N^o d'ordre : 4589

THÈSE

présentée à

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

par Mathieu Brulin

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Informatique

Analyse sémantique d'un trafic routier dans un contexte de vidéo-surveillance

---

Soutenue le : 25 octobre 2012

Après avis de :

M^me^	Christine FERNANDEZ-MALOIGNE	Professeur	Univ. de Poitiers	**Rapporteur**
M.	Marc GELGON	Professeur	Univ. de Nantes	**Rapporteur**

Devant la Commission d'Examen composée de :

M.	Marc CHAUMONT	Maître de conférences	Univ. de Montpellier	**Examinateur**
M.	Pascal DESBARATS	Professeur	Univ. de Bordeaux .	**Président**
M.	Christophe MAILLET	Gérant Adacis	Société Adacis	**Gérant Adacis**
M.	Henri NICOLAS	Professeur	Univ. de Bordeaux .	**Directeur**

- 2012 -

Analyse sémantique d'un trafic routier dans un contexte de vidéo-surveillance

Résumé :

Les problématiques de sécurité, ainsi que le coût de moins en moins élevé des caméras numériques, amènent aujourd'hui à un développement rapide des systèmes de vidéosurveillance. Devant le nombre croissant de caméras et l'impossibilité de placer un opérateur humain devant chacune d'elles, il est nécessaire de mettre en oeuvre des outils d'analyse capables d'identifier des évènements spécifiques. Le travail présenté dans cette thèse s'inscrit dans le cadre d'une collaboration entre le Laboratoire Bordelais de Recherche en Informatique (LaBRI) et la société Adacis. L'objectif consiste à concevoir un système complet de vidéo-surveillance destiné à l'analyse automatique de scènes autoroutières et la détection d'incidents. Le système doit être autonome, le moins supervisé possible et doit fournir une détection en temps réel d'un évènement.

Pour parvenir à cet objectif, l'approche utilisée se décompose en plusieurs étapes. Une étape d'analyse de bas-niveau, telle que l'estimation et la détection des régions en mouvement, une identification des caractéristiques d'un niveau sémantique plus élevé, telles que l'extraction des objets et la trajectoire des objets, et l'identification d'évènements ou de comportements particuliers, tel que le non respect des règles de sécurité. Les techniques employées s'appuient sur des modèles statistiques permettant de prendre en compte les incertitudes sur les mesures et observations (bruits d'acquisition, données manquantes, ...).

Ainsi, la détection des régions en mouvement s'effectue au travers la modélisation de la couleur de l'arrière-plan. Le modèle statistique utilisé est un modèle de mélange de lois, permettant de caractériser la multi-modalité des valeurs prises par les pixels. L'estimation du flot optique, de la différence de gradient et la détection d'ombres et de reflets sont employées pour confirmer ou infirmer le résultat de la segmentation.

L'étape de suivi repose sur un filtrage prédictif basé sur un modèle de mouvement à vitesse constante. Le cas particulier du filtrage de Kalman (filtrage *tout gaussien*) est employé, permettant de fournir une estimation *a priori* de la position des objets en se basant sur le modèle de mouvement prédéfini.

L'étape d'analyse de comportement est constituée de deux approches : la première consiste à exploiter les informations obtenues dans les étapes précédentes de l'analyse. Autrement dit, il s'agit d'extraire et d'analyser chaque objet afin d'en étudier son comportement. La seconde étape consiste à détecter les évènements à travers une coupe du volume 2d+t de la vidéo. Les cartes spatio-temporelles obtenues sont utilisées pour estimer les statistiques du trafic, ainsi que pour détecter des évènements telles que l'arrêt des véhicules.

Pour aider à la segmentation et au suivi des objets, un modèle de la structure de la scène et de ses caractéristiques est proposé. Ce modèle est construit à l'aide d'une

étape d'apprentissage durant laquelle aucune intervention de l'utilisateur n'est requise. La construction du modèle s'effectue à travers l'analyse d'une séquence d'entraînement durant laquelle les contours de l'arrière-plan et les trajectoires typiques des véhicules sont estimés. Ces informations sont ensuite combinées pour fournit une estimation du point de fuite, les délimitations des voies de circulation et une approximation des lignes de profondeur dans l'image. En parallèle, un modèle statistique du sens de direction du trafic est proposé. La modélisation de données orientées nécessite l'utilisation de lois de distributions particulières, due à la nature périodique de la donnée. Un mélange de lois de type von-Mises est utilisée pour caractériser le sens de direction du trafic.

Mots-clefs : Détection automatique d'incidents, Segmentation de mouvement, Suivi d'objets, Modélisation de la scène, Analyse de comportements.
Discipline : Informatique.

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
Université de Bordeaux 1
351 cours de la libération
33405 Talence FRANCE

Table des matières

Contexte industriel

Présentation du contexte

Le travail présenté dans cette thèse s'inscrit dans le cadre d'un contrat CIFRE et est issue d'une collaboration entre le LaBRI [1], dans lequel j'ai rejoint l'équipe Image et Son dans le thème de recherche Analyse et Indexation Vidéo, et la société Adacis [2] qui est une TPE bordelaise spécialisée dans la sécurité des systèmes d'information. Nous avons également travaillé en partenariat avec le ministère de l'Ecologie, du Développement durable et de l'Energie et l'équipe trevise (études et recherches appliquées à la vidéo, réseau, télécommunication), qui nous ont permis d'accéder au réseau de caméras de la rocade de Bordeaux.

Adacis est une structure créée en 2001 qui propose des services en ingénierie informatique, notamment dans le domaine de l'infrastructure. Les activités de l'entreprise sont variées et se déclinent en plusieurs services principaux :

- La conception des architectures réseaux sécurisés,
- L'intégration de solutions de réseaux et sécurité,
- L'organisation et la normalisation des processus de sécurité.
- L'administration et l'exploitation des réseaux et des équipements de sécurité,

La société adacis s'est spécialisée dans la mise en place d'architectures sécurisées, appliquée (entre autres) à la gestion des informations routières. Avec la hausse constante de la fréquentation sur les réseaux routiers, les exploitants doivent être capables de savoir à tout moment ce qui se déroule sur leurs réseaux pour prendre les mesures adéquates et le plus rapidement possible lorsqu'un incident se produit. Les incidents ont un impact direct sur la sécurité des usagers, et sur la capacité des réseaux et entraînent généralement des ralentissements , voir la saturation du réseau. Les congestions provoquent de nombreuses nuisances, telles que

- La dégradation des conditions de transport des usagers,
- La dégradation de la sécurité des usagers (notamment en queue de congestion),
- Des nuisances environnementales (pollutions, nuisances sonores).

L'extension des réseaux et la rapidité des évolutions du trafic lorsqu'un incident se produit rendent les approches purement manuelles insuffisantes et font de la Détection Automatique d'Incidents (DAI) une composante fondamentale des Systèmes d'Aide à l'Exploitation [Cohen 2000]. Dans ce domaine, on définira un incident comme un événement non prévu entrainant une dégradation de la capacité de l'infrastructure et la dégradation de son niveau de sécurité. On retrouve parmi les incidents possibles un véhicule arreté sur la chaussée, un véhicule à contre-sens, un piéton sur la chaussée, de la fumée dans un tunnel, objet sur la chaussée, ...

Objectifs et enjeux

L'objectif des industriels est donc de maintenir au maximum l'écoulement fluide du trafic et de maitriser les congestions sur les infrastructures chargées. Dans cet objectif,

1. Laboratoire Bordelais de Recherche en Informatique
2. www.adacis.net

le besoin en Système de Transport Intelligent (STI, système intégrant informatique et télécommunications appliqué au transport routier) a fortement augmenté. L'objectif est de fournir aide logicielle permettant de réduire au maximum de temps de traitement d'un incident. La chaîne de gestion d'un incident peut se décomposer en 4 étapes :

- La détection de l'incident,
- L'information des usagers (via des panneaux à messages variables par exemple),
- Le traitement de l'incident,
- Le retour à la normale du trafic

Des études statistiques montrent que le temps total passé dans une congestion créée par un incident varie comme le carré de la durée de l'incident [Cohen 2000]. De plus, des études démontrent que l'attention de la plupart des opérateurs chute après seulement 20 minutes à regarder et analyser les écrans de surveillance, et qu'un opérateur ne peut pas suivre attentivement une dizaine de caméras plus de 15 minutes [Gouaillier 2009]. On mesure ainsi l'enjeu d'une implémentation d'un système de detection automatique d'incidents (DAI), permettant la réduction du délai d'arrivée des secours, la réduction du temps de congestion, la réduction de la pollution atmosphérique, la réduction du nombre de collisions en fin de bouchon et l'aide aux opérateurs pour la réduction du temps de détection.

Projet Vizird

Dans le cadre de leurs activités, Adacis propose une solution sécurisée de visualisation d'informations routière, appelée Vizird [3]. Cette solution s'installe dans les centres de gestion du trafic qui exploitent les réseaux de caméras surveillant le trafic et possède plusieurs fonctionnalités principales, représentées sur la Figure 1 sous la forme de serveurs :

- Un serveur vidéo, qui gère l'acquisition sécurisées des flux vidéos,
- Un serveur affichage, permettant la gestion de l'affichage du mur d'image,
- Un serveur streaming, permettant à un utilisateur isolé d'accéder à un flux spécifique.
- Un serveur publication, permettant l'envoi vers un serveur distant de clips vidéos et d'informations routières relatives aux caméras.

L'objectif de ce travail consiste à intégrer au coeur du produit Vizird, une solution DAI permettant de traiter de façon logicielle un flux vidéo du réseau. Cette solution doit être capable de détecter en temps réel, à partir d'un capteur caméra, les incidents sur la chaussée. Nous nous plaçons ainsi dans un contexte de surveillance de scène extérieures à l'aide d'une seule caméra, et dans un objectif d'exploitation du réseau existant. Autrement dit, nous n'avons aucun contrôle sur le placement des caméras qui n'ont pas forcément été placé dans un objectif de détection automatique. Nous n'avons également aucunes connaissances ni sur les paramètres intrinsèques de la caméra (focale, dimension du capteur, ...), ni sur ses paramètres extrinsèques (hauteur de la caméra, angle de la caméra, ...).

3. Visualisation sécurisée d'Informations Routières Déportées

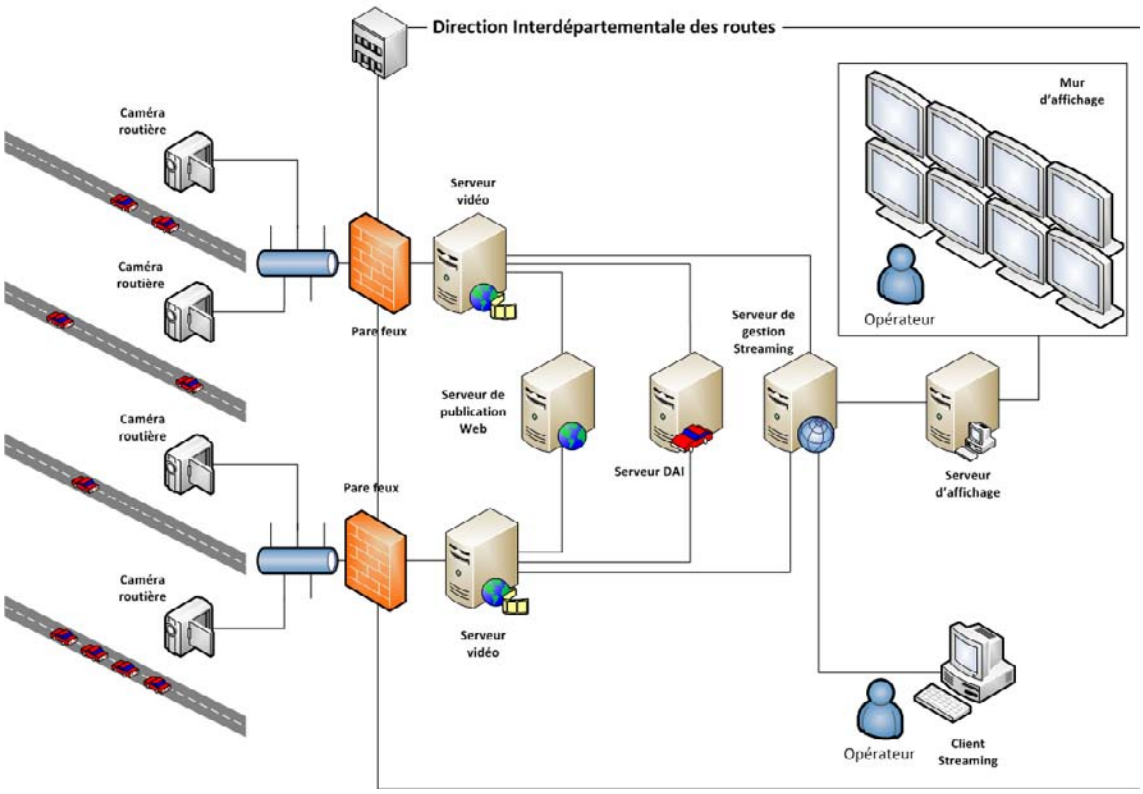


FIGURE 1: Architecture de la solution Vizird comportant les fonctionnalités d'acquisition sécurisée des flux vidéo (serveur vidéo), de gestion d'affichage sur le mur d'images (serveur d'affichage), de publication vers un serveur distant (serveur publication), de streaming d'un flux particulier vers un poste isolé (serveur streaming) et de détection automatique d'incidents (serveur dai).

Introduction générale

Introduction

Les systèmes de vidéo-surveillance jouent un rôle de plus en plus important dans la surveillance de sites sensibles ou de lieux publics et privés. Ses premières utilisations remontent historiquement aux années 1950 pour la surveillance de lancées de missiles. Cependant, la surveillance à l'aide de systèmes en circuit fermé (CCTV) ne s'est réellement développée qu'à partir des années 1970 avant de s'intensifier au cours des années 1990 [Gouaillier 2009]. Les attentats du 11 septembre 2001 aux Etats-Unis et de 2005 à Londres ont contribué au développement fulgurant du nombre de caméras installées par exemple pour la surveillance de sites. Ces systèmes se sont largement déployés pour surveiller des entrepôts ou des parkings afin de lutter contre le vol, pour filtrer les entrées et sorties dans les banques et réduire les risques de braquages, pour lutter contre le vol à l'étalage dans les magasins, pour repérer d'éventuelles tricheries dans les casinos, pour la surveillance du trafic routier ou d'un site industriel sensible et prévenir un incident.

Motivations

Devant le nombre croissant de caméras et l'impossibilité de placer un opérateur humain derrière chacune d'elles, la demande et le besoin d'outils d'analyse automatique des données récupérées a fortement augmenté. La répétitivité de la tâche et le faible nombre d'évènements ou de situations anormales entrainent une forte lassitude et baisse de l'attention des agents de sécurité.

Les efforts de recherche et la diminution du coût matériel des caméras ont ouvert la possibilité d'utilisation des systèmes de vidéo surveillance intelligents dans une large gamme d'applications à travers des fonctionnalités telles que la reconnaissance et le suivi automatique d'objets, l'interprétation de la scène et l'extraction ou l'indexation d'évènements particuliers. On retrouve des applications dans la surveillance de sites industriels (contrôle d'accès ou encore le contrôle de qualité de la production), dans la surveillance de lieux publics hautement fréquentés (gares, métro [Krausz 2010], commerces [Sicre 2010]), dans la surveillance et l'analyse d'activités de personnes agées (indexation d'activités [Karaman 2010] ou la détection de chute [Foroughi 2008]), dans le milieu sportif (football, golf), ...

De nombreux projets ont vu le jour afin d'évaluer l'efficacité et la faisibilité d'un système de vidéo-surveillance intelligent [Gouaillier 2009]. Un des précurseurs est le projet VSAM [4], dont l'objectif a été de fournir des outils d'analyse en temps réel pour l'intervention et la prévention d'incidents pour des applications militaires de surveillance de zones urbaines ou de combats. Citons également, par exemple, le projet européen ADVISOR [5] dont l'objectif a été de concevoir des outils logiciels pour la surveillance intelligente dans les transports publics et le projet ANR QUIAVU (Qualité des Images pour les Applications de VidéosUrveillance, 2009-2012) qui a pour objectif de fournir des outils de mesure de qualité des images obtenue par les systèmes de vidéo-surveillance.

Les systèmes de vidéo-surveillance sont généralement composés des étapes suivantes [Kastrinaki 2003] :

4. Video Surveillance and Monitoring, Etats-Unis, 1997-2000.
5. Annotated Digital Video for Surveillance and Optimised Retrieval, Communauté européenne, 2000-2003.

- **Détection de mouvement**. La détection de mouvement est généralement la base de tout système de vidéo-surveillance. Elle permet de déceler une activité dans la scène sous surveillance, comme le déplacement d'un objet, l'apparition ou la disparition d'un objet.

- **Extraction et classification des objets**. Une fois les objets detéctés, ils sont extraits et classés en différentes catégories (véhicule, piéton, poids-lourd, ...). De façon générale cette classification s'effectue à l'aide de primitives de niveau intérmédiaire, telles que les caractéristiques de forme d'un objet et ses propriétés de mouvement.

- **Suivi des objets au cours du temps**. Le suivi d'objet consiste localiser et maintenir l'identité des objets détectés au cours du temps. Suivre plusieurs objets simultanément présente plusieurs difficultés et de nombreux défis, notamment lorsqu'une occlusion se produit (région cachée par une autre) ou lorsque deux objets sont très proches.

- **Analyse de comportement et détection d'incidents**. Cette dernière étape consiste à interpréter les comportements des objets de la scène. Cette étape requiert une analyse sémantique souvent très dépendante du contexte d'application.

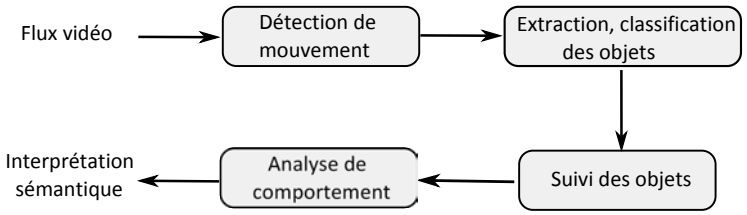


FIGURE 2: La conception d'un système de vidéo-surveillance contient généralement les 4 étapes suivantes : Détection d'objets, extraction et classification des objets, suivi des objets, analyse de comportement.

Initiallement développés pour des situations relativement simples ne faisant intervenir que peu d'objets, les systèmes actuels s'attaquent à des problèmes plus complexes dans des conditions bien moins contrôlées et plus proches de situations réelles.

Difficultés et problématiques

En pratique, la conception d'un système de vidéo-surveillance doit faire face à un certain nombre de containtes et de difficultés :

- **Contraintes techniques liées au matériel**, comme par exemple la résolution de l'image, le taux de rafraichissement de la vidéo, l'ajustement automatique de gain, le placement de la caméra, ...

- **Contraintes environnementales liées à la scène**, comme les changements de luminosité, les conditions climatiques, l'arrière-plan en mouvement, ...

– **Contraintes sémantiques liées au contexte de l'application visée**, il est très difficile de définir la notion de normalité en informatique sans en définir le contexte. Par exemple la présence d'un piéton sur la route n'est normale que s'il se trouve sur un passage piéton.

Ces contraintes entrainent de nombreux problèmes et difficultés dans les étapes de l'analyse. Les algorithmes doivent être robustes face à de nombreuses situations

– Une image est une représentation 2D d'une scène 3D. La projection perspective lors de la formation de l'image sur le capteur caméra entraîne une perte d'information relative à la profondeur dans l'image. Un même objet, observé selon un point de vue de la caméra différent, peut avoir une apparence très différente dans l'image.

– Les conditions d'acquisition de l'image peuvent varier d'un environnement à l'autre et les systèmes sont soumis aux conditions environnementales extérieures (météo par exemple).

– Lorsque les objets sont proches, il peut y avoir des occultations rendant difficile la tâche d'extraction et de suivi d'objets. Les algorithmes doivent faire face aux occlusions provoquées par la projection perspective lors de la formation de l'image.

En effet, dans l'objectif d'exploitation du réseau de caméra existantes, le matériel, ainsi que ses caractéristiques, sont imposées. La position de la caméra face à la scène n'a pas nécessairement été placé dans l'objectif d'une analyse automatique et les algorithmes de traitement doivent fonctionner sous plusieurs angles de vue. Les algorithmes d'extraction et de suivi d'objets doivent faire face aux déformations éventuelles et occlusions rencontrées lorsque deux objets sont proches par exemple. La résolution de l'image et le taux de rafraîchissement jouent également un rôle dans le choix des descipteurs pour la reconnaissance ou le suivi de l'objet. De plus, les algorithmes de détection d'objets doivent faire face à de nombreuses difficultés liées à la scène, telles que les changements de luminosité (locale ou globale), conditions climatiques, la présence d'un éventuel arrière-plan en mouvement, le problème de camouflage.

Contexte et objectifs

Cette thèse s'inscrit dans le cadre d'une convention CIFRE et dans un contexte fortement applicatif de vidéo-surveillance. Dans le cadre d'un projet appelé Vizird [6] financé par la société Adacis, l'objectif consiste à concevoir un système de vidéo-surveillance visant à aider et alléger la tâche fastidieuse des opérateurs pour la surveillance de scènes autoroutières. Une description détaillée d'une scène autoroutière est donnée dans la section 2.1.1. L'objectif consiste à exploiter le réseau de caméras existant et d'apporter une solution logicielle de surveillance intelligente. Cette solution doit être autonome, en temps réel, la moins supervisée possible, tout en étant la plus générique, afin d'être facilement déployable dans différentes configurations. Le système doit être capable de fournir des statistiques sur l'état du trafic ainsi que de détecter des évènements anormaux potentiellement dangereux tels que l'arrêt d'un véhicule, le contre-sens ou la présence d'un piéton sur la chaussée.

6. Visualisation Sécurisée d'Informations Routières Déportées

Stratégie envisagée

Les étapes présentées sur la Figure 2 s'exécutent de façon hiérarchique en partant du niveau des pixels, à celui des objets, pour atteindre l'échelle sémantique de comportement et d'interprétation de ce qui se déroule dans la scène. Ainsi, pour aborder le problème et répondre aux exigences et objectifs précédemment décrits, nous nous sommes orientés vers une structure générique composée de trois niveaux sémantiques comprenant l'ensemble des étapes généralement employées dans les systèmes de vidéo-surveillance : la détection des objets en mouvement, la classification des objets, leur suivi au cours du temps et l'analyse de leur comportement (Figure 3). Ces étapes permettent d'extraire à partir des données vidéos un contenu sémantique par une stratégie d'enrichissement des connaissances au fur et à mesure que l'on avance dans l'analyse. Ainsi la première étape consiste en une analyse des caractéristiques de bas-niveau ne contenant aucune information sémantique sur ce qui se déroule dans la scène. Les caractéristiques de bas-niveau correspondent généralement à la couleur, le gradient, la texture ou les vecteurs mouvement issus d'une estimation de flot optique. Une fois extraites, elles sont utilisées par l'analyse sémantique intermédiaire. Des caractéristiques de plus haut niveau sémantique sont extraites telles que la taille des objets, leurs formes ou leurs trajectoires. La dernière étape est souvent dépendante du contexte d'application et consiste à extraire une interprétation sémantique sur ce qui se déroule dans la scène.

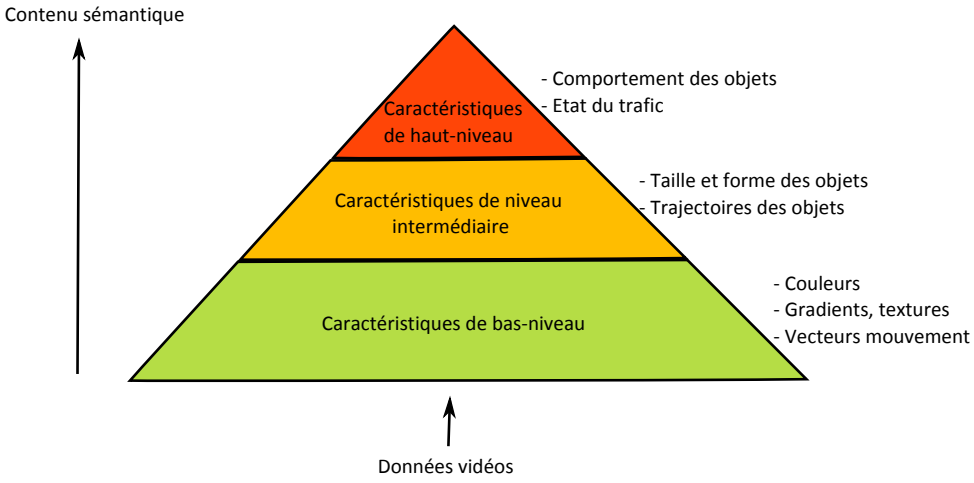


FIGURE 3: Les trois niveaux sémantiques pour l'analyse vidéo.

Le système proposé comporte une étape d'initialisation (ou d'apprentissage) durant laquelle une modèle de scène est construit à l'aide d'une séquence d'entrainement. Ce modèle est ensuite accessible depuis l'analyse de bas-niveau, de niveau intermédiaire et de haut-niveau, comme illustré sur la Figure 4.

