{i=1}^{N} \alpha_t^{\#}(i)\beta_t^{\#}(i)} \tag{4.51}$$

A noter que c'est le dénominateur de l'équation (4.16) qui assure l'invariant. Ce même terme permet aussi la normalisation de la séquence d'observations (le terme en $\frac{1}{P_k}$ de l'équation (4.30)) lors de l'utilisation des *Multi Observation HMMs*. Dans les cas des chaînes couplées, ce terme n'est plus nécessaire. En effet, la normalisation est implicitement contenue dans le terme $\beta_t^{\#}(i)$ conditionné. Aucun traitement supplémentaire n'est donc à réalisé pour exploiter plusieurs séquences d'observations lors de l'entaînement.

Pour l'algorithme de Viterbi aucune étape de mise à l'échelle n'est nécessaire. Le simple fait de travailler en échelle logarithmique suffit.

4.3.3 Implémentations

En pratique nous couplons la HMM associée à un bloc avec les HMMs des $N = 4$ blocs adjacents selon une 4-connexité. Nous verrons, par la suite, comment à partir de la théorie de couplage entre deux chaînes, vue dans la section 4.3.2.1, ce couplage entre la chaîne d'un bloc et les N voisins est réalisé au travers de deux manières différentes. Avant cela, un problème lié à la gestion du non-mouvement est mis en avant.

Synchronisation des données Rappelons que prendre en compte lors de la phase d'apprentissage les zones de non-mouvement n'est pas viable en raison de la prédominance de celles-ci sur le long terme. A cause du mécanisme de couplage, les couples de HMM doivent néanmoins être entrainés sur des séquences correspondant aux mêmes instants et avoir la même durée.

Il est donc nécessaire d'une part de compléter les séquences contenant du mouvement par un état représentant le non-mouvement et d'autre part, si à un instant T un bloc i (donc la chaîne i) ne perçoit aucun mouvement alors que le bloc adjacent $i+1$ en perçoit, une séquence sans mouvement $O^{[i]}$ sera associée à la chaîne i. Une illustration de cette synchronisation est donnée figure 4.6, où les chaînes H1 et H2 sont couplées entre elles. Les instants où des mouvements sont perçus au niveau des blocs associés à H1 et H2 sont représentés par des chronogrammes. Les instants où le couplage (H1,H2) doit être entrainé correspond à l'union de ces deux chronogrammes.

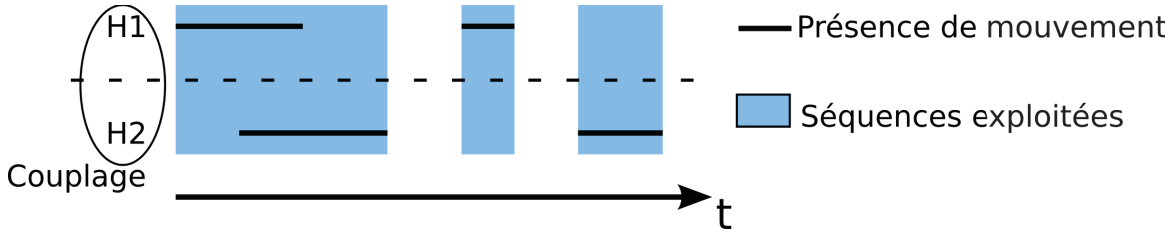


FIGURE 4.6 – Exemple de synchronisation à réaliser avant l'apprentissage.

Cependant cette synchronisation entraine une complexité algorithmique qui rend l'apprentissage extrêmement long. En effet, pour illustrer le phénomène, la figure 4.7 représente les chronogrammes de trois chaînes H1, H2, H3, la chaîne H2 étant couplée à la fois aux chaînes H1 et H3. Le couplage (H1, H2) devrait théoriquement être entrainé sur l'union des chronogrammes de H1 et H2, c'est-à-dire sur la partie en bleu alors que le couplage (H2, H3) devrait être entrainé sur la partie rouge. Cependant, le chaîne H2 ne peut pas être entrainé sur deux ensembles d'apprentissage différents. Il faut donc entrainer le couplage (H1,H2) et (H2,H3) sur le même ensemble, à savoir l'union des ensembles bleu et rouge représenté par l'ensemble jaune. En généralisant à notre cadre applicatif, cela signifierait que la présence d'un mouvement à un instant t dans n'importe quel bloc de la scène impliquerait d'ajouter à l'ensemble d'apprentissage tous les blocs de la scène les observations (même celles sans mouvements) de cet instant t. En général, dans les scènes qui nous intéressent, il y a quasiment toujours du mouvement en une zone de la scène, ce qui revient donc à apprendre en chaque bloc à tous les instants. C'est précisément ce que l'on cherche à éviter, toujours en raison de la prépondérance du non-mouvement sur le long terme pour les vidéos traitées, ce qui à tendance à fausser l'apprentissage des mouvements.

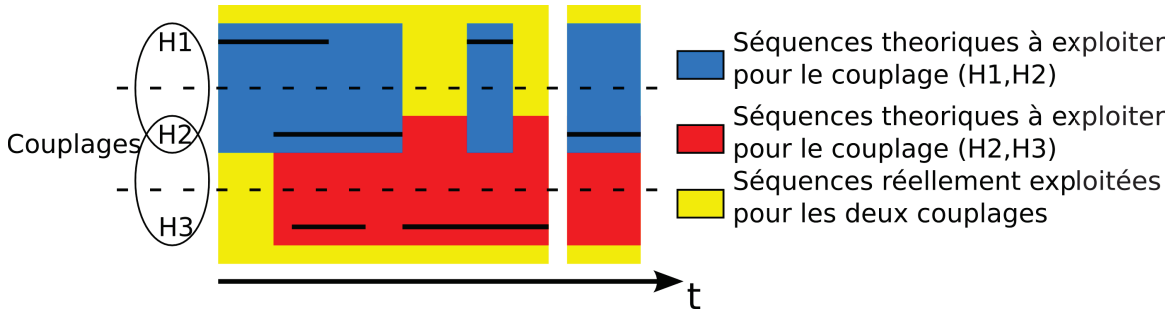


FIGURE 4.7 – Mise en évidence du problème de synchronisation par propagation de proche en proche.

Pour éviter ce problème, nous avons choisi de dupliquer les chaînes afin qu'une chaîne n'appartienne qu'à un couplage à la fois. En conséquence, à un même bloc sera associé N chaînes. Chaque chaîne appartenant à un couplage sera entrainée sur l'union des séquences associées à chacune des

chaînes de ce couplage et uniquement de ce couplage, comme représenté sur la figure 4.8. On constate qu'en dupliquant la chaîne H2 il est possible d'entrainer (H1,H2) d'une part sur l'ensemble bleu et (H2',H3) d'autre part sur l'ensemble rouge.

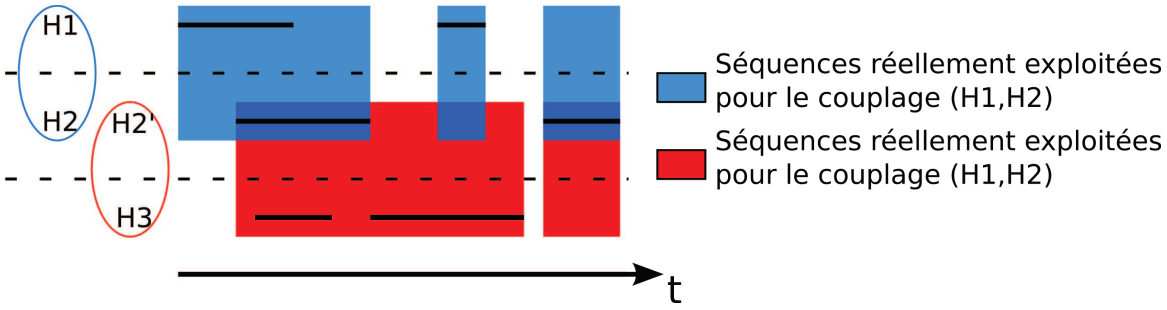


FIGURE 4.8 – Résolution du problème de synchronisation par duplication des chaînes à entrainer.

Ceci permet de réaliser un apprentissage des HMMs couplées équivalent à celui réalisé avec le SKDE ou les HMMs simples, c'est-à-dire un apprentissage qui ne prend pas en compte les périodes de non-mouvement de certaines zones de la scène. Cette mise à l'écart volontaire du non-mouvement est ce qui permet l'étude de densité de personnes ou de véhicules variable au cours du temps.

Méthodes de couplage Deux méthodes de couplage ont été mises à l'œuvre. La première, la plus basique est réalisée à partir de quatre couples, nous noterons ce couplage par la suite couplage « 4x2 » (cf. figure 4.9). Pour ce bloc i, la vraisemblance finale sera la moyenne géométrique de chacune des quatre vraisemblances de couple $P(O^{[i]}, O^{[j]}|\lambda_i, \lambda_j)$ afin d'avoir l'influence de l'ensemble du voisinage :

$$P(\tilde{O}|M_i) = \sqrt[n]{\prod_{j=1}^{N} P(O^{[i]}, O^{[j]}|\lambda_i, \lambda_j)} \tag{4.52}$$

Cette méthode implique d'avoir quatre HMMs différentes (une pour chaque couple) par bloc. Avec la contrainte de synchronisation vue précédemment, cela fait un total de huit chaînes à définir par bloc, ce qui est relativement coûteux.

L'autre méthode est d'étendre la théorie des *Coupled HMMs*, non plus avec un couplage point à point entre deux chaînes, mais via un couplage « 1-N ». Ce couplage brise la symétrie entre les chaînes. La chaîne centrale doit tenir compte des probabilités de transition croisées depuis les N « états amis » de ses voisins alors que pour ces derniers le couplage avec la chaîne centrale s'effectue de manière classique (cf. figure 4.10).

Cette approche nécessite moins de chaînes par bloc puisque qu'avec la synchronisation, seulement cinq chaînes par bloc sont à définir. Ceci permet une économie en terme de calcul, comme nous le verrons dans la chapitre 5.

4.3.4 Enrichissement de la détection

Les approches mises en place jusqu'à présent, à savoir le SKDE et les HMMs simples, ont des traitements par bloc complètement décorrélés les uns des autres. En ajoutant une modélisation spatiale grâce aux chaînes couplées, on est en droit de se demander si outre la modélisation de la cohérence

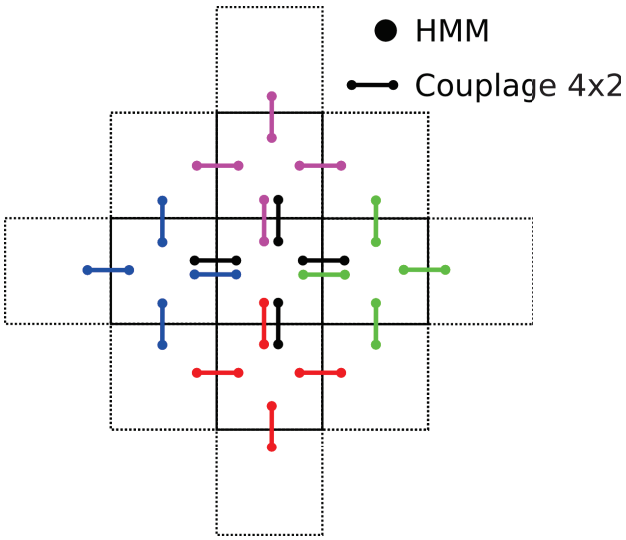


FIGURE 4.9 – Couplage 4x2. Le nombre moyen de chaînes par bloc est de 8 : 4 chaînes pour ses propres couplages avec ses voisins (points noirs dans le bloc central) et une de chacun de ses voisins (points voilet, vert, rouge et bleu dans le bloc central).

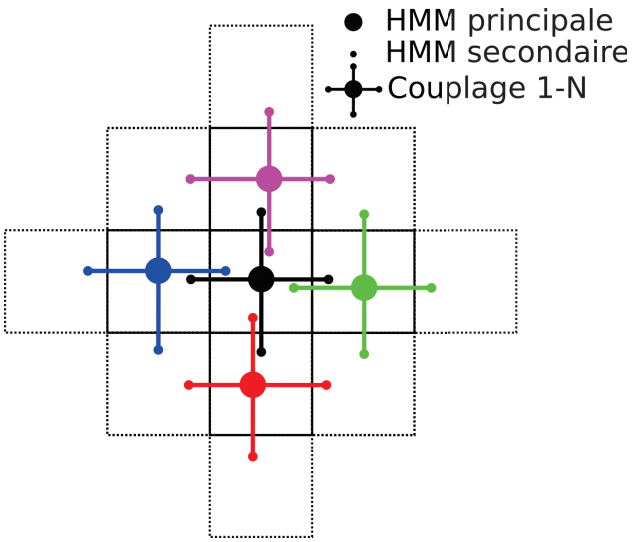


FIGURE 4.10 – Couplage 1-4. Le nombre moyen de chaînes par bloc est de 5 : la chaîne principale (gros point noir dans le bloc central) et une chaîne de chacun de ses voisins (points voilet, vert, rouge et bleu dans le bloc central).

spatiale, il n'est pas possible d'enrichir le panel d'évènements susceptibles d'être détectés. Ces nouvelles détections concernent des configurations d'ensembles de mouvements inhabituelles.

Dans notre contexte applicatif, mettre en avant de telles configurations est assez délicat. Cela correspond à des situations extrêmement rares qui n'ont pas été rencontrées durant la thèse, comme des scènes de carambolages ou des queues de poisson très serrées. Cependant, pour vérifier que les chaînes couplées possèdent cette faculté de modélisation permettant de détecter ce genre de situation, une vidéo de synthèse a été utilisée. Cette séquence de synthèse a été créée via un procédé expliqué au chapitre 5. Pour cette séquence, un croisement dangereux entre deux véhicules a été créé. Les deux véhicules passent très près l'un de l'autre, dans des directions antagonistes mais qui si elles sont prises indépendamment sont normales. Une illustration de ce type de scènes est présentée à la figure 4.11. Les deux premières lignes montrent des évolutions normales du mouvement. En revanche, la troisième ligne n'est pas une situation normale de par la coexistence des deux mouvements simultanément.

La première séquence (première ligne de la figure 4.11) représente le mouvement normal d'un flot de voiture traversant le carrefour dans un sens. La deuxième séquence (deuxième ligne de la figure 4.11) représente le passage normal d'une autre voiture traversant le carrefour selon la direction perpendiculaire à celle de la première séquence. Et enfin, la dernière séquence (troisième ligne de la figure 4.11) représente les deux passages simultanément, ce qui constitue une anomalie.

Afin de vérifier la réaction du système face à un tel évènement, l'évolution des vraisemblances de certains blocs est à l'étude. Nous espérons observer une diminution de la vraisemblance sur la séquence anormale seulement lors de l'utilisation des chaînes couplées. Pour cela, un intérêt particulier est porté sur les quatre blocs situés au niveau de la zone de croisement (cf. figure 4.12). En chacun de ces blocs, l'évolution de l'opposé de la log vraisemblance (OLV) à travers le temps est représentée sur les figures 4.12 et 4.13. Plus cette OLV sera grande, plus elle caractérisera une situation anormale. Les courbes rouges sont associées aux vraisemblances du premier mouvement normal,

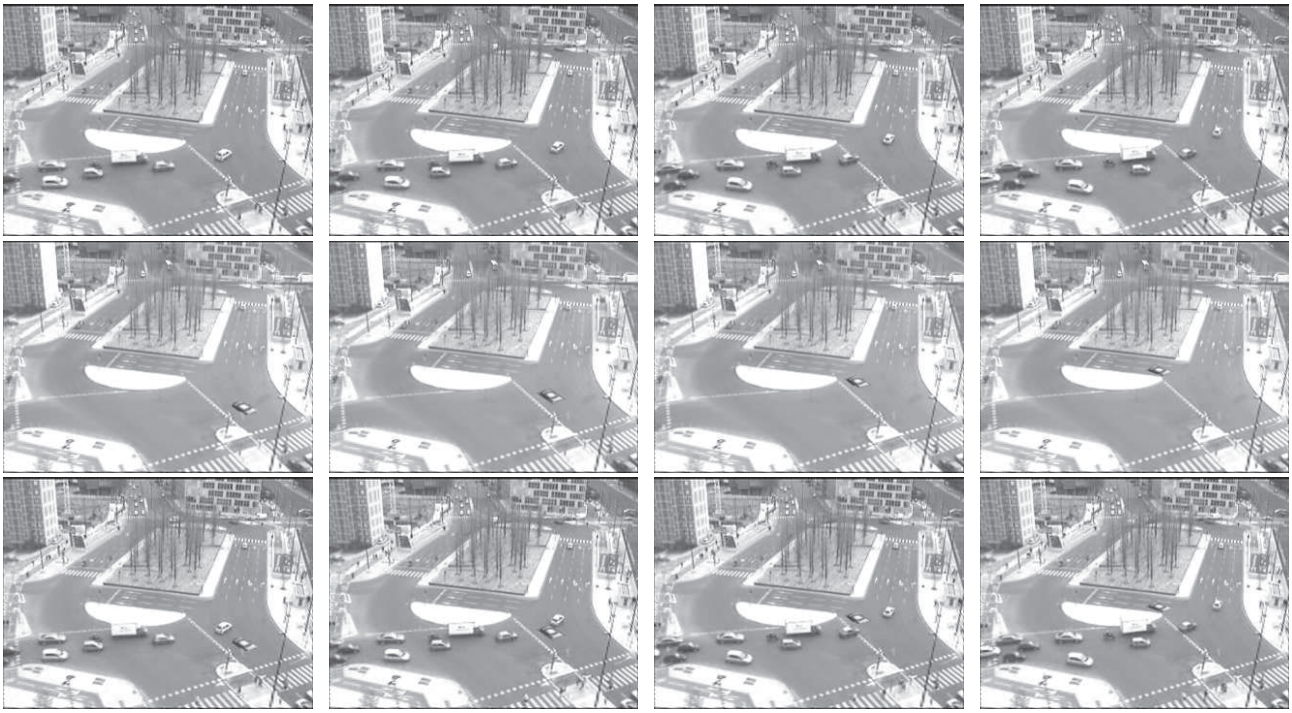


FIGURE 4.11 – Exemples type de configurations de mouvements anormaux. Les deux premières lignes représentent un mouvement de voiture normal. En revanche, la troisième ligne fait cohabiter ces deux mouvements simultanément, ce qui correspond à un comportement anormal.

les vertes à celles du deuxième mouvement normal et enfin les bleues sont associés au mouvement combiné (respectivement la première, deuxième et troisième ligne de la figure 4.11). Cette évolution des vraisemblances est représentée à la figure 4.13 pour des méthodes de classification basées sur les chaînes couplées, alors que la figure 4.12 représente le test témoin réalisé avec une machine d'apprentissage sans couplage, ici le SKDE. Les vraisemblances obtenues avec les HMMs simples n'ont pas été utilisées comme témoin car cette approche ne présente pas de résultats significatifs par rapport au SKDE, comme nous pourrons le voir au chapitre 5. Leur étude est cependant nécessaire pour mettre en place la création et l'exploitation des chaînes couplées.

Pour une courbe donnée, un OLV nul signifie qu'aucun mouvement, donc aucune vraisemblance en sortie de classifieur, n'est observé dans ce bloc à cet instant. Les courbes rouges et vertes servent de témoin pour évaluer l'amplitude de OLV associés à des mouvements normaux. La séquence combinée représentant une situation anormale, on doit donc constater à un instant t, une OLV égale à celle de la courbe rouge aux instants normaux et une OLV supérieure à celles des courbes rouges et vertes, au moment de l'anomalie, c'est à dire au moment où les OLV des courbes rouges et vertes sont toutes les deux non nulles (lors du passage). On peut constater que, comme anticipé, ceci n'est pas le cas avec le SKDE représenté sur la courbe 4.12 puisqu'aucune modélisation de la disposition spatiale du mouvement n'est faite.