L'**initialisation du système** (Chapitre 3) consiste en l'analyse d'une séquence d'apprentissage et l'extraction de caractéristiques de bas-niveau et de niveau intermédiaire dans l'objectif de construire un modèle sur la structure de la scène. Ce modèle est enrichi par une estimation approximative de la profondeur dans l'image par rapport à la caméra permettant un découpage de la zone d'intérêt en cellules contenant approximativement la même

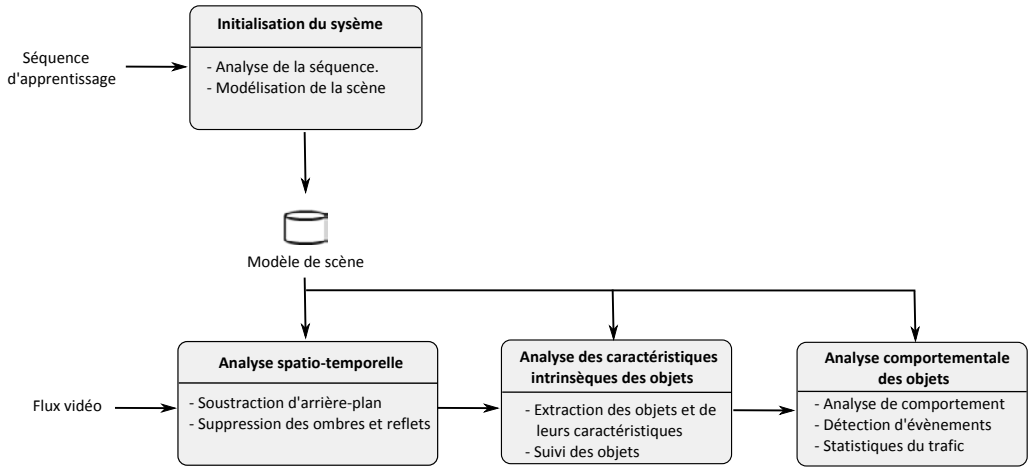


FIGURE 4: Les trois niveaux sémantiques pour l'analyse vidéo.

surface dans la scène. A chacune des cellules est attribué le sens de direction du trafic, tout écart à ce modèle de mouvement engendre une alerte remontée aux opérateurs. Cette étape non supervisée fournit les délimitations des voies, une estimation de la profondeur dans l'image, le modèle couleur d'arrière-plan de la scène ainsi que le sens de direction du trafic routier. En fonction du contexte d'application, chaque cellule peut être enrichie d'une information sémantique permettant d'identifier une situation anormale, comme par exemple l'interdiction de circulation sur la voie de gauche pour les poids lourds.

L'**analyse de bas-niveau** (Chapitre 4) consiste à extraire de l'image les régions en mouvement à partir de caractéristiques de bas-niveau. Cette étape repose sur la construction d'un modèle statistique couleur de l'arrière-plan de la scène sous surveillance. Initialisé pendant l'analyse de la séquence d'apprentissage, ce modèle nécessite une mise à jour régulière pour prendre en compte les changements de luminosité. Cette maintenance consiste à réestimer de façon récursive les paramètres caractérisant le modèle. Durant l'étape de segmentation des objets, tout pixel s'écartant du modèle couleur estimé est considéré comme étant potentiellement en mouvement. Pour prendre en compte les changements locaux de luminosité, une étape de détection d'ombres et de reflets est adoptée. Parallèlement à la segmentation couleur, une différence de gradient combinée à l'estimation du flot optique fournit un masque des vecteurs mouvement. Ce masque permet de valider ou d'invalider la présence des objets dans la scène.

L'**analyse de niveau intermédiaire** (Chapitre 5) consiste à exploiter les caractéristiques de bas-niveau afin d'en extraire une information sémantique de plus haut niveau. Cette analyse comporte les étape d'extraction des régions en mouvement et de leurs caractéristiques, ainsi que de leur suivi au cours du temps. L'identité des régions est maintenu au cours du temps et est modélisée sous la forme d'objets vidéo comportant aussi bien les caractéristiques de bas-niveau (couleur, texture, ...) que des caractéristiques de plus haut niveau telles que la classe d'objet ou les trajectoires des objets. Le suivi d'objets est abordé à l'aide d'une modélisation statistique de la configuration des objets (positions, vitesses) basée sur un modèle de mouvement à vitesse constante. L'association entre les régions détectées dans l'image courante et les objets vidéos s'effectue sous la forme d'hypothèses d'association. Les ambiguités d'association (division et fusion d'objets) sont résolues

à l'aide d'une analyse à plus long terme.

L'**analyse de haut-niveau** (Chapitre 6) consiste à interpréter les résultats issus des analyses précédentes. Cette étape fournit une information sémantique sur l'état du trafic et comporte un module de détection d'évènements permettant le comptage de véhicules, la détection d'arrêt ou de contre-sens, la détection de changement de voies, ...

Organisation du document

Ce document est divisé en six chapitres et s'organise de la façon suivante :

Le chapitre 1 dresse un état de l'art des méthodes couramment utilisées dans le domaine de la vidéo surveillance et traite des techniques de détection d'objets, de suivi d'objets et d'analyse de comportement.

Le chapitre 2 traite de l'architecture du système proposé. Après une description des caractéristiques d'une scène autoroutière, la chaine complète et les étapes du traitement sont présentées, ainsi que les métriques d'évaluation des performances du système.

Le chapitre 3 décrit la procédure d'initialisation utilisée pour construire un modèle de la scène sous surveillance.

Le chapitre 4 décrit le modèle statistique utilisé pour modéliser la couleur de l'arrière-plan de la scène ainsi que le sens de direction du trafic routier.

Le chapitre 5 traite de l'extraction des objets et leur suivi au cours du temps.

Le chapitre 6 décrit les techniques utilisées pour l'extraction de contenu sémantique dans la scène.

Chapitre 1

Etat de l'art

Sommaire

Introduction

La vidéo-surveillance intelligente (*smart video surveillance*) est un processus qui consiste à identifier automatiquement dans des séquences vidéo, des objets, des comportements ou des événements particuliers (prédéfinis par un utilisateur ou appris par le système). Elle analyse et transforme les données issues d'une (ou plusieurs) caméra en une interprétation sémantique directement exploitable par un opérateur humain. Par exemple, lorsqu'une anomalie est détectée, le système peut envoyer une alerte au personnel afin qu'il puisse prendre une décision sur l'intervention adéquate à mettre en place. Généralement, les systèmes de vidéo-surveillance sont composés des étapes suivantes :

- Détection d'objets en mouvement.
- Extraction, classification et suivi des objets.
- Analyse comportementale des objets.

La première section 1.1 est consacrée à la détection des objets en mouvement dans une séquence vidéo. La Section 1.2 est consacrée à l'extraction des objets (et leurs caractéristiques) et au suivi d'objets. Finalement, la Section 1.3 fournit un aperçu de quelques techniques utilisées pour l'analyse de comportement.

1.1 Détection d'objets en mouvement

Nous considérons le problème de détection de mouvement comme un problème de segmentation consistant à séparer ou classer les pixels en 2 classes distinctes, l'arrière-plan (*background*) et l'avant-plan (*foreground*). Les zones de l'arrière-plan font référence à toute structure ou objet situé dans le champ de vision de la caméra et ne subissant pas (ou peu) de changements au cours du temps, tandis que les régions du *foreground* correspondent aux éléments de la scène en déplacement (ou susceptible de l'être). L'estimation de l'arrière-plan est une étape importante dans de nombreuses applications de vidéo-surveillance. En pratique, certains éléments de l'arrière-plan peuvent changer d'apparence (conditions climatiques, changements de luminosité) et/ou être en mouvement (mouvement des branches d'un arbre). Les principales différences entre les méthodes de soustraction d'arrière-plan résident dans la modélisation de l'arrière-plan et la façon de calculer la différence entre l'image et le modèle. Ces méthodes ont un point commun, elles reposent sur l'analyse spatio-temporelle de l'intensité des pixels. Il s'agit d'analyser temporellement les valeurs des pixels afin d'en extraire une information, dite de bas-niveau, mais essentielle aux traitements de plus haut niveau (extraction et suivi d'objets, analyse de comportement). Nous présentons dans cette section les méthodes standards couramment utilisées dans la littérature.

1.1.1 Différences temporelles

Les approches basées sur la différence entre images (ou différence temporelle, DT) ne nécessitent pas de modèles d'arrière-plan et extraient les régions en mouvement par analyse de la variation temporelle de l'intensité lumineuse des pixels. L'approche la plus simple consiste à observer la différence absolue entre deux images d'entrée. Un seuillage permet ensuite de déterminer les changements dans la scène observée. Si I_t est l'intensité lumineuse de la t-ième image et (u, v) les coordonnées d'un pixel de cette image, alors la

différence en valeur absolue s'exprime par

$$\Delta_t(u,v) = |I_t(u,v) - I_{t-n}(u,v)| \tag{1.1}$$

avec généralement une valeur faible pour n compris entre 1 et 5. L'image des zones en mouvement (*foreground*), notée $M(u,v)$ est extraite par seuillage telle que

$$M_t(u,v) = \begin{cases} 1 & \text{si} \quad \Delta_t(u,v) \geq \tau \\ 0 & \text{sinon} \end{cases} \tag{1.2}$$

Cette approche s'adapte très rapidement aux changements de luminosité, mais est peu robuste face aux bruits d'acquisition. Les résultats de la segmentation dépendent uniquement du choix de la méthode de seuillage utilisée et souvent un simple seuillage ne permet pas d'extraire de façon précise les zones mobiles. Notamment, lorsque les objets sont uniformes en intensité et/ou qu'ils se déplacent lentement, cette méthode laisse apparaître à l'intérieur et à proximité des objets des erreurs de segmentation comme illustré sur la Figure 1.1(problème d'ouverture). Pour palier à ces problèmes, certains auteurs [Kameda 1996] suggèrent l'utilisation d'une double différence à partir de 3 images consécutives. Les régions en mouvement sont extraites à partir de l'image résultante de la double différence de la façon suivante. Dans un premier temps, deux différences entre images sont calculées et un seuillage est effectué afin d'extraire deux masques binaires. Une opération *ET logique* permet ensuite d'obtenir l'image de double-différence (binaire). Malgré les limitations des méthodes basées sur la différence temporelle, elles permettent une extraction rapide du mouvement à travers l'analyse de l'intensité lumineuse des pixels. Elles ne sont cependant pas adaptées lorsque les objets se déplacent lentement ou lorsque les objets sont homogènes en couleur. Elles sont donc rarement employées seules et sont utilisées afin d'extraire une information de bas-niveau sur le mouvement dans la scène.

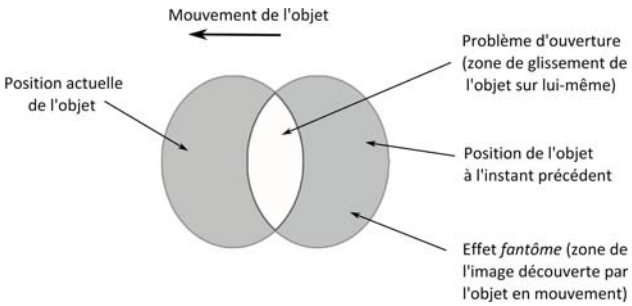


FIGURE 1.1: Illustration du problème d'ouverture (adaptée de [Migliore 2006]) des algorithmes basés sur la différence entre images. Lorsque l'objet se déplace lentement et qu'il possède une couleur uniforme, la soustraction laisse apparaître un effet *fantôme* (surface anciennement recouverte par l'objet), et une zone d'ouverture au centre de l'objet.

Des méthodes hybrides ont été proposées combinant cette information avec une soustraction d'arrière-plan utilisant un modèle de fond. Citons par exemple la méthode proposée par [Spagnolo 2006] où une double différence permet d'obtenir les régions en mouvement. Une fois estimées, une soustraction d'arrière-plan à l'aide d'une image de référence est utilisée pour affiner les résultats. Une approche semblable a été proposée dans [Migliore 2006],

où la différence inter-image est combinée à une différence avec une image d'arrière-plan afin de résoudre le problème d'ouverture et d'effet *fantôme*. La description des méthodes de soustraction d'arrière-plan basées sur l'estimation d'une image de référence fait l'objet de la section suivante.

1.1.2 Soustraction d'arrière-plan

Utilisation d'une image

Dans sa forme la plus simple, l'arrière-plan consiste en une image B dans laquelle la scène est représentée sans objet. La segmentation consiste à étudier chaque pixel de l'image et à les comparer avec l'image de référence B. Si la différence est supérieure à un seuil τ, le pixel est classé en tant que *foreground*, sinon il est classé en tant que *background*. Le résultat de la segmentation est une image binaire M donnée par

$$M_t(u,v) = \begin{cases} 1 & \text{si } |I_t(u,v) - B(u,v)| \geq \tau \\ 0 & \text{sinon} \end{cases} \tag{1.3}$$

Il est souvent difficile d'obtenir l'image d'arrière-plan et il est nécessaire de la mettre à jour régulièrement pour prendre en compte les changements de luminosité. L'estimation de l'image B peut être effectuée de différentes façons. Elle peut être représentée par la valeur moyenne estimée de façon incrémentale à l'aide de l'Equation 1.4 ou 1.5.

$$B_t(u,v) = \frac{t-1}{t} B_{t-1}(u,v) + \frac{1}{t} I_t(u,v) \tag{1.4}$$

De façon générale, la valeur moyenne récursive permet d'estimer la moyenne récursive (*mean average*) à l'aide du filtrage adaptatif suivant

$$B_t(u,v) = \alpha B_{t-1}(u,v) + (1-\alpha) I_t(u,v) \tag{1.5}$$

Le paramètre α est appelé le taux d'apprentissage et permet de contrôler la vitesse de mise à jour de l'image d'arrière-plan. Une valeur élevée aura pour conséquence de ne prendre que très peu en compte la nouvelle image, tandis qu'une valeur très faible permettra une adaptation très rapide de l'arrière-plan (lorsque $\alpha = 0$, l'image d'arrière-plan correspond à l'image précédente).

Une approche alternative consiste à estimer la valeur médiane des intensités des pixels à partir des n derniers échantillons d'une fenêtre temporelle telle que

$$B_t(u,v) = \text{Median}\left\{I_{t-n+1}(u,v), \ldots, I_{t-1}(u,v), I_t(u,v)\right\} \tag{1.6}$$

L'inconvénient majeur de cette approche est la nécessité de conserver en mémoire les images précédentes pour l'estimation. Une implémentation récursive a été proposée dans [McFarlane 1995] où la valeur médiane est incrémentée de 1 si le pixel est supérieur à sa valeur, et décrémentée de 1 le cas échéant. L'inconvénient est une lente adaptation et demande par conséquent une longue période d'apprentissage. De manière générale, l'incrémentation (ou la décrémentation) de la valeur médiane s'effectue à l'aide d'une constance notée c telle que :

$$B_t(u,v) = \begin{cases} B_t(u,v) + c & \text{si } I_t(u,v) > B_{t-1}(u,v) \\ B_t(u,v) - c & \text{si } I_t(u,v) < B_{t-1}(u,v) \\ B_t(u,v) & \text{sinon} \end{cases} \tag{1.7}$$

Modèle gaussien

Dans [Wren 1997], les auteurs font l'hypothèse suivante : les pixels d'arrière-plan sont indépendamment distribués selon une distribution gaussienne $\mathcal{N}$. Cette approche probabiliste permet de prendre en compte les faibles variations d'intensité lumineuse considérées comme étant des bruits de mesures dans le modèle. Le mode de la distribution (moyenne) caractérise la couleur dominante prise par le pixel (couleur d'arrière-plan) et la variance caractérise la variabilité autour de cette valeur. Pour prendre en compte les changements de luminosité, les paramètres de moyenne et de variance sont mis à jour régulièrement de façon récursive à l'aide d'un paramètre $0 \leq \alpha \leq 1$ appelé *taux d'apprentissage* et permettant de régler la vitesse d'adaptation. L'arrière-plan et sa variance sont estimés, pour chaque image, avec

$$\begin{aligned} B_t &= \alpha.B_{t-1} + (1-\alpha)I_t \\ V_t &= \alpha.V_{t-1} + (1-\alpha)(B_t - I_t)^2 \end{aligned} \tag{1.8}$$

L'utilisation d'un tel modèle permet de définir la vraissemblance $\mathcal{L}$ pour chaque pixel d'appartenir à l'arrière-plan selon

$$\mathcal{L}(I_t) = \mathcal{N}\left(I_t | B_t, V_t\right) \tag{1.9}$$

L'image *foreground* est construite en définissant les pixels d'arrière-plan comme étant ceux suffisamment éloignés de la valeur moyenne. Généralement, la valeur de la variance est directement exploitée dans la décision de la façon suivante :

$$M_t(u,v) = \begin{cases} 1 & \text{si } |I_t(u,v) - B_t(u,v)| < 2.5\sqrt{V_t(u,v)} \\ 0 & \text{sinon} \end{cases} \tag{1.10}$$

Filtrage prédictif

Dans [Toyama 1999], les auteurs proposent un algorithme basé sur un filtrage prédictif à l'aide d'un filtre de Wiener. Une prédiction linéaire basée sur l'historique des anciennes valeurs est effectuée ; si un pixel s'écarte de la prédiction, alors il est déclaré en tant que pixel en mouvement. La prédiction de la valeur du pixel est donnée par

$$x_t = -\sum_{k=1}^{p} a_k x_{t-k} \tag{1.11}$$

avec a_k les coefficients de prédiction du filtre, x_t la prédiction de la valeur du pixel à l'instant t. Le filtre utilise les p échantillons les plus récents de l'historique pour effectuer la prédiction, et les coefficients a_k sont déterminés à partir de la covariance des valeurs de x_t [Makhoul 1975]. La décision de classification utilise l'erreur de prédiction e, définie par

$$E[e_t^2] = E[s_t^2] + \sum_{k=1}^{p} a_k E[s_t s_{t-k}] \tag{1.12}$$

Pour chaque pixel, cette erreur est évaluée et si un pixel s'écarte de plus de $4.0\sqrt{E[e_t^2]}$ de la prédiction, il est considéré comme étant en mouvement. Le filtrage de Kalman est une autre

approche prédictive fournissant une solution optimale lorsque le système dynamique relatif au problème est représentable sous forme linéaire et qu'il est perturbé par un bruit supposé gaussien. Dans le cadre de la modélisation d'arrière-plan, de nombreuses versions différentes ont été proposées ([Ridder 1995], [Gao 2001], [Zhong 2003], [Lei 2010], [Ahmad 2011]) et se différencient généralement par le choix des caractéristiques utilisées. Notons x_t le vecteur d'état d'un pixel p à l'instant t. Ce vecteur décrit la valeur d'intensité lumineuse du pixel ainsi que sa dérivée temporelle et s'écrit $x_t = \left[I_t, \dot{I}_t\right]^T$. Chaque pixel de l'image est mis à jour récursivement selon

$$\begin{bmatrix} I_t \\ \dot{I} \end{bmatrix} = A.\begin{bmatrix} I_{t-1} \\ \dot{I}_{t-1} \end{bmatrix} + K.\left(I_t - H.A.\begin{bmatrix} I_{t-1} \\ \dot{I}_{t-1} \end{bmatrix}\right) \tag{1.13}$$

avec A la matrice d'évolution décrivant la dynamique de l'arrière-plan et H la matrice d'observation décrivant la relation entre la mesure et l'état. Le gain K du filtre caractérise généralement le *taux d'apprentissage* et s'écrit $K = \begin{bmatrix}\alpha & \alpha\end{bmatrix}^T$.

Dictionnaire de mots visuels

L'utilisation d'un dictionnaire de mots visuels pour modéliser l'arrière-plan a par exemple été proposée dans [Kim 2004], [Li 2006], [Zhang 2009], [Shah 2011]. Cette approche consiste à construire pour chaque pixel un modèle représenté par un ensemble de variables appelées mots visuels caractérisant son état actuel.

Dans le modèle $W4$ (Minimum-Maximum filter) [Haritaoglu 2000], chaque pixel est caractérisé par un jeu de 3 valeurs, le minimum d'intensité (Min), le maximum d'intensité (Max) et la différence maximum d'intensité entre 2 images consécutives (Diff). Ces valeurs sont initialement estimées durant une phase d'apprentissage et mises à jour régulièrement au cours du temps. Un pixel est considéré comme étant en mouvement si une des deux conditions suivantes est remplie :

$$|\mathrm{Min}_t - I_t| > D_t \qquad \text{ou} \qquad |\mathrm{Max}_t - I_t| > D_t \tag{1.14}$$

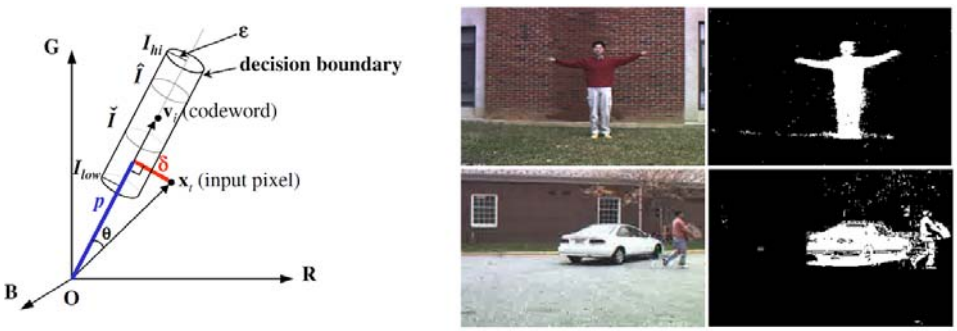


FIGURE 1.2: Modélisation par dictionnaire de mots visuels présentée dans [Kim 2004]. (À gauche) L'ensemble des mots visuels forment un cylindre dans l'espace couleur caractérisant l'arrière-plan. (À gauche) Résultats obtenus par les auteurs en utilisant ce modèle.

Dans [Kim 2004], le dictionnaire est enrichi par la fréquence d'occurrence du mot f, la durée maximale durant laquelle le mot n'a pas été sollicité pendant l'apprentissage λ

et le premier et dernier accès au mot visuel p et q. Ces valeurs sont utilisées pendant la période d'apprentissage pour construire le modèle. La classification *background-foreground* s'effectue en calculant la différence de couleur et de luminosité (*brightness*) selon la Figure 1.2 : Si la valeur du pixel est contenue dans le cylindre formé par I_low, I_high et par la distance δ sur la Figure 1.2, alors il est considéré comme appartenant à l'arrière-plan.

1.1.3 Segmentation par mélange de gaussiennes

Lorsque l'arrière-plan est fortement dynamique (par exemple lorsque la scène contient des branches d'arbres en mouvement) la variance des pixels devient rapidement élevée et il n'est plus possible de représenter la couleur de l'arrière-plan à l'aide d'une seule gaussienne. Pour prendre en compte la multi-modalité de ce type de variation, Stauffer *et al.* proposent dans [Stauffer 1999] l'utilisation d'un modèle de mélange de gaussiennes (*Gaussian Mixture Model*, GMM) dont les paramètres sont estimés à l'aide d'un algorithme de type *Expectation-Maximization* (EM). Pour permettre une analyse en temps réel, une version récursive de l'algorithme EM est proposée, dont les approximations ont été identifiées dans [Power 2002]. L'algorithme original a été étudié intensivement depuis son apparition et de nombreuses variantes ont été proposées dans la littérature [Bouwmans 2008].

L'intensité pour chaque pixel de l'image est modélisée par un mélange de K distributions gaussiennes, avec K le nombre de composantes du mélange généralement compris entre 3 et 5. La probabilité de voir apparaître un pixel d'intensité x_t à l'instant t est estimée par

$$P(x_t) = \sum_{k=1}^{K} w_{k,t} \mathcal{N}(x_t | \mu_{k,t}, \Sigma_{k,t}) \tag{1.15}$$

avec $\mu_{k,t}$, $\Sigma_{k,t}$, $w_{k,t}$ respectivement les moyennes, covariances et poids associés à la composante k à l'instant t. Les paramètres des distributions sont mis à jour de façon récursive en approximant l'algorithme EM par un algorithme *K-mean*. Chaque pixel est comparé à l'ensemble des composantes du modèle. Le poids des distributions sont mis à jour selon

$$w_{k,t} = w_{k,t-1} + \alpha(o(k,t) - w_{k,t-1}) \tag{1.16}$$

avec $o(k,t) = 1$ si la composante k correspond au pixel, $M(k,t) = 0$ sinon. Lorsqu'une composante est sélectionnée, ses paramètres de moyennes et variances sont mis à jour selon

$$\begin{aligned} \mu_t &= \mu_{t-1} + (\alpha / w_{k,t})(x_t - \mu_{t-1}) \\ \sigma_t^2 &= \sigma_{t-1}^2 + (\alpha / w_{k,t})(x_t - \mu_t)^T (x_t - \mu_t) \end{aligned} \tag{1.17}$$

avec α une variable définissant la vitesse de mise à jour du modèle (*taux d'apprentissage*). Les K distributions sont classées selon le ratio $w_{k,t}/\sigma_{k,t}^2$ et les B premières distributions sont considérées comme représentatives de l'arrière-plan

$$B = \text{argmin}_b \left(\sum_{k=0}^{b} w_k > T \right) \tag{1.18}$$

avec T un seuil prédéfini. La décision et la classification s'effectue en comparant chaque pixel aux B distributions, si la différence est inférieure à 2,5 fois la variance de la composante, alors il est considéré comme étant en mouvement.

Dans [Zivkovic 2004], Z. Zivkovic *et al.* proposent une estimation automatique du nombre de composantes à chaque image et supposent une densité *a priori* de Dirichlet à coefficients c négatifs sur les poids $w_{k,t}$ telle que

$$f(w_{1,t}, w_{2,t}, \ldots, w_{K,t}) = \frac{1}{a} \prod_j w_{j,t}^{-c} \tag{1.19}$$

avec a un coefficient de normalisation et c un paramètre de contrôle du nombre d'échantillons nécessaires pour considérer qu'une composante soit visible. Les travaux de l'auteur ont conduit à l'introduction d'une nouvelle variable c_T dans la mise à jour des poids des composantes, l'équation 1.16 devient

$$w_{k,t} = w_{k,t-1} + \alpha(o(k,t) - w_{k,t-1}) - \alpha c_T \tag{1.20}$$

L'ajout du paramètre c_T permet de supprimer une composante dont le poids deviendrait négatif. La figure 1.3 montre quelques exemples du nombre variable de composantes dans la modélisation de plusieurs scènes.

FIGURE 1.3: Illustration du nombre de composantes variables dans différentes scènes [Zivkovic 2004]. Le modèle de mélange comporte un nombre variable de composantes gaussiennes en fonction de la dynamique de la scène.

1.1.4 Estimation basée sur un noyau

L'estimation basée sur un noyau (*Kernel Density estimation*, KDE) est une technique d'estimation de densité de probabilité à partir d'un ensemble d'échantillons sans aucune hypothèse sur la forme de la distribution dont ils sont issus. Soit un ensemble d'échantillons $S = \{x_i\}_{i=1\ldots N}$ distribué selon la loi de densité $p(x)$. Une estimation de $p(x)$ peut être évaluée en utilisant

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(x - x_i) \tag{1.21}$$

avec K_σ est une fonction noyau (fenêtrage) de largeur σ. Dans [Elgammal 2000], les auteurs proposent d'estimer l'arrière-plan à l'aide des n valeurs les plus récentes.

En utilisant un noyau gaussien et un vecteur couleur x de dimension $d = 3$, la densité est estimée selon

$$p(x) = \frac{1}{N}\sum_{i=1}^{N}\prod_{c=1}^{d}\frac{1}{\sqrt{2\pi\sigma_c^2}}\exp -\frac{(x_{c,t} - x_{c,i})^2}{2\sigma_c^2} \tag{1.22}$$

La largeur du noyau σ est estimée à partir de la valeur médiane m des différences entre images consécutives $|I_t - I_{t-1}|$ [Elgammal 2000] :

$$\sigma = \frac{m}{0.68\sqrt{(2)}} \tag{1.23}$$

1.1.5 Classification par analyse en composantes principales

L'utilisation de méthodes d'analyse de données telles que l'ACP (Analyse en Composantes Principales) a également été appliquée pour la modélisation de l'arrière-plan [Oliver 2000], [Li 2003], [Verbeke 2007]. L'objectif consiste à créer une matrice de données à partir d'un ensemble d'images d'apprentissage afin d'y appliquer une ACP et d'en dégager une base de vecteurs propres (appelée *eigenbackground*).

En pratique, la construction du modèle d'arrière-plan est réalisée à partir d'un ensemble de N images d'apprentissage prises à des instants différents dans la vidéo. Afin d'obtenir une plus grande représentativité de l'arrière-plan, ces images sont prises à des instants non consécutifs. À partir de ces images est construite une image moyenne I_μ et une matrice de covariance C calculée sur le vecteur X_I représentant le réarrangement des images d'entrainement sous la forme d'un vecteur 1D. Si w, h et c sont respectivement la largeur, la hauteur et le nombre de canaux couleur, alors chaque image de la base d'apprentissage est représentée sous forme d'un vecteur 1D de taille (wxhxc) et la matrice contenant l'ensemble des N images est de taille (wxhxcxN). Une fois calculée, la matrice de covariance est diagonalisée pour obtenir une base de vecteurs propres Φ et un matrice diagonale Λ comportant les valeurs propres associées (seuls les vecteurs propres associés aux K plus grandes valeurs propres sont conservés).

Une fois l'apprentissage terminé, chaque nouvelle image I est projetée dans l'espace de dimension réduite (défini par la base de vecteurs propres). Les objets en mouvement sont extraits en calculant la distance entre l'image d'entrée I et l'image reconstruite à partir de sa projection notée I_Φ et donnée par

$$I_\Phi = (I - I_\mu)\Phi\Phi^T + I_\mu \tag{1.24}$$

En utilisant une mesure de similarité euclidienne, la carte de distance D est obtenue à l'aide de l'Equation 1.25 qui est ensuite seuillée pour obtenir l'image des pixels en mouvement (*foreground*).

$$D = \sqrt{(I - I_\Phi)^2} \tag{1.25}$$

1.1.6 Autres méthodes

Plus récemment, les auteurs de [Wang 2011a] proposent un schéma multi-résolution à 3 niveaux (low, middle, high) à l'aide d'une pyramide d'images. Une fois la pyramide générée, l'image de basse résolution I_{low} est dans un premier temps utilisée pour identifier les régions

d'intérêts par soustraction d'arrière-plan. Un modèle gaussien basé sur l'intensité des pixels est construit durant une phase d'apprentissage. Les paramètres moyennes et écart-types des gaussiennes sont estimés à l'aide de l'Equation 1.8 dans laquelle les auteurs proposent une pondération (fonction de l'écart de l'intensité à la valeur médiane) afin d'éliminer les valeurs extrêmes. La distance de Mahalanobis entre la valeur du pixel et le modèle permet finalement d'obtenir le masque *foreground* $\mathcal{M}_{low}$ de basse résolution. L'image de moyenne résolution I_{mid} est ensuite analysée pour affiner le masque $\mathcal{M}_{low}$ en utilisant 3 critères de saillance basés sur la couleur (S_{color}), l'ombre (S_{shad}) et le contour (S_{edge}). Le critère couleur consiste à comparer les composantes r, g et b des pixels avec un modèle gaussien (similaire à l'analyse basse résolution). Le critère d'ombre est basé sur la distorsion chromatique (Section 4.2.2) et le critère contour est estimé par comparaison des gradients de l'image et de l'arrière-plan. Ces critères sont combinés et comparés au masque de basse résolution afin d'obtenir celui de résolution moyenne $\mathcal{M}_{mid}$. Finalement, l'analyse haute-résolution exploite de nouveau le critère contour et combine les résultats avec le masque de moyenne résolution $\mathcal{M}_{mid}$. Cette approche est attractive pour son coté multi-résolution, cependant aucun processus de mise à jour n'est proposé.

Dans les travaux issus de [Wang 2011b], les auteurs proposent une version étendue de la mixture de gaussienne appelée *Local-Patch Gaussian Mixture Model* (LPGMM) dans laquelle la relation spatiale des pixels est pris en compte. Les paramètres moyenne et matrice de covariance sont estimés à partir des moyennes de l'entourage spatial pour chaque pixel. Un critère basé sur la distance de Mahalanobis permet d'extraire les zones en mouvement. En guise de post-traitement, un détecteur d'ombre basé SVM affine les résultats et fournit le masque final d'avant-plan.

1.2 Extaction et suivi d'objets

Le suivi d'objets consiste à associer les objets détectés dans l'image courante avec ceux détectés aux images précédentes. Il s'agit de maintenir l'identité des objets et l'évolution temporelle de leurs positions (ou d'une autre caractéristique). Ce problème peut être vu comme un problème de localisation spatiale et temporelle des objets présents dans la scène. De nombreuses approches de suivi d'objets se basent sur l'apparence d'un objet. L'utilisation de ces méthodes nécessite une représentation pertinente de l'objet possédant des primitives fiables pour décrire son contenu. Le choix d'un modèle d'apparence est une étape cruciale et joue un rôle central dans les techniques de suivi visuel d'objet. Les performances de la reconnaissance de l'objet au cours de temps sont fortement liées au choix des primitives visuelles utilisées.

1.2.1 Représentation des objets

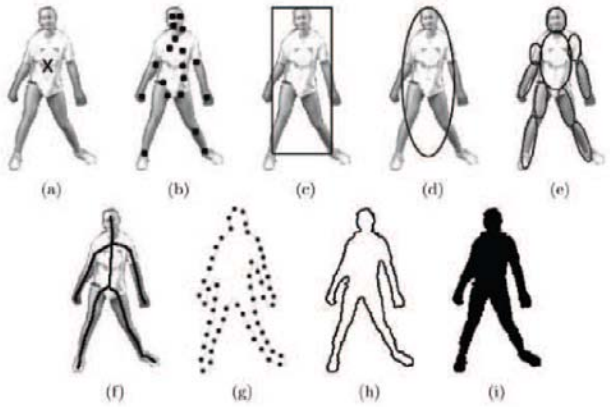


FIGURE 1.4: Exemples de représentation de la forme d'un objet (adapté de [Yilmaz 2006]). (a)-(b) Représentation à l'aide d'un ensemble de points, (c)-(d)-(e) Représentation à l'aide d'un ensemble de formes géométriques, (f) Représentation à l'aide du squelette, (g)-(h)-(i) Représentation à l'aide du contour (partiel ou non) et de la silhouette d'un objet.

Les objets peuvent être représentés de nombreuses façons et le choix de la représentation d'un objet dépend fortement du domaine d'application. Nous reprenons dans cette section la classification proposée dans [Yilmaz 2006].

Représentation de la forme d'un objet

Les représentations basées sur la forme d'un objet sont nombreuses (Figure 1.4) : un ensemble de points, une forme géométrique (ex. un rectangle, une ellipse), un contour, une silhouette, un modèle 2D ou 3D, ...

- **Points** - Un objet peut être représenté par un point. Il peut s'agir par exemple de son centre de masse, du centre de sa boite englobante, ou tout autre point caractéristique de la forme. Il s'agit d'une représentation simple de la localisation 2D (ou 3D) de l'objet. Cette représentation se généralise à un ensemble de points auxquels peuvent être associés des descripteurs locaux de couleur, de texture ou de mouvement.

- **Formes géométriques** - L'objet est représenté par une forme géométrique, par exemple un rectangle ou une ellipse, permettant une description de la dimension de l'objet. Le mouvement des objets associés est généralement modélisé à l'aide de transformations de translations, affines ou projectives. Cette représentation se généralise par des modèles articulés, composés d'un ensemble de formes géométriques 2D ou 3D particulièrement utilisé dans la modélisation du corps humain [Brox 2010], [Yang 2011].

- **Contours** - La représentation d'un objet par son contour permet une description plus complète de la forme d'un objet. Un contour peut être vu comme étant un

ensemble de points ordonnés généralement estimé à l'aide d'une analyse du gradient d'intensité au voisinage d'un pixel. La région interne du contour est appelée silhouette de l'objet et peut être utilisée conjointement à l'information de contour pour le suivi d'objets [Rosenhahn 2005], [Yilmaz 2004].

– **Squelette** - Le squelette d'un objet peut être extrait pour caractériser la forme d'un objet ou d'une forme géométrique. Ce modèle est utilisé en tant que descripteur de forme pour la reconnaissance d'objets [Herda 2001]. Cette représentation peut être aussi bien utilisée sur des objets déformables que des objets rigides. Dans [Aziz 2011], les auteurs utilisent le squelette des formes obtenues par un module de détection de mouvement afin d'estimer la position de la tête et d'effectuer une tâche de comptage (voir Figure 1.5).

FIGURE 1.5: Exemple d'utilisation du squelette pour le représentation des piétons [Aziz 2011]. Le squelette des formes détectées est représenté sous la forme d'un graphe afin de détecter la position de la tête des piétons présents dans la scène et de mettre en place un processus de comptage de piétons.

Représentation de l'apparence d'un objet

Les caractéristiques d'apparence sont généralement utilisées conjointement aux caractéristiques de formes dans l'objectif de compléter la représentation de l'objet à suivre. Elles ont pour objectif de résumer l'information contenue dans le signal lumineux (image). Parmi les méthodes existantes, on retrouve les fonctions de densité de probabilité (estimateurs à noyaux, histogrammes ou modèle de mélange de gaussiennes) les patrons (*template*) ou encore les modèles dynamiques d'apparence (*Active Appearance Models*).

– **Densité de probabilité d'apparence** - L'apparence d'un objet peut être modélisée à l'aide de la répartition des valeurs des couleurs qu'il contient (ou de toute autre caractéristique) sous forme de densité de probabilité. La fonction de densité de probabilité peut être estimée par un estimateur à noyau (*Kernel Density Estimator*, *KDE*) [Huang 2007], représentée sous la forme d'un histogramme [Gevers 2004], ou encore sous une forme paramétrique à l'aide d'une gaussienne ou d'un mélange de gaussiennes [McKenna 1999].

– **Patrons** - Cette représentation considère directement le signal lumineux dans les images. Dans le cadre du suivi d'objet, les méthodes basées sur un patron (*Template-based matching*) effectuent directement la mise en correspondance 2D sur une partie de l'image sans passer par une phase d'extraction de caractéristiques. La recherche des paramètres de la transformation se fait généralement en optimisant un critère de

corrélation [Pressigout 2005].

– **Modèle dynamique d'apparence** - Les modèles dynamiques d'apparence (*Active appearance models*) modélisent généralement les objets à travers des caractéristiques de formes et d'apparence des objets. Ces méthodes ont pour objectif de prendre en compte les variations d'apparence d'un objet. Ces variations peuvent être de deux types différents, intrinsèque ou extrinsèque. La déformation de la forme ou le changement de pose d'un objet est considéré comme une variation intrinsèque, tandis que les variations causées par le changement de luminosité, le mouvement de la caméra ou l'occlusion sont considérés comme une variation extrinsèque. Ces modèles nécessitent généralement une phase d'apprentissage permettant d'apprendre à partir d'un jeu d'exemples la forme et l'apparence d'un objet [Sun 2010], [Wang 2010].

1.2.2 Primitives pour le suivi d'objets

La sélection des primitives visuelles joue un rôle essentiel dans les techniques de suivi d'objets. Ces primitives ont pour objectif de décrire les propriétés visuelles de l'objet dans l'image. Elles sont ensuite utilisées pour détecter ou suivre les objets à l'aide d'une métrique de comparaison. Ces primitives peuvent être choisies manuellement par l'utilisateur en fonction de l'application ou encore sélectionnées de façon automatique à l'aide, par exemple, d'une analyse en composante principale (PCA).

Couleur

La couleur est sans doute la primitive la plus utilisée pour décrire un objet. Elle est directement accessible depuis le signal lumineux (image) et fournit une description intuitive de l'apparence d'un objet. La couleur prédominante d'un objet peut être utilisée directement en recherchant dans l'image les pixels de mêmes valeurs ou, de façon plus générale, rechercher l'objet dans l'image à l'aide de sa distribution couleur. Par exemple, dans [Lu 2001], les auteurs représentent les objets à l'aide d'un histogramme couleur. Dans [Perez 2002] les auteurs proposent de partitionner spatialement l'objet afin de le modéliser à l'aide d'un ensemble histogrammes associés à chacune des régions obtenues. Ou encore dans [Xie 2011] où les auteurs proposent un modèle paramétrique (mélange de gaussiennes) pour représenter la répartition de la couleur de la peau.

Généralement, l'information couleur est représentée dans l'espace de couleur RGB. L'inconvénient majeur d'une telle représentation est sa forte corrélation entre les composantes couleurs et sa non uniformité dans la perception faite par l'humain (la différence entre les couleurs dans l'espace RGB ne correspond pas à la différence perçue par l'oeil [Paschos 2001]). De nombreuses évaluations ont été faites pour évaluer les performances entre les différents espaces de couleur (exemple [Van de Sande 2008]). L'espace HSV possède par exemple un certain degré d'invariance contre les changements d'illumination, et l'espace L*a*b est un espace de couleur approximativement uniforme (perceptuellement) [Van de Sande 2008].

Gradient

L'information de gradient spatial des objets a été largement utilisé pour caractériser la forme et le contour d'un objet. Cette information est extraite à partir de l'analyse spatiale de l'intensité lumineuse de l'image. Une propriété importante du gradient est sa sensibilité plus faible aux changements de luminosité comparée aux caractéristiques couleurs. Les contours issus du gradient sont exploités dans de nombreuses approches de suivi d'objet. Le gradient permet de définir des points caractéristiques dans les objets (détecteur de Morravec, Harris, ...). L'algorithme CONDENSATION [Isard 1998] (*Conditionnal density propagation*) consiste à initialiser une courbe *spline* sur les contours, et un filtre à particules est utilisé pour mettre à jour les paramètres de la courbe paramétrée. Des techniques de minimisation d'énergie le long des contours des objets ont également été proposées pour suivre les objets sous certaines contraintes de régularisation (*snakes* et contours actifs [Fang 2011]). Les histogrammes d'orientations de gradient (HOG) ont été utilisés en tant que primitives pour la construction de certains descripteurs [Mikolajczyk 2005]. Par exemple, les descripteurs SIFT [Lowe 2004] (*Scale Invariant Feature Transform*) qui combinent un détecteur et un descripteur invariants à l'échelle basés sur la distribution du gradient. Les orientations des gradients dans un voisinage pondéré sont représentées sous forme d'histogrammes (Figure 1.6).

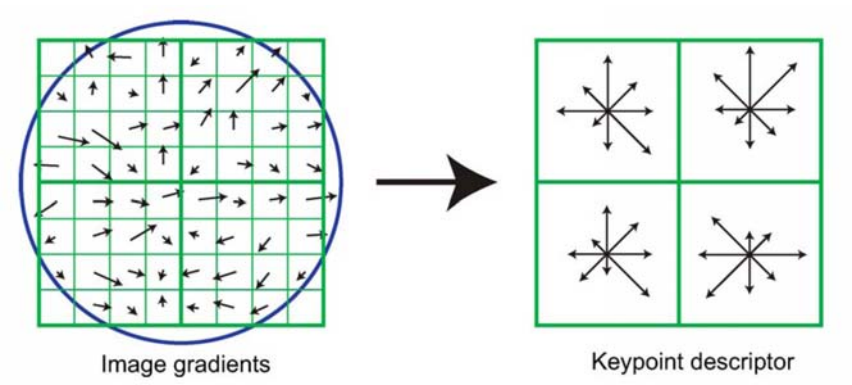


FIGURE 1.6: Construction du descripteur SIFT à l'aide de la norme et de l'orientation du gradient pondérée par noyau gaussien (représenté par le cercle bleu). Les valeurs sont accumulées dans un histogramme d'orientations qui résume l'information contenue dans le voisinage (figures d'après [Lowe 2004]).

Dans l'objectif de diminuer le temps de calcul, le descripteur SURF (*Speed-Up Robust Features* a été proposé dans [Bay 2006]. Les auteurs proposent une approche dans laquelle le détecteur DoG (*Difference of Gaussian*) de SIFT a été remplacé par un détecteur fast-héssien. Quant au descripteur, il se base sur l'utilisation des ondelettes de Haar. Un exemple de détection de points d'intérêt est montré sur la Figure 1.7. Pour une description détaillée, le lecteur intéressé pourra se référer à [Bay 2006].

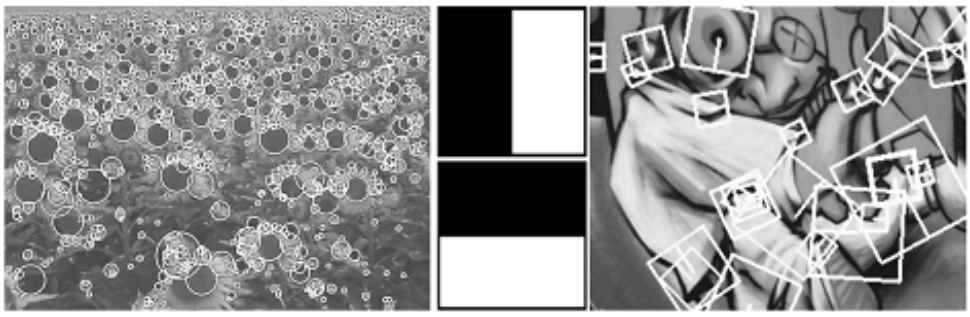


FIGURE 1.7: Détection de points d'intérêt à l'aide du descripteur SURF [Bay 2006]).

Texture

La texture d'un objet est également une caractéristique utilisée pour modéliser et suivre les objets. Les méthodes de modélisation de texture peuvent être classées dans quatre catégories : les modèles statistiques, les modèles structurels et les modèles fondés sur des filtres (spatiaux et/ou fréquentiels). Les modèles statistiques mesurent la distribution spatiale des valeurs des pixels (histogrammes [Boukouvalas 1999], matrices de co-occurence [Haralick 1973], auto-corrélation, ...). Les méthodes structurelles, la texture est représenté par une répétition d'éléments structurels et la texture est modélisée comme étant un arrangement spatial de ces éléments [Vilnrotter 1986]. Quant aux modèles fondés sur les filtres, ils consistent à appliquer un ensemble de filtres à l'image afin d'en étudier la réponse. On retrouve les filtres dans le domaine spatial (filtres de Sobel, de Canny, Robert, ...), ceux dans le domaine fréquentiel (filtre de Fourier) ou dans le domaine spatio-temporel (filtre de Gabor, transformée en ondelettes, ...). Pour une description détaillée des méthodes, le lecteur intéressé pourra se référer à [Xie 2010].

1.2.3 Techniques de suivi d'objets

Il existe de nombreux états de l'art dans la littérature traitant du sujet de suivi d'objets. Nous regroupons dans ce paragraphe les méthodes principalement utilisées par les algorithmes de suivi. Pour une description approfondie des différentes classifications possible, voir [Yilmaz 2006], [Moeslund 2006] et [Hu 2004a]. Dans [Hu 2004a], les auteurs classent les algorithmes de suivi d'objets dans 4 catégories : algorithmes basés régions, basés contours actifs, basés caractéristiques (*features*) et basés sur un modèle. Dans [Yilmaz 2006], les auteurs classent les algorithmes en 3 catégories : suivi de points, suivi à noyaux et le suivi de silhouette. Ces classifications ne sont pas strictes et certaines approches peuvent être représentées dans plusieurs catégories.

Nous reprenons dans cette section la classification proposée dans [Yilmaz 2006]. Nous dissocions cependant les méthodes fondées sur les modèles prédictifs pour leurs attraits et leurs grandes popularités. Ces méthodes appartiennent au suivi à noyaux dans la classification de Yilmaz.

Approches basées sur l'apparence

L'utilisation de modèles d'apparence est sans doute une des approches les plus utilisées pour la détection et le suivi d'objet. Ces approches se basent sur les descripteurs présentés

dans la section 1.2.1 et sur la définition d'une métrique de comparaison.

L'approche par ***Template Matching*** (appariement de gabarit) consiste à comparer l'intensité des pixels entre l'image candidate et le *template*. Les métriques de mesures utilisées sont généralement la norme L1, la norme L2 ou le coefficient de cross-corrélation. Généralement, l'intensité ou les composantes couleurs de l'image sont utilisées, ce qui rend ces méthodes sensibles aux variations de luminosité. Plutôt que d'utiliser une information colorimétrique, [Birchfield 1998] proposent par exemple l'utilisation d'une image gradient pour former le *template*.

L'algorithme ***Mean-Shift*** est sans doute la méthode la plus populaire qui utilise une représentation de l'apparence d'un objet sous forme d'histogrammes. Il s'agit d'une méthode non paramétrique qui maximise de façon itérative la similarité entre l'apparence d'un objet et celle d'un candidat autour d'une position estimée. Initialement présentée dans [Fukunaga 1975], cette méthode consiste à estimer localement une densité de probabilité à l'aide d'un estimateur non paramétrique à noyau (fenêtre de Parzen). Une méthode de montée de gradient permet à l'algorithme *mean-shift* d'estimer de façon itérative les modes d'une distribution d'un ensemble de points définis dans R^d.

Appliqué au suivi d'objets, l'algorithme *mean-shift* est utilisé suivant deux approches [Collins 2003]. La première consiste à construire une image de vraisemblance dans laquelle la valeur des pixels est proportionnelle à la probabilité d'appartenance du pixel à l'objet à suivre. L'algorithme est appliqué sur l'image de vraisemblance afin de déterminer le maximum local par une méthode de montée de gradient. La seconde approche consiste à modéliser la distribution couleur (ou toute autre caractéristique) à l'aide d'un histogramme. Une mesure de similarité entre l'histogramme du modèle et les histogrammes des régions candidats (sélectionnés autour de la dernière position connue de l'objet à suivre) est effectuée, l'algorithme *mean-shift* est appliquée sur la surface résultante de la mesure de similarité et dans une fenêtre de recherche (voir Figure 1.8, 1.9).

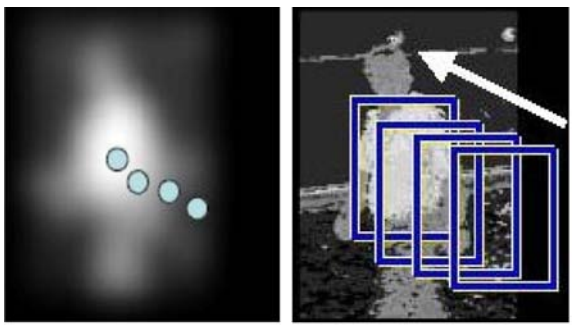


FIGURE 1.8: Illustration de l'algorithme *mean-shift* : une montée de gradient est appliquée sur la carte de vraisemblance (à gauche) afin d'estimer de façon itérative la nouvelle position de l'objet correspondant au maximum de la carte [Collins 2003].

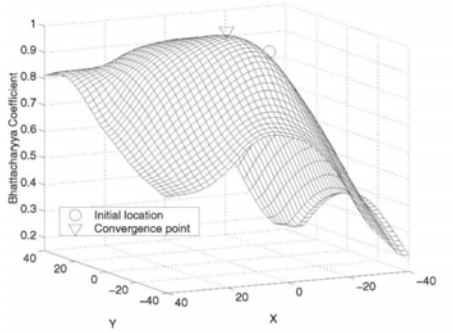


FIGURE 1.9: Exemple de surface obtenue par une mesure de similarité (distance de Battacharyya) entre les histogrammes couleurs du modèle et des candidats dans un voisinage proche de l'image suivante [Comaniciu 2002].

Le modèle d'apparence est représenté sous la forme d'un histogramme couleur et la mesure de similarité est définie à l'aide du coefficient de Battacharyya.

L'algorithme **Camshift** [Allen 2004] est une version étendue dans laquelle une étape de mise à jour des histogrammes permet à l'algorithme de s'adapter aux changements d'apparence des objets.

FIGURE 1.10: Exemple de suivi d'objet à l'aide de l'algorithme CamShift [Allen 2004].

Le ***Kanade Lucas Tracker*** est une méthode de suivi basée sur un ensemble de points caractéristiques invariants. Ces points sont détectés et suivis dans la séquence vidéo.

D'autres points d'intérêts ont été utilisés comme l'algorithme SIFT (*Scale invariant feature transform*) [Lowe 1999] ou SURF (*Speed up robust features*) [Bay 2008] pour détecter certaines caractéristiques locales dans les images.