Les approches couplées ont été évaluées au travers de la méthode de couplage « 1-N » qui met en place un couplage beaucoup plus complexe que la méthode « 4x2 » qui ne superpose que 4 couples de manière indépendantes. La méthode d'entrainement des chaînes couplées fait ici, l'objet d'un intérêt particulier. En effet, comme vu à la section 4.2.1.2 bien que la méthode d'entrainement classique pour les modèles de Markov soit celle de Baum-Welch, il existe cependant une autre méthode baptisé le *Viterbi Path Counting*. Cette méthode ne permet pas une réestimation complète de la chaîne, en

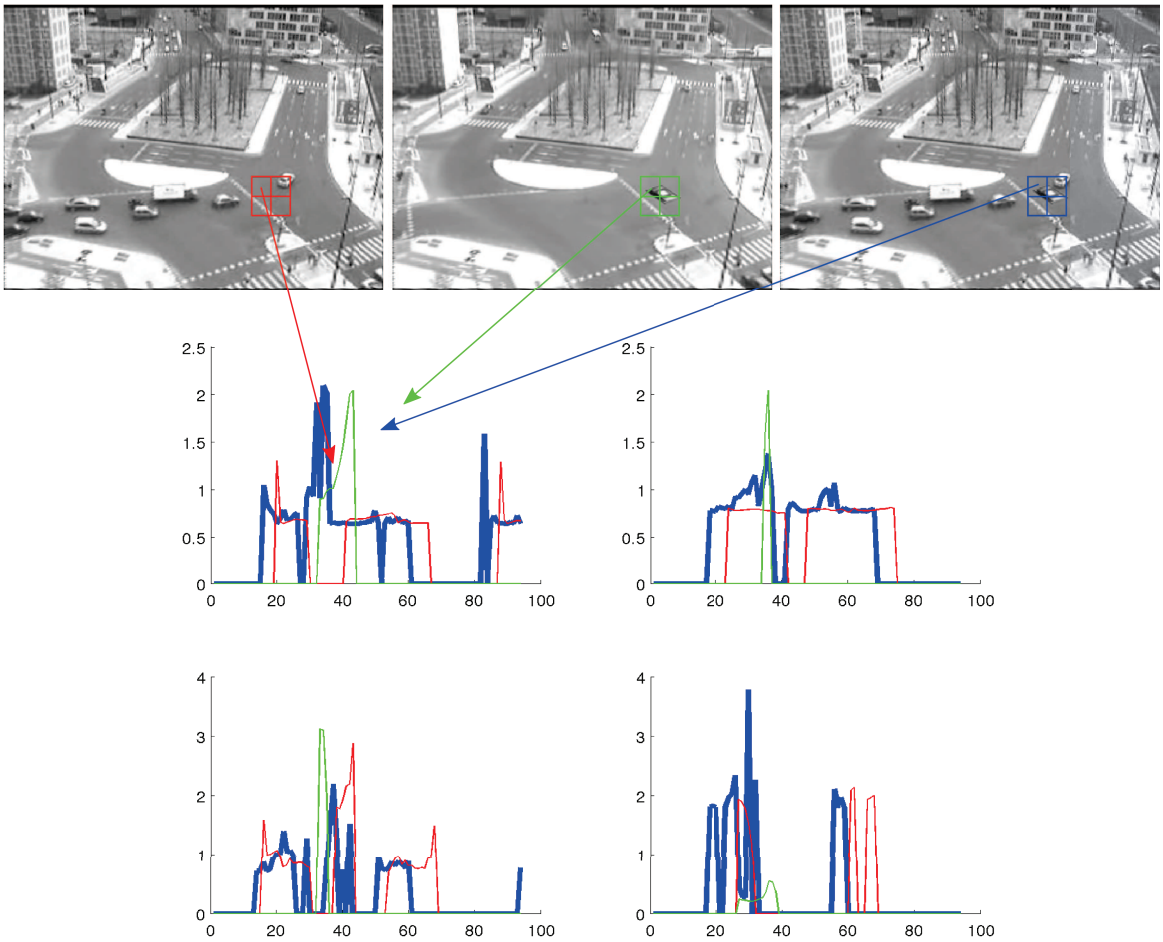
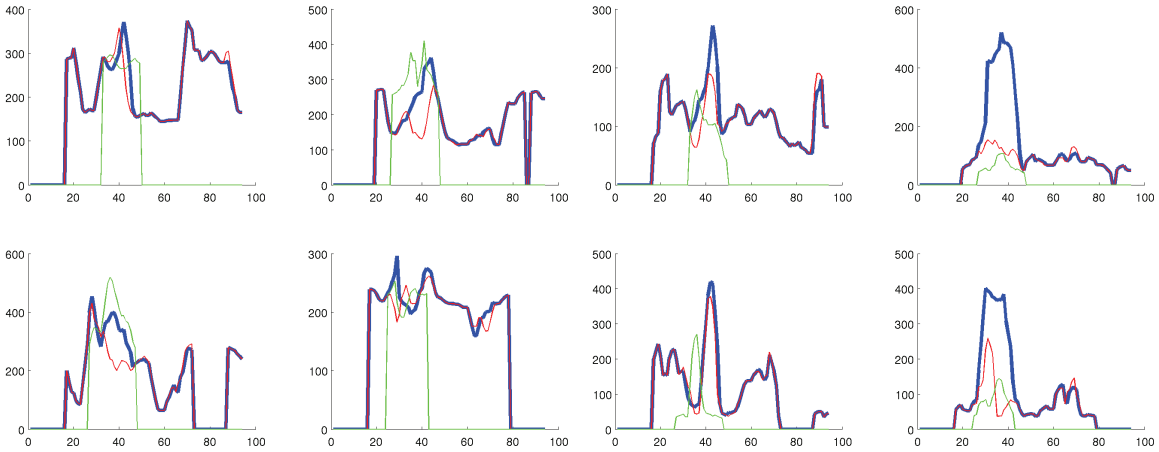


FIGURE 4.12 – Évolution temporelle des OLV du SKDE pour quatre blocs situés au niveau d'un croisement. Les courbes rouges représentent les vraisemblances de véhicules franchissant le carrefour dans une direction normale, les courbes vertes les vraisemblances d'un véhicule incrusté franchissant le carrefour dans une autre direction normale et les courbes bleues les vraisemblances de véhicules franchissant le croisement en même temps (situation anormale). Les vraisemblances issues du SKDE ne font pas ressortir d'anomalie particulière au niveau des courbes bleues.

revanche elle est réputée pour mieux converger. Elle consiste à exploiter le chemin déterminé par l'algorithme de Viterbi pour réajuster les probabilités de transition et d'émission le long de ce chemin. Dans le cadre de la détection de configuration de mouvements anormaux dont il est question ici, cette méthode apprentissage s'avère plus appropriée, comme on peut le constater sur la courbe bleue de la figure 4.13(b) (en comparaison avec celles de la figure 4.13(a)) qui montre de manière plus nette l'augmentation de la OLV souhaitée par rapport aux courbes rouges et vertes.

Ce constat est fait afin de souligner ce qu'il est possible d'exploiter d'une telle modélisation. Il est vrai que dans notre cadre applicatif, ce type d'évènement est assez rare et n'a pas été rencontré. C'est pourquoi une étude plus approfondie du phénomène n'a pas été réalisée. Cependant, dans d'autres cadres applicatifs, il peut être intéressant d'être capable de détecter de telles configurations, chose qui est rendue possible par le biais des chaînes couplées.



(a) Apprentissage de Baum-Welch (b) Apprentissage *Viterbi Path Counting*

FIGURE 4.13 – Évolution temporelle des OLV pour les modélisations couplées pour les quatre blocs situés au niveau du croisement. Dans le cas de l'apprentissage via l'algorithme du *Viterbi Path Counting*, la OLV de la situation anormale (courbe bleue) est très nettement supérieure à celle des situation normale.

4.4 Conclusion

L'objectif de ce chapitre était la mise en place de machine d'apprentissage plus complexes permettant des modélisations plus fines de la dynamique des variables aléatoires et de leur éventuelles dépendances, les HMMs pour ajouter une cohérence temporelle à la modélisation du mouvement et les *Coupled HMMs* pour la cohérence spatiale.

Ce type de modèle nécessite des pré-requis assez stricts qui imposent une mise en place particulière. Nous avons vu dans ce chapitre différentes stratégies pour répondre à ces prérequis. L'influence de ces différentes approches sera évaluée en termes de performance de classification dans le chapitre 5 après avoir mis en place un protocole expérimental permettant de les comparer.

Il est cependant possible de mettre en avant l'un des avantages propres aux chaînes couplées. La modélisation spatiale permet d'enrichir le panel des évènements détectables en définissant une notion de causalité dans la propagation du mouvement. Ce type de modélisation a été illustré dans le cadre de ces travaux, sur un exemple précis mais il pourrait très facilement être généralisé dans d'autres cadres applicatifs plus spécifiques nécessitant des modélisations de processus interagissant. Pour ce qui est de la mise en place concrète de tels mécanismes, nous avons constaté que là encore, des précautions quant à la méthode d'entrainement choisie doivent être prises.

Chapitre 5

Comparaison et Évaluation

Ce chapitre a pour objectif de faire la synthèse sur l'apport en termes de classification d'évènements anormaux, des différentes briques algorithmiques mises en place, aussi bien au niveau du descripteur que de la machine d'apprentissage. Pour évaluer chacun de ces apports, un critère d'évaluation ainsi qu'un protocole expérimental précis doivent avant tout être fixés. L'analyse comparative consistera ensuite à constater le gain en performance de classification entre les différents descripteurs, les différentes stratégies d'utilisation des machines d'apprentissage et entre les machines d'apprentissage elles-mêmes.

5.1 Analyse de performance

Nous avons défini l'anormal au sens de notre application comme étant des mouvements ne correspondant pas à ce qui peut être observé de manière fréquente. La question qu'il reste à se poser est de définir ce que « ne correspond pas » signifie. Cette notion bien plus subjective qu'il n'y parait est particulièrement difficile à caractériser. Si tout le monde peut se mettre d'accord pour considérer qu'un mouvement à contresens, lorsque celui-ci n'a jamais été aperçu, est anormal, des variations plus fines sont plus délicates à classer.

Ce problème d'évaluation est d'autant plus crucial, qu'il n'existe pas de norme dans la communauté pour évaluer les performances de ce genre d'application. Les travaux en la matière, vus au chapitre 1 font clairement état de systèmes très différents. Ils ne s'appuient pas sur les mêmes informations de bas-niveau, ne s'intéressent pas strictement aux mêmes évènements, fournissent des réponses qui peuvent être continues sur l'image, par blocs ou tout simplement par image, etc. En résumé, établir un protocole expérimental légitime, représentant correctement la réalité de la détection est une tâche très difficile.

5.1.1 Critère d'évaluation

Nous considérons comme anormal, toute déviation du mouvement normal. On peut citer par exemple :

▷ Au niveau d'un péage routier, des véhicules déboîtant d'une file à l'autre.

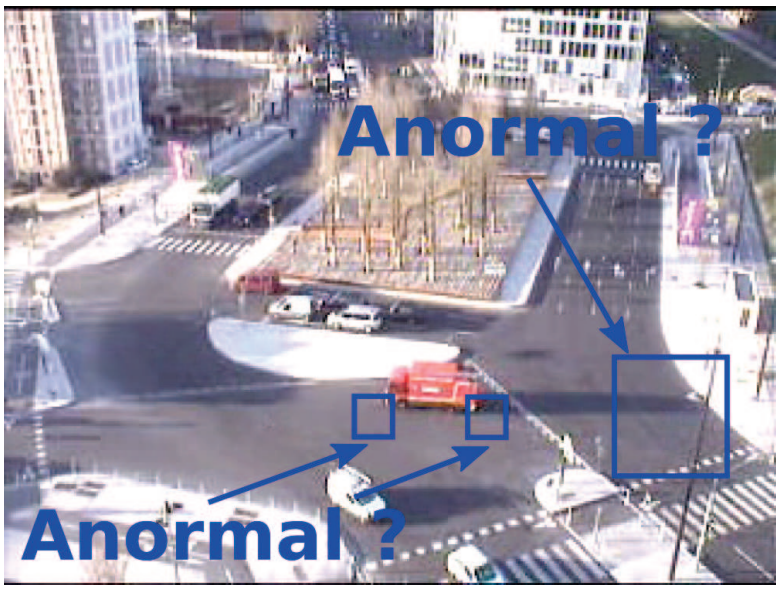


FIGURE 5.1 – Comment définir la vérité terrain ? L'extrémité de l'ombre suit le mouvement du camion de pompier qui circule à contresens. Pourtant dans cette zone de la scène les mouvements vers la gauche ne sont pas anormaux. Doit-on considérer cette zone comme anormale ? Les blocs limitrophes du véhicule, sur lesquels une infime partie du véhicule apparait, doivent-ils aussi être considérés comme anormaux ?

- ▷ Des mouvements à contresens de tous types.
- ▷ Des écarts suspects.

Afin d'évaluer nos résultats de classification, une vérité terrain a dû être établie. Mais là encore, établir cette vérité terrain est une tâche assez subjective. En effet, au vu de notre application, définir avec exactitude quel bloc doit être considéré comme normal ou anormal à chaque instant n'est pas possible comme illustré à la figure 5.1 qui représente la séquence du camion de pompier allant à contresens. L'ombre du camion va aussi à contresens, cependant doit-on la considérer ? Si oui, celle-ci s'étale dans une zone où les mouvements vers la gauche sont normaux, doit-on considérer ces blocs comme anormaux ? Les blocs entourant le camion qui ne contiennent qu'une infime partie de celui-ci, doivent-ils être considérés comme anormaux alors qu'un être humain s'il ne voyait la scène qu'à travers ce bloc ne saurait pas en distinguer le mouvement ?

Tous ces cas litigieux montrent à quel point établir une vérité terrain est délicat. Lorsque l'on demande d'ailleurs, à différents opérateurs de définir une vérité terrain, les différences peuvent être assez surprenantes. Ces différences vont de la variation d'interprétation de certains comportements à l'oubli pur et simple de certains évènements. Ajouté à cela, le fonctionnement des algorithmes eux-mêmes introduit un certain biais sur la position spatiale et temporelle de la détection. Tout d'abord, parce que la caractérisation du mouvement en chaque bloc peut varier selon les paramètres choisis pour le calcul du flot optique ou du descripteur spatio-temporel (la largeur de noyau lors du calcul des gradients par la méthode de Canny par exemple peut étaler plus ou moins le mouvement). Ensuite, nos différentes stratégies pour la mise en place des machines d'apprentissage ont aussi tendance à retarder ou étaler les détections. Les HMMs par exemple, avec leur mécanisme séquentiel donnent une réponse différée de quelques images, ce qui produit un effet « comète » sur les détections (la réponse sur un bloc n'est fournie qu'après quelques images : dans notre cas 5 images). Les chaînes couplées quant à elles, étalent les détections car elles sont susceptibles de réagir à des évènements

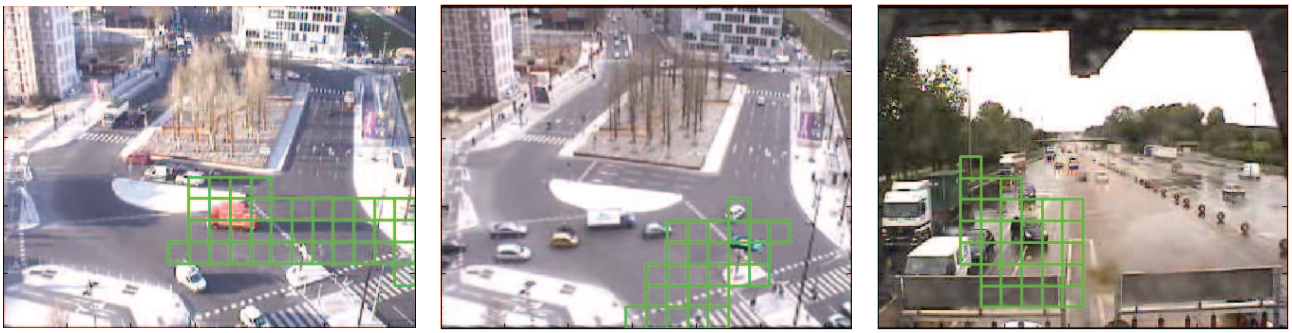


FIGURE 5.2 – Exemple de vérités terrain. Les blocs verts qui représentent les blocs anormaux sont volontairement sélectionnés de manière large pour tenir compte des biais de détection des différentes approches comparées. Les contraintes physiques de la scène sont aussi prises en compte, comme les ombres, car elles sont issues de véhicules possédant des comportements anormaux.

provoqués par une anomalie dans leur voisinage.

Toutes ces détections, bien qu'elles soient légèrement retardées ou étalées n'en sont pas fausses pour autant, elles sont les conséquences d'une anomalie bien réelle. Il est donc nécessaire de définir des vérités terrain capables d'englober l'ensemble de ces détections quelque soit la méthode utilisée. Les vérités terrain ont donc été définies à la main de manière large comme illustré à la figure 5.2. Les blocs considérés comme anormaux sont donc tous ceux entourant le véhicule ou la personne responsable de l'anomalie durant l'ensemble de sa « manœuvre », même aux instants où le mouvement n'est pas assez prononcé pour être considéré comme anormal.

Une fois cette vérité terrain définie, il faut maintenant définir la notion de détection du système. Là encore, les choix sont multiples, plus ou moins légitimes selon ce que l'on cherche à montrer.

- ▷ En termes d'application, un comptage naturel serait un comptage par évènement. Un évènement étant défini par une période de T_{evt} images dans laquelle une entité (personne ou voiture) adopte un comportement anormal. D'après la vérité terrain que nous avons définie, cela correspond à un ensemble de blocs connexes se propageant de manière ininterrompue d'une image à l'autre.
  On considère qu'un évènement est correctement détecté lorsque sur au moins K images successives des blocs sont classés anormaux sur un voisinage, dans la zone de la vérité terrain. Cependant un tel comptage rend difficilement compte de la qualité de la détection et ne permet pas de pleinement percevoir l'apport de chaque brique algorithmique.
- ▷ Un autre comptage plus fin est possible, le comptage par image. Ce comptage est plus à même de différencier les différentes briques algorithmiques puisque la notion de durée d'évènement apparait. Une détection franche sur plusieurs images sera donc plus valorisée qu'une détection éphémère. Il faut cependant faire attention car au vu de la vérité terrain définie de manière large, le rapport entre le temps de détection et la durée de l'anomalie d'après la vérité terrain est faible, ce qui abaisse le **taux de bonnes détections**. Rappellons que cette « souplesse » dans la définition de la vérité terrain ne peut pas être réduite *a priori*.
- ▷ Un comptage par bloc pourrait aussi être envisagé mais, tout comme le comptage par image, les résultats seraient pénalisés par notre manière de définir la vérité terrain. Le rapport entre le nombre de blocs détectés et le nombre total de blocs de la vérité terrain « largement » choisis est bien trop faible pour être significatif et donc permettre de nuancer l'apport des briques algorithmiques.

FIGURE 5.3 – Exemples de conditions climatiques et d'illuminations rencontrées.

A noter que quelque soit la manière de comptabiliser une bonne détection, il est nécessaire de comptabiliser les fausses alarmes de la même manière. Dans le cas d'un comptage par évènement, il est ainsi possible de définir ce que nous appellerons un taux de « dérangement moyen ». Ce taux peut être vu comme la fréquence où un opérateur sera dérangé à tort, ce qui peut être une information particulièrement utile pour ce dernier.