Approches basées sur la forme géométrique

Les **contours actifs** ([Yilmaz 2004], [Torkan 2010]) permettent de prendre en compte la complexité des contours dans le suivi. Appelé également *snake*, un contour actif est une structure dynamique d'un ensemble de points mobiles qui évoluent itérativement dans l'image afin d'épouser au mieux la forme d'un objet d'intérêt. L'idée de cette méthode consiste à déplacer les points pour les rapprocher des zones à forts gradients, tout en

conservant certaines caractéristiques de forme sur le contour (disposition entre les points). La dynamique de déplacement des points est basée sur une notion d'énergie associée au contour.

Les contours actifs ont été introduits dans [Terzopoulos 1988] pour permettre la modélisation précise d'objets à l'aide de courbes décrites par un ensemble de vecteurs $v(s) = (x(s), y(s))^T$, avec s l'abscisse curviligne d'un point du contour actif telle que $0 < s < 1$. L'évolution du contour est régie par la minimisation de son énergie associée E_{totale}. Cette énergie se décompose en deux termes, une énergie interne E_{int} et une énergie externe E_{int}, telles que $E_{\text{totale}} = E_{\text{int}} + E_{\text{ext}}$. L'energie interne a pour objectif de donner une certaine régularité au contour en imposant des contraintes sur la forme (courbure par exemple) ou la régularité des points autour du contour. Elle ne dépend pas de l'image ni de la forme à segmenter mais uniquement des points du contour actif (courbure, espacement entre les points ou autres contraintes liées à la disposition des points). Elle est généralement décomposée en deux termes, une énergie interne de courbure E_{courb} et une énergie interne d'élasticité E_{elast}. Quant à l'énergie externe, elle fait appel aux données et tente de rapprocher les points du contour vers les zones à fort gradient d'intensité. Lorsque le contour épouse parfaitement la forme de l'objet, cette énergie est théoriquement minimale. L'énergie totale d'un contour actif s'écrit

$$E_{\text{total}} = \underbrace{\alpha E_{elast} + \beta E_{courb}}_{\text{Energie interne}} + \underbrace{\gamma E_{ext}}_{\text{Energie externe}} \tag{1.26}$$

avec α, β et γ les pondérations apportées aux énergies permettant de contrôler l'effet de chacune des composantes de l'énergie totale.

Certains auteurs ont proposé l'ajout d'une énergie supplémentaire, appelée énergie de contexte, permettant d'introduire des connaissances *a priori* sur ce qui est recherché [Cohen 1991].

FIGURE 1.11: Exemple de suivi de contours des objets issu des travaux de [Yilmaz 2004]. Les caractéristiques utilisées pour la minimisation d'énergie sont la couleur et la texture.

Approches fondées sur des modèles prédictifs

L'utilisation de modèles d'évolution des objets permet de prédire la position d'un objet dans l'image suivante. Cette étape de prédiction est une caractéristique importante pour les systèmes de suivi d'objets puisqu'elle permet d'aider à la mise en correspondance des objets et de maintenir une cohérence temporelle de la trajectoire grâce aux contraintes du modèle.

Le **modèle de mouvement dans sa version la plus simple** consiste à prédire la position de l'objet à l'instant suivant à partir de sa vitesse et sans prise en compte de l'accélération. Ce modèle s'exprime par :

$$\begin{cases} x_{t+1} = x_t + v_{x,t} \\ y_{t+1} = y_t + v_{y,t} \end{cases} \tag{1.27}$$

où (x_{t+1}, y_{t+1}) est la prédiction de la nouvelle position, (x_t, y_t) est la position actuelle et (v_x,v_y) les composantes du vecteur vitesse à l'instant t. La vitesse peut être estimée soit à l'aide de la position précédente (Equation 1.28), soit à l'aide d'une valeur plus ancienne de l'historique (Equation 1.29), ou encore à partir d'une estimation de sa moyenne (Equation 1.30)

$$\begin{cases} v_{x,t} = x_t - x_{t-1} \\ v_{y,t} = y_t - y_{t-1} \end{cases} \tag{1.28}$$

$$\begin{cases} v_{x,t} = \dfrac{x_t - x_{t-N}}{N} \\ v_{y,t} = \dfrac{y_t - y_{t-N}}{N} \end{cases} \tag{1.29}$$

$$\begin{cases} v_{x,t} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} x_i - x_{i-1} \\ v_{y,t} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} y_i - y_{i-1} \end{cases} \tag{1.30}$$

avec N la taille de l'historique utilisée. Le modèle réagit plus rapidement lorsque la valeur de l'historique est faible (ou lorsqu'aucun historique n'est utilisé), mais rend la prédiction sensible aux erreurs de location des objets (mauvaise segmentation par exemple) qui peut fournir des positions prédites peu fiables.

Le **filtrage de Kalman** [Kalman 1960] a été utilisé de façon intensive dans les algorithmes de suivi d'objets [Zou 2007], [Du 2009], [Alin 2011]. Il s'agit d'un filtre linéaire prédictif qui fournit une solution simple et efficace (dans des conditions particulières) pour estimer et prédire la position d'un objet. Ce filtre s'inscrit dans un cadre d'estimation récursive bayésienne dans laquelle l'estimation de la position est effectuée conditionnellement aux mesures et aux états précédents. L'algorithme de Kalman n'est équivalent à l'estimation bayésienne en terme d'optimisation que si les processus mis en jeu dans le modèle sont des processus markoviens et gaussiens, ce qui n'est plus le cas lorsque les systèmes sont non linéaires. Le filtre de Kalman nécessite la description du modèle d'évolution du

vecteur d'état à l'aide d'un modèle de mesure (observation) linéaire, auquel est ajouté un bruit blanc gaussien :

$$x_t = A_{t-1}X_{t-1} + w_t \tag{1.31}$$

$$z_t = H_t X_t + u_t \tag{1.32}$$

où A_{t-1} la matrice de transition du système (traduisant l'évolution du vecteur d'état) et H_t la matrice d'observation traduisant la relation entre l'observation z_t et le vecteur d'état x_t. Les bruits w_t et u_t sont supposés indépendants, gaussiens centrés en zéro et de matrice de covariance Q_{t-1} et R_t. Le filtre de Kalman comporte 3 étapes, une étape de prédiction, une étape d'innovation et une étape de correction (voir Figure 1.12). Une description plus complète du filtre de Kalman est donnée au chapitre 5. Lorsque le modèle d'évolution est non-linéaire, des techniques de linéarisation ont été proposées, comme le filtre de Kalman étendu (EKF) [Ribeiro 2004]. Ce filtrage consiste à linéariser les équations autour de la moyenne de l'état prédit à l'aide d'une approximation au premier ordre.

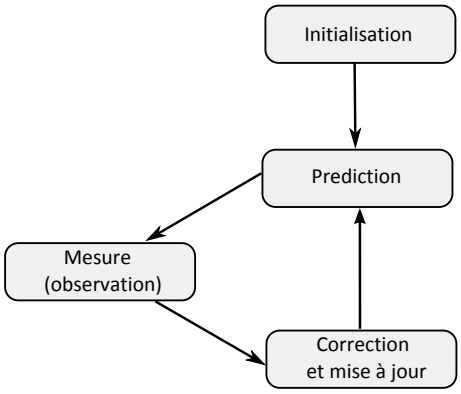


FIGURE 1.12: Les 3 étapes (hors initialisation) du filtrage de Kalman : Prédiction de l'état, Mesure de l'observation et Correction de la prédiction.

Le **filtrage particulaire** est une généralisation du filtrage de Kalman dans laquelle la distribution n'est plus contrainte à être gaussienne. Il s'agit d'une méthode de simulation séquentielle de type Monte Carlo, dans laquelle des échantillons pondérés appelés particules explorent l'espace d'état et interagissent sous l'effet d'un mécanisme de sélection qui concentre automatiquement les particules dans les régions d'intérêt de l'espace d'état [Legland 2003]. Les particules font office de description de la distribution et sont mises à jour régulièrement dans un schéma similaire au filtrage de Kalman à l'aide d'une étape de prédiction, d'une étape de mesure et d'une étape de correction de l'état.

1.2.4 Suivi multi-cible

Les sections précédentes présentent plusieurs méthodes permettant de détecter et suivre un objet dans une séquence vidéo. Dans un contexte de poursuite mono-cible, l'algorithme de suivi est vu comme une mise en correspondance d'un objet de l'image précédente avec un objet de l'image courante. Certaines méthodes présentées précédemment ne prennent pas en

compte la possibilité d'avoir plusieurs observations pouvant correspondre à la même cible, ni la possibilité de n'avoir aucune observation pour une cible. Ces situations sont pourtant fréquentes lorsque les objets entrent ou sortent de la scène. De plus, dans un contexte multi-objets, il est nécessaire de traiter les ambiguités d'association lorsque plusieurs objets sont proches ou lorsqu'ils sont occultés. De même une cible contenant plusieurs objets peut se diviser si les objets contenus dans la cible prennent des trajectoires différentes. Chercher à résoudre ces ambiguités revient à résoudre un problème d'association de données. Les techniques d'association sont généralement combinées à un filtrage prédictif selon le schéma de la Figure 1.13.

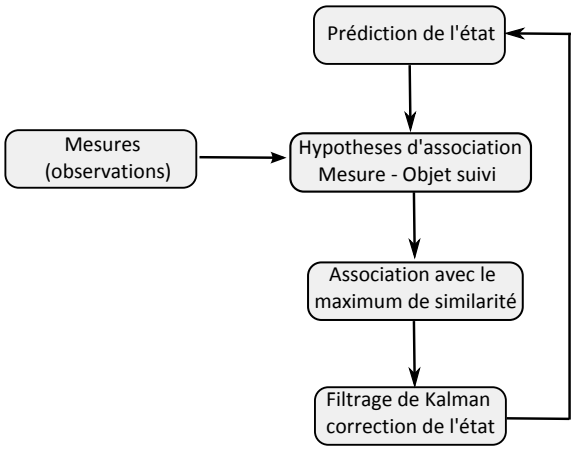


FIGURE 1.13: Suivi multi-cible à l'aide d'un filtrage prédictif et d'un module de génération d'hypothèses.

Association de données

Le problème d'association de données n'est pas un problème nouveau. Les techniques de résolution ont été initialement prévues pour résoudre un problème de poursuite de cibles issues de détecteurs tels que les radars ou les sonars. Dans cette section, nous décrivons quelques méthodes fréquemment utilisées pour résoudre ce problème. Pour une description plus détaillée des méthodes et algorithmes, se référer à [Pulford 2005].

La solution la plus intuitive consiste à associer chaque objet au candidat le plus proche en terme de similarité. Cette méthode, appelée méthode du plus proche voisin (*Nearest Neighbor*) considère que chaque détection a été générée par la cible la plus proche dans l'espace d'état. Les cibles dans l'espace d'état sont caractérisées par une zone de confiance caractérisant la probabilité que la cible ait généré l'observation. Si on considère les densités de probabilité gaussiennes, la similarité s'exprime à l'aide de la distance de Mahalanobis. Une fois la similarité estimée, seules les mesures statistiquement proches d'une cible sont utilisées pour mettre à jour son vecteur d'état. L'inconvénient de cette méthode est qu'elle ne prend pas en compte les ambiguités d'associations, et puisque qu'une détection ne peut être associée qu'à une seule cible, on risque de voir se propager les erreurs d'association.

Le *Probabilist Data Association Filter* (PDAF) est une méthode qui calcule les probabilités d'association pour chaque observation proche d'une cible. Sous l'hypothèse de distributions gaussiennes, ce filtre est similaire à un filtrage de Kalman.

Dans l'algorithme du plus proche voisin global (*Global nearest neighbor, GNN*) toutes les distances entre les mesures et les cibles sont prises en compte. La résolution de la mise en correspondance est résolue à l'aide d'un algorithme d'optimisation combinatoire Kuhn-Munkres [Kuhn 1955], appelé aussi algorithme hongrois (*hungarian algorithm*). Cet algorithme consiste à construire une matrice C de taille cxl appelée matrice de coût, dans laquelle le nombre de lignes l correspond au nombre de mesures et le nombre de colonnes c représente le nombre de cibles. Chaque élément $C(i, j)$ de la matrice représente un coût d'association de la cible i avec la mesure j. L'algorithme Kuhn-Munkres détermine l'association optimale pour chaque ligne (mesures) et chaque colonne (cibles) en réarrangeant les lignes et les colonnes telle que la somme des éléments de la diagonale principale soit minimale. Pour une description détaillée de l'algorithme, se référer à [Frank 2005].

L'algorithme MHT (*Multiple Hypothesis Tracking*) [Reid 1979] utilise l'historique complet des objets. Contrairement aux méthodes précédentes, celle-ci consiste à maximiser la probabilité des enchainements d'association de toutes les détections depuis leurs apparitions. La structure mise en oeuvre pour représenter les associations est un arbre. A chaque nouvelle détection, un nouvel étage de l'arbre est créé en prolongeant chaque feuille par l'ensemble des possibilités de correspondance. Les arcs sont pondérés par la probabilité que la nouvelle affectation soit vérifiée, compte tenu des hypothèses précédentes. Cet algorithme énumère de façon exhaustive l'ensemble des configurations possibles. Un exemple d'arbre des hypothèses est montré sur la Figure 1.14. Pour réduire la complexité exponentielle du MHT, les auteurs de [Cox 1996] proposent l'utilisation des k-meilleures hypothèses. Une version incrémentale de la propagation des probabilités est proposée.

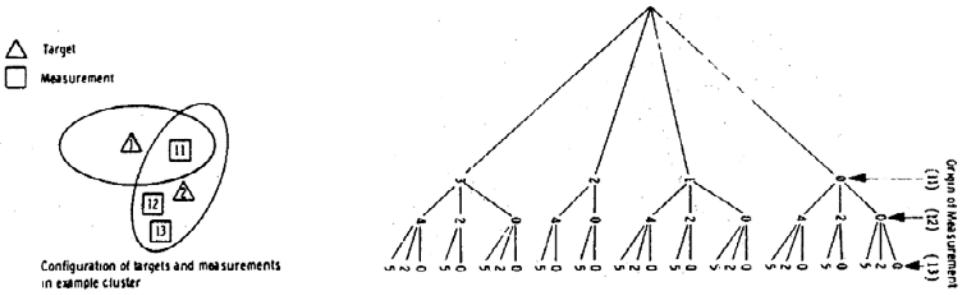


FIGURE 1.14: Illustration d'un arbre à hypothèses multiples pour le suivi de cibles [Reid 1979].

Traitement des occlusions et consistence temporelle

Généralement, on suppose que le déplacement des objets dans la scène est faible, dû à la rapidité de vitesse d'acquisition des caméras actuelles (format vidéo comprenant 25 images par seconde). Cette hypothèse permet de réduire la zone de recherche dans un voisinage immédiat de la dernière détection. La taille du voisinage dépend du nombre d'images par seconde et de la vitesse de l'objet.

Il s'agit, dans cette analyse, d'extraire les objets vidéos de la séquence d'image. Les objets vidéos sont directement représentatifs du contenu sémantique de la vidéo. L'extraction de ces objets est essentielle pour une analyse sémantique du contenu (analyse de

haut-niveau). Cette étape permet donc de préparer les données à fournir au module de haut-niveau sémantique. Puisque l'on travaille sur une séquence d'images, l'objet vidéo possèdera un caractère temporel représentatif de son évolution dans la scène.

1.3 Analyse de comportement

L'analyse de comportement est la dernière étape du traitement et fournit une interprétation sémantique de ce qui se déroule dans la scène, telle que l'identification d'une action ou la détection d'un évènement anormal [Candamo 2010]. Cette étape est généralement effectuée à partir des informations obtenues dans les étapes précédentes de l'analyse (détection de mouvement, suivi et classification d'objets). Dans les systèmes de vidéo-surveillance, la reconnaissance d'évènements depend généralement du contexte de la scène ; le même comportement peut avoir des significations différentes en fonction de l'environnement et du contexte d'analyse. En fonction des objectifs et de l'application, de nombreuses caractéristiques ont été proposées pour détecter des évènements [Lavee 2009], [Turaga 2008]. Parmi les plus utilisées, on retrouve les trajectoires des objets [Stauffer 2003], la forme et la silhouette des objets [Bissacco 2004], l'information de mouvement des objets [Adam 2008], ... Cette Section fournit un aperçu rapide des méthodes et techniques utilisées pour l'apprentissage et la détection d'évènements.

1.3.1 Représentation d'un évènement

Dans l'objectif d'établir une interprétation sémantique d'un évènement, il est nécessaire d'en choisir une représentation. Cette dernière définit l'extraction et la transformation des caractéristiques de bas-niveau en une représentation abstraite exploitable par le module de détection d'évènements. La sélection de la représentation dépends fortement du domaine d'application (surveillance maritime [Seibert 2006], aide à la personne agée [Karaman 2011], surveillance de magasins [Sicre 2010], [Trinh 2011], détection de bagages abandonnés [Kra], surveillance de foules [Andrade 2006b], [Benabbas 2011]). Les techniques de reconnaissance d'actions se sont focalisées initialement sur la détection d'actions élémentaires basée sur une analyse indépendante des objets de la scène et de leurs trajectoires. Le principal problème rencontré par ces approches est la forte sensibilité des résultats aux erreurs de trajectoires ; le système de reconnaissance depend fortement du processus de suivi et d'extraction des caractéristiques utiles à la reconnaissance de l'évènement.

Analyse d'actions élémentaires

Les actions élémentaires peuvent généralement être directement extraites de l'analyse des *blob*. Les caractéristiques des objets extraits telles que leur vitesse, leur direction, leur taille, ... peuvent être utilisées en les comparant par exemple à un ensemble de règles définissant l'anormalité d'un évènement. Par exemple dans [Ribeiro 2005], les auteurs utilisent l'information du flot optique des objets détectés pour détecter les évènements élémentaires suivants : *blob* actif/inactif, marche, course. De façon relativement similaire, dans [Nascimento 2005] les auteurs utilisent les caractéristiques mouvements des objets pour détecter l'entrée, la sortie, ou l'intérêt portée à une vitrine devant un magasin. Citons également les travaux de [Bodor 2003], qui portént sur la détection d'objets dans une zone

interdite prédéfinie. La position de l'objet est analysée afin de déterminer si la présence de l'objet est anormale. En ajoutant les informations de suivi d'objets, les auteurs proposent également la détection de chute d'une personne.

Analyse d'une séquence d'actions élémentaires

Certains évènements plus complexes, comportant des intéractions entre objets par exemple, ne peuvent pas être détecté par une analyse simple des trajectoires des objets. Généralement, les évènements complexes sont modélisés sous la forme d'un ensemble d'évènements élémentaires qui se déroulent séquentiellement. Des modèles statistiques plus sophistiquées telles que les réseaux de neurones, les modèles de Markov cachés ou les réseaux bayésiens, sont utilisés dans la littérature [Lou 2002] pour détecter ce type d'évènements. Par exemple dans [Ivanov 1999], les auteurs proposent un générateur d'évènement, qui traduit les informations issues du processus de suivi en évènements élémentaires, telles que *objet détecté*, *objet perdu*, *objet sorti de la scène*, ... Ces évènements élémentaires sont ensuites analysés à l'aide d'un *parser* d'évènements afin de construire des évènements plus complexes.

Analyse des trajectoires

La représentation d'une activité à l'aide des trajectoires des objets a été utilisée dans de nombreuses approches [Shah 1997]. Les trajectoires peuvent être utilisées pour apprendre celles qui sont usuelles (fréquentes). Toute trajectoire déviant des trajectoires usuelles sont ensuite considérées comme anormales. Par exemple, dans [Johnson 1996], les auteurs proposent la représentation des trajectoires des objets sous la forme d'une séquence de vecteurs mouvements. Un réseau de neurones est utilisé pour quantifier et apprendre les trajectoires typiques en différentes classes. Dans [Hu 2004b] les auteurs modélisent les vecteurs mouvement à l'aide de cartes auto-adaptives. Les trajectoires apprises peuvent ensuite être utilisées dans le cadre d'une détection d'évènements en identifiant les trajectoires qui s'écartent des modèles. Stauffer et Grimson [Stauffer 2000] proposent l'utilisation d'un algorithme de type *k-mean* pour classifier les trajectoires en fonction de l'information sur la taille des objets. Dans [Makris 2005], une utilisation plus précise des trajectoires est présentée, dans laquelle des régions sémantiques sont définies dans l'image (zones d'entrées, de sorties, d'arrêts, chemins, routes, ...) et appris à l'aide d'un mécanisme d'apprentissage.

Analyse du mouvement

D'autres méthodes tentent de détecter les évènements à travers l'analyse de caractéristiques dans sa globalité, autrement dit sans prendre en compte les objets individuellement. On retrouve leurs applications dans, par exemple, l'analyse de comportements dans une foule [Andrade 2006a]. Ces méthodes sont généralement basées sur la caractéristique mouvement, telle que les vecteurs mouvements issus d'une estimation de flot optique.Le champ de vecteur obtenu fournit une description concise des régions dans l'images contenant du mouvement ainsi que leur direction et leur amplitude.

Dans [Benabbas 2011], les auteurs estiment le champ de vecteurs mouvement à travers l'estimation du flot optique. Ce champ est utilisé pour modéliser une carte d'orientations et

de normes des déplacements. Après un filtrage spatial, tout vecteur mouvement s'écartant du modèle est considéré comme étant anormal.

Autres caractéristiques

Certains auteurs proposent l'utilisation de plusieurs caméra pour aider et rendre plus robuste la détection et le suivi d'objets. Par exemple, dans [Zelniker 2008] un système multi-caméra est utilisé afin de créer des trajectoires globales à travers l'ensemble des caméras et d'en détecter des comportements anormaux.

Citons également les travaux de [Wang 2009] dans lesquels les auteurs proposent une méthode non supervisée d'apprentissage basée sur un modèle hiérarchique et dans un contexte bayésien. Les activités élémentaires sont modélisées à l'aide de caractéristiques de bas-niveau, tandis que les intéractions entre évènements sont modélisées à l'aide de distributions basées sur ces activités élémentaires.

1.3.2 Surveillance du trafic routier

La détection d'évènements pour la surveillance du trafic routier a fait l'objet de nombreuses recherches [Kastrinaki 2003], [Buch 2011], [Tian 2011].

Dans [Pucher 2010], les auteurs combinent les informations issues du processus de détection et de suivi d'objets avec une information issue de capteurs sonores dans l'objectif de détecter les situations de contre-sens, d'arrêt et d'embouteillage. L'information sonore est utilisée pour augmenter la précision de la détection d'incidents.

Détection de véhicules à l'arrêt

Dans [Melli 2005], les auteurs utilisent les informations issues du suivi d'objets pour détecter l'arrêt dans des zones interdites. Le processus de suivi, basé sur un filtrage prédictif de Kalman, assigne un état aux objets en fonction de leur déplacemement dans l'image. Lorsqu'une occlusion est détectée, Un objet est considéré à l'arrêt s'il est détecté dans l'image courante, et qu'il a été en mouvemement pendant suffisamment longtemps dans les images précédentes.

Dans [Huang 2009], également, la direction du mouvement et la position des véhicules issues du processus de suivi d'objets sont utilisées pour reconnaître les évènements telles qu'un freinage brutal (arrêt), le changement de voie ou la détection de contre-sens. La détection de freinage est basée sur l'historique des cinq dernières vitesses estimées ; un véhicule est considéré comme étant en train de freiner si sa vitesse moyenne (sur l'historique) est inférieure à un seuil. La détection de contre-sens est basée sur la comparaison du mouvement du véhicule avec le déplacement moyen observé, et la détection de changement de voie s'appuie sur l'analyse des trajectoires des objets.

Dans [Porikli 2007], les auteurs proposent un système basé sur l'exploitation de deux images d'arrière-plan pour détecter les objets stationnaires. Les arrière-plans sont construits à fréquences différentes : une pour les évènements à court terme et l'autre pour la détection à plus long terme. Cette technique permet de s'affranchir de l'utilisation des trajectoires, souvent sujettes aux erreurs de détection et de suivis difficiles à corriger.

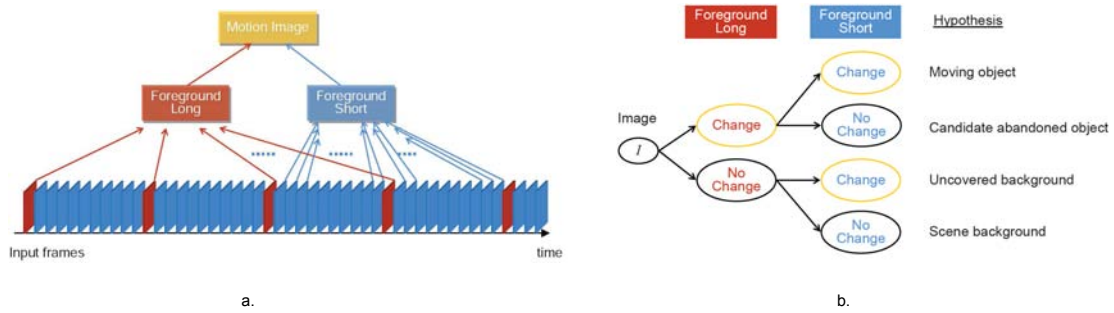


FIGURE 1.15: Illustration du processus de détection d'arrêt issu de [Porikli 2007]. (a) Procédure d'estimation de l'arrière-plan utilisant deux fréquences différentes. (b) Procédure de décision pour la détection d'objets statiques.

Dans [Albiol 2011], les auteurs utilisent l'estimation du déplacement de points d'intérêt (coins) afin d'identifier les véhicules garés ou à l'arrêt. En se basant sur cette détection, les coins statiques appartenant à l'arrière-plan sont éliminés à l'aide d'un filtrage temporel et ceux appartenant à de potentiels objets à l'arrêt sont utilisés pour construire une carte spatio-temporelle. Cette carte est ensuite analysée pour estimer les véhicules à l'arrêt.

Détection de véhicules en contre-sens

Dans [Huang 2009], la détection de contre-sens s'appuie sur le mouvement des véhicules estimé par le processus de suivi d'objets. La direction moyenne prise par les véhicules est calculée et utilisée comme référence afin de détecter tout véhicule dont la direction s'éloigne de plus de 90 degré.

Dans [Monteiro 2007], les auteurs utilisent l'information issue du flot optique afin de construire un modèle du sens de direction du trafic. Le champ de vecteurs pour chaque image est ensuite comparé au modèle afin d'en identifier les régions en contre-sens. Le modèle utilisé est un mélange de distributions gaussiennes. Pour valider la détection et supprimer les fausses alarmes, une procédure de classification des régions détectés est effectuée.

Dans [Luvison 2012], les auteurs proposent l'utilisation d'un descripteur spatio-temporel, appelé *Separated Selected Correlation* (SSC) permettant d'estimer un champ de vecteurs caractéristique du mouvement dans l'image. Le processus d'apprentissage proposé est non supervisé et consiste en une somme pondérée de fonctions noyaux. L'estimation de la vraisemblance d'une observation à ce modèle (appelé *Sequential Kernel Density Estimation* - SKDE par les auteurs) permet d'identifier les comportements anormaux des véhicules tels que des changements de voies ou des véhicules en contre-sens.

1.4 Conclusion

Nous avons vu dans ce chapitre les méthodes et techniques couramment utilisées pour les différentes étapes d'un système de vidéo-surveillance. La première Section a été consacrée à la détection d'objets à l'aide de méthodes telles que la différence temporelle, la soustraction d'arrière-plan ou l'estimation du flot optique. Appliquées dans un contexte

de vidéo-surveillance, ces méthodes sont généralement employées pour détecter les régions dans l'image susceptible de contenir des objets d'intérêt. Pour faire face à la complexité des environnements sous surveillance, telles que les changements de luminosité et les différentes conditions climatiques, les techniques de segmentation se sont orientées vers des modèles plus complexes qui s'adaptent et prennent en compte la multi-modalité de l'arrière-plan. En particulier, les modèles statistiques telles que les modèles de mélange de lois ont été utilisées de façon intensives ces dernières années.

La seconde section concerne la problématique de représentation des objets et des caractéristiques utilisées pour le suivi d'objets. Le processus de suivi consiste à conserver l'identité des objets détectés et de suivre leurs évolutions au cours du temps. Le suivi d'objets est par conséquent dépendant de la représentation des objets et de l'incertitude sur les observations. Parmi les approches existantes, on retrouve les méthodes basées sur l'apparence des objets, d'autres basées sur la forme géométrique (contours actifs) et d'autres basées sur un modèle prédictif (filtrage de Kalman ou à particules). Les modèles prédictifs ont été proposés pour prendre en compte les incertitudes sur les observations. Le principe de filtrage bayésien, en particulier le filtrage de Kalman et le filtrage à particules, ont été fortement utilisés pour suivre les objets. Enfin, le suivi de plusieurs cibles simultanément nécéssite l'utilisation d'un processus d'association afin de prendre en compte les éventuelles intéraction entre objets, telles que les occlusions.

Quant à la troisième section, elle a été consacrée aux méthodes de détection d'évènements dans des séquences vidéos. Certaines méthodes utilisent l'information issue du processus de suivi des objets afin d'analyser les trajectoires et d'en déduire les comportements anormaux. D'autres méthodes consistent à analyser le mouvement dans l'image dans sa globalité, sans tenir compte du comportement individuel des objets.

Chapitre 2

Présentation de l'approche

Sommaire

2.1 Introduction

Ce chapitre décrit l'architecture du système proposé pour l'interprétation de scènes autoroutières et l'analyse du trafic. Il s'agit de concevoir un système de vidéo-surveillance autonome capable d'identifier dans la scène sous surveillance des objets, des comportements ou des évènements spécifiques. Le système doit être capable de fournir en temps réel une information sur l'état du trafic (densités et statistiques de circulation) et d'enclencher une alarme lorsqu'un évènement particulier est détecté.

Dans une première section, nous décrivons les caractéristiques d'une scène autoroutière ainsi que la façon dont nous modélisons les données (scène sous surveillance et objets y évoluant). La Section 2.2 fournit une description de l'architecture générale du système. La Section 2.3 conclue ce chapitre en introduisant les métriques et la procédure d'évaluation des performances du système permettant de valider l'approche proposée.

2.1.1 Caractéristiques d'une scène autoroutière

Une scène autoroutière est un environnement dynamique dans lequel des objets (véhicules) se déplacent essentiellement sur une zone (route) réservée à la circulation des véhicules motorisés. Cette zone contient une ou plusieurs chaussées définissant le sens de circulation (à sens unique). Chaque chaussée est elle-même composée d'une ou plusieurs voies de circulation. Dans le cas d'une autoroute ou d'une voie rapide, elle peut également contenir sur le coté extérieur une bande d'arrêt d'urgence pour permettre aux usagers de s'arrêter (en cas d'urgence) sans gêner la circulation. Les chaussées sont le plus souvent séparées par un terre-plein central ou des glissières de sécurité permettant de limiter les chocs frontaux.

La composition du revêtement des routes (bitume pour 95% du réseau autoroutier français ou goudrons pour les voies anciennes) donne une couleur à la route généralement grise, peu texturée. Certains éléments de la scène, extérieurs à la route, peuvent altérer la détection des véhicules (Figure 2.1). Il s'agit par exemple d'arbres, de lampadaires ou de panneaux d'affichages, ... Ces éléments externes ont une influence directe sur l'analyse de la scène, puisqu'ils peuvent perturber les algorithmes de détection lorsqu'ils occultent partiellement la route, qu'ils sont en mouvement et qu'ils projettent des ombres dans la zone sous surveillance.

FIGURE 2.1: Exemples d'éléments extérieurs à la route et pouvant altérer la détection des objets.

L'environnement sous surveillance est une scène extérieure, soumise à des conditions

climatiques variables telles que la pluie, la neige, le brouillard, etc. Par conséquent, de nombreux changements de luminosité peuvent être provoqués et le simple déplacement du soleil dans le ciel ou le passage de nuages peuvent en être responsables. Plusieurs effets sont visibles sur l'image : par exemple la pluie fait apparaître des taches sur l'image et réduit l'intensité lumineuse. Quant au brouillard, il réduit le contraste ainsi que la visibilité dans l'image. En ce qui concerne le soleil, celui-ci peut aveugler le système lorsque l'angle de vue de la caméra est trop faible.

FIGURE 2.2: Exemples de changements de luminosité causés par le passage d'un nuage.

La position du soleil influe non seulement sur la direction mais également sur la forme des ombres portées des différents objets de la scène. Le passage des nuages contribue aussi, que ce soit localement ou globalement, aux changements de luminosité. De nuit, la scène peut être très obscure, par la non présence d'éclairage, ou au contraire moins sombre grâce à un éclairage public, artificiel. Dans ce dernier cas de figure, la lumière est diffuse et est *relativement* stable.

FIGURE 2.3: Exemples de conditions d'éclairage différentes.

Nous appellerons *objets d'intérêt* les véhicules circulant sur la route. Plusieurs types de véhicules coexistent et leur longueur varie entre 3m et 15m pour les camions ou les bus par exemple. Ils peuvent être de n'importe quelle couleur, néanmoins, les véhicules clairs sont statistiquement dominants. La majorité des véhicules présentent des parties plus sombres que la route, elles correspondent aux bas de caisse, aux pare-brises et aux roues. D'autres parties peuvent s'avérer plus claires (comme la carrosserie ou le reflet du soleil sur les pare-brises par exemple). De nuit et avec les feux de route des véhicules, de fortes réflexions au sol apparaissent. Mais les véhicules ont surtout la particularité d'être en mouvement (ou sont en tout cas supposés l'être hors bouchon) et leur vitesse est considérée comme constante. Ils peuvent changer de voie de circulation et possèdent une zone d'entrée et une zone de sortie de voie (ou tout du moins une limite de détection).

Influence de la position de la caméra

Lorsque la scène est projetée sur le plan image de la caméra, des déformations perspectives indésirables apparaissent :

- La forme des véhicules subit une transformation en fonction des paramètres intrinsèques et extrinsèques de la caméra. L'arrière et l'avant du véhicule ne forment pas un rectangle et sa hauteur est projetée soit sur le coté droit (vue de gauche), soit sur le coté gauche (vue de droite) soit vers l'avant dans le cas d'une surveillance dans une position centrale. En conséquence, la forme et la taille des véhicules ne sont pas constantes dans l'image et dépendent de leurs distances à la caméra.
- La vitesse apparente d'un véhicule n'est pas constante et dépend également de sa position à la caméra. Les véhicules ont une vitesse apparente plus élevée lorsqu'ils sont proches de la caméra et semblent se déplacer plus lentement lorsqu'ils s'en éloignent.
- Les véhicules peuvent être en partie masqués par d'autres circulant sur la route. Généralement, le phénomène d'occlusion apparaît avec les véhicules de grandes tailles (types camionnettes ou camions). Lors d'un trafic dense, l'arrière des véhicules peut être masqué.

D'autres phénomènes indésirables peuvent se produire au niveau de la caméra.

- Le vent peut provoquer une vibration de la caméra, toute la scène est alors déplacée dans le sens de la vibration.
- L'ajustement automatique de gain des caméras pour compenser les niveaux de gris lors des changements de luminosité entrainent des changements brutaux de luminosité qui interfèrent avec les algorithmes de détection de changement. Ce phénomène intervient généralement lorsqu'un véhicule de grande taille occupe une grande surface dans l'image.

2.1.2 Modélisation des données

Dans l'objectif de suivre l'évolution du comportement des véhicules dans la zone sous surveillance, une modélisation de la structure de la scène ainsi que des objets est définie. Les modèles sont appris à partir d'une séquence d'apprentissage et permettent de définir les caractéristiques et les informations qui seront utiles à l'analyse du trafic.

Structure spatiale de la scène

Le modèle de la structure de la scène se divise en différentes zones définies sur l'image (Figure 2.7). Chaque zone autorise ou au contraire interdit des comportements spécifiques à l'intérieur de la scène. Nous définissons dans un premier temps la région active, correspondant à la projection de la route sur l'image. Cette région regroupe l'ensemble des pixels sur lesquels les objets sont succeptibles d'évoluer. La région complémentaire est appelée région inactive et regroupe les pixels sur lesquels les objets ne peuvent pas apparaître. Les pixels appartenant à cette zone ne sont pas traités par le système, ce qui permet de réduire le temps de calcul. La région active comporte une ou plusieurs voies, qui sont elles-mêmes découpées en sous-zones en fonction de la profondeur dans l'image, comme illustré sur la Figure 2.4.

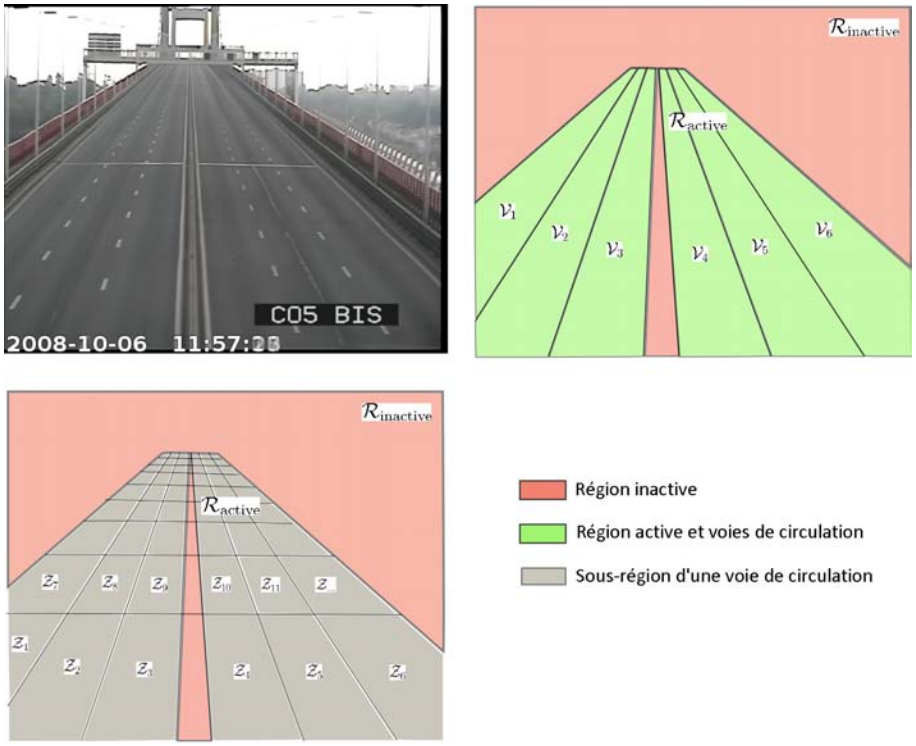


FIGURE 2.4: Illustration du modèle de la structure de la scène sous surveillance. L'image est découpée en sous-zones relatives aux voies de circulations auxquelles elles appartiennent et en fonction de la profondeur dans l'image.

Formellement, ce découpage s'exprime de la façon suivante. Soit I l'image de la scène et $\mathcal{R}_{\text{active}}$, $\mathcal{R}_{\text{inactive}}$ respectivement la région active et inactive de l'image, telle que $\mathcal{R}_{\text{inactive}}$ soit complémentaire de $\mathcal{R}_{\text{active}}$. Autrement dit, nous avons

$$I = \{\mathcal{R}_{\text{active}}, \mathcal{R}_{\text{inactive}}\}, \qquad \text{avec} \qquad \begin{array}{l} \mathcal{R}_{\text{active}} \cup \mathcal{R}_{\text{inactive}} = I \\ \mathcal{R}_{\text{active}} \cap \mathcal{R}_{\text{inactive}} = \varnothing \end{array} \tag{2.1}$$

La région active est découpée en différentes régions succeptibles de contenir les objets d'intérêt. Il peut s'agir d'une zone de circulation mais également d'une zone de stationnement (parking) ou d'arrêt (bande d'arrêt d'urgence), ceci dépendant du contexte applicatif. Dans le cadre de notre application, nous considérons le découpage de la région active en un ensemble de zones relatives aux voies de circulation que nous noterons $\mathcal{V}$ telles que

$$\bigcup_i \mathcal{V}_i = \mathcal{R}_{\text{active}} \qquad \forall i \in [1, N_v] \tag{2.2}$$

avec N_v le nombre total de voies. Finallement, les voies sont découpées en sous-zones notées $\mathcal{Z}^k$, avec k l'index de la voie correspondante, telles que

$$\bigcup_i \mathcal{Z}_i^k = \mathcal{V}_k \qquad \forall i \in [1, N_z^k] \tag{2.3}$$

avec N_z^k le nombre de sous-zones de la voie $\mathcal{V}_k$ concernée.

Sémantiques associées

Nous différençions quatre types de sous-zones différentes représentant les différents contextes sémantiques généralement rencontrés.

- Zone d'entrée ($\mathcal{Z}_{entree}$). Les zones d'entrée correspondent aux régions dans lesquelles les objets apparaissent dans la scène. Si un nouvel objet apparaît hors de la zone d'entrée, il y a de fortes chances qu'il s'agisse d'une erreur de détection (division d'un groupe d'objets par exemple).
- Zone de sortie ($\mathcal{Z}_{sortie}$). Les zones de sortie définissent la limite de détection des objets dans la scène.
- Zone de circulation ($\mathcal{Z}_{circulation}$). Les zones de circulations représentent les régions dans lesquelles les objets évoluent. Ces zones sont placées entre les zones d'entrée et les zones de sortie.
- Zone interdite ($\mathcal{Z}_{interdite}$). Il s'agit d'une zone dans laquelle un objet n'est pas autorisé à circuler (par exemple la bande d'arrêt d'urgence). Si cela se produit, une alarme est générée.

Notons que cette liste peut être complétée en fonction du type d'application et des besoins de l'utilisateur. Il est par exemple possible de rajouter une restriction sur le type de véhicule circulant sur une voie et de définir une zone interdite aux poids lourds. Une description complète du processus de modélisation de la scène utilisée dans notre système est détaillée au chapitre 3.

Modélisation des objets

La modélisation des objets permet de décrire de façon abstraite les données extraites de l'analyse. L'utilisation d'un modèle d'objet possède un double objectif : d'une part pour définir les informations qui seront extraites par le système et d'autre part pour fournir une représentation exploitable des données à une étape d'analyse automatique de comportement (par la formulation de requêtes par exemple). Dans cette section, nous définissons un modèle d'objet comme étant une description de la position, des paramètres internes et des relations avec l'extérieur d'un objet vidéo. Le choix des caractéristiques utilisées dépend du domaine d'application. Plus cette description est précise et plus la compréhension pourra l'être également (généralement au détriment d'une complexité calculatoire plus grande). Nous définissons les caractéristiques d'un objet $\mathcal{O}$, comme étant composées d'un terme ξ_{forme} regroupant les caractéristiques de formes, d'un terme $\xi_{\text{apparence}}$ regroupant les caractéristiques d'apparence et un terme $\xi_{\text{sémantique}}$ regroupant les caractéristiques sémantiques de l'objet.

$$\mathcal{O} = \{\xi_{\text{forme}},\ \xi_{\text{apparence}},\ \xi_{\text{sémantique}}\} \tag{2.4}$$

- **Caractéristiques de formes** - Les caractéristiques de formes fournissent une représentation 2D de l'objet et de sa forme telle qu'elle est perçue par la caméra. Il s'agit de décrire de façon structurelle la forme visuelle de l'objet, sa position, ses dimensions, ...
- **Caractéristiques d'apparences** - Les caractéristiques d'apparences ont pour objectif de caractériser l'apparence de l'objet généralement à l'aide de sa couleur, de sa texture ou toute autre caractéristique visuelle représentative de l'apparence.

– **Caractéristiques sémantiques** - Les descripteurs sémantiques contiennent une description sur la fonction et la sémantique de l'objet modélisé. Il peut s'agir par exemple de représenter le comportement d'un objet ou encore la classe d'objet (piéton, véhicule, poids lourd, ...).

Notons que ce modèle d'objet est un modèle dynamique évoluant au cours du temps. L'évolution des caractéristiques et le modèle sous-jacent n'est pas représenté ici mais est implicitement contenu dans les descripteurs. Par exemple, l'évolution de la position de l'objet (contenu dans le descripteur de forme) fournit une information sur la vitesse de l'objet qui peut notamment être utilisée pour décrire le type d'objet (contenu dans le descripteur sémantique). Plusieurs classes d'objets sont définies : piéton, cycliste, moto, véhicule léger et poids lourd. Les caractéristiques permettant de discriminer les classes sont principalement des descripteurs de formes, mais également un indicateur sur la vitesse de circulation pour, par exemple, différencier les piétons et les vélos des motos. Notons également que cette classification n'est pas stricte et que la limite entre les classes n'est pas forcément nette et évidente (entre une camionnette, un mini-van et un camion par exemple).

2.2 Architecture générale du système

Nous venons de présenter la modélisation utilisée pour représenter la scène sous surveillance et les objets y évoluant. Cette section présente l'analyse effectuée pour construire les modèles et les maintenir au cours du temps. Il s'agit d'extraire, à partir de l'analyse de la vidéo, un contenu sémantique de ce qui se déroule dans la scène. Notre démarche correspond à une utilisation croissante des connaissances en termes de richesse sémantique. Ainsi, les modèles des objets vidéos manipulés s'enrichissent au fur et à mesure que l'on avance dans l'analyse. Notre approche comporte quatre étapes (dont une d'initialisation) et peut être résumée sur la Figure 2.5.

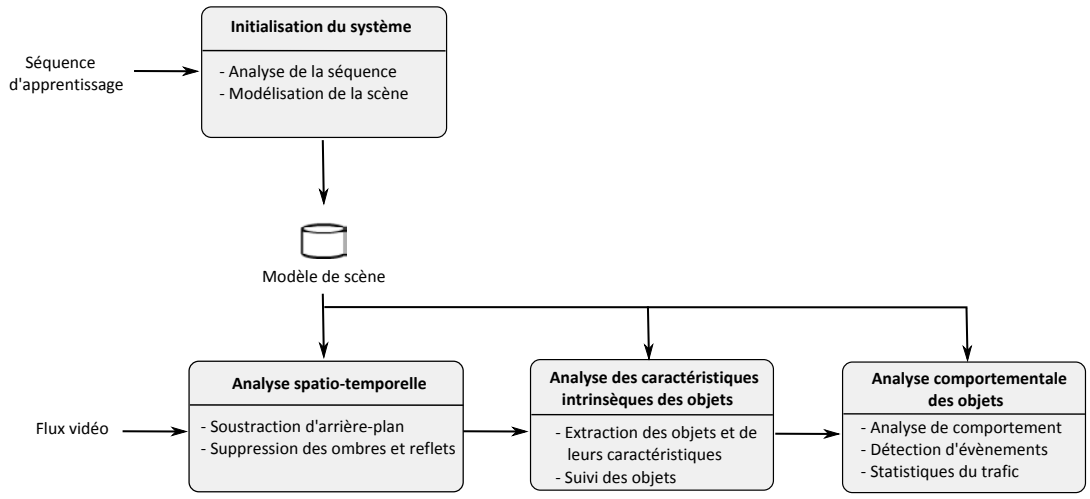


FIGURE 2.5: Architecture générale du système proposé

– **Initialisation du système**. L'objectif de cette étape consiste à construire un modèle

de la scène comportant des informations relatives à sa structure et aux comportements des objets. Il s'agit d'une étape d'apprentissage non supervisée durant laquelle une analyse spatio-temporelle de la séquence d'apprentissage permet d'extraire une estimation du sens de direction du trafic ainsi que les régions en mouvement succeptibles de contenir des objets d'intérêt. Ces objets sont mis en correspondance image après image dans l'objectif d'extraire les trajectoires typiques des véhicules. L'ensemble de ces informations est fusionnée pour construire le modèle de scène tel qu'il est présenté dans la Section 2.1.2.

- **Analyse spatio-temporelle**. Il s'agit d'une analyse de la vidéo à partir de caractéristiques bas-niveau, i.e. qui ne dépendent d'aucun modèle et ne contennant aucune information sémantique sur ce qui se déroule dans la scène. Les caractéristiques de bas-niveau sont extraites à partir des données issues de la caméra (valeurs des pixels de l'image) sans aucune information *a priori* sur la scène ou sur les objets et dont on ne fait aucune interprétation sur leur signification dans le contexte applicatif. Il s'agit par exemple d'une analyse du mouvement, de couleur ou de texture des données brutes de la vidéo. Ces primitives sont obtenues par une analyse spatiale (gradient), temporelle (évolution des valeurs) ou spatio-temporelle (flot optique) des valeurs des pixels. Cette analyse permet notamment d'estimer les pixels de l'image en mouvement succeptibles de contenir des objets d'intérêt.

- **Analyse des caractéristiques intrinsèques des objets**. L'analyse des caractéristiques intrinsèques des objets consiste à interpréter les caractéristiques de bas-niveau afin d'en extraire une information sémantique sur ce qui se déroule dans la scène. Cette interprétation suppose implicitement l'utilisation d'un modèle pour l'extraction des objets. Il peut s'agir, par exemple, de définir les contours des objets ou plus généralement ses caractéristiques de formes à partir d'une segmentation de bas-niveau basée sur le mouvement, la couleur et/ou la texture. Ainsi, les connaissances *a priori* sur la scène et les objets sont utilisées pour extraire les caractéristiques intrinsèques des objets telles que la forme, l'apparence et leurs sémantiques associées (type de véhicule, comportement, voie de circulation sur laquelle le véhicule circule, ...). Une fois extraits, les objets sont suivis dans le temps et leurs trajectoires sont estimées.

- **Analyse comportementale des objets**. La dernière étape consiste, à partir des informations issues des étapes précédentes, à identifier et à analyser le comportement des objets. Toute l'information *a priori* sur la scène, les objets et leurs comportements est intégrée afin de fournir une information sémantique (comportement normal ou anormal par exemple) sur ce qui se déroule dans la scène. Le plus souvent, cette analyse dépend fortement du contexte applicatif.

Dans le cadre de l'approche proposée, chacune des analyses possède plusieurs étapes de traitement qui seront développées dans les chapitres suivants. L'information sur la structure de la scène est intégrée au système, et chaque niveau sémantique à accès à cette information. La construction du modèle de la scène est effectuée lors de l'initialisation du système et fait l'objet de la section suivante.

2.2.1 Initialisation du système

Durant l'étape d'initialisation, un modèle de la scène sous surveillance est construit dans l'objectif de fournir aux algorithmes de traitement une information sur le contexte et la structure de la scène. L'apport de cette information permet de fournir une aide à la segmentation et au suivi d'objets. La description complète de cette étape est présentée au Chapitre 3. La construction du modèle est basée sur l'analyse d'une séquence d'apprentissage afin d'en extraire des caractéristiques de bas-niveau (couleur, mouvement) et des caractéristiques de niveau intermédiaire (trajectoire et sens de direction du trafic). Les caractéristiques recueillies sont ensuite utilisées pour la construction du modèle de la scène. Le schéma complet de l'approche est présenté sur la Figure 2.6.