5.1.2 Protocole expérimental

Présentation des bases Les briques algorithmiques ont été évaluées sur 2 bases labellisées. La première base, que nous noterons B_{CVN} (pour Carrefour de la Vache Noire à Arceuil [1]), est issue d'une webcam d'un carrefour routier déjà présenté au chapitre 1. La seconde, que nous noterons B_{PDPP} (pour Péage Dozulé Paris-Province [2]), est aussi issue d'une webcam située au niveau d'un péage d'autoroute. Les vidéos récupérées de ces webcams ont une résolution de 320x240 en 12 images par seconde. Les séquences récupérées sont des vidéos d'une trentaine de secondes acquises à intervalle de 15 minutes. La qualité des vidéos est relativement mauvaise, comme on peut en juger sur les images de la figure 5.3. Les difficultés proviennent aussi bien de la faible résolution, que des conditions climatiques très variables allant d'un temps très ensoleillé à la pluie ou des conditions d'illumination très différentes, de plein jour à la nuit en passant par des reflets de phares sur une chaussée humide, etc. Ceci permet de se placer dans des conditions proches d'une application réelle. Les mouvements normaux associés à chacune des deux bases sont représentés sur les figures 5.4(a) et 5.4(b) par de grandes flèches qui respectent le code couleur introduit au début de ce manuscrit dans la partie notation page 3.

Différentes sous-bases sont constituées pour chacune des scènes : une pour la phase d'apprentissage, une pour la phase validation qui consiste à déterminer le critère de décision, une base comportant des anomalies et enfin une base sans aucune anomalie qui servira à déterminer un taux de dérange-

1. Vidéos obtenues via le site **www.viewsurf.com** propriété de la société **Orange**
2. Vidéos obtenues via le site **www.viewsurf.com** propriété de la société **Autoroutes Trafic**

(a) Base B_{CVN}. (b) Base B_{PDPP}.

FIGURE 5.4 – Vérités terrains des bases de test. Les flèches représentent les directions normales de déplacement.

Bases	Apprentissage	Validation	Test	Sans évènement
B_{CVN}	33	24	135	51
B_{PDPP}	30	40	37	42

TABLE 5.1 – Comptabilisations des séquences vidéo dans chacune des bases utilisées.

ment moyen. Les évènements rencontrés englobent aussi bien les mouvements à contresens, que les marche-arrières, les écarts de trajectoire, les changements de file sur l'autoroute à l'approche du péage, etc. Le détail du nombre de vidéos dans chacune des sous-bases est récapitulé dans le tableau 5.1.

Création de la base de synthèse Pour la sous-base d'évènement de B_{CVN}, afin d'avoir suffisamment d'évènement anormaux pour pouvoir établir des statistiques viables, des évènements de synthèse ont été générés. En incrustant sur des séquences sans anomalie, des formes texturées selon une trajectoire anormale, un panel de comportements dangereux a été créé. L'incrustation n'est pas photo-réaliste car aucune modélisation 3D de la scène de base n'était disponible. En revanche, elle respecte la perspective de la scène comme on peut le constater sur les images de la figure 5.5. Grâce à ce procédé, 15 trajectoires anormales ont été définies et pour chaque trajectoire 9 textures différentes ont été appliquées sur l'objet en déplacement.

Outils d'évaluation Pour la comparaison des méthodes, nous nous appuyons sur une représentation via des courbes ROC [3]. Ces courbes sont les plus couramment utilisées pour évaluer les performances de classification. Un descriptif des courbes ROC est fourni en annexe A.

Pour mieux se rendre compte de la répartition temporelle des détections, des chronogrammes bruts sont utilisés. Un exemple est fourni sur la figure 5.6. Cette représentation permet d'intégrer sur la dimension spatiale de la détection en représentant seulement les temps où au moins un bloc a été classifié comme anormal. La partie en jaune représente les instants anormaux d'après la vérité terrain,

3. *Receiver Operating Curves*.

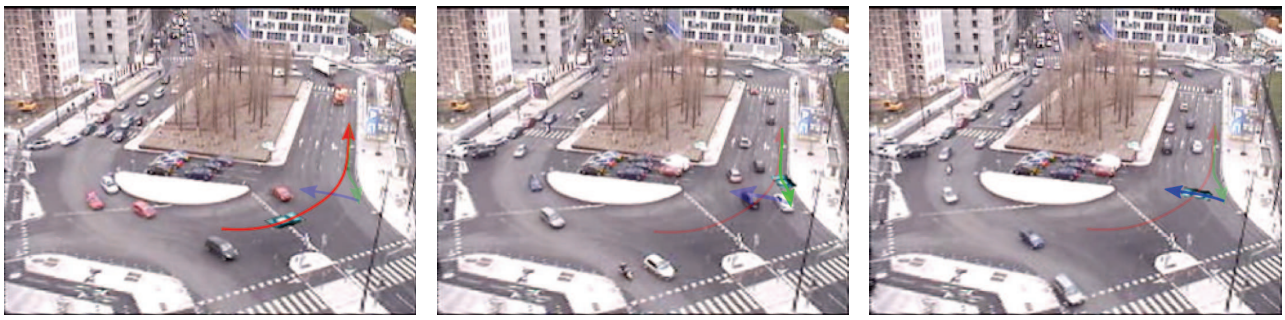


FIGURE 5.5 – Exemples de réalité augmentée générée pour constituer une base de test d'évènements. La trajectoire est définie par une spline par morceaux, le long de laquelle une texture est incrustée en respectant une transformation homographique permettant de prendre en compte la perspective de la scène.

les fronts rouges représentent les instants où au moins un bloc a été classifié à juste titre comme anormal (à noter que ces fronts rouges ne peuvent être présent que dans les parties jaunes) et les fronts verts correspondent aux instants ou au moins un bloc a été classifié à tort comme anormal. Le chronogramme en question a été obtenu à partir d'un seuil de détection assez permissif. Comme on peut le constater, de nombreuses fausses alarmes peuvent apparaître.

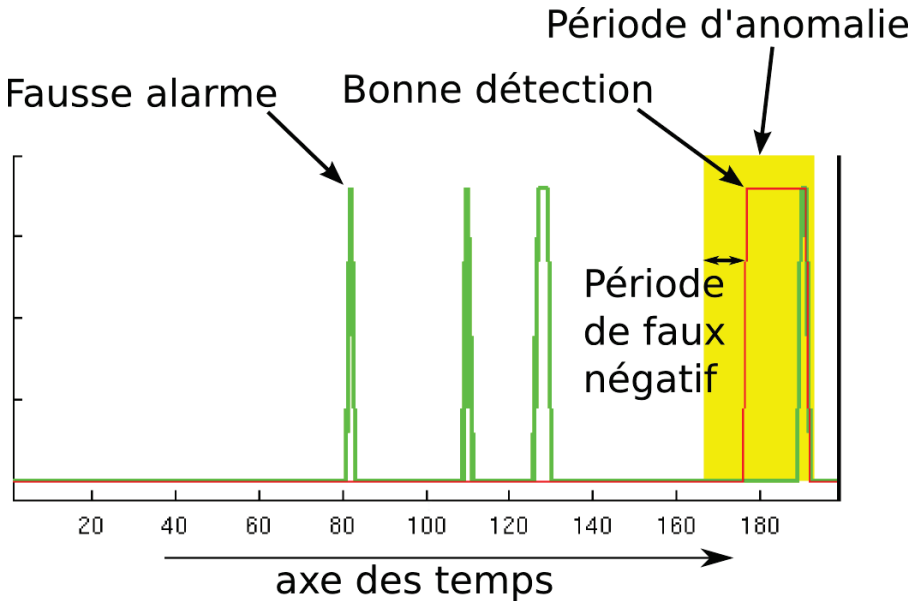


FIGURE 5.6 – Exemple de chronogramme. Les zones jaunes représentent les instants anormaux au sens de la vérité terrain. Les fronts verts représentent les instants où au moins un bloc de la vérité terrain est considéré comme anormal. Les fronts rouges représentent les instants où au moins un bloc n'appartenant pas à la vérité terrain est considéré comme anormal.

Critère d'évaluation La nature de la classification binaire dans notre cas sera une classification par image. Comme pour le chronogramme 5.6, un vrai positif sera comptabilisé lorsqu'à un instant t au moins un bloc a été classifié à juste titre comme anormal et un faux positif lorsqu'à un instant t au moins un bloc a été classifié à tort comme anormal. Le paramètre de classification à faire varier sera, non pas le seuil de détection brut mais le quantile de l'algorithme de confiance présenté au chapitre

3. Pour valider ce choix, l'apport du seuil basé sur l'algorithme de confiance par rapport à un seuil de classification statique identique pour tous les blocs, est illustré à la figure 5.7. Les deux bases ont été utilisées pour la comparaison ; les courbes vertes représentent la classification obtenue avec un seuil statique et les courbes rouges, celles obtenues avec le critère de confiance. Ces courbes ont été réalisées avec le même descripteur (CSD), la même machine d'apprentissage (SKDE) et le même jeu de paramètres pour cette dernière. Les seules différences sont :

- ▷ Seuil basique : le seuil de classification pour chaque bloc a été fixé de manière empirique et égal pour tous.
- ▷ Seuil de confiance : les seuils de classification sont déterminés pour chaque bloc grâce à l'algorithme de confiance. Le paramètre de la courbe ROC sera le quantile $\mathcal{Q}$.

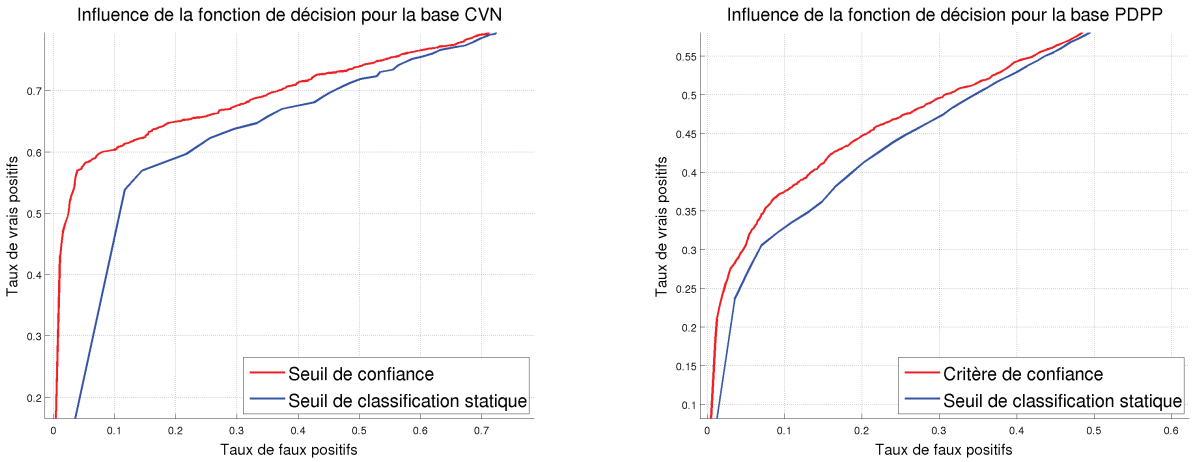


FIGURE 5.7 – Courbes ROC pour la comparaison de performance de la fonction de décision. La comparaison est effectuée sur les bases B_{CVN} et B_{PDPP} avec le descripteur CSD et le SKDE. Les courbes vertes sont associées à un seuil empirique pour tous les blocs,les courbes rouges sont obtenues avec un seuil de classification par bloc déterminé grâce au critère de confiance.

On constate que l'apport du calcul de seuil automatique via le critère de confiance par rapport à un seuil empirique est réellement positif en terme de performance. Ceci confirme bien l'idée que le seuil déterminant si un descripteur est conforme au modèle dépend de la forme de la distribution.

On peut remarquer sur la courbe ROC que les performances de classification semblent modestes comme on pouvait le présager sur le chronogramme 5.6. Ceci est lié à la définition large de la vérité terrain. Aussi ces courbes ROC doivent exclusivement servir à **comparer l'apport des différentes briques algorithmiques**.

5.2 Expérimentations

Le nombre de briques algorithmiques et de variantes à chacune d'entre elles sont très nombreuses. Pour rappel, il est nécessaire de comparer :

- ▷ Le descripteur issu du flot optique / le descripteur CSD.
- ▷ Les deux stratégies de construction des chaînes de Markov cachées simples et couplées.
- ▷ Les deux méthodes d'apprentissage des chaînes de Markov cachées simples et couplées.
- ▷ Les deux stratégies de couplage des chaînes couplées.

▷ Les trois machines d'apprentissage mises en place tout au long de ce manuscrit, à savoir le SKDE, les HMMs simples et les HMMs couplées.

Une comparaison exhaustive des résultats de classification de tous les couplages de *descripteur/machines d'apprentissage* sous toutes leur variantes n'est pas possible. Les comparaisons sont donc réalisées étape par étape en terme de performance de classification sans réel rattachement aux performances applicatives en temps que telles, afin de déterminer de manière incrémentale les meilleurs approches. À la fin de ce chapitre, seules les associations de méthodes les plus pertinentes sont comparées, cette fois-ci en termes de classification mais aussi et surtout en termes de performances de détection d'évènements plus significative pour un utilisateur.

Pour l'ensemble des tests réalisés par la suite, les blocs de la grille d'analyse du système ont une taille de 16x16. Cette taille est un compromis entre les résolutions et la taille des acteurs des différentes scènes étudiées.

5.2.1 Evaluation des descripteurs

La comparaison de performance entre les descripteurs a été réalisée en se servant de la machine d'apprentissage SKDE car elle est la plus simple et la plus souple à utiliser. Les mouvements prépondérants par bloc du flot optique, que nous appellerons par la suite « OF », ont été comparés au descripteur CSD proposé. Les descripteurs OF sont obtenus en déterminant par bloc le mouvement avec la vraisemblance maximale au sein de ce bloc grâce à l'algorithme SKDE, comme présenté au chapitre 3.

Concernant la paramétrisation pour le calcul des descripteurs spatio-temporels, les cuboïdes ont une taille de 16x16x5. La profondeur temporelle des descripteurs spatio-temporels a été expérimentalement choisi égale à $T = 5$ afin de filtrer suffisamment, tout en assurant autant que possible que les mouvements contenus dans les cuboïdes soient élémentaires. La variance des gaussiennes pour le calcul des gradients de Canny est choisie égale à 1. A noter que pour limiter l'effet comète, les descripteurs spatio-temporels sont recalés par rapport à l'image centrale du cuboïde dont ils sont issus.

Des jeux de paramètres comparables pour les deux descripteurs ont été choisis pour le SKDE afin d'avoir une comparaison légitime. Ainsi, dans le cas du descripteur OF, la variance choisie pour les noyaux est de 0.47 et de 0.3 dans le cas du descripteur CSD. Dans les deux cas, ces variances couvrent la même portion de l'intervalle dans lequel les distances associées prennent leurs valeurs. En effet, dans le cas du descripteur OF, la distance angulaire entre vecteur déplacement définie par l'équation (2.17) du chapitre 2 est à valeur dans $[0, \pi]$ et la distance associée au descripteur CSD définie à l'équation (2.33) est à valeur dans $[0, 2]$. Les écarts-type sont choisies de manière à avoir $\frac{\pi}{0.47} \approx \frac{2}{0.3}$. Ce choix est justifié car il a été vu au chapitre 3 que la largeur du noyau est un paramètre critique pour l'approximation de densités quelconques. Ici ce paramètre à une signification physique bien précise. Il représente la taille de l'influence angulaire d'une observation, définissant la normalité associée à cette observation. A noter que pour le descripteur OF, la variance des noyaux utilisée pour déterminer le mouvement prépondérant est aussi choisie égale à 0.47. Le quantile d'arrêt de la procédure d'approximation en phase d'apprentissage est choisi égal à $\mathcal{Q}l = 0.99$

On constate sur la courbe ROC de la figure 5.8, un apport net en terme de performance de classification pour l'utilisation du descripteur CSD. Ce gain de performance est essentiellement dû à la diminution du nombre de fausses alarmes causées par le problème d'ouverture. De plus, l'intégration temporelle du descripteur CSD permet une caractérisation du mouvement plus continue et par

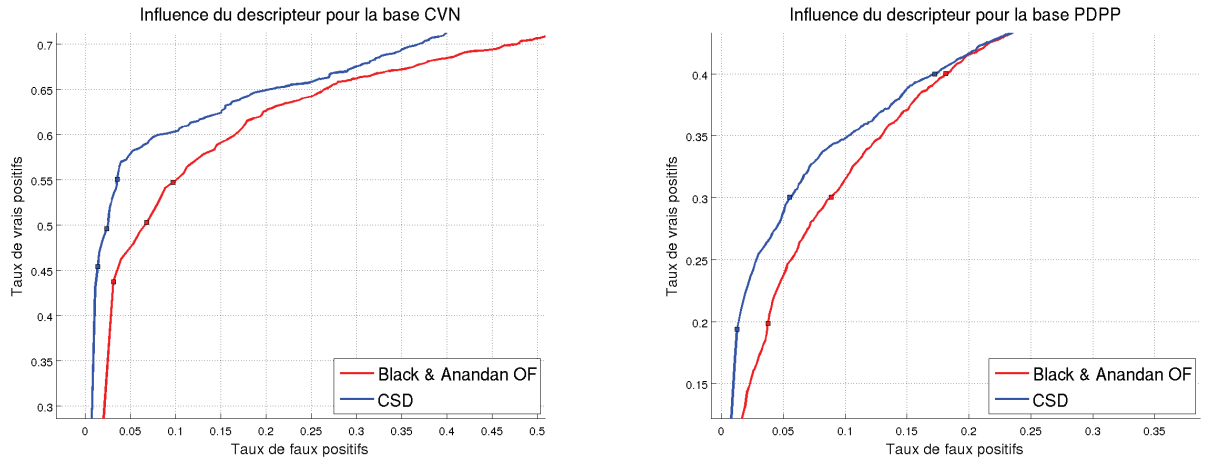


FIGURE 5.8 – Courbes ROC pour la comparaison de performance des descripteurs. La comparaison est effectuée sur les bases B_{CVN} et B_{PDPP} avec le SKDE et le critère de confiance pour la détermination du seuil de classification. Les courbes rouges sont obtenues avec le descripteur OF. Les courbes bleues sont issues de l'utilisation du descripteur CSD.

la même occasion d'éviter des erreurs de caractérisation ponctuelle qui se traduisent en général par des fausses alarmes pour le descripteur OF. On peut mieux se rendre compte de ce filtrage sur les chronogrammes de la figure 5.9 et 5.10 qui représentent la répartition des bonnes détections et des fausses alarmes pour quelques séquences de la base concernée. Les quantiles de détection sont choisis de manière à avoir un taux de vrais positifs égal pour les différentes approches (représenté par des points sur les courbes ROC de la figure 5.8).

Les évaluations réalisées dans cette section permettent de montrer que le descripteur CSD proposé dans ce manuscrit surclasse les approches basées sur le flot optique, grâce à une intégration temporelle et un traitement des cas dégénérés qui permet de filtrer des erreurs d'estimation qui se traduisent très souvent par une erreur de classification. Dans la suite des tests, c'est le descripteur CSD proposé qui sera utilisé.

5.2.2 Utilisation des chaînes de Markov

Concernant la brique algorithmique des HMMs, il a été vu au chapitre 4 que différentes méthodes d'apprentissage ainsi que différentes stratégies de création des chaînes pouvaient être utilisées pour l'adaptation des HMMs et part conséquences des CHMMs dans notre application.

Pour rappel, deux stratégies de création sont possibles. La première consiste à utiliser des densités d'émission continues avec un *a priori* sur le nombre d'état et sur la forme de ces densités d'émission. La seconde consiste à utiliser des densités discrètes permettant de bénéficier complètement du mécanisme d'apprentissage des chaînes de Markov et de ne faire aucun *a priori* sur le nombre d'état, celui-ci étant égal au nombre de symboles de l'espace discret (8 dans notre cas).

Les deux méthodes d'apprentissage que nous évaluons, sont la traditionnelle approche de Baum-Welch et le *Viterbi Path Counting*.