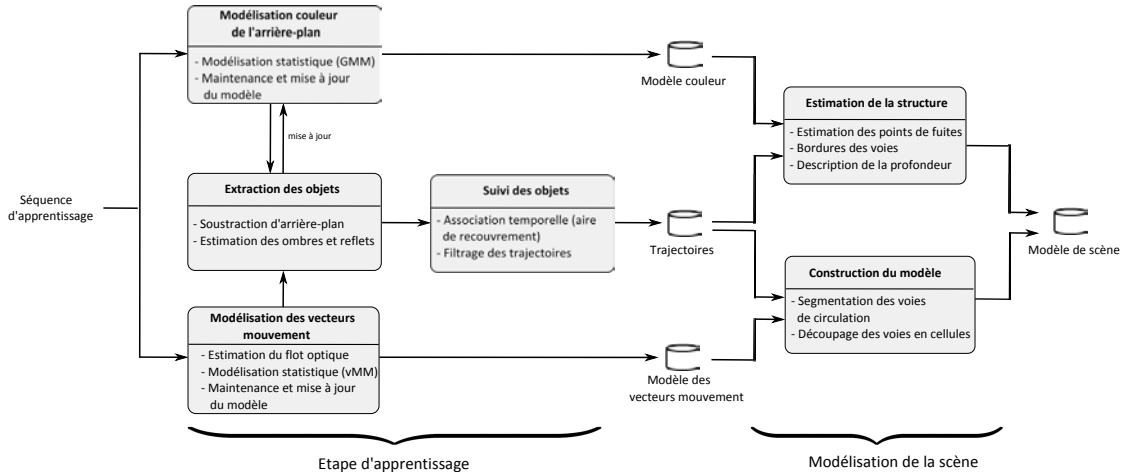


FIGURE 2.6: Présentation de l'approche utilisée pour la modélisation de la scène. Deux étapes sont nécessaires : une étape d'analyse de la séquence d'apprentissage et une étape de construction du modèle.

Étape d'apprentissage

La première étape consiste en une analyse de la séquence d'apprentissage afin d'en extraire une modélisation de la couleur, du mouvement estimé par flot optique ainsi que des trajectoires typiques des véhicules détectés. L'extraction de l'ensemble de ces informations nécessite des traitements de bas-niveau (analyse spatio-temporelle) ainsi que des traitements de plus haut niveau sémantique (extraction des caractéristiques intrinsèques des objets) :

- **Modèle couleur de l'arrière-plan**. Un modèle statistique couleur est utilisé pour représenter l'arrière-plan. Pour cette étape d'apprentissage, nous utilisons un modèle gaussien qui sera ensuite utilisé pour initialiser le modèle plus complexe présenté au Chapitre 4.

- **Extraction de *blobs***. Cette étape consiste à effectuer une comparaison de la nouvelle image avec le modèle (soustraction d'arrière-plan) afin d'en extraire les pixels couleurs s'écartant de la modélisation. Avant d'être regroupés par analyse en composantes connexes, une étape de soustraction d'ombre est effectuée sur l'ensemble des

pixels du masque issu de la soustraction d'arrière-plan.

- **Modèle de mouvement**. Un modèle statistique du sens de direction du trafic routier est construit à l'aide des vecteurs mouvement issus de l'estimation dense du flot optique dans l'image.

- **Suivi de *blobs***. Une mise en association des *blobs* détectés dans l'image courante avec ceux des images précédentes permet la construction des trajectoires des objets évoluant dans la scène. L'objectif ici consiste à estimer les trajectoires typiques (ie. les plus rencontrées) des véhicules dans la scène. Le processus de suivi ne comporte aucune gestion des ambiguités d'association (fusion ou division d'objets) et seules les trajectoires suffisamment longues et n'ayant subit aucune ambiguité sont conservées.

Construction du modèle de scène

Une fois l'analyse de la séquence d'apprentissage terminée, les informations extraites sont utilisées pour estimer les caractéristiques spatiales de la scène dans l'image. L'estimation des bordures des voies, du point de fuite et de la profondeur dans la scène permettent de construire un partitionnement de la scène en cellules qui, en théorie, couvrent la même surface dans le monde réel. À chaque cellule est associé un ensemble de caractéristiques sémantiques permettant l'analyse de comportement des objets.

- **Estimation de la structure**. La structure de la scène est extraite à l'aide du modèle couleur d'arrière-plan et de l'estimation des trajectoires des objets. Les bordures des voies, le point de fuite et les lignes de profondeur dans l'image sont estimés et caractérisent la structure spatiale de la scène.
- **Modèle de la scène**. Le modèle de la scène est construit en partitionnant spatialement l'image à l'aide des bordures des voies et des lignes de profondeur. Les cellules issues de ce partitionnement contiennent des caractéristiques sémantiques telles que le sens de direction du trafic ou le type de cellule décrit dans la Section 2.1.2 (zone d'entrée, de circulation, de sortie, ...).

À la fin de cette étape d'apprentissage, la structure de la scène est caractérisée par :

- Une zone d'intérêt caractéristique des régions succeptibles d'être en mouvement.
- La délimitation des voies de circulations.
- Une description de la profondeur dans l'image.

Cette structure est ensuite enrichie par des caractéristiques sémantiques déduites de l'analyse de la séquence d'apprentissage : le sens de direction du trafic, les points de fuite, les zones d'entrée, de circulation et de sortie, les statistiques sur la taille des objets. L'ensemble de ces informations est stocké dans le modèle selon la procédure décrite dans la Section 2.1.2. Un exemple de modèle de scène obtenu est illustré sur la Figure 2.7 dans laquelle le sens de direction est affiché à l'aide de la palette de couleur en haut à droite de l'image.

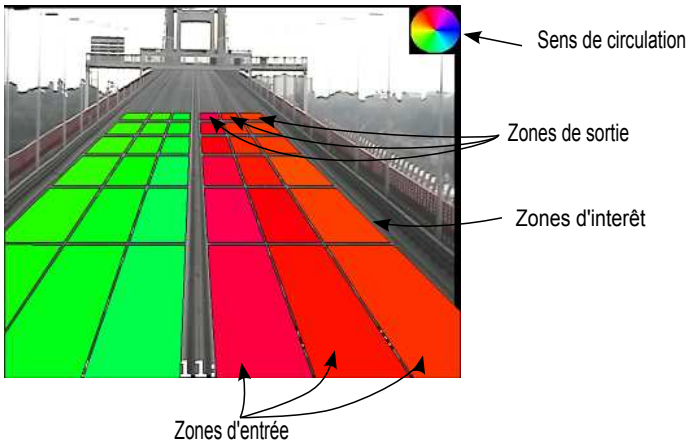


FIGURE 2.7: Exemple de modèle de scène comportant l'information sur le sens de circulation des voies. La palette de couleur utilisée pour représenter l'orientation du sens du trafic est affichée en haut à droite de l'image.

2.2.2 Analyse spatio-temporelle

L'analyse spatio-temporelle s'appuie sur l'extraction de caractéristiques non sémantiques dans l'objectif d'identifier les régions dans l'image contenant des objets en mouvement. L'extraction consiste à recueillir un jeu de caractéristiques (couleurs, gradients et vecteurs mouvement issus du flot optique) à partir de l'image courante de la vidéo. L'information couleur est utilisée pour détecter tout changement d'intensité lumineuse par rapport à celle de l'arrière-plan, tandis que les caractéristiques de gradient et de flot optique permettent de valider la présence potentielle d'un objet dans une région dont la couleur s'écarte du modèle. L'arrière-plan est représenté sous la forme d'un modèle statistique régulièrement mis à jour pour prendre en compte les perturbations extérieures (passage de nuages par exemple). La vitesse d'adaptation de l'arrière-plan lorsqu'un changement de luminosité apparaît entraine généralement des fausses détections si la couleur est utilisée seule pour la segmentation. L'ajout d'une information supplémentaire issue du gradient spatial et de l'estimation du flot optique dans l'image, permet de valider ou non la présence d'un objet. Le processus complet est illustré sur la Figure 2.8 et fera l'objet du Chapitre 4.

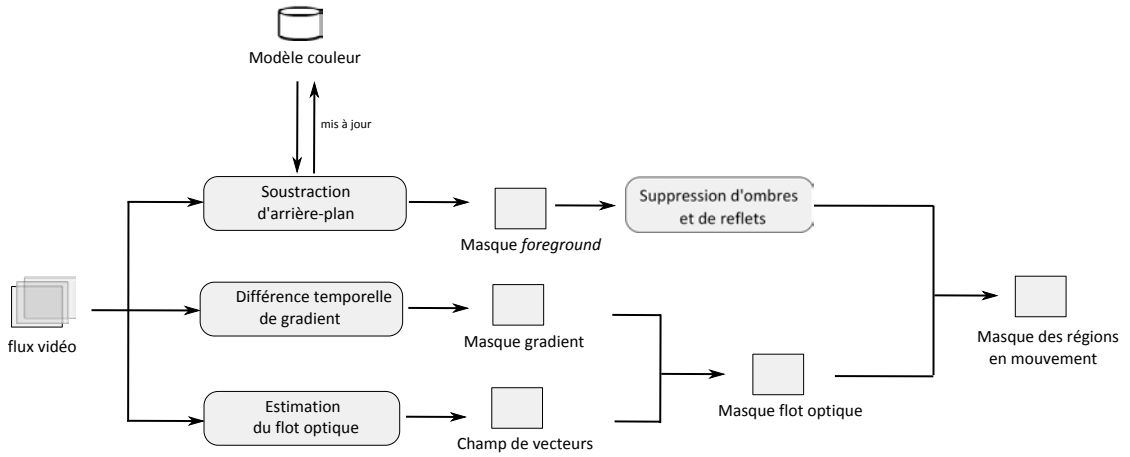


FIGURE 2.8: Analyse spatio-temporelle : estimation des régions susceptibles de contenir des objets en mouvement à partir de caractéristiques de bas-niveau (couleur, gradient et vecteurs mouvement).

2.2.3 Analyse des caractéristiques intrinsèques des objets

Une fois l'analyse spatio-temporelle effectuée, l'étape suivante consiste à extraire les caractéristiques intrinsèques des objets. L'objectif de cette étape est la construction d'objets à contenu sémantique de plus haut niveau. Chaque objet vidéo possède les informations de bas-niveau (couleurs, gradient et mouvement) mais également les informations temporelles issues du suivi d'objet et permettant d'analyser l'évolution des objets dans la scène. Deux étapes sont nécessaires à leur construction : une étape d'extraction d'objets, durant laquelle les masques obtenus dans l'analyse bas-niveau sont exploités et combinés pour la segmentation, et une étape de suivi d'objets permettant de maintenir l'identité des objets évoluant dans la scène. L'extraction des objets s'appuie sur une modélisation implicite des véhicules. Une région en mouvement issue de l'analyse spatio-temporelle est considérée comme étant un objet d'intérêt si sa taille est comprise entre une taille minimum et une taille maximum autorisée. L'évolution temporelle de la position, de la forme et d'apparence permet d'enrichir le contenu sémantique des objets vidéo, qui sera analysé dans la dernière étape du traitement. Le processus complet est illustré sur la Figure 2.9 et fera l'objet du Chapitre 5.

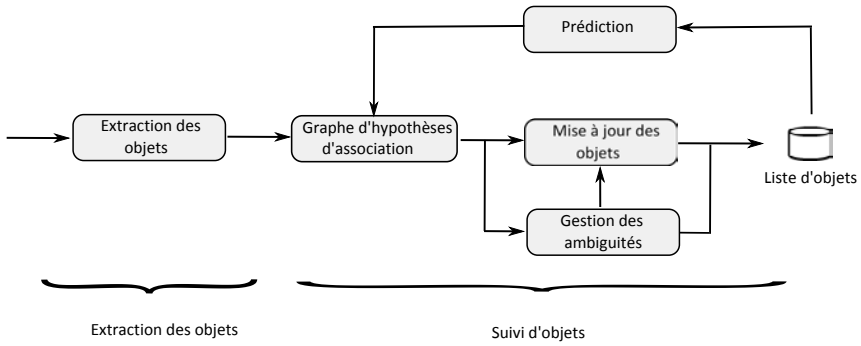


FIGURE 2.9: Analyse des caractéristiques intrinsèques des objets : les étapes d'extraction et de suivi d'objets permettent de maintenir l'identité des objets présents dans la scène. La structure de la scène est utilisée pour aider à la segmentation et au suivi des objets.

2.2.4 Analyse comportementale des objets

L'extraction du contenu sémantique est la dernière étape du traitement dont l'objectif est d'analyser et de détecter des comportements spécifiques dans la vidéo. Il s'agit de fournir une information sémantique sur ce qui se déroule dans la scène et de, par exemple, déclencher une alerte lorsqu'un événement particulier se produit. On considère plusieurs types d'évènements à détecter, contenant des informations sémantiques plus ou moins complexes :

- **Détection d'objets ou d'intrusions**. La présence d'un objet dans une zone spécifique peut-être considérée comme un évènement, particulièrement lorsqu'il s'agit d'une zone sous restriction (bande d'arrêt d'urgence par exemple). Une alarme est générée lorsqu'un évènement de ce type est détecté.

- **Détection d'objets à l'arrêt**. Dans une scène autoroutière, aucun véhicule n'est autorisé à s'arrêter. La détection d'objet à l'arrêt permet de générer une alarme lorsque ce cas arrive. Que l'objet soit sur une bande d'arrêt d'urgence ou sur une voie de circulation, si un véhicule à l'arrêt est détecté, il s'agit d'un incident.

- **Détection d'objets en contre-sens**. Les voies de circulation possèdent un sens de direction du trafic, tout objet en contre-sens est considéré comme un incident (même si généralement, ce cas particulier aboutit rapidement à un accident qui génère une détection d'arrêt).

- **Détection de changements de voies**. Le changement de voies des véhicules n'est pas un incident à proprement parlé, mais est considéré comme un évènement. Cet évènement peut-être couplé à la détection d'arrêt par exemple, pour valider la présence d'un véhicule arrêté (si tous les véhicules appartenant à la voie sur laquelle un véhicule est arrêté changent de voies par exemple).

- **Détection de bouchons**. La détection de bouchons est une fonctionnalité importante pour caractériser le trafic routier. Elle permet d'améliorer la logistique et l'aménagement du réseau routier afin de limiter les bouchons et un trafic dense. La détection de formation d'un bouchon n'est pas un incident à proprement parlé, mais constitue un évènement important à reconnaître. Notons que la formation d'un bouchon peut-être provoqué par la présence d'un incident sur une des voies de circulation.

2.3 Évaluation des performances

Ces dix dernières années, de nombreux systèmes de vidéo-surveillance ont été proposés dans la littérature. Chaque nouveau système proposé vise à augmenter la robustesse et la performance des résultats face aux nombreuses difficultés rencontrées en vision par ordinateur. L'évaluation des résultats devient impératif, particulièrement lorsque les algorithmes font face à des problèmes encore non résolus. D'un point de vue expérimental, la résolution d'un problème nécessite la validation par un protocole d'évaluation bien défini, permettant ainsi la comparaison avec d'autres algorithmes afin d'en identifier les faiblesses et les points forts.

La mise en place d'un protocole d'évaluation est une étape difficile due à la complexité du système lui-même, qui est composé de nombreux modules (détection, extraction d'objets, suivi temporel, ...). De nombreux efforts de recherche ont été faits dans l'évaluation des performances des systèmes de vidéo surveillance (PETS, CAVIAR, ETISEO). Ces travaux proposent des métriques caractérisant des aspects particuliers d'une tâche demandée. L'utilisation d'une seule mesure ne suffit pas à caractériser l'ensemble d'un système et un trop grand nombre de métriques est difficilement exploitable puisque cela demande une expertise supplémentaire des résultats obtenus. Chacune de ces méthodes évalue les performances grâce à un certain nombre de mesures fondées sur la comparaison des résultats de l'algorithme avec une vérité-terrain.

La génération de la vérité-terrain est une étape indispensable qui nécessite la définition de certaines règles d'annotation. Des hypothèses sur les observations doivent être établies, par exemple, le temps autorisé à un véhicule à l'arrêt avant de considérer qu'une alarme doit être générée. Dans l'exemple d'un comptage de véhicule, il est nécessaire de bien définir la zone d'intérêt, et de définir à partir de quand un objet est considéré comme présent ou non (partiellement présent).

Appliquer l'algorithme sur des séquences différentes donnera des performances différentes, il est donc nécessaire de bien choisir le jeu de vidéos de test en fonction de la tache à analyser. La question de la représentativité des vidéos choisies face aux problèmes réels peut être posée, puisque généralement il s'agit de courtes séquences.

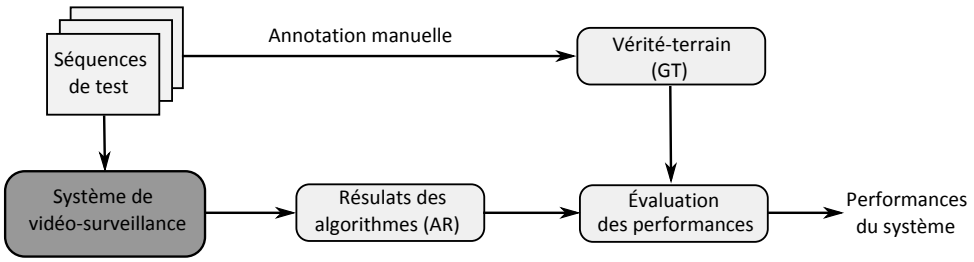


FIGURE 2.10: Vue d'ensemble d'un processus d'évaluation des performances d'un système de vidéo-surveillance.

2.3.1 Notations et définitions standards

Les définitions usuelles suivantes sont utilisées dans le calcul de nombreuses métriques qui seront développées dans la Section 2.3.3.

True Positive (TP)	Le système a détecté une situation réelle. La situation existe aussi bien dans le résultat de l'algorithme que dans la vérité terrain.
True Negative (TN)	La situation n'existe ni dans l'algorithme, ni dans la vérité-terrain.
False Negative (FN)	Une situation réelle a été ratée par l'algorithme. La situation n'existe pas dans le résultat de l'algorithme tandis qu'elle existe dans la vérité-terrain.
False Positive (FP)	Le système a détecté une situation qui n'est pas réelle. Cette situation existe dans le résultat de l'algorithme mais n'existe pas dans la vérité-terrain.

TABLE 2.1: Définitions standards : True Positive, True Negative, False Positive, False Negative.

Ces mesures permettent de caractériser un algorithme vis à vis du nombre de réussites et d'échecs (TP, TN, FP, FN) face à un problème donné. Les définitions précédentes permettent d'estimer quatre scores de performances, définis par :

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ Specificity &= \frac{TN}{FP+TN} \\ F-score &= \frac{2.Precision.Recall}{Precision+Recall} \end{aligned} \tag{2.5}$$

L'utilisation des mesures définies par les équations 2.5 permettent d'obtenir des caractéristiques ramenées à l'ensemble des résultats obtenus.

2.3.2 Comparaison des résultats avec la vérité-terrain

Pour déterminer le nombre de succès et d'échecs d'un algorithme, il est nécessaire de définir une ou plusieurs métriques de comparaison entre les résultats de l'algorithme (AR) et la vérité-terrain (GT). Les distances utilisées doivent prendre en compte à la fois l'aspect spatial et l'aspect temporel des observations fournies par l'algorithme. L'association est déterminée à l'aide du recouvrement spatial et temporel entre l'observation et la vérité-terrain (Figure 2.11).

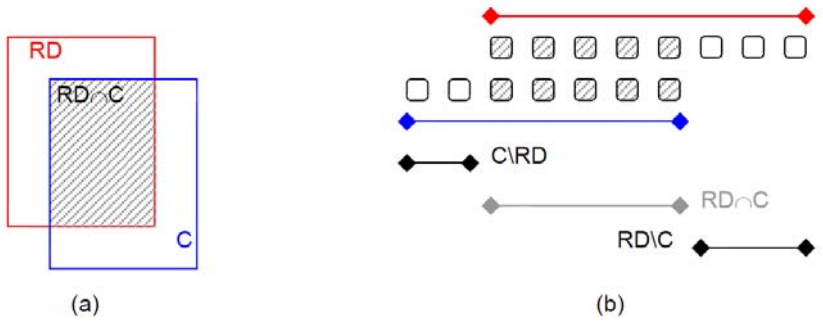


FIGURE 2.11: Mise en correspondance d'un résultat de l'algorithme (AR) avec la vérité-terrain (GT) en utilisant un critère de recouvrement spatial et temporel.

Dans ETISEO [Nghiem 2007], quatre mesures de distances sont définies (notées E1-E4 [1]) (Figure 2.11), et peuvent être utilisées pour évaluer aussi bien le recouvrement spatial que le recouvrement temporel entre l'observation et la vérité terrain.

$$\begin{aligned} E1 &= \frac{2.Card(GT \cap AR)}{Card(GT) + Card(AR)} \\ E2 &= \frac{Card(GT \cap AR)}{Card(GT)} \\ E3 &= \frac{Card(GT \cap AR)^2}{Card(GT).Card(AR)} \\ E4 &= \max\left(\frac{Card(AR \cap GT)}{Card(AR)}, \frac{Card(GT \cap AR)}{Card(GT)}\right) \end{aligned} \tag{2.6}$$

où $Card(E)$ représente le cardinal (nombre d'éléments) de l'ensemble E. Ces mesures fournissent un score qui peut éventuellement être seuillé pour obtenir une décision binaire de mise en correspondance. Pour l'évaluation de notre système, nous utiliserons la mesure E2 (normalisée par la vérité-terrain GT).

2.3.3 Métriques d'évaluation

Le système proposé est évalué en utilisant un ensemble d'outils d'évaluation basé sur les métriques définies dans ETISEO [Nghiem 2007]. Il s'agit d'un projet de mise en oeuvre d'une plateforme d'évaluation de performance pour les systèmes de vidéo surveillance. Nous nous intéressons à l'évaluation des performances de la segmentation de mouvement, de la détection d'objets, du suivi d'objets et de la détection d'évènements.

Segmentation de mouvement

Les résultats de l'algorithme de segmentation de mouvement sont évalués en termes de TPR (*True Positive Rate*) de FAR (*False Alarm Rate*) et de F-Score. Ces valeurs

1. http://www-sop.inria.fr/orion/ETISEO/iso_album/eti-metrics_definition-v2.pdf

sont déterminées en comptabilisant le nombre de pixels correctement détecté (#TP *True Positive*), le nombre de pixels incorrectement détectés (#FP, *False Positive*) et le nombre de pixels de la vérité terrain qui n'ont pas été detectés (#FN, *False Negative*).

Le TPR correspond à une mesure de Rappel (voir Section 2.3.1) et reflète le taux de pixel correctement détectés (#TP) parmi l'ensemble des pixels de la vérité-terrain (#TP+#FN). Sa valeur doit être aussi élevée que possible et est donnée par :

$$TPR = \underbrace{\frac{\#TP}{\#TP + \#FN}}_{\text{Rappel}} \tag{2.7}$$

La mesure du FAR (*False Alarm Rate*) consiste à comptabiliser le nombre de pixels incorrectement détectés (#FP) parmi l'ensemble des pixels détectés par l'algorithme (#TP+#FP). Cette valeur reflète le nombre de faux positifs (FP) détecté par l'algorithme et est donnée par :

$$FAR = \frac{\#FP}{\#TP + \#FP} = 1 - \underbrace{\frac{\#TP}{\#TP + \#FP}}_{\text{Precision}} \tag{2.8}$$

Quant à la mesure de F-Score, elle permet de fournir une mesure scalaire obtenue en combinant les valeurs de TPR et de FAR. Cette valeur doit être aussi élevée que possible pour refléter un bon taux de TPR et un faible FAR. La mesure de F-score est donnée par :

$$F - Score = \frac{2.(1 - FAR).TPR}{(1 - FAR) + TPR} = \frac{2.\text{Precision.Rappel}}{\text{Precision} + \text{Rappel}} \tag{2.9}$$

Finalement, la métrique utilisée pour l'évaluation de la segmentation des pixels en mouvement s'écrit

$$\text{Det}_{pixel} = \begin{cases} \text{TPR} & = \dfrac{\#TP}{\#TP + \#FN} \\ \text{FAR} & = \dfrac{\#FP}{\#TP + \#FP} \\ \text{F-score} & = \dfrac{2.(1 - FAR).TPR}{(1 - FAR) + TPR} \end{cases} \tag{2.10}$$

Détection des objets

L'évaluation des performances de la détection d'objets est une analyse qui suit logiquement l'analyse de la segmentation des pixels. Il s'agit de quantifier les performances de la détection de l'algorithme à l'échelle des objets. Pour déterminer si un objet est correctement détecté, une association avec la vérité-terrain est effectuée à l'aide de la mesure E2 présentée dans la Section 2.3.2. Si l'aire de recouvrement entre l'objet détecté par l'algorithme et l'objet de la vérité-terrain est supérieure à un seuil, alors il est considéré comme correctement détecté. Notons qu'un seul objet de l'algorithme ne peut être associé à un objet de la vérité-terrain. Si deux objets correspondent, alors celui ayant l'aire de recouvrement la plus grande lui est associé tandis que les autres objets sont considérés comme étant des faux positifs. Cette mise en association des objets de l'algorithme avec la vérité-terrain

permet de définir les mêmes métriques de détection d'objet que dans la section précédente (à l'échelle du pixel) :

$$\text{Det}_{obj} = \begin{cases} \text{TPR} & = \dfrac{\#TP}{\#TP + \#FN} \\ \text{FAR} & = \dfrac{\#FP}{\#TP + \#FP} \\ \text{F-score} & = \dfrac{2.(1 - FAR).TPR}{(1 - FAR) + TPR} \end{cases} \tag{2.11}$$

Suivi des objets

Les métriques d'évaluation pour le suivi d'objets sont définies en termes de Précision, de Rappel et de F-Score. Le nombre d'objets correctement suivis est déterminer en analysant les trajectoires issues de l'algorithme avec celles de la vérité-terrain. La mise en association s'effectue à l'aide de la mesure E2 définie dans la Section 2.3.2 : si un objet de l'algorithme est associé au même objet de la vérité-terrain pendant la majorité de sa durée de vie, alors il est considéré comme correctement suivi. Si deux objets sont associés au même objet de la vérité-terrain, alors celui ayant le plus grand recouvrement temporel lui est associé et le second objet est considéré comme étant un faux positif (FP). Cette mise en association permet de définir un jeu de métrique pour l'évaluation des performances du suivi d'objets donné par :

$$\text{Sui}_{obj} = \begin{cases} \text{Precision} = & \dfrac{TP}{TP + FP} \\ \text{Recall} = & \dfrac{TP}{TP + FN} \\ \text{F-score} = & \dfrac{2.\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \end{cases} \tag{2.12}$$

Détection d'événements

L'évaluation de la détection d'événements est effectué de façon qualitative et quantitative en comptabilisant le nombre d'événements correctement détectés. Nous nous intéressons aux cas de détection de véhicule à l'arrêt, de détection de véhicules en contre-sens, et de détection de changements de voies. Lorsque le nombre de détections est comptabilisé, les résultats sont évalués en termes de Précision et de Rappel. Pour l'évaluation de la détection d'arrêt et de contre-sens, l'évaluation est donnée qualitativement à l'aide d'exemples d'images détectées contenant un évenement particulier.