En raison de ses performances, le descripteur retenu est le descripteur CSD. On peut constater sur les courbes de la figure 5.11 que pour les deux bases, les conclusions sont les mêmes. Tout d'abord, la création des chaînes grâce à des distributions discrètes donne de meilleurs résultats qu'avec des densités continues. Restreindre le mécanisme d'apprentissage des HMMs avec des densités d'émis-

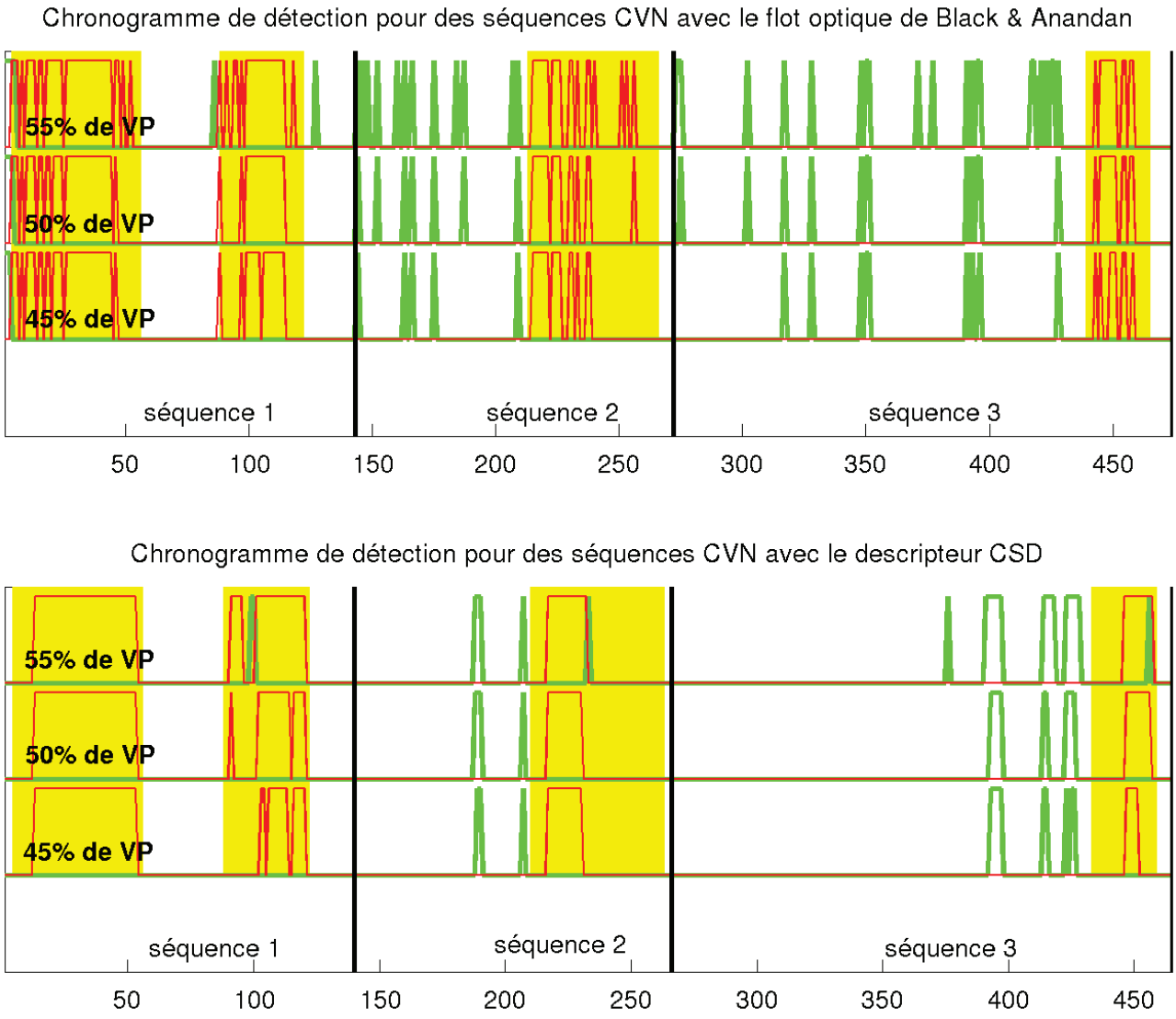


FIGURE 5.9 – Chronogramme de comparaison de l'influence du descripteur pour la base B_{CVN}, associés aux courbes ROC de la figure 5.8. Les détections avec le flot optique apparaissent essentiellement sous forme de diracs synonymes de détections très hachées. Le descripteur CSD en revanche filtre la détection pour la rendre plus continue gommant au passage beaucoup de fausses alarmes.

sion fixes comme proposé par Kratz et Nishino (2009), diminue les performances de classification par rapport à un apprentissage complet même si celui-ci est réalisé sur des densités d'émission plus grossières. L'autre conclusion qu'il est possible de tirer de ces courbes concerne la méthode d'apprentissage à utiliser. Bien que le *Viterbi Path Counting* donne de meilleurs résultats pour ce qui est de mettre en avant les anomalies de mouvement dans un voisinage proche comme on a pu le constater dans le chapitre 4 sur un exemple précis, à partir d'évaluation plus conséquente on constate que la tendance s'inverse nettement pour des évènements plus classiques. La réestimation complète de la chaîne via l'algorithme de Baum-Welch permet une modélisation beaucoup plus robuste et fiable que celle du *Viterbi Path Counting* (cf. partie 4.3.4).

Les conclusions tirées dans cette section ont été obtenues grâce aux HMMs simples. Elles restent vraies pour les chaînes couplées. En effet, les chaînes couplées reposent fondamentalement sur le

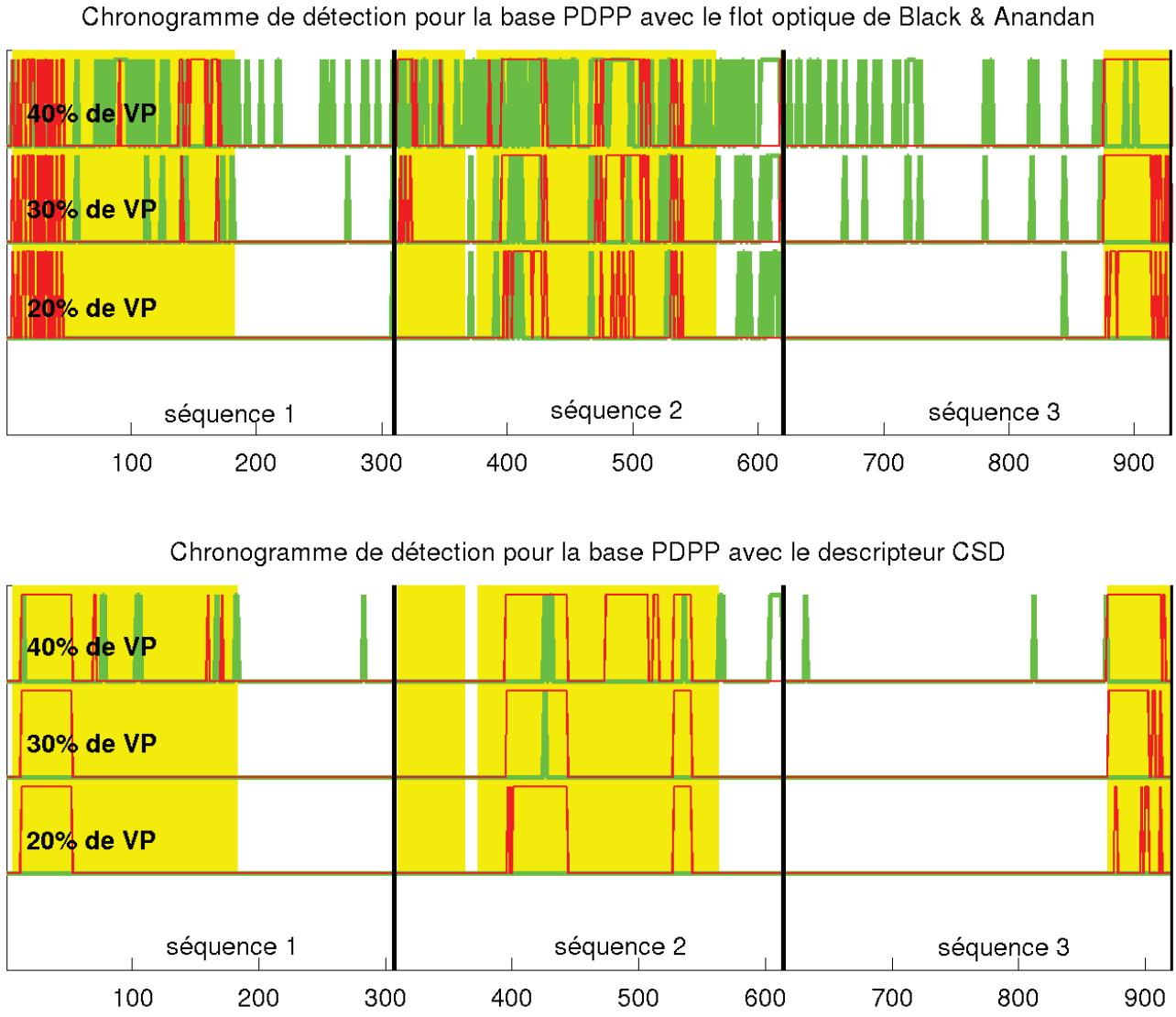


FIGURE 5.10 – Chronogramme de comparaison de l'influence du descripteur pour le base B_{PDPP}, associés aux courbes ROC de la figure 5.8. Ici encore, le descripteur CSD montre une détection beaucoup plus continue avec beaucoup moins de fausses alarmes.

même formalisme que les HMMs simples, seul l'ajout de transitions croisées supplémentaires les différencient. En conséquence dans la suite des évaluations, la création des chaînes simples ou couplées est exclusivement réalisée avec des densités discrètes et leur entrainement effectué via l'algorithme de Baum-Welch.

5.2.3 Stratégie de couplage

Les deux méthodes de couplages sont implémentées dans les deux cas pour les mêmes quatre voisins de chaque bloc. On peut constater sur la courbe 5.12 qu'en terme de classification, les deux approches se valent. En pratique, le couplage spatial à essentiellement vocation à filtrer les détections en tenant compte du comportement avoisinant. Cette fonction de filtrage, au vu des flux laminaires considérés, fonctionne aussi bien avec la méthode « 4x2 » qui est un mécanisme de couplage faible.

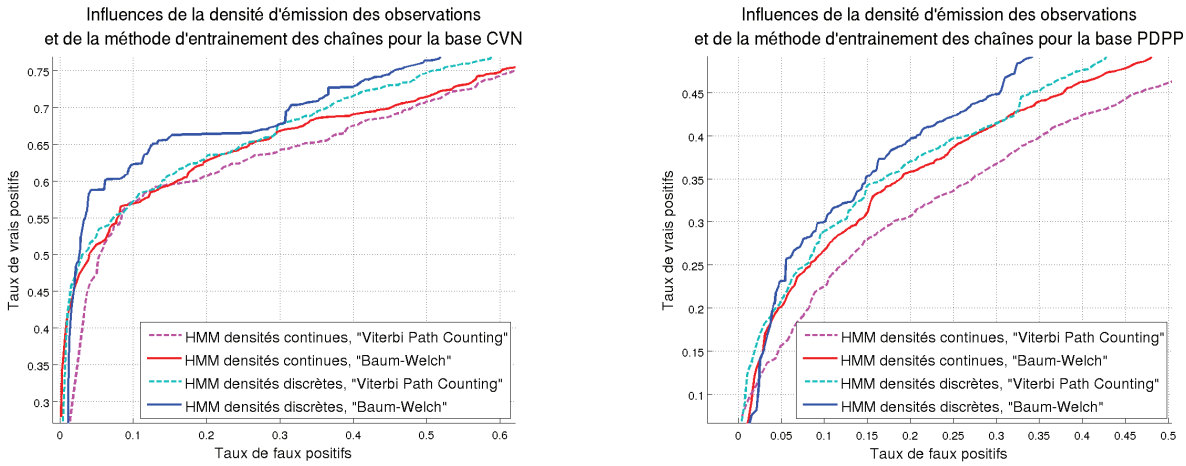


FIGURE 5.11 – Courbes ROC pour la comparaison de performance des différentes manières d'entraîner les HMMs. La comparaison est effectuée sur les bases B_{CVN} et B_{PDPP}. Les courbes en trait plein sont obtenues avec l'algorithme de Baum-Welch est celles en tiret avec l'algorithme *Viterbi Path Counting*. Les courbes bleues représentent des chaînes construites grâce à des densités d'émission discrètes alors que la courbe rouge et magenta correspondent à des densités continues.

En revanche, le couplage asymétrique « 1-N » met en œuvre moins de HMM comme il a été vu au chapitre 4, permettant d'économiser du temps de calcul aussi bien à l'apprentissage que pendant la phase de test comme nous le verrons plus loin.

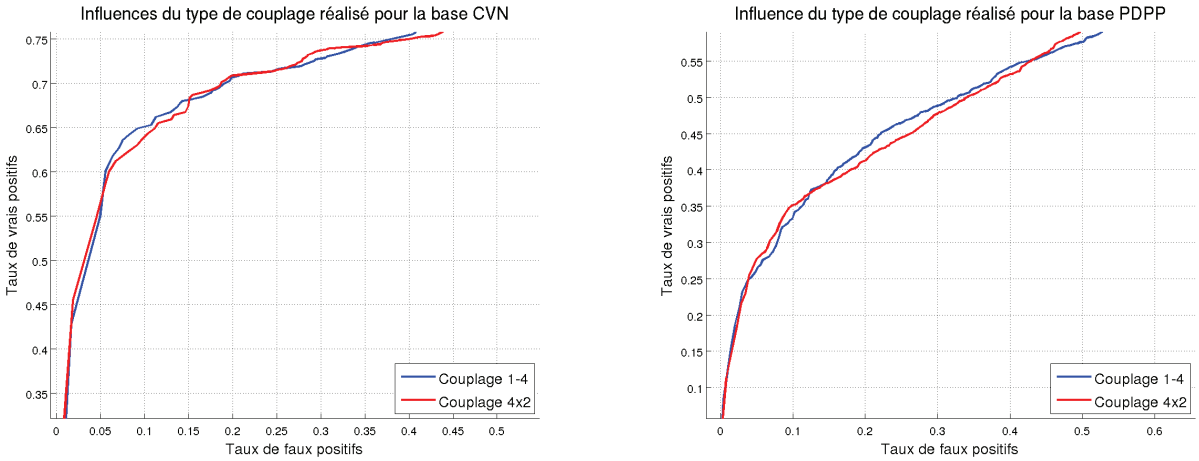


FIGURE 5.12 – Courbes ROC pour la comparaison des stratégies de couplage des chaînes. Les courbes bleues sont obtenues avec un couplage total dissymétrique de type 1-N alors que les courbes rouges représentent les résultats obtenues pour quatre couplages symétriques entre deux chaînes.

Au travers des différentes comparaisons qui ont été menées jusqu'ici, seules quelques combinaisons de méthodes ont été conservées. Dans tous les cas, c'est le descripteur CSD qui est utilisé avec le seuil de confiance. Pour les réseaux bayésiens, HMMs simples ou couplées, ce sont des densités d'émission discrètes qui sont utilisées avec ce que cela implique (cf. chapitre 4), et les chaînes sont entrainées via l'algorithme de Baum-Welch. Enfin pour les chaînes couplées, le couplage asymétrique « 1-N » est utilisé.

5.2.4 Comparaisons applicatives

Les évaluations réalisées maintenant visent à comparer les performances obtenues avec les différentes machines d'apprentissage mises en place tout au long de ce manuscrit, chacune des machines étant utilisée dans sa version la plus performante comme expliqué précédemment. Ces évaluations sont effectuées, tout d'abord avec un comptage image par image afin d'estimer les performances pures de classification puis avec un comptage évènementiel qui permet de rendre mieux compte des performances dans le cadre applicatif. Enfin, des résultats qualitatifs terminent cette comparaison de méthodes en montrant des exemples précis de détection.

5.2.4.1 Évaluations quantitatives

Comptage par image La comparaison en terme de classification des différentes machines d'apprentissage qui ont été présentées dans ce manuscrit est représentée à la figure 5.13. On constate sur la base synthétique B_{CVN} que plus les méthodes assurent de cohérence (temporelle ou spatiale) plus les résultats de classification sont bons. Ainsi les chaînes couplées surpassent les HMMs qui dépassent elles-mêmes le SKDE. On peut remarquer que pour des taux de fausses alarmes très faibles, les chaînes couplées sont légèrement inférieures aux autres méthodes car la cohérence spatiale a tendance à lisser et renforcer la vraisemblance de certaines fausses alarmes favorisant leur détection même pour des seuils de détection très lâches. Nous reviendrons sur ce phénomène de manière plus claire avec le comptage par évènements. Concernant la base B_{PDPP}, les résultats sont moins flagrants. Sur cette base réelle ou les trajectoires sont souvent rectilignes, les différentes cohérences intégrées ne permettent pas de montrer un réel gain. On constate que le SKDE permet d'obtenir des résultats meilleurs que les HMMs alors que les chaînes couplées parviennent à surpasser le SKDE pour des seuils de détection plus stricts.

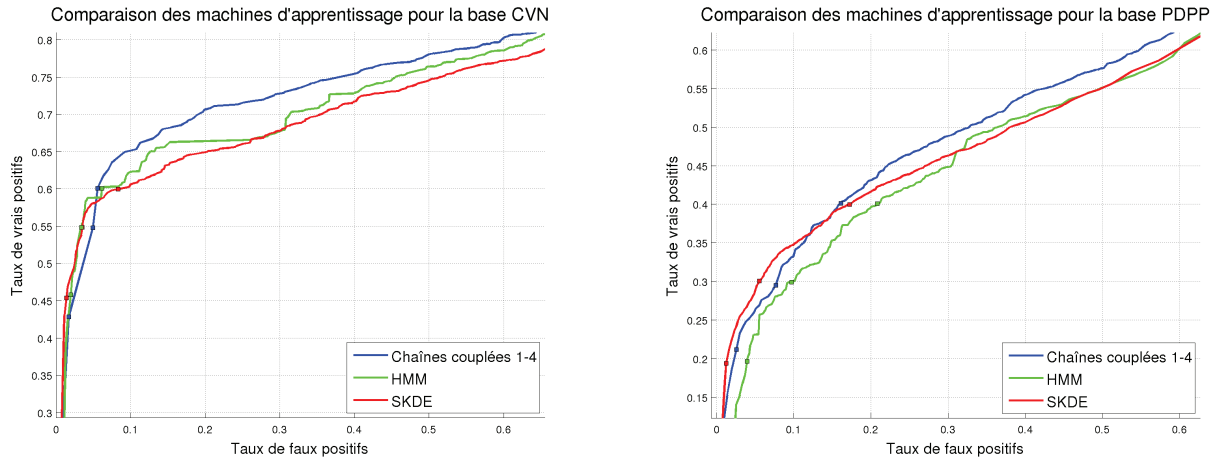


FIGURE 5.13 – Courbes ROC pour les différentes machines d'apprentissage chacune évaluées dans leur meilleur configuration. Les courbes bleues sont obtenues avec un couplage total dissymétrique de type 1-N, les courbes vertes avec des HMMs simples alors que les courbes rouges sont obtenues avec le SKDE.