2.3.4 Corpus de test

L'évaluation des performances est effectuée sur un ensemble de vidéos issues d'acquisition de scènes autoroutières dans des conditions réelles. Parmi ces vidéos, des séquences ont été sélectionnées pour l'évaluation des performances représentant certaines conditions particulières. Les configurations utilisées pour l'évaluation des performances sont décrites dans les chapitres consacrés à la détection d'objets, au suivi d'objets et à l'analyse de

	C5	C21	Lyon	C15	CE11	Highway	I-LIDS	C10
Segmentation mouvement (échelle pixel)	√	√	√	√				
Détection d'objets (échelle objet)	√	√	√	√				
Suivi d'objets	√		√		√	√		
Comptage d'objets	√	√	√	√	√	√		
Détection de changements de voies	√		√		√			
Détection d'arrêts	√						√	√
Détection de contresens						√		

TABLE 2.2: Récapitulatif des traitements effectués sur les séquences de test.

comportement. Dans le tableau 2.2 sont répertoriées les vidéos utilisées pour chaque type d'évaluation des performances du système.

Dans les chapitres suivant, l'analyse des performances du système utilisera des séquences de tests parmi les vidéos suivantes (voir Figure 2.12) :

- **C5** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues). Cette séquence a été choisie pour les nombreux changements de luminosité qu'elle contient causés par le passage de nuages. Cette séquence de test est utilisée pour évaluer les performances de la détection et du suivi d'objets.
- **Lyon** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues). Cette scène contient une voie d'insertion rejoignant un ensemble de trois voies. La caméra est reculée par rapport à la route et les objets sont de petites tailles comparés aux autres séquences de test.
- **C21** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues). Il s'agit d'une séquence contenant deux voies de circulation et une voie de sortie. La route sous surveillance est légèrement courbée. Cette séquence est utilisée pour évaluer les performances de détection et de suivi d'objets.
- **C9** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues). La scène comporte trois voies de circulation ainsi qu'une voie de sortie. Cette dernière est considérée comme étant une bande d'arrêt d'urgence pour évaluer la détection de présence d'un objet dans une zone interdite. Cette séquence est également utilisée pour évaluer la détection d'objets.
- **CE11** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues). La scène sous surveillance comporte trois voies sur la chaussée de droite et deux voies sur la chaussée de gauche. Cette séquence est utilisée pour évaluer la détection et le suivi d'objets.
- **Highway-II** - Scène contenant des véhicules légers et poids lourds. Il s'agit d'une

séquence disponible publiquement [2]. Cette séquence est utilisée pour estimer les performances de la détection et du suivi d'objets.

- **I-LIDS-Easy** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues) ainsi que des piétons circulant sur le trottoir et traversant la route. Il s'agit d'une séquence disponible publiquement [3] utilisée pour la détection d'évènement (arrêts de véhicules) et pour le comptage d'objets.
- **I-LIDS-Medium** - Scène contenant des véhicules de tout type (légers, poids lourds et deux roues) ainsi que des piétons circulant sur le trottoir et traversant la route. Il s'agit d'une séquence disponible publiquement [4] utilisée pour la détection d'évènement (arrêts de véhicules) et pour le comptage d'objets.
- **C10** - Scène contenant des véhicules légers et poids lourds. Il s'agit d'une scène de nuit comportant trois voies sur la chaussée de droite et de gauche. Cette séquence est utilisée pour évaluer la détection et le suivi d'objets dans des conditions nocturnes.

2.4 Conclusion

Ce chapitre a présenté l'architecture du système utilisé pour l'interprétation de scènes autoroutières et l'analyse du trafic. La démarche envisagée consiste en une analyse et une utilisation croissante des connaissances en termes de richesse sémantique. Ainsi, à travers l'analyse de caractéristiques de bas-niveau (couleur, texture, mouvement) les régions susceptibles de contenir des objets en mouvement sont estimées. Ces régions sont ensuite analysées afin d'extraire les objets et leurs caractéristiques. Un processus de mise en association est finalement utilisé pour suivre les objets au cours du temps. Enfin, l'étape d'analyse de comportement permet d'extraire les comportements suspects (non fréquents) et les statisiques du trafic. Ces étapes sont précédées d'une étape d'initialisation, visant à extraire des informations sur la structure de la scène et le comportement des objets qui y évoluent. L'estimation du modèle de scène s'effectue au travers une étape d'apprentissage, dans laquelle une séquence d'entrainement est analysée. Cette étape fait l'objet du chapitre suivant.

2. http://cvrr.ucsd.edu/aton/shadow/data/highwayII_raw.AVI
3. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html
4. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

(a) C5

(b) CE11

(c) Lyon

(d) C9

(e) C21

(f) C10

(g) Highway-II

(h) I-LIDS-Easy

(i) I-LIDS-Medium

FIGURE 2.12: Exemple d'images extraites des séquences de test

Chapitre 3

Initialisation et modélisation de la scène

Sommaire

Introduction

Ce chapitre présente l'approche utilisée pour modéliser la scène sous surveillance. Cette approche fait également office d'étape d'initialisation du système et des modèles qu'il comporte. L'objectif consiste à obtenir un modèle du comportement des objets et des caractéristiques de la scène afin de guider les étapes de détection et de suivi des objets. Ainsi, un modèle de la structure de la scène est construit dans l'objectif de segmenter les éléments de la scène en fonction de leurs comportements (ou de ceux qu'ils contiennent). Il s'agit typiquement de détecter la zone représentant la chaussée, les délimitations des voies, les zones d'entrée et de sortie, le sens de circulation, etc ...

Ces éléments peuvent être sélectionnés manuellement par l'utilisateur, nous proposons néanmoins une approche automatique qui se déroule en deux étapes (Figure 3.1) :

- Analyse d'une séquence d'apprentissage.
- Construction du modèle de scène.

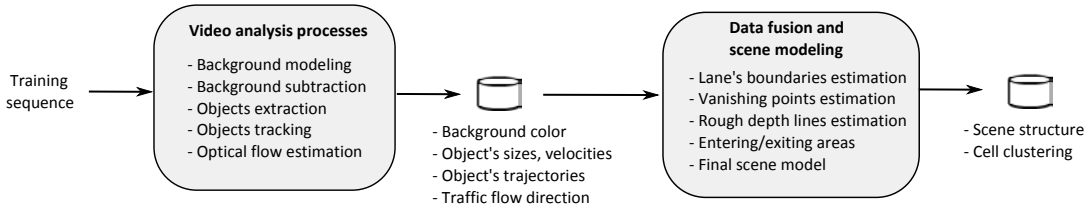


FIGURE 3.1: L'approche proposée se décompose en deux étapes : une procédure d'analyse d'une séquence d'apprentissage et une étape d'extraction des informations issues de l'analyse.

Ainsi, un processus complet d'analyse vidéo est utilisé pour obtenir l'ensemble des informations nécessaires à la construction de notre modèle, comportant les étapes de modélisation d'arrière-plan, d'estimation des objets en mouvement et de suivi d'objets. Nous supposerons dans cette section que les séquences d'apprentissage sont représentatives d'une circulation fluide et sans changement de luminosité. Plus la séquence d'apprentissage est complexe, plus les traitements devront être robustes aux difficultés rencontrées.

Les opérations présentées dans cette section sont adaptées aux séquences de test choisies et ne traitent pas des difficultés que l'on peut rencontrer dans des situations plus complexes (changements de luminosité, occlusions des objets, ...). La gestion de ces difficultés et le processus complet d'analyse de notre système seront présentés aux Chapitres 4, 5 et 6.

3.1 Analyse d'une séquence d'apprentissage

Cette étape consiste à extraire des informations sur la scène et sur le comportement général des objets à partir d'une séquence d'apprentissage. L'objectif est d'obtenir une représentation de l'arrière-plan, de définir les trajectoires typiques des véhicules et d'estimer le sens de direction du trafic. Ces informations sont exploitées dans l'étape suivante (Section 3.2) pour construire un modèle de la structure de la scène.

Le déroulement de l'analyse de la séquence d'apprentissage est présenté sur la Figure

3.2. Il s'agit d'en extraire les caractéristiques couleurs de l'arrière-plan ainsi que les caractéristiques de mouvement à travers l'estimation des vecteurs de déplacement de points d'intérêt dans l'image. L'étape d'extraction des objets en mouvement et de suivi au cours du temps permet finalement d'extraire et de définir les trajectoires typiques des véhicules.

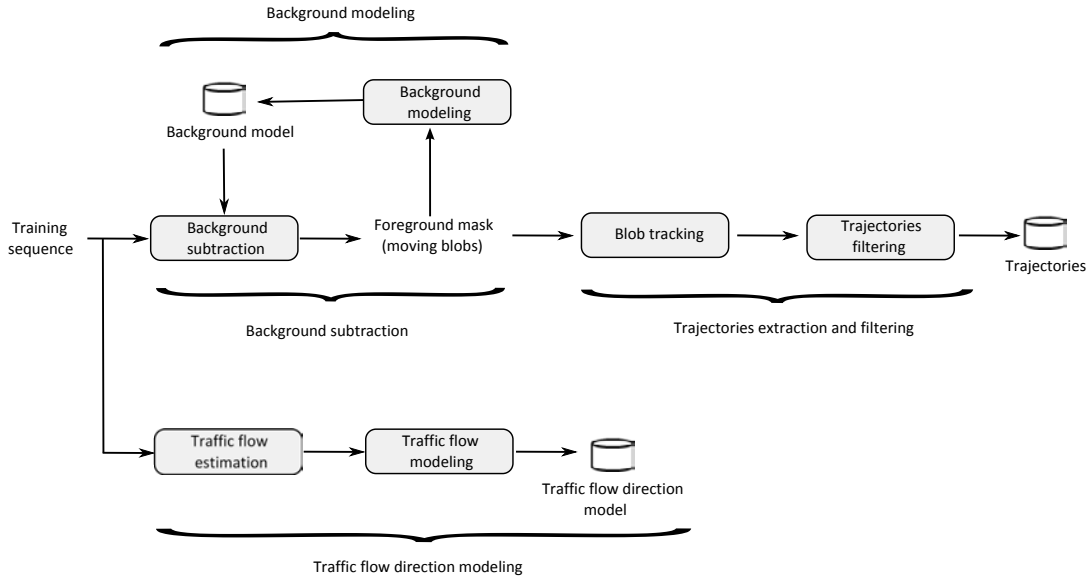


FIGURE 3.2: Procédure d'analyse de la séquence d'apprentissage permettant d'extraire les caractéristiques couleurs de l'arrière-plan, la taille des objets, leurs trajectoires et le sens de direction du trafic.

Nous supposerons que la séquence d'apprentissage ne comporte aucun évènement anormal, que les véhicules circulent dans le sens de direction du trafic et que les changements de luminosité sont faibles. Notons également que le traitement présenté dans cette Section est effectué pour toutes les images, autrement dit avec une fréquence de rafraichissement de la vidéo de 25 fps.

3.1.1 Modélisation de l'arrière-plan

La modélisation de l'arrière-plan est une étape essentielle dans la majorité des systèmes de vidéo-surveillance [Moeslund 2006]. Il s'agit de créer un modèle d'arrière-plan de la scène ne contenant aucun objet mobile. Il existe de nombreuses méthodes de modélisation et la littérature est abondante sur le sujet ([Toyama 1999], [McIvor 2000], [Piccardi 2004], [Moeslund 2006], [Yilmaz 2006], [Bouwmans 2010]). Un aperçu des méthodes existantes et une description complète du modèle utilisé par le système seront présentés au Chapitre 4. Nous nous intéressons ici simplement à la construction d'une image d'arrière-plan de la scène afin d'en étudier la structure (contours de l'image). Notons que ce modèle sera utile pour initialiser le modèle plus complexe présenté au Chapitre 4.

Parmi les nombreuses méthodes existantes, nous avons envisagé une approche simple et rapide pour estimer l'image de fond. Une méthode récursive a été privilégiée : l'image d'arrière-plan est mise à jour pour chaque nouvelle image de la séquence d'apprentissage.

Nous supposons que la séquence d'apprentissage n'est pas soumise à des changements de luminosité et que les objets se déplacent continuellement dans la scène (les pixels d'arrière-plan sont majoritairement représentés).

L'approche employée consiste en une estimation récursive de l'arrière-plan à l'aide d'un modèle statistique représentant la distribution de couleur pour chaque pixel de l'image. Chaque pixel de l'image est représenté à l'aide d'une distribution gaussienne paramétrée par sa moyenne μ et son écart-type σ, comme suggéré dans [Wren 1997], [McKenna 2000] ou [François 1999]. Chaque pixel est modélisé à l'aide d'un vecteur contenant les caractéristiques couleur rgb s'écrivant $[\mu_r, \mu_g, \mu_b, \sigma_r, \sigma_g, \sigma_b]$. Pour chaque nouvelle observation x, les paramètres sont mis à jour récursivement selon

$$\begin{cases} \mu = (1-\alpha_t)\mu + \alpha_t x \\ \sigma^2 = \max\left((1-\alpha_t)\sigma^2 + \alpha_t(x-\mu)^2, \sigma^2_{min}\right) \\ \alpha_t = \max(\alpha_{min}, 1/t) \end{cases} \tag{3.1}$$

avec x la caractéristique couleur rgb d'un pixel et $\alpha(t)$ un paramètre appelé *taux d'apprentissage* et qui permet de contrôler la vitesse de mise à jour du modèle. Le paramètre α est rendu variable et décroissant pour les premières images de la séquence d'apprentissage jusqu'à une valeur limite α_{min}. Le choix de cette valeur est défini empiriquement et dépend de la séquence vidéo ainsi que de la fréquence d'acquisition. Plus la valeur de α_{min} est élevée, plus la vitesse de mise à jour est grande. Une valeur élevée du paramètre α_{min} prend majoritairement en compte la valeur de la nouvelle observation plutôt que les anciennes données contenues dans le modèle. Nous utilisons dans notre implémentation une valeur de $\alpha_{min} = 0.05$. La valeur minimum de l'écart-type σ_{min} est introduite en tant que seuillage du bruit, ce qui permet d'éviter à l'écart-type du modèle d'atteindre une valeur inférieure à σ_{min}.

L'image d'arrière-plan B est estimée à partir des moyennes μ pour chaque pixel de l'image. Le calcul des écarts à la moyenne σ^2 va permettre d'effectuer la tache de soustraction d'arrière-plan, et plus particulièrement va permettre de s'affranchir du choix d'un seuil global pour la segmentation.

3.1.2 Soustraction d'arrière-plan

L'opération de soustraction d'arrière-plan est une opération qui suit de façon logique la modélisation d'arrière-plan. L'objectif consiste à détecter les pixels en mouvement en comparant l'image courante de la vidéo à un modèle d'arrière-plan. Si le modèle est une image, une différence en valeur absolue est généralement employée. S'il s'agit d'un modèle statistique, la probabilité qu'un pixel appartienne à l'arrière-plan est estimée en testant la valeur observée dans le modèle. Une faible probabilité d'appartenance signifie que le pixel observé appartient à un objet en mouvement. La carte des distances (ou des probabilité) est ensuite seuillée afin d'obtenir une classification binaire *background* ou *foreground*.

La comparaison entre une nouvelle observation x de l'image et la distribution gaussienne (de moyenne μ et de variance σ) est définie par la distance euclidienne :

$$d_M(x, \mu) = \sqrt{(x_r - \mu_r)^2 + (x_g - \mu_g)^2 + (x_b - \mu_b)^2} \tag{3.2}$$

Un pixel est considéré comme étant en mouvement si la distance au modèle est supérieure à un seuil. Ce seuil est directement lié à la variance de la gaussienne considérée, en prenant la valeur minimum de la variance parmi les composantes couleurs, un pixel est classé selon :

$$\begin{array}{lll} F = 0 & \text{Si} & d_M < 2.5 \min(\sigma_r, \sigma_g, \sigma_b) \\ F = 1 & \text{Sinon} & \end{array} \tag{3.3}$$

L'utilisation de la variance permet de s'affranchir du choix d'un seuil global dans l'image. Les pixels en mouvement sont regroupés et les régions sont étiquetées à l'aide d'une analyse en composantes connexes (voisinnage 8-pixels) [Suzuki 1985]. Chaque région (appelée *blob* - *binary large object*) est représentée par la position et la taille de sa boite englobante associée à la région. Le résultat fourni par cette étape est une liste des *blobs* issus de masque *foreground*.

3.1.3 Estimation des trajectoires

Une fois les *blobs* extraits du masque *foreground*, une liste d'objets suivis appelés *tracked blobs* est créée afin de maintenir l'identité des régions en mouvement au cours du temps. A chaque objet est associée une trajectoire stockée sous la forme d'une liste de points 2D (coordonnées de l'objets dans l'image). Ces trajectoires sont obtenues à l'aide d'une mise en correspondance simple, dans laquelle les objets sont associés à l'aide de l'aire de recouvrement existante entre deux objets à l'instant t et $t-1$. L'objectif ici n'est pas d'obtenir les trajectoires précises des objets, mais simplement d'en estimer leurs directions.

Ainsi chaque *tracked blob* de la liste des objets suivi (à l'instant $t-1$) est mis en correspondance avec les *blobs* de la liste des nouveaux objets détectés (instant t). Si un objet suivi ne possède aucun candidat, alors il est supprimé de la liste. Les situations de division-fusion (*merge-split*) ne sont pas prises en compte : lorsque deux nouveaux *blobs* sont simultanément associés à un seul *tracked blob* (fusion), seul celui comportant le plus grand taux de recouvrement lui est associé, tandis que le second *blob* est supprimé. De même, lorsqu'un nouveau *blob* est associé à deux *tracked blobs* simultanément (division), seul celui comportant le plus grand taux de recouvrement lui est associé, tandis que le second est supprimé. Enfin, tout nouveau *blob* non associé à un *tracked blob* est sauvegardé dans la liste des objets suivis en tant que *tracked blob*.

Les règles définies précédemment réduisent considérablement la longueur des trajectoires, mais permettent de limiter les erreurs de suivi d'objets et de ne conserver que les morceaux de trajectoires obtenus sans ambiguïté d'association. Un filtrage sur le nombre de points N est effectué afin de ne conserver que les trajectoires suffisamment longues ($N > 10$). Malgré la simplicité de l'approche, elle est suffisante pour obtenir des trajectoires (ou morceaux de trajectoires) qui seront utilisées durant l'estimation du point de fuite dans l'image (Figure 3.3). La procédure complète du processus de suivi des véhicules est présentée au Chapitre 5.

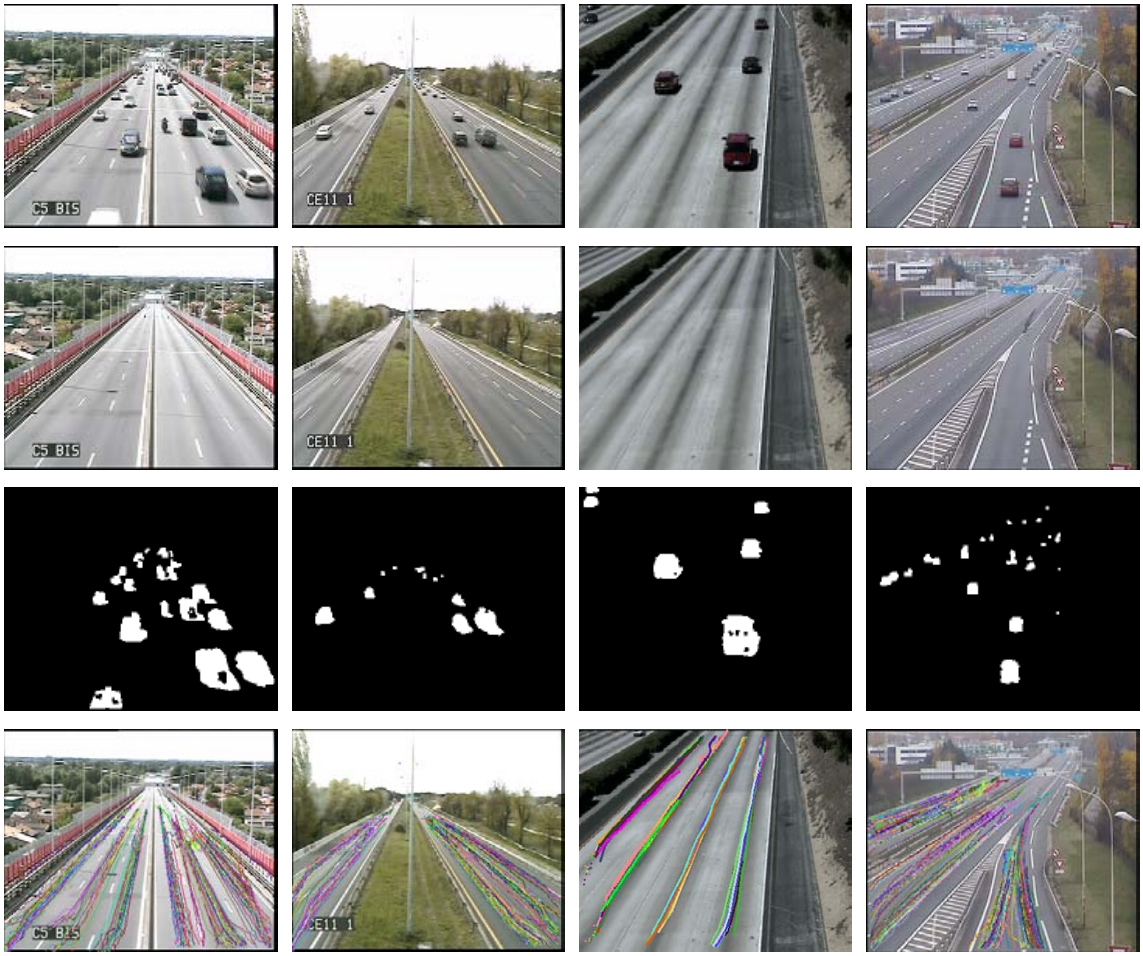


FIGURE 3.3: (1ère ligne) Première image des séquences vidéos d'apprentissage. (2nde ligne) Estimation de l'arrière-plan de la scène. (3ème ligne) Estimation du masque des objets en mouvement (*foreground*). (4ème ligne) Estimation des trajectoires des objets.

La Figure 3.3 présente les résultats de l'analyse de la séquence d'apprentissage. La première ligne montre une capture d'écran des séquences de test. La seconde ligne montre l'estimation de l'image d'arrière-plan qui ne contiennent aucun objet en mouvement. L'estimation obtenue reflète visuellement bien la scène sous surveillance avec cependant quelques artefact sur la séquence 4. La troisième ligne est une estimation des objets en mouvement de la capture d'écran. Enfin, la quatrième ligne montre les résultats obtenus pour le suivi d'objets qui fournissent un ensemble de trajectoires du comportement global des véhicules.

3.1.4 Estimation des vecteurs mouvements

L'estimation du sens de direction du trafic consiste en une méthode de détection et de suivi de points d'intérêt tels que les coins ou les contours (par exemple [Moravec 1980], [Harris 1988], [Shi 1994]), ou toute autre caractéristique pertinente (par exemple [Lowe 1999], [Bay 2008]) pour le suivi. Les objets sont indirectement suivis à travers l'estimation du déplacement des points d'intérêt qui les composent. Notons que le regroupement de points

d'intérêt pour former les objets est une tâche qui peut se révéler difficile. L'objectif ici est d'obtenir et de modéliser le déplacement moyen de l'ensemble des points d'intérêt détectés dans l'image. Le processus complet se décompose en plusieurs étapes (Figure 3.4).

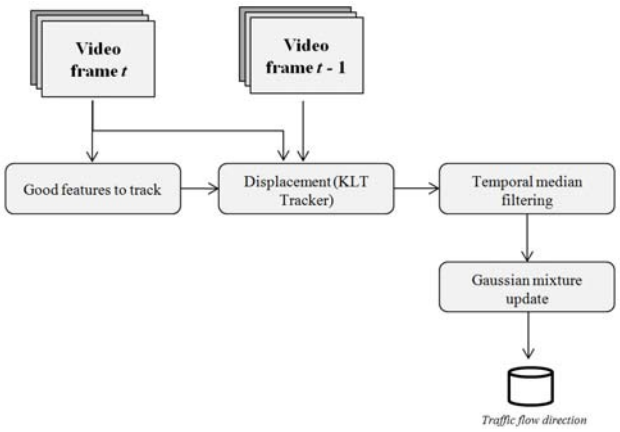


FIGURE 3.4: Processus d'estimation des vecteurs mouvements à partir du déplacement des points caractéristiques.

La première étape consiste à estimer les points d'intérêt dans l'image. Nous avons sélectionné le détecteur proposé dans [Shi 1994] qui fournit un ensemble de points d'intérêt appelés *good features to track*. Le déplacement des points d'intérêt est estimé à l'aide d'un algorithme d'estimation de flot optique proposé dans [Takeo 1991]. Pour améliorer le temps de calcul et prendre en compte les déplacements de faibles amplitudes, nous utilisons une version pyramidale de l'algorithme décrit dans [Bouguet 2001]. Une fois le champ de déplacement estimé, le modèle de déplacement est mis à jour de façon récursive à l'aide d'un mélange de lois statistiques. Le modèle utilisé est décrit dans le paragraphe suivant.