Une combinaison des machines d'apprentissage ? Une question légitime est de savoir si les détections des différents systèmes sont complémentaires. Pour vérifier cela, une étude des chronogrammes

de détection est réalisée. On constate sur les figures 5.14 et 5.15 que les détections pour chacune des méthodes, qu'elles soient bonnes ou mauvaises, s'effectuent à peu près au même instant. Par exemple, on peut voir sur la figure 5.14, pour la séquence 2, trois instants bien précis avant l'anomalie sont susceptibles d'être anormaux. Les instants aux alentours de l'image 220, 250 et 270. La différence entre les méthodes se manifeste ensuite sur leur capacité à bien classifier les anomalies à ces instants. Rappelons que la vérité terrain a été définie de manière large et qu'en conséquence, certains instants considérés comme anormaux par l'homme ne le seront pas par la machine quelque soit le descripteur ou la méthode de classification choisie et ce pour des raisons très variées (mouvement de trop faible amplitude, variation de la trajectoire masquée par la géométrie de la scène, etc.). On peut tout de même remarquer que de manière générale les bonnes détections obtenues grâce aux chaînes couplées englobent les détections obtenues avec les autres méthodes.

Les chronogrammes permettent aussi de confirmer les résultats énoncés à partir des courbes ROC de la figure 5.13. L'ajout d'une modélisation temporelle voir spatiale de l'évolution de mouvement permet dans l'ensemble de diminuer le nombre de fausses alarmes ponctuelles, de lisser les détections mais aussi en contrepartie de faire de même pour les fausses alarmes plus conséquentes. On peut s'en rendre compte sur la séquence 2 et 3 de la base B_{PDPP} (figure 5.15) où quelques fausses détections ponctuelles obtenues avec la méthode SKDE sont gommées avec la méthode des chaînes couplées. Les bonnes détections sont lissées comme on peut le voir sur la première anomalie de la séquence 2 entre les images 310 et 420. En revanche, certaines fausses détections sont aussi lissées, ce qui peut être assez gênant comme nous allons le voir par la suite.

Comptage par évènement Les évaluations menées jusqu'ici ont eu pour but de refléter la qualité de la classification avec des détections les plus continues possibles, les plus nettes possible, des fausses alarmes les moins fréquentes possible, etc. Cependant, au vu de l'application finale visée, à savoir la vidéo-protection d'espaces publics, la composante utilisateur doit être prise en compte. En effet, un système qui alerte à tort un utilisateur d'une menace trop régulièrement est un système que l'on éteint. En conséquence, une évaluation en termes d'évènements et de dérangements a été réalisée.

Comme annoncé au début de ce chapitre, un comptage par évènement consiste à identifier durant la durée de l'anomalie au moins un ensemble de blocs connexes se propageant de manière ininterrompue d'une image à l'autre caractérisant le véhicule ou la personne concerné. Au vu des courbes ROC et des chronogrammes présentés précédemment, on s'aperçoit bien que des fausses alarmes persistent. Ce qui différencie ces fausses alarmes des véritables détections est en général la longueur de la détection. Ainsi, une détection d'évènements est considérée lorsque K images successives ont mis en évidence des blocs anormaux d'un même voisinage.

Ce type de comptage est représenté sur les courbes des figures 5.16(a) et 5.16(b) en trait plein. Les couleurs des courbes sont les mêmes que celles utilisées sur les courbes ROC de la figure 5.13. Pour ne pas fixer de manière empirique un paramètre supplémentaire, l'évolution des détections évènementielles est représentée en fonction du paramètre K. Le choix des quantiles de classification est réalisé à partir des courbes ROC de la figure 5.13, de manière à avoir un taux de vrai positif constant dans un comptage image par image, pour les trois méthodes comparées.

On remarque que les différentes méthodes sont assez proches les unes des autres et détectent un assez fort pourcentage des évènements à détecter, de 60 à 90 % (145 évènements au total pour la base B_{CVN} et 45 pour la base B_{PDPP}).

Outre les performances de détection des différents systèmes, une mesure du taux de dérangement est mise en place. Un faux évènement sera comptabilisé si K images successives ont mis en évidence

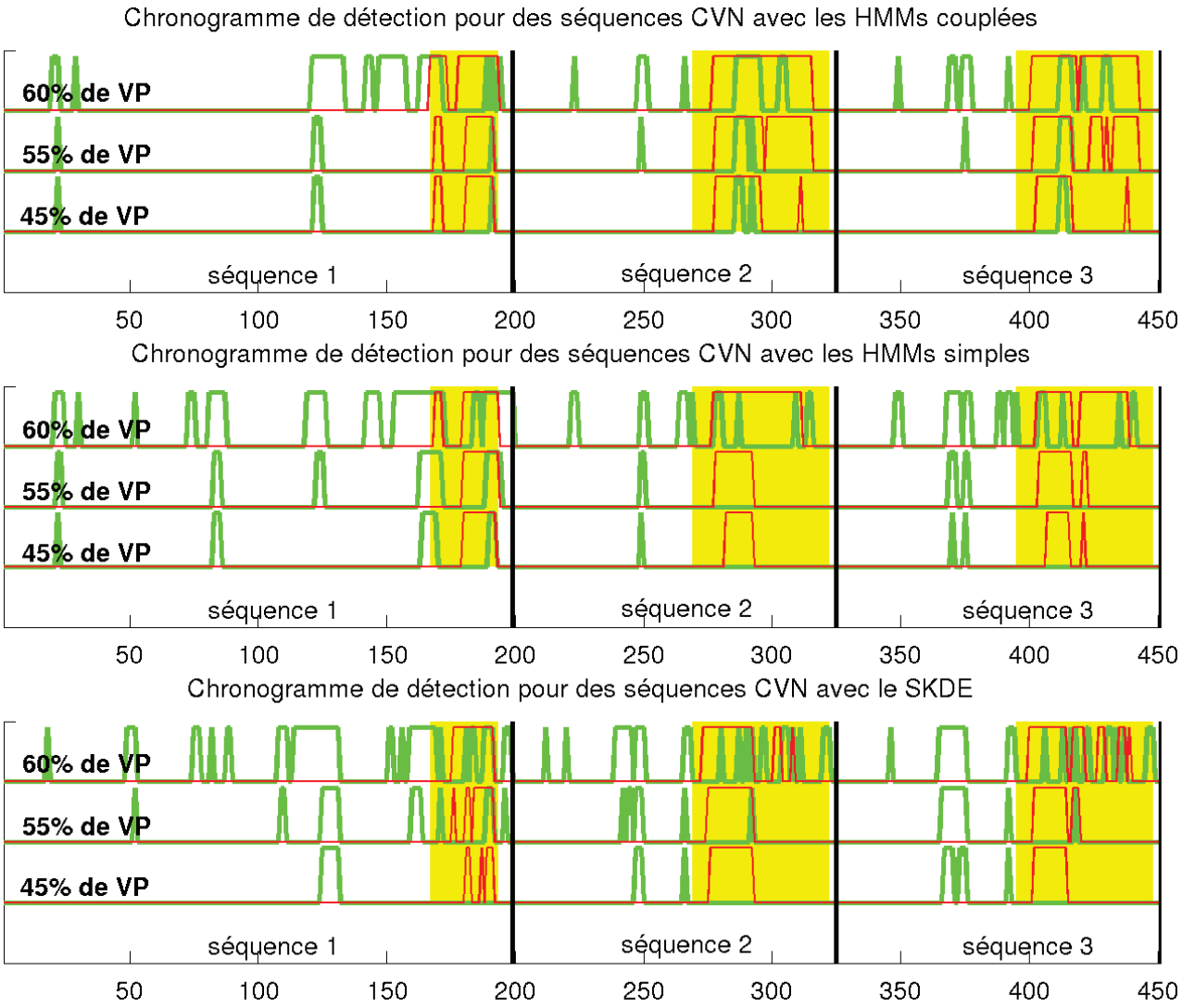


FIGURE 5.14 – Chronogramme de comparaison des différentes machines d'apprentissage pour la base B_{CVN}, associés aux courbes ROC de la figure 5.13. L'ajout de cohérence (temporelle et spatiale) a tendance à lisser la détection, gommer certaines fausses alarmes rémanentes mais aussi renforcées d'autres fausses alarmes.

des blocs anormaux d'un même voisinage à tort dans une période de temps fixe. La période de temps est fixée en rapport avec la durée moyenne des évènements de la vérité terrain pour chacune des bases (4 secondes pour la base B_{CVN} et 7 pour la base B_{PDPP}). Ce taux de dérangement est représenté par les courbes en tirets sur les figures 5.16(a) et 5.16(b).

On constate sur la figure 5.16(a) que pour la base B_{CVN}, la méthode avec les chaînes couplées obtient le taux de détection le plus élevé des trois méthodes. Ce taux est aussi particulièrement stable quelque soit le quantile de détection choisi contrairement aux autres approches. Cette stabilité est aussi vérifiée pour le taux de dérangement qui comme énoncé auparavant, est plus élevé pour les HMMs couplées avec des quantiles de détection très lâches. Ceci est la conséquence du lissage de certaines fausses alarmes par l'ajout des différentes cohérences qui ne sont plus filtrées par le critère des K détections successives. Pour des quantiles de détection plus stricts, la méthode des HMMs couplées devient la méthode ayant le plus faible taux de dérangement avec le meilleur taux de bonne

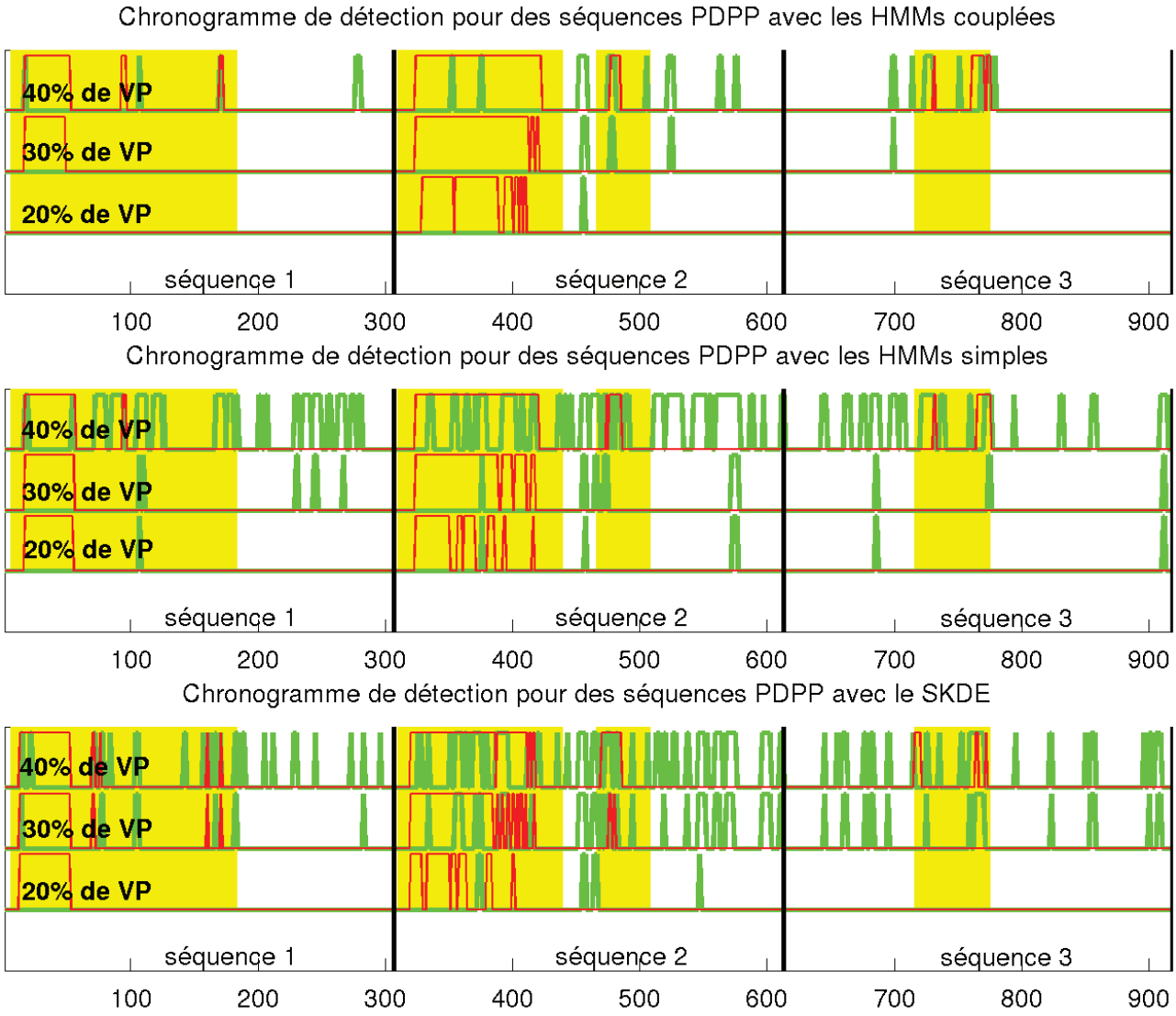
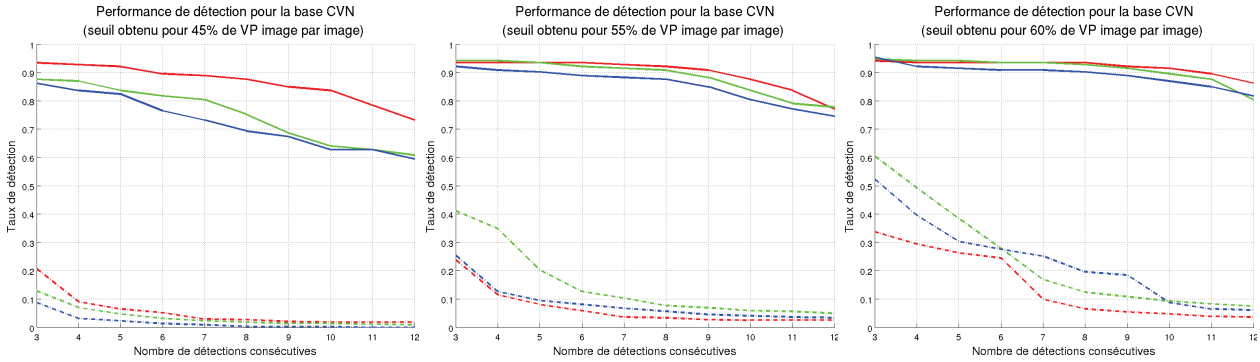
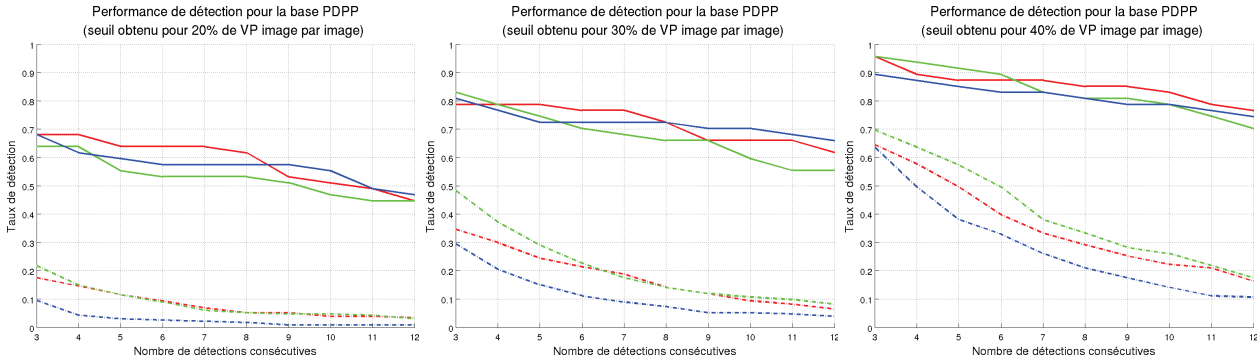


FIGURE 5.15 – Chronogramme de comparaison des différentes machines d'apprentissage pour la base B_{PDPP}, associés aux courbes ROC de la figure 5.13. L'apport de la cohérence temporelle et spatiale se remarque essentiellement au niveau du filtrage des fausses alarmes ponctuelles.

détection. Dans une optique de détection évènementielle maximale, il est possible d'obtenir plus de 90% de bonnes détections pour moins de 3 % de taux dérangement ($K = 9$ avec un taux de VP image par image de 55%), ce qui représente en moyenne $0.03 \times \frac{3600}{4} = 27$ alertes erronées pour une heure de détection sans anomalies. Dans cette configuration, l'objectif est de déterminer le maximum d'évènements, même les plus fugaces. Il en résulte un taux de dérangement qui peut sembler important avec une détection en moyenne toute les deux minutes. Dans un cadre applicatif légèrement différent, où l'objectif serait non plus la détection temps réel mais l'indexation de vidéos d'évènements anormaux en vu d'une phase de rejeu des enregistrements *a posteriori*, ce taux de bonnes détections serait le critère prédominant et les 27 fausses alertes par heure ne seraient alors, plus si pénalisantes. Pour en revenir au cadre applicatif qui est le nôtre, il est aussi possible de rechercher le taux de dérangement minimal tout en conservant la détection des évènements les plus grossiers. Dans ce cas, l'approche SKDE avec un quantile de détection permissif est plus adaptée puisqu'il est



(a) Base B_{CVN}.



(b) Base B_{PDPP}.

FIGURE 5.16 – Taux de détection évènementielle et taux de dérangement. Les courbes rouges, vertes et bleues sont respectivement associées aux HMMs couplées, aux HMMs simples et aux SKDE. Les taux sont représentés pour différents quantiles de classification de plus en plus strict. Ces quantiles sont déterminés à partir des courbes ROC de la figure 5.13 de manière à avoir le même taux de vrai positif.

possible d'obtenir un taux de détection de 70% pour un taux de dérangement très faible de 0.2% ($K = 8$ avec un taux de VP image par image de 45%), ce qui représente en moyenne moins de 2 alertes par heure.

Pour la base B_{PDPP}, comme pour les courbes ROC de la figure 5.13, les résultats sont moins marqués. Les HMMs couplées permettent toujours d'obtenir un taux de bonnes détections légèrement supérieur aux autres méthodes mais aussi un taux de dérangement plus élevé que le SKDE. Au vu de ces courbes, la méthode SKDE est à préconiser pour ce type de scène car il permet d'obtenir un faible taux de dérangement avec un taux de détection très proche de celui des HMMs couplées. Cette scène de péage ne montre pas de mouvements très complexes contrairement à la scène de la base B_{CVN}. Les mouvements par bloc sont relativement monomodaux et dans ces conditions l'apport de la cohérence spatiale des HMMs couplées n'est pas réellement significative.

A noter que nous n'avons pas insisté sur les résultats des HMMs simples car ceux-ci sont quasiment toujours inférieurs aux HMMs couplées que ce soit en terme de détection ou de dérangement. Rappelons que les HMMs simples ont été introduites uniquement comme une étape dans la mise en place les HMMs couplées.

On peut donc conclure que l'approche SKDE ou celle basée sur les chaines couplées sont deux approches viables pour répondre à la problématique de vidéo assistance d'espace possiblement dense. L'utilisation de l'une ou l'autre méthode sera plutôt guidée par les besoins utilisateur finaux. En effet,

si le but recherché est le meilleur taux de détection possible, les chaînes couplées sont plus adaptées. Au contraire, pour un taux de dérangement le plus faible possible tout en conservant les détections les plus grossières, la méthode SKDE est à choisir.

Temps de calcul Parce que l'application a pour but final l'analyse en temps réel, une discussion sur les temps de calcul est effectuée ici. Calculés pour la base B_{CVN}, les temps d'apprentissage sont donnés en minutes à titre indicatif seulement. Pour rappel, une grille de 300 blocs est utilisée pour la scène CVN. 33 séquences de 30 secondes encodées en 12 images par secondes pour un total de 12375 images ont servi de base à l'apprentissage. Or comme le non mouvement n'est pas pris en compte, le nombre moyen de primitives réellement apprises par bloc est de 2000 pour les algorithmes SKDE et HMMs simples. Ce nombre est plus important pour les HMMs couplées à cause du couplage qui impose l'ajout de séquences sans mouvement pour la synchronisation (cf. chapitre 4).

On peut constater, tout d'abord, sur le tableau 5.2 qu'en phase d'évaluation, tous les algorithmes fonctionnent en temps réel (les chaînes couplées 4x2 qui est la méthode la plus lente met 20 ms sur un processeur cadencé à 2.86 GHz). On peut aussi remarquer que la méthode du SKDE est vraiment très peu coûteuse par rapport aux autres algorithmes. Comme annoncé précédemment le couplage dissymétrique « 1-N » permet de diminuer le coût de calcul par rapport au couplage symétrique « 4x2 ». Concernant les temps d'apprentissage, comme cette phase est réalisée hors ligne, leur impact est moindre, on peut cependant noter que l'utilisation de méthode de modélisation plus complexe tel que les HMMs et les *Coupled-HMMs* ont tout de même un coût calculatoire important puisqu'on peut voir sur le tableau 5.2 qu'un facteur dix existe entre le temps d'apprentissage du SKDE et des HMMs ainsi qu'entre les HMMs et les chaînes couplées de type « 1-N ».

	SKDE	HMM	Couplage 4x2	Couplage 1-4
Temps d'apprentissage (en minutes)	3	32	442	342
Temps d'évaluation (en million de cycles)	1	3.58	58.06	42.33

TABLE 5.2 – Ordre de grandeur des temps d'apprentissage et temps d'exécution en phase de test.

5.2.4.2 Évaluations qualitatives

D'autres scènes ont été utilisées pour évaluer notre système. Cependant à cause d'un manque de séquences avec anomalies, elles n'ont pas été utilisées pour une analyse quantitative. En revanche, il est possible de donner des résultats qualitatifs sur ces séquences. Afin de mieux évaluer les raisons des anomalies présentées, une carte des mouvements normaux (représentés par des flèches vertes) est fournie pour chacune de ces séquences. Les méthodes du SKDE ou des chaînes couplées ont été utilisées pour ces différentes séquences selon les meilleurs résultats visuels obtenus.

CVN réel Quelques évènements réels associés à la base B_{CVN} ont été repérés. Trop peu pour constituer une base de test, le système met tout de même en évidence quelques comportements anormaux. Ces évènements sont de nature variée, du camion de pompier allant à contresens, qui a très largement été utilisé à titre d'exemple tout au long de ce manuscrit, au piéton traversant hors zone piétonne, en

passant par un véhicule effectuant une marche arrière ou des scooters quittant le trottoir de manière inhabituelle, comme il est possible de voir à la figure 5.17.

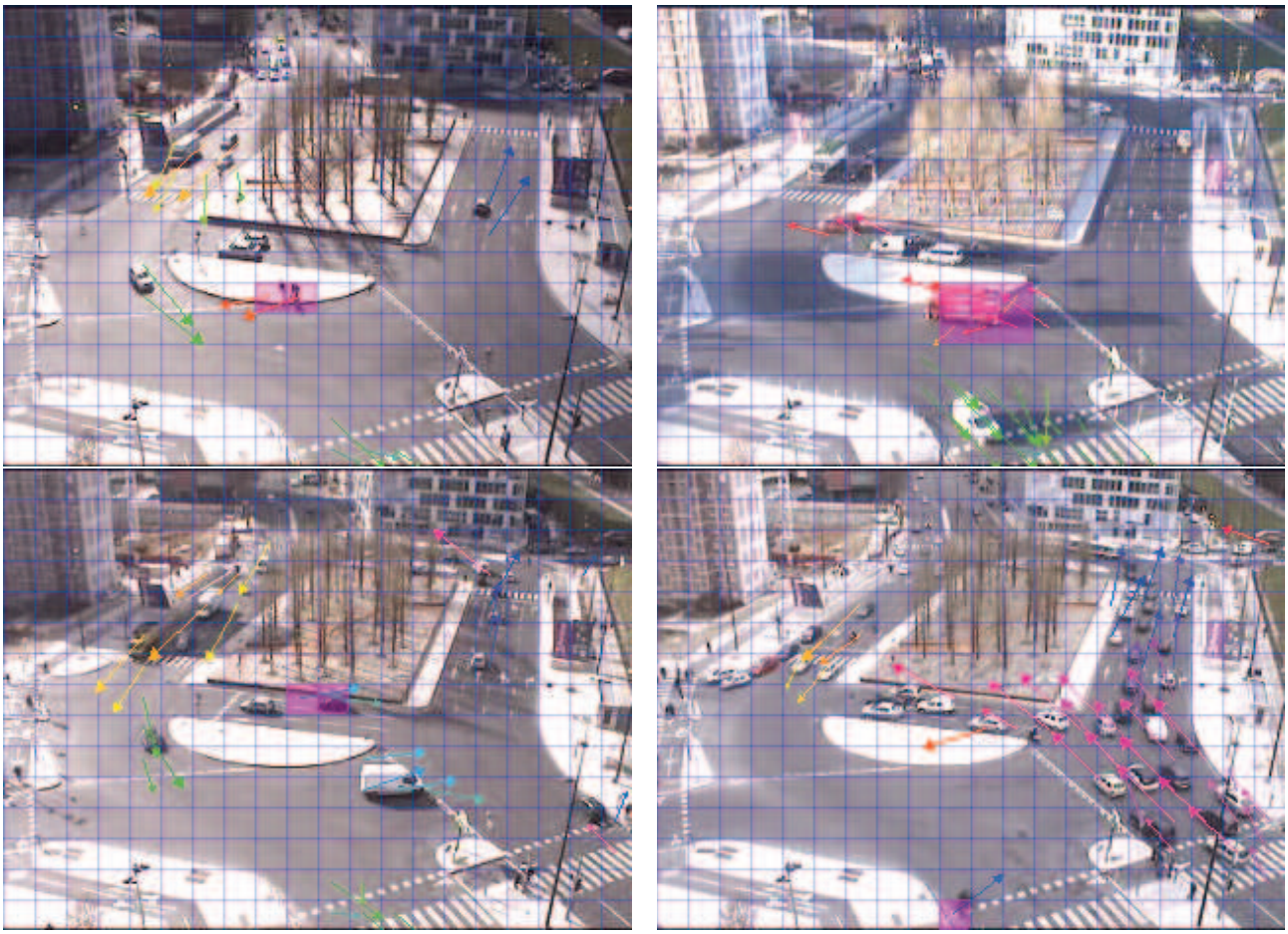


FIGURE 5.17 – Exemples de détection d'évènements réels sur la séquence routière utilisée pour réaliser la base B_{CVN}. Les détections sont représentées par des carrés magentas semi-transparents superposés sur les blocs analysés comme anormaux.

PDPP Quelques évènements de la base B_{PDPP} qui ont permis de calculer les différentes statistiques présentées lors de l'analyse quantitative sont représentés sur la figure 5.18. Ces évènements représentent pour la plus part des changements de file plus ou moins grossiers ou des marches arrières pour changer de file, caractéristique de personnes voulant quitter ou éviter un poste de péage encombré. Des descentes de voiture peuvent aussi être détectées comme représenté sur la première image de la figure 5.18 où une personne regagne son véhicule après être allée voir un autre conducteur.

Peachtree Sur une séquence d'un carrefour routier [4] représenté sur la figure 5.19, relativement peu d'anomalies ont été observées, hormis six personnes traversant hors du passage piéton. Sur ces six personnes, les six ont été détectées à un moment ou à un autre de leur traversée. Ces traversées ont eu lieu en se faufilant entre des voitures à l'arrêt, en deux temps en s'arrêtant au milieu de la chaussée à cause de la circulation ou à travers la chaussée déserte (cf. figure 5.20)

4. http ://ngsim-community.org/

FIGURE 5.18 – Exemples de détection d'évènements réels sur la séquence routière utilisée pour réaliser la base B_{PDPP}.

Hall Dans cette scène d'intérieur, on analyse cette fois-ci des flux de personnes, plus ou moins denses (cf. figure 5.21). Différents évènements sont à mettre en avant ici, tels qu'une discussion houleuse entre personnes suivie d'un mouvement à contresens d'une personne, le mouvement à contresens de deux personnes et la chute d'une personne provoquant un phénomène de regroupement induisant des mouvements localement anormaux 5.22. Parce que la perception du comportement et du mouvement dans une foule est difficile, des images supplémentaires représentant la scène quelques instants avant et après la détection sont produites afin de mieux percevoir le déroulement de l'anomalie.

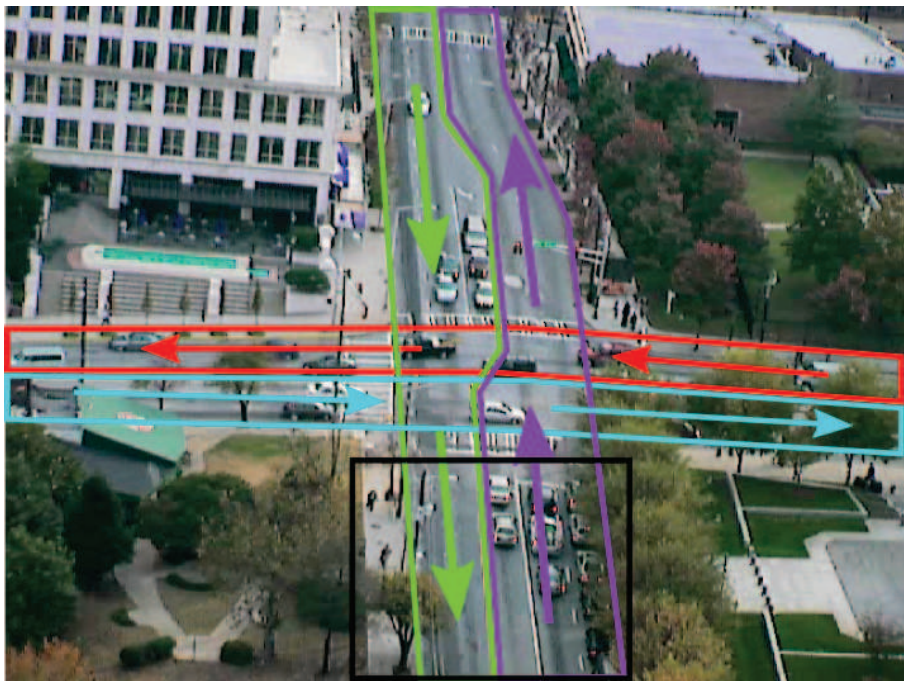


FIGURE 5.19 – Croisement routier Peachtree avec vérité terrain. Le rectangle noir représente une zone ou plusieurs anomalies ont été observées (cf. figure 5.20).

FIGURE 5.20 – Exemples de détection sur la séquence routière Peachtree. Différents piétons traversent en dehors des passages cloutés. Certains piétons se faufilent au travers de voiture à l'arrêt.

On peut remarquer que si la résolution est suffisante pour permettre la caractérisation du mouvement d'une seule personne, le système proposé parvient à détecter l'anomalie même dans une foule.

Marathon Cette autre scène met en avant l'analyse d'un très grand nombre de personnes, très proches les unes des autres. Malgré le longueur de la séquence relativement courte, environ 45 secondes en tout, l'apprentissage à été réalisé sur 10 secondes. Durant les 35 secondes restantes, trois anomalies situées à différents endroits de la scène (cf. figure 5.23) peuvent être observées : un joggeur quittant la course ainsi qu'un couple quelques instant après et une personne du comité d'organisation qui remonte le cortège à contresens le long de la barrière de sécurité, présentés au travers de quelques images sur la figure 5.24. Ces trois évènements sont correctement perçus par notre système (figure 5.25)

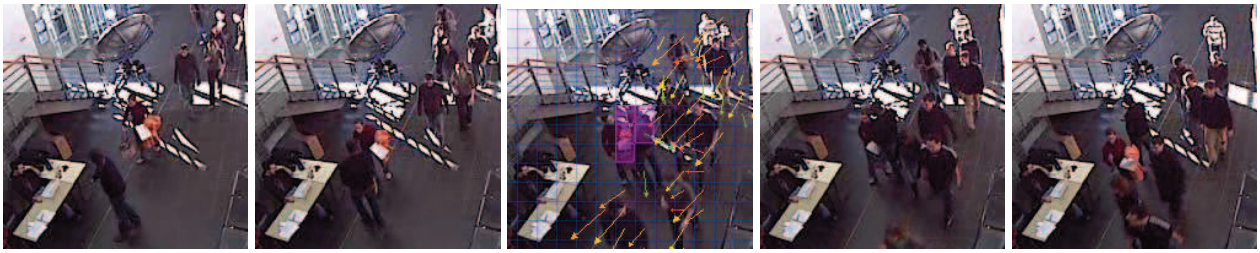
FIGURE 5.21 – Hall intérieur pour l'analyse de foule.

5.3 Conclusion

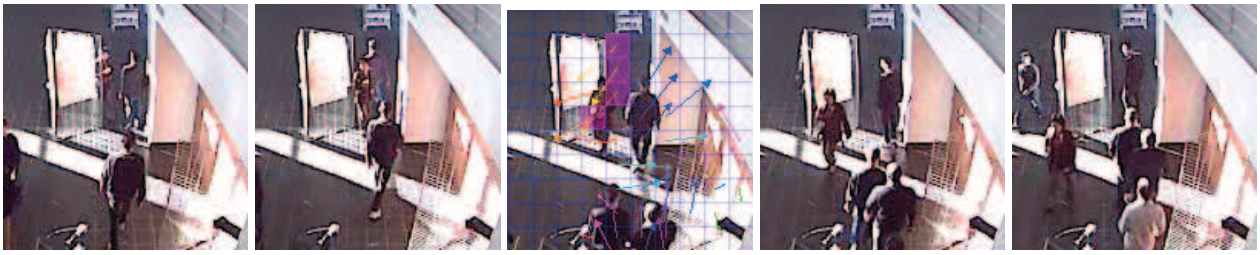
Ce chapitre d'évaluation vient faire le bilan sur les différentes solutions proposées pour répondre au problème de l'analyse d'espaces possiblement denses de manière automatique, non supervisée et sans calibrage.

Les évaluations ont été menées sur des vidéos difficiles issues de situations réelles. Les évènements rencontrés correspondent à des cas réels observés dans des conditions elles aussi bien réelles. En effet, tous types de conditions climatiques et d'illumination ont été rencontrées, ainsi que les désagréments qui leurs sont souvent associés tel que les reflets sur une chaussée humide, les lentilles salies par les traces de pluie, etc. Les différents tests permettent tout d'abord de montrer l'apport des différentes versions de chacune des briques algorithmiques mises en place dans ce manuscrit. Ainsi de manière assez nette, nous avons montré l'intérêt d'utiliser le critère de confiance pour le calcul de seuil automatique sur l'ensemble des blocs permettant de prendre en compte l'incertitude dans la classification pour des zones où le mouvement peut être quasiment quelconque. L'apport du descripteur proposé a aussi été prouvé, à deux niveaux. Tout d'abord en termes de performance, en comparaison avec le flot optique, le descripteur CSD permet d'obtenir de bien meilleures performances grâce à l'intégration temporelle qui permet une meilleure caractérisation du mouvement. L'autre intérêt de notre descripteur présenté au chapitre 4, réside dans la possibilité de pouvoir le discrétiser de manière efficace en assez peu de symboles contrairement aux descripteurs de Shechtman et Irani (2005) et Kratz et Nishino (2009), ce qui permet d'utiliser les chaînes de Markov de manière optimale sans en brider le mécanisme d'apprentissage comme avec l'approche proposée par Kratz et Nishino (2009) et ainsi obtenir de meilleures performances.

La deuxième série de tests qui vise à déterminer quelle machine d'apprentissage répond le mieux à notre problématique, a permis de mettre en avant deux des trois machines d'apprentissage proposées. La machine fondée sur les chaînes de Markov cachées classiques n'apporte pas réellement d'amélio-

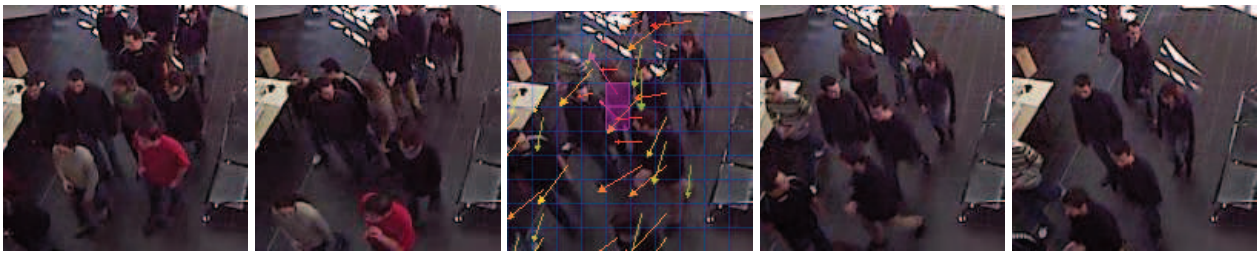


(a) Perturbation du flux de déplacement due à une dispute entre personnes.



(b) Passage d'une personne à contresens.

(c) Perturbation du flux de déplacement suite à la chute d'une personne et aux personnes venues pour l'aider.



(d) Demi-tour d'une personne au milieu de la foule.

FIGURE 5.22 – Exemples de détection d'évènements de foule dans un hall en intérieur.

ration par rapport au SKDE et aux chaînes couplées. Leur mise en place était pourtant nécessaire pour celle de chaînes couplées. Pour les deux méthodes restantes, les différents tests n'ont pas mis en avant une approche plutôt qu'une autre. La conclusion à en tirer est plus complexe. L'ajout des cohérences temporelle et spatiale par le biais des chaînes couplées permet certes de lisser et robustifier la détection d'une part et de filtrer certaines fausses alarmes d'autre part mais cela contribue aussi à renforcer certaines fausses alarmes. En pratique, pour une application de vidéosurveillance les chaînes couplées ne sont pas systématiquement la solution à adopter. Le choix dépendra beaucoup de l'application. En effet, si le but recherché est le minimum de fausses alarmes tout en étant capable de détecter les évènements grossiers, alors la méthode à adopter est le SKDE. Ses détections très ponctuelles facilement filtrables ne concernent dans ce cas que les fausses alarmes ou les évènements subtils alors que les évènements grossiers qui perdurent eux résisteront au filtrage. En revanche, si le but recherché

FIGURE 5.23 – Surveillance d'un marathon. Trois anomalies apparaissent dans cette scène. Leurs positions sont représentées par des rectangles noirs. La nature des anomalies est détaillée à la figure 5.24.

est le maximum de détections possible, alors les chaînes couplées sont à choisir au prix d'un taux de dérangement plus important. Nous pouvons étayer ce dernier point en rappellant que seules ces approches basées sur les chaînes couplées permettent de mettre en évidence des évènements complexes de mouvements relatifs entre blocs adjacents anormaux (cf. chapitre 4). Ces évènements sont, certes extrêmement rares mais ils représentent aussi des situations très intéressantes à detecter, telles que des manœuvres de queue de poisson très dangereuses ou des carambolages.

Enfin, les résultats plus qualitatifs montrent qu'en situation réelle, pour des anomalies plus « communes » non caractérisées par des mouvements relatifs entre blocs adjacents, il est possible de détecter des évènements très variés. Ces comportements anormaux concernent aussi bien sur des scènes de surveillance routière que des scènes d'espaces intérieurs ou extérieurs de foule plus ou moins dense. Le panel de détections présenté va de la détection de mouvement à contresens de véhicule ou de personne, de piéton traversant les voies de circulation en zone inhabituelle, à la rupture de trajectoire pour changer de file à l'approche d'un péage en passant par tout flux localement chaotique dû à des évènements inhabituels tels que des disputes entre personnes, ou l'agitation occasionnée autour d'une personne ayant chuté.

Ces résultats répondent bien à la problématique initialement posée, à savoir la faisabilité d'un système de vidéo assistance capable de détecter des anomalies de comportement pour une scène donnée sans avoir à donner d'information *a priori* sur la scène et sans être limité par le nombre de personnes ou de véhicules présents dans la scène.

(a) Passage à contresens d'une personne le long de la barrière de sécurité.

(b) Couple quittant la course.

(c) Courreur quittant la course.

FIGURE 5.24 – Détails des différentes anomalies à détecter pour la séquence du marathon.

FIGURE 5.25 – Détections des 3 évènements de la séquence du marathon.

Conclusion et Perspectives

Les scènes possiblement denses telles que les vidéos de foule ou de trafic routier sont particulièrement difficiles à analyser en raison de la grande quantité de données à traiter et de la complexité de la scène. Les approches basées sur l'identification et le suivi des acteurs de la scène (véhicule ou piéton) sont mises en difficultés par la quantité de cibles à traiter. Il apparaît alors pertinent de travailler sur des informations plus globales telles que le mouvement apparent.

Ces travaux ont pour but de répondre à la question « est-il possible des détecter des évènements automatiquement dans les espaces publis ? ». Pour cela, un nouveau système de vidéo assistance a été proposé. Ce système peut être décomposé en deux parties, la caractérisation du mouvement global d'une part et la procédure d'apprentissage et de classification des schémas de déplacement locaux (sur les blocs d'une grille fixe sur la scène) d'autre part. Ce système ne nécessite ni calibration de caméra, ni modélisation 3D de la scène. C'est une propriété forte pour des systèmes de vidéosurveillance car une calibration est un exercice très délicat. Cela nécessite des conditions favorables d'illumination et de résolution afin d'obtenir une calibration précise. Certaines positions et orientations de caméra peuvent même être proscrites. De plus, la calibration est une étape à réaliser pour chaque caméra, ce qui peut devenir extrêmement fastidieux pour des réseaux importants. Enfin, l'approche basée apprentissage de la méthode rend le système adaptatif et l'aspect non supervisé de l'apprentissage facilite son exploitation. Toutes ces propriétés rendent le système proposé extrêmement simple à déployer et à utiliser, ce qui est une caractéristique essentielle pour un système de vidéosurveillance.

Les travaux présentés dans ce manuscrit mettent en avant deux contributions, une étude approfondie sur l'utilisation de réseaux bayésiens pour l'étape de classification ainsi qu'une synthèse comparative sur les différentes machines d'apprentissage étudiées.

Première contribution : une nouvelle caractérisation du mouvement. Le mouvement global de la scène est caractérisé grâce à un nouveau descripteur basé sur une structure spatio-temporelle. Ce descripteur a montré des performances dépassant celles des autres descripteurs spatio-temporels en termes de distinction de mouvements. Comparé au flot optique, le descripteur CSD proposé se montre aussi beaucoup plus robuste aux erreurs ponctuelles, conséquences entre autres du problème d'ouverture, mais aussi beaucoup plus lisse grâce à l'intégration temporelle. La détection est donc beaucoup plus propre et bien moins sujette aux fausses alarmes erratiques contribuant au bruit de détection. De plus, ce descripteur est invariant aux changements affines d'illumination ce qui est particulièrement intéressant pour l'analyse de scènes extérieures contrairement au descripteur de Kratz et Nishino (2009). Enfin, nos descripteurs sur la grille d'analyse complète sont déterminés plus rapidement qu'avec le flot optique ou même les autres descripteurs spatio-temporels étudiés.

Deuxième contribution : une nouvelle méthode pour l'estimation de densité de probabilité. Cette méthode peut être utilisée dans des cadres applicatifs bien différents de celui de nos travaux ;

elle vise à estimer une densité quelconque uniquement à partir d'observations tirées de cette densité. Cette estimation est mise en œuvre via une représentation parcimonieuse des fenêtres de Parzen, adaptée pour des évaluations en temps réel. Cette méthode donne des résultats de qualité équivalente aux méthodes classiques avec un meilleur compromis en terme de précision, de parcimonie et de temps d'estimation. Une autre propriété intéressante de l'estimation proposée réside dans la possibilité de répertorier de manière automatique les « états » possibles du modèle. En regardant de plus près le modèle obtenu, on constate qu'il est constitué de points de contrôle qui ont individuellement une signification à part entière puisqu'ils représentent un « aspect » (ou un « état ») possible du modèle. Cette propriété n'est au contraire pas acquise pour des approches telles que le *One Class SVM*, où les vecteurs supports représentent des points « sentinelles » autour du modèle et ne sont donc pas représentatifs individuellement d'un quelconque « état » du modèle.

Étude approfondie : exploitation des réseaux bayésiens pour la classification de mouvement En terme de machines d'apprentissage, ce genre d'estimation de densité est utilisé dans notre cas pour déterminer la vraisemblance du mouvement en chaque bloc de la scène. Une autre approche possible consiste à utiliser des réseaux bayésiens et plus particulièrement des chaînes de Markov cachées. L'étude de ces approches constitue un autre grand pan des travaux menés durant cette thèse. Initialement proposée par Kratz et Nishino (2009), l'utilisation de HMMs vise à ajouter de la cohérence temporelle à la modélisation des schémas de déplacement. Nous avons amélioré les résultats de cette approche en modifiant la procédure de création de ces chaînes afin de ne plus brider le mécanisme d'apprentissage comme le faisaient Kratz et Nishino (2009). Enfin, l'ajout d'une cohérence spatiale modélisée grâce à une extension des chaînes de Markov cachées qui vise à coupler ces dernières, a aussi été mise en place. Une discussion des différentes stratégies de couplage a été réalisée pour mettre en avant une modélisation plus concise avec une complexité algorithmique plus faible. Ce type de couplage permet d'ajouter de la cohérence spatiale et de détecter des combinaisons de mouvements anormales. Concernant la fonction de décision pour l'étape de classification, quelle que soit la machine d'apprentissage utilisée, un critère de confiance est utilisé. Ce critère permet de tenir compte de la forme de la distribution dans la prise de décision et ainsi insérer une notion d'incertitude qui permet d'améliorer les résultats en terme de classification.

Comparaison des approches d'apprentissage. Les résultats comparatifs des différentes machines d'apprentissage ont montré que l'ajout de cohérences (temporelle et spatiale) permet de lisser l'analyse en gommant de nombreuses fausses alarmes et en lissant les détections. Cependant certaines fausses alarmes plus récalcitrantes se retrouvent aussi renforcées et il devient alors plus difficile de les filtrer. Aussi, en situation d'exploitation réelle où des filtrages supplémentaires sur la sortie du classifieur sont mis en place pour ne lever d'alerte que pour des évènements de durée minimale, les fausses alarmes lissées deviennent indifférentiables des bonnes détections. Les résultats ont montré des taux de détection très corrects qui font émerger deux approches pour répondre efficacement à ce problème. La première, fondée sur les chaînes couplées, privilégie la détection afin de manquer le moins possible d'évènements. La seconde, fondée sur l'approche bayésienne proposée (SKDE), privilégie le taux de dérangement minimal tout en détectant les principaux évènements.

Concernant la variété des scènes exploitables, là encore la méthode est très adaptative. L'information traitée (le mouvement) est globale à l'image et très robuste aux changements d'illumination et de conditions climatiques. Couplé à l'approche basée apprentissage, ce système est donc capable de

fonctionner aussi bien en intérieur qu'en extérieur, pour la surveillance de trafic routier ou de foule. La densité de la foule peut aussi être variable, contrairement aux travaux de Kratz et Nishino (2009) qui ne prennent pas en compte le non mouvement lors de l'apprentissage. Les évènements détectables sont aussi très variés. Toute perturbation même subtile du flux global est susceptible de déclencher une alarme. Ces perturbations vont du mouvement à contresens, aux personnes traversant hors des passages piétons en passant par des perturbations causées par des disputes ou des attroupements autour d'une personne ayant chuté.

Le système développé tout au long de ce manuscrit avait pour contrainte de fonctionner en temps réel. L'objectif derrière cette contrainte était de pouvoir signaler des alertes en direct à des opérateurs pour que ces derniers puissent agir en conséquence. Cet objectif a été atteint et ce independamment de la densité de la scène contrairement à des approches basées sur du suivi d'objet. Dans ce cas d'application le taux de fausses alarmes est important et favorise légèrement la méthode SKDE si toutefois le système est autorisé à être plus permissif. En revanche, il serait aussi possible d'utiliser notre système à des fins d'indexations en temps réel de vidéos pour le contrôle *a posteriori* des évènements. Dans ce genre d'application, où le taux de fausses alarmes vient en second plan, l'approche fondée sur les chaînes coupleés est dans intéressante car elle limite les risques d'échecs de détection et peut permettre de détecter des évènements plus complexes.

Ce système est la preuve de la faisabilité d'un système autonome pour la vidéosurveillance en temps réel d'espace publics. Néanmoins, dans un souci d'amélioration et pour reprendre la formulation de Zhong *et al.* (2004), la détection d'évènements anormaux est un problème « difficile à décrire mais facile à vérifier ». Nous avons pu vérifier ce fait au travers des différentes approches qui ont été évaluées. En effet, un gros gain de performance a été obtenu grâce au travail effectué sur le descripteur. C'est un point qu'il est important de garder à l'esprit : il est en effet préférable de « soigner la mesure plutôt que de vouloir la corriger *a posteriori* ». Ainsi, un travail sur l'information de bas niveau afin que celle-ci soit la plus fiable possible est donc vital. Le choix de la machine d'apprentissage a aussi un impact sur les performances de détections car grâce à des mécanismes plus élaborés mettant en œuvre des modélisations plus complexes tenant compte de la dynamique du mouvement, il est possible de filtrer en partie des erreurs récalcitrantes. Cependant, ces modélisations plus complexes peuvent et surtout doivent servir à élargir le panel de détections afin de pouvoir répondre au mieux à cette problématique extrêmement vague de la détection de comportement anormaux.

Perspectives

Les applications de vidéosurveillance intelligentes connaissent actuellement un véritable essor faisant apparaître des systèmes très variés analysant des scènes elles-mêmes très variées. Au terme du travail qui a été réalisé durant ces trois années, de nombreuses pistes d'évolution viennent à l'esprit. Parmi celle-ci certaines on fait l'objet de quelques essais et évaluations, d'autres feront l'objet de travaux ultérieurs. Quelques-unes de ces pistes sont évoquées ci-dessous.

La multi-résolution Une structure fixe par blocs a été choisie. Il serait légitime de vouloir étendre cette analyse à une analyse multi-résolution. En effet, toujours dans l'optique d'être générique et non dépendant de la scène étudiée, une analyse multi-résolution, pourrait aller dans ce sens en permettant une analyse plus adaptée pour des scènes avec une forte contrainte géométrique impliquant des changements d'échelles importants ou des scènes avec des vitesses de déplacement très élevées qui nécessiteraient un sous échantillonnage spatial pour pouvoir être ramenées dans des plages raisonnables.

Ce type d'approche a été abordé mais n'a pas fait l'objet d'une étude approfondie, c'est pourquoi il sera présenté ici en vue de développements ultérieurs. Une approche multi-résolution consiste, comme il a été dit au chapitre 2, à construire une pyramide de résolution pour en extraite des informations plus grossières aux faibles résolutions pour éventuellement corriger les informations aux résolutions plus fortes.

Le système a donc été étendu, non plus a une simple grille, mais à une pyramide de grille comme représenté sur le schéma 6.1. La grille de résolution d'un niveau $N+1$ étant deux fois moins haute et large que la grille de niveau N. A cette pyramide de grille est aussi fournit une pyramide d'observations issue du calcul des descripteurs à différentes résolution.

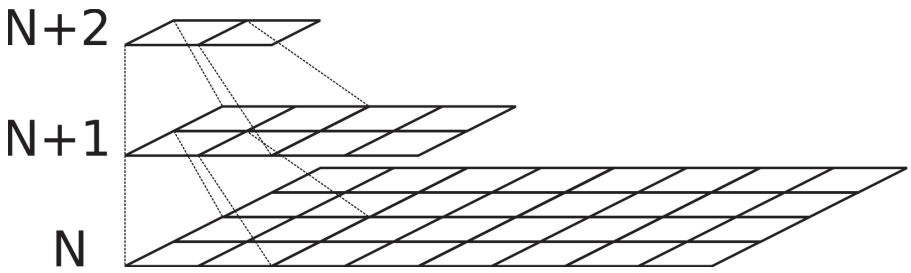


FIGURE 6.1 – Pyramide de grille de l'approche multi-résolution.

Un apprentissage et la détermination de la fonction de décision associée sont ensuite réalisés sur l'ensemble des blocs de la pyramide de résolution. La phase délicate consiste à fusionner l'ensemble de ces informations obtenues à différentes échelles pour obtenir une réponse finale. Dans notre cas d'application, cette étape de fusion est réalisée via une heuristique. Notre approche est une intégration adaptative du type *Coarse To Fine*. Elle consiste à propager ou non l'analyse à une résolution plus forte N en fonction de la cohérence, les caractérisations de mouvement entre les niveaux $N+1$ et N. Pour cela, on considère pour un bloc de faible résolution et ses quatre fils de la résolution supérieure. Notons m_{FR} le mouvement au niveau de faible résolution $N+1$ et m_{HR} le mouvement du niveau de résolution supérieur N. L'heuristique proposée est de distinguer différents cas :

- ▷ Le mouvement m_{FR} est en accord avec les quatre mouvements m_{HR} : l'analyse peut être limitée à celle de m_{FR}.
- ▷ Le mouvement m_{FR} est en désaccord total avec les quatre mouvements m_{HR} : la caractérisation du mouvement est supposée erronée à la résolution supérieure à cause d'un mouvement trop rapide ou d'une géométrie de scène peu favorable (objet trop proche de la caméra). L'analyse est donc limitée à celle de m_{FR}.
- ▷ Le mouvement m_{FR} est en accord avec certains des quatre mouvements m_{HR} : la faible résolution est trop grossière, l'analyse doit être propagée aux quatre blocs fils de la résolution supérieure. Ce type de cas est typique lorsque la taille du bloc de la résolution $N+1$ n'est pas adaptée à celle des objets, gommant ainsi la finesse de détection.
- ▷ Le mouvement m_{FR} n'est pas perceptible : l'analyse est propagée aux quatre blocs fils de la résolution supérieure

Une illustration de l'adaptation à la géométrie recherchée est présentée à titre illustratif à la figure 6.2. La taille des carrés cyans représente la résolution à laquelle l'analyse est effectuée, plus un carré est gros, plus la résolution d'analyse est faible. On constate qu'avec une intégration adaptative de ce genre, il est possible de déterminer automatiquement la taille d'analyse du bloc pour correspondre au mieux à celle d'un camion circulant sur une autoroute et observé sous en angle de vue très propice

aux changements d'échelle.

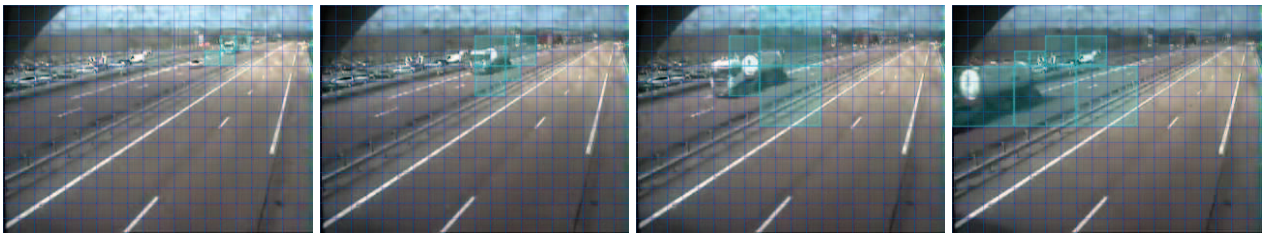


FIGURE 6.2 – Exemples d'adaptation de la taille du bloc d'analyse grâce à une approche multi-résolution de type *Coarse To Fine*. La taille des carrés cyans correspond à la résolution d'analyse. Un gros carré signifie une analyse à une résolution faible alors qu'un petit carré concerne une analyse à la résolution maximale.

Enfin pour ce qui concerne l'impact de la multi-résolution sur les temps de calcul, celui-ci est mineur. En effet, tout d'abord lors de l'estimation des gradients spatio-temporels qui rappelons-le constituent l'essentiel du temps de calcul lors de l'analyse en temps réel, l'application du filtre passe-bas nécessaire avant le sous-échantillonnage de la multi-résolution est aussi réalisée lors du calcul du gradient temporel par la méthode de Canny (cf. annexe B). Il est donc possible de calculer des gradients spatio-temporels avec la méthode Canny à différentes résolutions de manière efficace. De plus, le temps nécessaire à la classification en phase d'analyse est réduit avec une approche *Coarse To Fine* car les classifications de plusieurs blocs en mono-résolution sont susceptible d'être remplacées par une seule classification d'un bloc parent à plus faible résolution.

Vers des architectures de couplages plus complexes Une étude des architectures de couplage pourrait aussi être effectuée. Le couplage réalisé ici respectait le schéma d'une 4-connexité. On pourrait envisager des couplages supplémentaires avec des couplages à d'autres résolutions afin de caractériser une modélisation par plage de la vitesse normale ou de la taille d'analyse normale. On pourrait aussi envisager des couplages non symétriques dépendant d'une étude préalable de la scène, pour ne conserver que des couplages pertinents entre blocs.

Exploitation plus approfondie du non mouvement Outre l'information globale étudiée dans ce manuscrit à savoir le mouvement, l'information obtenue par les algorithmes de soustraction de fond serait aussi particulièrement utile au vue des scènes traitées. Cette information permettrait de distinguer le fond de la scène statique et les objets immobiles qui dans les deux cas correspondent à des zones de non mouvement indifférentiables par notre système.

La notion d'anormalité est extrêmement vague. Une définition exhaustive de celle-ci est impossible car elle dépend très fortement du contexte de la scène. Les travaux réalisés dans cette thèse sont une première tentative pour répondre à ce projet ambitieux qu'est la vidéosurveillance intelligente d'espaces publics. Cependant, les voies de recherche qu'ils restent à exploiter sont nombreuses.

Annexe A

Les courbes ROC (*Receiver Operating Curves*)

Ces courbes sont plus courantes pour évaluer les performances de classification binaire. Elles représentent le taux de bonnes détections en fonction du taux de fausses alarmes, soit les vrais positifs en fonction des faux positifs. Pour tracer cette courbe paramétrique, deux critères doivent être fixés. Quel est la nature de la classification binaire ? Et quel est le paramètre de classification qui va varier ?

En pratique, le critère à faire varier est le seuil de décision qui permet de distinguer les deux classes de la classification binaire (par exemple, la classification d'une sous-zone de l'image comme piéton ou non piéton (Leyrit (2010))). En faisant varier ce seuil de décision, on calcule pour chaque valeur, le taux de bonnes détections et de fausses alarmes. Sur la figure A, un classifieur parfait est représenté par la courbe rouge : 100% de bonnes détections pour 0% de fausses larmes. Une courbe correspondant à la diagonale (droite bleue sur la figure) est un classifieur qui n'est pas meilleur que le hasard. On peut ainsi évaluer les performances d'un classifieur : plus il s'approchera du cas idéal, meilleur il sera.

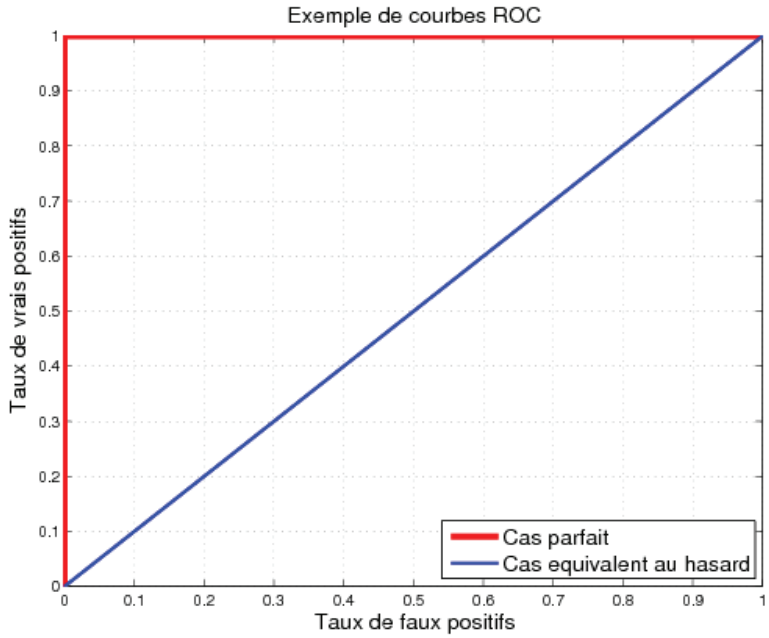


FIGURE A.1 – Exemples de courbes ROC. Deux courbes ROC. La courbe rouge corresponde à un cas idéal. La courbe ROC bleue correspond à un classifieur qui n'est pas meilleur que le hasard.

Annexe B

Les gradients de Canny 3D

L'algorithme de Canny développé en 1986 est initialement utilisé pour la détection de contour Canny (1986). Les contours correspondant aux zones de l'image où le gradient spatial est maximal, un calcul préalable de ces gradients d'intensité est nécessaire. C'est ce dont il est question ici.

Un gradient d'intensité est obtenu en convoluant une image avec un filtre dérivateur, comme par exemple $d = [-1\ 0\ 1]$. Cependant le bruit naturel des images conduit à des prétraitements de lissage sur celle-ci. Les variantes que l'on peut rencontrer se distinguent sur le filtre passe-bas utilisé. Si la méthode de Prewitt utilise un filtre rectangulaire et celle de Sobel utilise un filtre triangulaire, l'approche de Canny s'appuie sur un filtrage gaussien. Un calcul de gradient de Canny dans le cas 2D consiste donc à appliquer à l'image un flou gaussien de type :

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \tag{B.1}$$

ce qui concrètement revient à appliquer filtre 2D de taille NxN (N dépendant de la variance de la gaussienne choisie). L'étape suivante consiste à appliquer le filtre dérivateur d.

En pratique, dans un souci d'efficacité et grâce à la séparabilité des filtres, ce n'est pas le filtre gaussien 2D qui est appliqué mais deux filtres gaussiens 1D, l'un horizontalement et l'autre verticalement. De plus, l'application du filtre gaussien et du filtre dérivateur peut être combiné en appliquant uniquement la dérivé du filtre gaussien. Finalement le gradient de Canny dans le cas 2D pour une direction i est obtenu en convoluant l'image par un filtre gaussien selon une l'autre direction puis par un filtre de la dérivé de la gaussienne dans la direction i.

Le calcul des gradient de Canny se généralise exactement de la même manière en trois dimensions x, y, t (cf. algorithme 5. Nous noterons $F^{(i)}$ le filtre 1D F appliqué selon la dimension i et @ l'opérateur de convolution.

ENTRÉES: I une collection images successives en niveau de gris
ENTRÉES: H un filtre gaussien 1D
ENTRÉES: D_H un filtre dérivatif gaussien 1D
$G_x = I@H^{(y)}@H^{(t)}@D_H^{(x)}$
$G_y = I@H^{(t)}@H^{(x)}@D_H^{(y)}$
$G_t = I@H^{(x)}@H^{(y)}@D_H^{(t)}$
retourne Les gradients spatio-temporel G_x, G_y, G_t

Algorithme 5 : Gradient de Canny 3D

Bibliographie

A. ADAM, E. RIVLIN, I. SHIMSHONI et D. REINITZ : Robust real-time unusual event detection using multiple fixed-location monitors. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, March 2008.

E.L. ANDRADE et R.B. BLUNSDEN, S.and Fisher : Hidden markov models for optical flow analysis in crowds. *In Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 460–463, 2006.

E.L. ANDRADE, S. BLUNSDEN et R.B. FISHER : Modeling crowd scene for event detection. *In Proceedings of the International Conference on Pattern Recognition*, volume 01, pages 175–178, 2006.

F. BARDET, T. CHATEAU et D. RAMASASAN : Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous tracking and classification. *In Proceedings of the International Conference on Computer Vision*, 2009.

J.L. BARRON, D.J. FLEET, S.S. BEAUCHEMIN et T.A. BURKITT : Performance of optical flow techniques. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 92, pages 236–242, 1992.

L.E. BAUM : An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1970.

M.J. BLACK et P. ANANDAN : The robust estimation of multiple motions : Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

S. BLUNSDEN, E. ANDRADE et R. FISHER : Non parametric classification of human interaction. *In Pattern Recognition and Image Analysis*, pages 347–354, 2007.

S. BOUCHAFA, D. AUBERT et S. BOUZAR : Crowd motion estimation and motionless detection in subway corridors by image processing. *In Proceedings of the International Conference on Intelligent Transportation Systems Conference*, 1997.

M. BOZZOLI, L. CINQUE et E. SANGINETO : A statistical method for people counting in crowded environments. *In Proceedings of the International Conference on Image Analysis and Processing*, 2007.

M. BRAND : Coupled hidden markov models for modeling interacting processes. Rapport technique, MIT, 1997.

M.D. BREITENSTEIN, H. GRABNER et L. VAN GOOL : Hunting nessie : Real time abnormality detection from webcams. *In Proceedings of the International Conference on Computer Vision - Workshop on Visual Surveillance*, 2009.

G.J. BROSTOW et R. CIPOLLA : Unsupervised bayesian detection of independant motion in crowds. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2006.

M.Z. BROWN, D. BURSCHKA et G.D. HAGER : Advances in computational stereo. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

A. BUGEAU et P. PÉREZ : Detection and segmentation of moving objects in complex scenes. *Computer Vision and Image Understanding*, 113(4):459–476, 2009.

C.J.C. BURGES : A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

J. CANNY : A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

A.B. CHAN et N. VASCONCELOS : Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30:909–926, 2008.

D. COMANICIU, V. RAMESH et P. MEER : Kernel-based object tracking. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.

T. CORPETTI, E. MEMIN et P. PEREZ : Dense estimation of fluid flows. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(3):365–380, March 2002.

F.C. CROW : Summed-area tables for texture mapping. *In Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 207–212, 1984.

H.M. DEE et D.C. HOGG : Navigational strategies and surveillance. *In Proceedings of the European Conference on Computer Vision - Workshop on Visual Surveillance*, pages 73–81, 2006.

H.M. DEE et S.A. VELASTIN : How close are we to solving the problem of automated visual surveillance ? : A review of real-world surveillance, scientific progress and evaluative mechanisms. *Machine Vision Applications*, 19(5-6):329–343, 2008.

A.P. DEMPSTER, N.M. LAIRD et D.B. RUBIN : Maximum likelihood from incomplete data via the em algorithm (with discussion). *Royal Statistical Society*, B 39:1–38, 1977.

L. DONG, V. PARAMESWARAN, V. RAMESH et I. ZOGHLAMI : Fast crowd segmentation using shape indexing. *In Proceedings of the International Conference on Computer Vision*, 2007.

R.O. DUDA, P.E. HART et D.G. STORK : *Pattern Classification*. John Wiley & Sons Inc., 2001.

M.A.F. FIGUEIREDO et A.K. JAIN : Unsupervised learning of finite mixture models. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24:381–396, March 2002.

G.D. FORNAY : The viterbi algorithm. *Proceedings of the* IEEE, 61(3):268–278, 1973.

Matthieu FRADET, Philippe ROBERT et Patrick PÉREZ : Clustering point trajectories with various life-spans. *In Proceedings of the European Conference on Visual Media Production*, pages 7–14, 2009.

S. GEMAN et D.E. MCCLURE : Statistical methods for tomographic image reconstruction. *In Proceedings of the International Statistical Institute*, 1987.

Z. GHAHRAMANI et M.I. JORDAN : Factorial hidden markov models. *Machine Learning*, 29(2-3):245–273, November 1997.

V. GOUAILLIER et A.E. FLEURANT : La vidéosurveillance intelligente : promesses et défis. rapport de veille technologique et commerciale. Rapport technique, CRIM and Technopôle Défense et Sécurité, 2009.

B. HAN, D. COMANICIU, Y. ZHU et L.S. DAVIS : Sequential kernel density approximation and its application to real-time visual tracking. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30:1186–1197, July 2008.

C. HARRIS et M. STEPHENS : A combined corner and edge detector. *In Proceedings of the Alvey Vision Conference*, pages 147–151, 1988.

B.K.P. HORN et B.G. SCHUNCK : Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

T. HOSPEDALES, S. GONG et T. XIANG : A markov clustering topic model for mining behaviour in video. *In Proceedings of the International Conference on Computer Vision*, 2009.

W. HU, X. XIAO, Z. FU, D. XIE, T. TAN et S. MAYBANK : A system for learning statistical motion patterns. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 28:1450–1464, September 2006.

J.R. JAIN et A.K. JAIN : Displacement measurement and its application in interframe image coding. IEEE *Transactions on Communications*, 29:1799–1806, 1981.

Fan JIANG, Junsong YUAN, Sotirios A. TSAFTARIS et Aggelos K. KATSAGGELOS : Video anomaly detection in spatiotemporal context. *In Proceedings of the International Conference on Image Processing*, pages 705–709, 2009.

M.I. JORDAN, Z. GHAHRAMANI et L.K. SAUL : Hidden markov decision trees. *In Advances in Neural Information Processing Systems*, volume 9, pages 501–507, 1996.

I.N. JUNEJO et H. FOROOSH : Trajectory rectification and path modeling for video surveillance. *In Proceedings of the International Conference on Computer Vision*, 2007.

Y. KE, R. SUKTHANKAR et M. HEBERT : Event detection in crowded videos. *In Proceedings of the International Conference on Computer Vision*, October 2007.

P. KELLY, N.E. O'CONNOR et A.F. SMEATON : Robust pedestrian detection and tracking in crowded scenes. *Image and Vision Computing*, 27:1445–1458, Septembre 2009.

J. KIM et K. GRAUMAN : Observe locally, infer globally : a space-time mrf for detecting abnormal activities with incremental updates. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2009.

T. KOGA, K. IINUMA, A. HIRANO, Y. IIJIMA et T. ISHIGURO : Motion compensated interframe coding for video conferencing. *In Proceedings of the National Telecommunications Conference*, pages G5.3.1–G5.3.5, 1981.

L. KRATZ et K. NISHIMO : Tracking with local spatio-temporal motion patterns in extremely crowded scenes. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010.

L. KRATZ et K. NISHINO : Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, 2009.

D. KÜTTEL, M.D. BREITENSTEIN, L. VAN GOOL et V. FERRARI : What's going on ? discovering spatio-temporal dependencies in dynamic scenes. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010.

E. LANGFORD, N. SCHWERTMAN et M. OWENS : Is the property of being positively correlated transitive ? *The American Statistician*, 55(4):322–325, 2001.

L. LEYRIT : *Reconnaissance d'objets en vision artificielle : Application à la reconnaissance de piétons*. Thèse de doctorat, École Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand, 2010.

J. LI, S.G. GONG et T. XIANG : Scene segmentation for behaviour correlation. *In Proceedings of the European Conference on Computer Vision*, volume 4, pages 383–395, 2008.

Y. LI et H. AI : Fast detection of independent motion in crowds guided by supervised learning. *In Proceedings of the International Conference on Image Processing*, 2007.

N. LIU et B.C. LOVELL : Gesture classification using hidden markov models and viterbi path counting. *In Proceedings of Digital Image Computing : Techniques and Appications*, volume 1, pages 273–282, January 2003.

X. LIU, P.H TU, J. RITTSCHER, A. PERERA et N. KRAHNSTOEVER : Detecting and counting people in surveillance applications. *In Proceedings of the International Conference on Advanced Video and Signal Based Surveillance*, 2005.

B.D. LUCAS et T. KANADE : An iterative image registration technique with an application to stereo vision. *In Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

B. LUVISON, T. CHATEAU, P. SAYD, Q.C. PHAM et J.T. LAPRESTÉ : Méthode d'apprentissage non supervisée pour la détection d'évènements inattendus. *In CORESA*, 2009a.

B. LUVISON, T. CHATEAU, P. SAYD, Q.C. PHAM et J.T. LAPRESTÉ : Méthode d'apprentissage pour la classification à partir d'exemples positifs. *In ORASIS*, 2009b.

B. LUVISON, T. CHATEAU, P. SAYD, Q.C. PHAM et J.T. LAPRESTÉ : An unsupervised learning based approach for unexpected event detection. *In VISAPP*, 2009c.

B. LUVISON, T. CHATEAU, P. SAYD, Q.C. PHAM et J.T. LAPRESTÉ : Estimation parcimonieuse de densité par fonctions noyaux : Application à la détection temps réel d'evènements rares. *In RFIA*, 2010.

B. LUVISON, T. CHATEAU, P. SAYD, Q.C. PHAM et J.T. LAPRESTÉ : *Automatic detection of unexpected events in dense areas for videosurveillance applications*. INTECH, 2011. Soumis en 2010.

R. MA, L. LI, W. HUANG et Q. TIAN : On pixel count based crowd density estimation for visual surveillance. *In Proceedings of the International Conference on Cybernetics and Intelligent Systems*, volume 1, pages 170–173, December 2004.

V. MAHADEVAN, W. LI, V. BHALODIA et N. VASCONCELOS : Anomaly detection in crowded scenes. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010.

R. MEHRAN, A. OYAMA et M. SHAH : Abnormal crowd behavior detection using social force model. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 0, pages 935–942, 2009.

A. MONITZER : Using video surveillance to detect dangerous situations in underground stations by computer vision. *In Proceedings of Scientific Presentation and Communication*, 2006.

G. MONTEIRO, M. RIBEIRO, J. MARCOS et J.P. BATISTA : A framework for wrong way driver detection using optical flow. *In Proceedings of the International Conference on Image Analysis and Recognition*, pages 1117–1127, 2007.

P.J. MORENO, P.P. HO et N. VASCONCELOS : A kullback-leibler divergence based kernel for svm classification in multimedia applications. *In Advances in Neural Information Processing Systems*, 2004.

E. MÉMIN et P. PÉREZ : Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, February 2002.

K.M. NAM, J.S. KIM, R.H. PARK et Y.S. SHIM : A fast hierarchical motion vector estimation algorithm using mean pyramid. IEEE *Transactions on Circuits and Systems for Video Technology*, 5 (4):344 –351, August 1995.

C. NORRIS, McCahill M. et D. WOOD : Editorial : The growth of cctv : A global perspective on the international diffusion of video surveillance in publicly accessible space. *Surveillance Society*, 2(2/3):110–135, 2004.

N.M. OLIVER, B. ROSARIO et A.P. PENTLAND : A bayesian computer vision system for modeling human interactions. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22:831–843, 2000.

P. PAALANEN, J.K. KAMARAINEN, J. ILONEN et H. KÄLVIÄINEN : Feature representation and discrimination based on gaussian mixture model probability densities : Practices and algorithms. *Pattern Recognition*, 39:1346–1358, 2006.

X. PAN : *Computational Modeling Of Human And Social Behaviors For Emergency Egress Analysis*. Thèse de doctorat, Stanford University, 2006.

S. PARK et M.M. TRIVEDI : A track-based human movement analysis and privacy protection system adaptive to environmental contexts. *In Proceedings of the International Conference on Advanced Video and Signal Based Surveillance*, 2005.

Q.C. PHAM, L. GOND, J. BEGARD, N. ALLEZARD et P. SAYD : Real time posture analysis in a crowd using thermal imaging. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.

L.M. PO et W.C. MA : A novel four-step search algorithm for fast block motion estimation. IEEE *Transactions on Circuits and Systems for Video Technolology*, 6:313–317, 1996.

A. PURI, H.M. HANG et D.L. SCHILLING : An efficient block-matching algorithm for motion compensated coding. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 25.4.1–24.4.4, 1987.

V. RABAUD et S. BELONGIE : Counting crowded moving objects. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2006.

L.R. RABINER : A tutorial on hidden markov models and selected applications in speech recognition. *In Readings in Speech Recognition*, pages 267–296. 1990.

B.D. RIPLEY : *Pattern recognition and neural networks*. Cambridge Univ Pr, 1996.

A. SAAD et M. SHAH : A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.

I. SALEEMI, L. HARTUNG et M. SHAH : Scene understanding by statistical modeling of motion patterns. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010.

I. SALEEMI, K. SHAFIQUE et M. SHAH : Probabilistic modeling of scene dynamics for applications in visual surveillance. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, PP, Issue 99:1–1, 2008.

L.K. SAUL et M.I. JORDAN : Boltzmann chains and hidden markov models. *In Advances in Neural Information Processing Systems*, pages 435–442, 1995.

Bernhard SCHÖLKOPF et Alexander J. SMOLA : *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

E. SHECHTMAN et M. IRANI : Space-time behaviour based correlation. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

O. SIDLA et Y. LYPETSKYY : Pedestrian detection and tracking for counting applications in crowded situations. *In Proceedings of the International Conference on Advanced Video and Signal Based Surveillance*, 2006.

E.P. SIMONCELLI, E.H. ADELSON et D.J. HEEGER : Probability distributions of optical flow. *In CVPR*, pages 310–315, 1991.

S. SINGH, R. MACHIRAJU et R. PARENT : Visual analysis of trajectory clusters for video surveillance. Rapport technique, Ohio State University, 2007.

C. STAUFFER et W.E.L. GRIMSON : Learning patterns of activity using real-time tracking. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22:747–757, 2000.

M. TARON, N. PARAGIOS et M.P. JOLLY : Registration with uncertainties and statistical modeling of shapes with variable metric kernels. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 31:99–113, January 2009.

M.E. TIPPING : Sparse bayesian learning and the relevance vector machine. *Machine Learning Research*, 1:211–244, 2001.

A. TYAGI, M. KECK, J.W. DAVIS et G. POTAMIANOS : Kernel-based 3d tracking. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.

Vladimir N. VAPNIK : *Statistical Learning Theory*. Wiley-Interscience, September 1998.

J. VARADARAJAN et J.M. ODOBEZ : Topic models for scene analysis and abnormality detection. *In Proceedings of the International Conference on Computer Vision - Workshop on Visual Surveillance*, Kyoto, October 2009.

S.A. VELASTIN, B.A. BOGHOSSIAN et M.A. VICENCIO-SILVA : A motion-based image processing system for detecting potentielly dangerous situations in underground railway stations. *Transportation Research Part C : Emerging Technologies*, 14, May 2006.

X. WANG, X MA et W.E.L. GRIMSON : Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 31(1):539–555, 2009.

X. WANG, K. TIEU et W.E.L. GRIMSON : Correspondence-free activity analysis and scene modeling in multiple camera views. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 32 (1):56–71, 2010.

S. WU, B.E. MOORE et M. SHAH : Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2010.

Q. YU et G. MEDIONI : Multiple-target tracking by spatiotemporal monte carlo markov chain data association. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2196–2210, December 2009.

B. ZHAN, D.N. MONEKOSSO, P. REMAGNINO, S.A. VELASTIN et L.Q. XU : Crowd analysis : A survey. *Machine Vision Applications*, 19:345–357, 2008.

T. ZHAO et R. NEVATIA : Tracking multiple humans in complex situations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 26:1208–1221, September 2004.

H. ZHONG, J.B. SHI et M. VISONTAI : Detecting unusual activity in video. *In Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 819–826, 2004.

Z. ZHONG, M. YANG et S. WANG : Energy methods for crowd surveillance. *In Proceedings of the International Conference on Information Acquisition*, pages 504–510, 2007.

Yifan ZHOU, H. NICOLAS et J. BENOIS-PINEAU : A multi-resolution particle filter tracking in a multi-camera environment. *In Proceedings of the International Conference on Image Processing*, pages 4065–4068, November 2009.