3.1.5 Apprentissage du sens de direction du trafic

Cette section s'appuie sur le modèle statistique présenté au Chapitre 4 (Section 4.1)dans lequel un mélange de distribution de probabilité est utilisé pour modéliser les distributions couleurs des pixels. Nous nous intéressons ici à l'application d'une telle modélisation à un champ de vecteurs issu de l'estimation du flot optique. Plus précisément, nous nous intéressons à l'orientation des vecteurs (donnée périodique) permettant d'estimer le sens de direction du trafic.

Les données angulaires sont une classe de données particulières définies sur un domaine circulaire (défini sur une sphère) différent du domaine linéaire classique (défini dans l'espace euclidien). Il s'agit généralement de mesures liées à l'orientation ou à la direction d'un phénomène. Ces mesures peuvent également représenter un phénomène périodique dans le temps (heures et fréquences des visites dans un hôpital par exemple). Il s'agit dans notre cas de l'orientation des vecteurs mouvements caractérisant le déplacement des objets dans la scène. L'ensemble de ces directions peut être représenté sur le cercle et être mesuré par rapport à une certaine *direction zéro* et un sens de rotation. Dû au caractère périodique des données, il est nécessaire d'utiliser des méthodes statistiques qui ne dépendent pas du choix arbitraire de la *direction zéro*. Par conséquent, les techniques définies dans le

domaine linéaire sont généralement inadaptées aux données périodiques. Dans le domaine de la statistique angulaire, il existe plusieurs distributions parmi lesquelles on retrouve la distribution uniforme, la distribution Wrapped Normal et la distribution von-Mises [Mardia 2000].

Distribution von-Mises

La distribution von-Mises (vM) est un cas particulier de la distribution von-Mises Fisher [Mardia 2000] de dimension p dans R^p. Il s'agit d'une fonction périodique dans $\mathbb{R}^2$ ($p = 2$), autrement dit, la densité de probabilité correspondante p_{vM} est périodique de période 2π telle que pour toute variable aléatoire circulaire prenant ses valeurs sur la circonférence du cercle unité :

$$p_{vM}(\mathbf{x} + \omega 2\pi) = p(x), \forall \omega \in Z$$

La densité von-Mises possède certaines caractéristiques analogues à la distribution gaussienne, notamment :

- Elle est entièrement décrite par deux paramètres : une direction moyenne et un paramètre de concentration autour de cette moyenne.
- Elle est symétrique par rapport à la direction moyenne.
- Elle est uni-modale sur une période et son mode correspond à la direction moyenne.

La loi von-Mises $V(\mathbf{x}; \mu, \kappa)$ permet de décrire statistiquement la distribution d'une variable aléatoire circulaire $\mathbf{x}$ de moyenne μ et de variance homogène à $1/\kappa$. Le paramètre de concentration κ varie entre 0 et l'infini et lorsque κ tend vers 0, la distribution tend vers une distribution uniforme dans laquelle aucune direction n'est privilégiée. Lorsque κ augmente, la distribution tend vers une distribution gaussienne de variance $1/\kappa$. La densité de probabilité d'une distribution von-Mises est donnée par

$$p_{vM}(\mathbf{x}|\mu, \kappa) = \frac{\exp^{(\kappa cos(\mathbf{x}-\mu))}}{2\pi I_0(\kappa)} \qquad , -\pi - \mu <= \mathbf{x} <= \pi + \mu \tag{3.4}$$

avec I_0 la fonction de Bessel modifiée d'ordre 0 donnée par :

$$\begin{aligned} I_0(\kappa) &= \frac{1}{2\pi} \int_0^{2\pi} \exp^{(\kappa \cos(\phi)))} \, d\phi \\ &= \sum_{l=0}^{\infty} \frac{1}{(l!)^2} \left(\frac{\kappa}{2}\right)^{2l} \end{aligned} \tag{3.5}$$

L'analyse du maximum de vraisemblance fournit les équations suivantes pour l'estimation de la moyenne μ et de la concentration κ [Jammalamadaka 2001] :

$$\begin{aligned} \mu &= \arctan\left(\frac{\sum_{i=1}^{N} \sin x_i}{\sum_{i=1}^{N} \cos x_i}\right) \\ A(\kappa) &= \frac{I_1(\kappa)}{I_0(\kappa)} = \frac{1}{N} \sum_{i=1}^{N} cos(x_i - \mu) \end{aligned} \tag{3.6}$$

Mélange de distributions von-Mises

Nous considérons à présent le cas d'une modélisation des données à l'aide d'un mélange de K lois de type von-Mises. L'utilisation d'un mélange de lois pour la modélisation des données est étudiée dans la Section 4.1. Il s'agit d'une combinaison linéaire de K lois de type von-Mises (p_{vM}) pondérées par un poids noté w_k. La densité de probabilité du mélange, noté p s'écrit

$$p(x|\mu,\kappa) = \sum_{k=1}^{K} w_k p_{vM}(x|\mu_k,\kappa_k) \tag{3.7}$$

Dans le cas d'un mélange de lois, l'estimation des paramètres $\Phi = (\mu,\kappa)$ à l'aide du maximum de vraisemblance est difficile (voir Section 4.1.1) et les paramètres du mélange sont estimés à l'aide d'une méthode d'optimisation itérative. L'algorithme EM [Dempster 1977] est particulièrement adapté à l'estimation des paramètres d'un mélange de lois lorsque l'on fait les hypothèses suivantes :

- Un échantillon ne peut être issu que d'une seule composante du mélange.
- Il existe une variable aléatoire dite *cachée* (car non observée), noté $\mathbf{z}$, qui exprime de quelle composante est issue l'échantillon x.

L'algorithme EM est un algorithme itératif basé sur deux étapes (voir Section 4.1.2) : une étape d'*Expectation*, durant laquelle la densité de probabilité conditionnelle $p(\mathbf{z}|\mathbf{x},\Phi^t)$ est estimée, et une étape de *Maximization*, qui consiste à re-estimer les paramètres du modèle Φ^{t+1} en s'appuyant sur la probabilité $p(\mathbf{z}|\mathbf{x},\Phi^t)$ précédemment estimée [Banerjee 2006].

Expectation :

$$p(\mathbf{z}=k|x_i,\Phi^t) = \frac{w_k p(x_i|\mu_k^t,\Sigma_k^t)}{\sum_{l=1}^{K} w_l p(x_i|\mu_l^t,\Sigma_l^t)} \tag{3.8}$$

Maximization :

$$\begin{cases} w_k^{t+1} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t) \\ \mu_k^{t+1} = \arctan\left(\dfrac{\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)\sin x_i}{\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)\cos x_i}\right) \\ A(\kappa_k) = \dfrac{I_1(\kappa_k)}{I_0(\kappa_k)} = \dfrac{\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)\cos(x_i-\mu_k^{t+1})}{\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)} \end{cases} \tag{3.9}$$

Le ratio des fonctions de Bessel $A(\kappa)$ ne peut pas être obtenu de façon analytique et doit être estimé numériquement [Hill 1981]. Dans [Banerjee 2006] et [Roy 2008], les auteurs proposent cependant l'utilisation de l'approximation suivante :

$$\kappa_k = \frac{2A(\kappa_k) - A(\kappa_k)^3}{1 - A(\kappa_k)^2} \tag{3.10}$$

Ou encore, dans [Carta 2008] les auteurs estiment les paramètres d'un mélange de distribution von-Mises à l'aide de la méthode des moindres carrés, et proposent l'estimation

suivante :

$$\kappa_k = \left(23.29041409 - 16.8617370\sqrt{A(\kappa_k)} - 17.4749884\exp^{-A(\kappa_k)^2}\right)^{-1} \tag{3.11}$$

avec

$$\begin{aligned} A(\kappa_k) &= \frac{I_1(\kappa_k)}{I_0(\kappa_k)} = \left(\overline{s}^2 + \overline{c}^2\right)^{\frac{1}{2}} \\ \overline{s} &= \frac{\sum_{i=1}^{N} \sin x_i}{N} \\ \overline{c} &= \frac{\sum_{i=1}^{N} \cos x_i}{N} \end{aligned} \tag{3.12}$$

Construction récursive du modèle

Les Equations 3.9 et 3.11 fournissent une méthode d'estimation des paramètres du mélange dans le cas de distributions de type von-Mises. Ces équations font intervenir l'ensemble des échantillons observés et demande un grande quantité de mémoire lorsque le nombre d'échantillons N devient grand. Il est possible d'obtenir une version récursive de ces équations telle que pour chaque nouvel échantillon x_t, on a

$$\begin{cases} w_k^{t+1} & = (1-\alpha)w_k^t + \alpha p_{vM}(k|x_t, \Phi^t) \\ \mu_k^{t+1} & = \arctan(\frac{\overline{s}_k^{t+1}}{\overline{c}_k^{t+1}}) \\ \kappa_k^{t+1} & = \frac{2A(\kappa_k^{t+1}) - A(\kappa_k^{t+1})^3}{1 - A(\kappa_k^{t+1})^2} \end{cases} \tag{3.13}$$

avec

$$\begin{cases} \overline{s}_k^{t+1} & = (1-\rho_k)\overline{s}_k^t + \rho_k \sin(x_t) \\ \overline{c}_k^{t+1} & = (1-\rho_k)\overline{c}_k^t + \rho_k \cos(x_t) \\ A(\kappa_k)^{t+1} & = ((\overline{s}_k^{t+1})^2 + (\overline{c}_k^{t+1})^2)^{\frac{1}{2}} \end{cases} \tag{3.14}$$

L'algorithme de mise à jour correspondant est illustré [**?**] :

3.1.6 Résultats de l'apprentissage

Nous supposons que les objets circulent dans le sens de direction du trafic et qu'aucun véhicule ne circule en contre-sens. Le processus d'estimation du sens de direction du trafic est illustré sur la Figure 3.4. Le sens de direction est modélisé à l'aide d'un mélange de lois von-Mises [Mardia 2000]. L'utilisation d'un mélange permet de prendre en compte plusieurs directions ce qui est utile dans le cas d'entrées ou de sorties de voies. Ceci permet également de modéliser le mouvement des véhicules qui changent de voies ou qui dépassent d'autres véhicules.

Le modèle de mouvement est appris à partir du champ des vecteurs mouvements obtenu à l'aide de la méthode de Lucas-Kanade-Tomasi décrite dans la Section 3.1.4. Un mélange

Algorithme 1: Mise à jour des paramètres Φ d'un mélange de lois von-Mises

Initialisation des paramètres w_k, μ_k, κ_k du modèle
for each *pixel* $\boldsymbol{x}$ **do**
  for each *distribution von-Mises* k *du mélange* **do**

$$r_k = \begin{cases} w_k p_{vM}(\mathbf{x}, \mu_k, \kappa_k) & \text{si} \quad \min(\mathbf{x} - \mu_k, 2\pi - (\mathbf{x} - \mu_k)) <= \tau \\ 0 & \text{sinon} \end{cases}$$

    if $r_k \neq 0$ **then**
      $r_k = r_k / \sum_k r_k$
      $w_k^+ = (1 - \alpha) w_k + \alpha r_k \; - \alpha . c$
      $\overline{s}_k^+ = (1 - \rho)\overline{s}_k + \rho \sin(\mathbf{x})$
      $\overline{c}_k^+ = (1 - \rho)\overline{c}_k + \rho \cos(\mathbf{x})$
      $\kappa_k^+ = (2A(\kappa_k^+) - A(\kappa_k^+)^3)/(1 - A(\kappa_k^+)^2)$
    else if $\sum_k r_k \neq 0$ **then**
      $w_k^+ = (1 - \alpha) w_k - \alpha . c \quad (k \neq j)$
    else
      $j = \arg\min_k(w_k)$
      $w_j = \alpha$, $\mu_j = \mathbf{x}$, $\kappa_j = \kappa_0$

de K lois de probabilité de type von-Mises est utilisé et constitue le modèle statistique des vecteurs d'orientation.

L'apprentissage est effectué sur quatre séquences de test : C5, CE11, Highway-II et Lyon pour une durée d'apprentissage d'environ 15 minutes (22000 images) excepté pour la séquence Highway-II qui est une séquence courte de 32 secondes (800 images). Le taux d'apprentissage α est fixé à 0.05 et $\rho = 1/c_i$, avec c_i le nombre d'occurrences de mise à jour du pixel x_i considéré.

La Figure 3.5 montre le sens de direction estimé par la première composante du mélange utilisée. La palette de couleurs des orientations est affichée en haut à droite de l'image. Ces résultats montrent une bonne estimation du sens de direction du trafic représenté par la première composante du modèle. La carte d'orientation des séquences C5 et Lyon sont parfaitement estimées, avec une représentation précise des orientations relevées grâce au trafic dense des séquences utilisées. La séquence CE11 contient davantage de bruit particulièrement en bordure d'image et contient un trafic moins dense, résultant en une carte moins précise. Quant à la séquence Highway, sa courte durée résulte en une carte incomplète au niveau pixel. Le sens de direction du trafic est malgré tout parfaitement représenté.

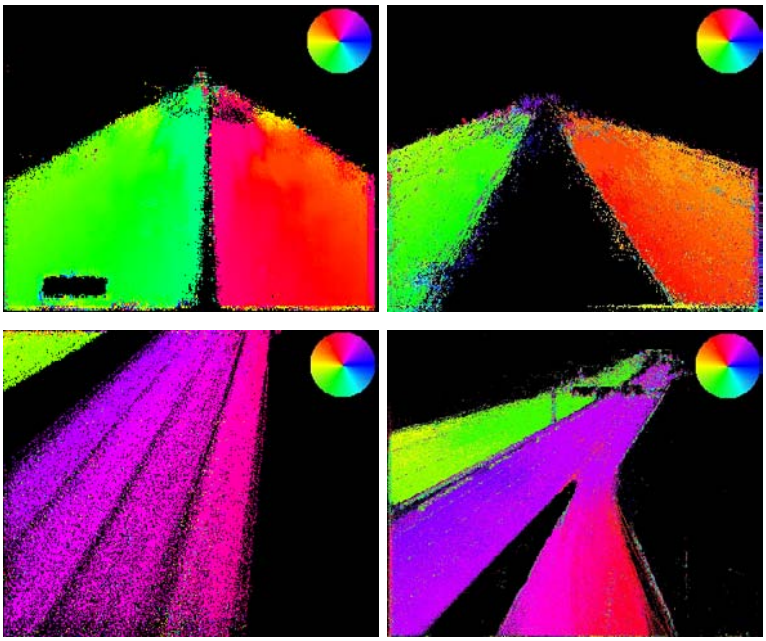


FIGURE 3.5: Sens de direction du trafic donné par la première composante du mélange de distribution von-Mises. La palette de couleur utilisée pour représenter les angles est affichée en haut à droite de l'image.

3.2 Construction du modèle de scène

L'objectif consiste à partitionner la scène en régions contenant des informations de plus haut niveau sémantique, telles que les délimitations des voies de circulation, le point de fuite des trajectoires des objets qui y circulent ou encore l'estimation de la profondeur de la scène sous surveillance.

3.2.1 Détection des bordures des voies

La détection présentée ici est basée sur l'algorithme CHEVP (Canny Hough Estimation of Vanishing Point) proposé dans [Wang 2004]. L'algorithme se base sur l'image d'arrière-plan, ce qui permet de travailler sur une image ne contenant aucun objet. Pour réduire le bruit, un filtre médian (préservant les contours) est appliqué. L'algorithme se décompose en plusieurs étapes (Figure 3.6).

Tout d'abord, l'information de contours est extraite à l'aide du détecteur de contours de Canny. Une fois la carte de contours connue, les lignes sont estimées à l'aide de la transformée de Hough standard [Illingworth 1988]. Cette méthode a été introduite par Paul Hough dans [Hough 1962] pour détecter des formes géométriques prédéfinies telles que des droites, des cercles ou des ellipses. La seule condition nécessaire pour son utilisation est la possibilité de représenter la forme recherchée sous une forme paramétrée. L'idée principale de la transformée de Hough est le transfert des informations de la carte des contours vers un espace des paramètres. La dimension de l'espace des paramètres correspond au nombre

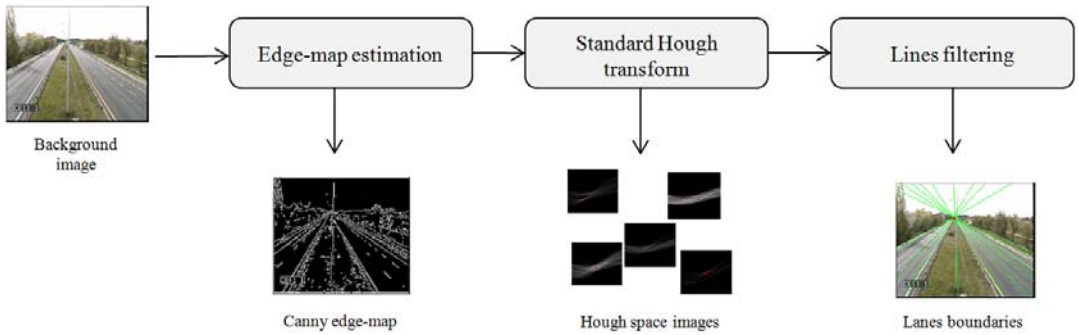


FIGURE 3.6: Procédure de détection des délimitations des voies.

de paramètres de la forme à détecter. Un processus de vote est effectué à l'aide d'un accumulateur pour chaque point de l'image contour.

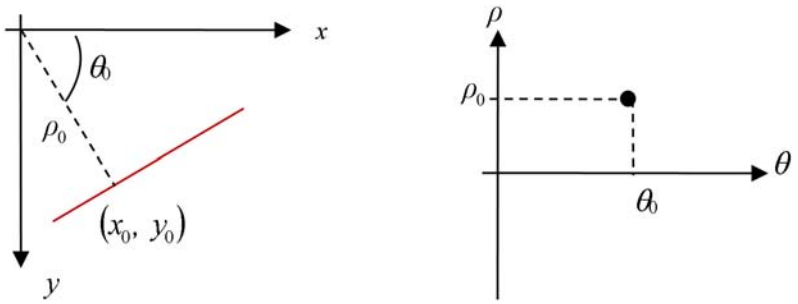


FIGURE 3.7: (À gauche) Représentation d'une droite paramétrée par ρ_0 et θ_0 dans l'espace euclidien. (À droite) Représentation de la droite paramétrée par ρ_0 et θ_0 dans l'espace de Hough.

Sous sa forme cartésienne $y = a.x + b$, une droite est définie par 2 paramètres : le coefficient directeur a et son ordonnée à l'origine b. L'ensemble des points de la droite $y = a.x + b$ est représenté par le point (a, b) dans l'espace des paramètres. A cause de la non-uniformité de l'espace des paramètres en utilisant la forme cartésienne d'une droite (la probabilité d'avoir un coefficient directeur entre $[0; 1]$ est la même que d'obtenir ce coefficient entre $[1; \infty]$), la forme paramétrée polaire est généralement utilisée : $x\cos(\theta) + y\sin(\theta) = \rho$. La valeur de θ est comprise entre 0 et π et la valeur de ρ est comprise entre 0 et la distance diagonale de l'image, ce qui définit les dimensions de l'espace des paramètres.

La transformée de Hough est un processus de vote. Soit P_0 un point défini sur la carte des contours de coordonnées cartésiennes (x_0, y_0) et de coordonnées polaires (ρ_0, θ_0). Toutes les lignes passant par ce point vérifient l'équation

$$x_0\cos(\theta) + y_0\sin(\theta) = \rho \tag{3.15}$$

Puisque $x_0 = \rho_0\cos(\theta_0)$ et $y_0 = \rho_0\sin(\theta_0)$, nous pouvons écrire

$$\begin{aligned} \rho =& \rho_0\left(\cos(\theta_0)\cos(\theta) + \sin(\theta_0)\sin(\theta)\right) \\ \rho =& \rho_0\left(\cos(\theta - \theta_0)\right) \end{aligned} \tag{3.16}$$

Ainsi, chaque point de la carte des contours est représenté dans l'espace des paramètres par une sinusoïde. De façon similaire, un ensemble de n points alignés dans la carte des contours est représenté par un ensemble de n sinusoïdes. L'intersection de ces sinusoïdes fournit un point (ρ_i, θ_i) correspondant aux paramètres de la droite passant par cet ensemble de points. L'idée de la transformée de Hough consiste à déterminer localement les intersections des sinusoïdes dans l'espace des paramètres afin d'obtenir l'ensemble des paramètres des droites contenus dans l'image d'entrée.

Dans l'objectif d'effectuer une détection locale, l'image est découpée en blocs de même taille, qui sont ensuite analysés à l'aide de la transformée de Hough. Une fois l'ensemble des espaces de Hough calculés, les maximum locaux pour chaque bloc sont extraits, correspondant aux paramètres des droites qu'ils contiennent (Figure 3.8).

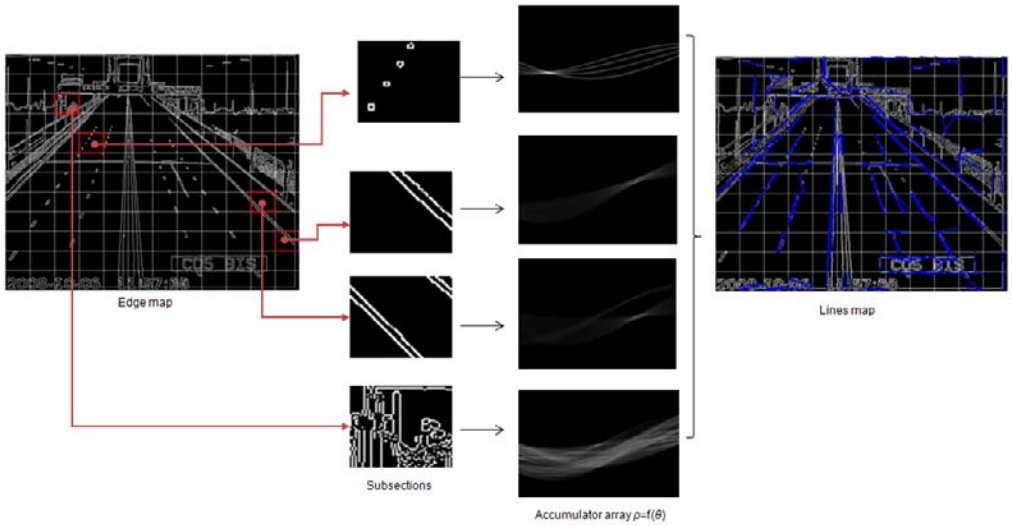


FIGURE 3.8: Illustration de la transformée de Hough sur la carte de contour de l'image d'arrière-plan. La transformée de Hough est appliquée sur chaque bloc de l'image (détection locale). A chaque point de contour correspond une sinusoïde dans l'espace de Hough, les coordonnées des maximum d'intersection des sinusoïdes fournissent les paramètres des droites associées.

3.2.2 Estimation du point de fuite

En théorie et si la caméra n'est pas perpendiculaire à la route, tous les véhicules circulent en direction du point de fuite dans l'image. De plus, les bordures des voies convergent également vers le point de fuite. En supposant que ces hypothèses sont satisfaites, le point de fuite dans l'image est estimé à l'aide des trajectoires. Celui-ci est ensuite utilisé pour valider les bordures des voies précédemment estimées.

Utilisation des bordures des voies

Le point de fuite est estimé à l'aide de l'espace de Hough, comme suggéré dans [Matessi 1999]. L'approche consiste à exploiter l'espace de Hough (en représentation polaire) à travers une

méthode des moindres carrés. Il s'agit de minimiser la quantité suivante :

$$\min_{x_0,y_0} \sum_{i=1}^{n} W_i \left(\rho_i - x_0 \cos(\theta_i) - y_0 \sin(\theta_i)\right)^2 \tag{3.17}$$

avec $W_i = v_i/V$, v_i est le nombre de vote de l'espace des paramètres pour la droite paramétrée par (ρ_i, θ_i) et V est le nombre total de vote. En dérivant l'équation précédente, on obtient

$$\begin{aligned} \sum_{i=1}^{n} \cos(\theta_i) \left(\rho_i - x \cos(\theta_i) - y \sin(\theta_i)\right) = 0 \\ \sum_{i=1}^{n} \sin(\theta_i) \left(\rho_i - x \cos(\theta_i) - y \sin(\theta_i)\right) = 0 \end{aligned} \tag{3.18}$$

En notant

$$\begin{aligned} A &= \sum_{i=1}^{n} W_i \cos^2(\theta_i), & B &= \sum_{i=1}^{n} W_i \sin^2(\theta_i) \\ C &= \sum_{i=1}^{n} W_i \cos(\theta_i)\sin(\theta_i), & D &= \sum_{i=1}^{n} W_i \rho_i \cos(\theta_i) \\ E &= \sum_{i=1}^{n} W_i \rho_i \sin(\theta_i) \end{aligned} \tag{3.19}$$

La solution est obtenue en résolvant le système d'équation suivant :

$$\begin{cases} Ax_0 + Cy_0 = D \\ Cx_0 + By_0 = E \end{cases} \tag{3.20}$$

Ce processus est répété jusqu'à convergence des résultats : en notant $\overline{\rho_i}$ le paramètre de la sinusoïde correspondant au point de fuite estimé, l'erreur résiduelle est définie par

$$\sigma^2 = \sum_{i=1}^{n} W_i (\rho_i - \overline{\rho_i})^2 \tag{3.21}$$

Les droites de paramètres ρ_i telles que $|\rho_i - \overline{\rho_i}| > 2.5\sigma^2$ sont considérées comme étant des valeurs extrêmes (*outliers*), le processus est répété tant qu'il existe des *outliers*[1]. Une fois le point de fuite estimé, il est utilisé pour valider la détection des bordures des voies. Puisque les délimitations des voies de circulation convergent théoriquement vers le point de fuite, seules les droites issues de l'espace de Hough et passant aux alentours du point de fuite sont conservées. Les résultats de l'estimation du point de fuite et des bordures des voies sont illustrées sur la Figure 3.9.

Plusieurs points de fuite définies par les bordures des voies peuvent exister, comme on peut le constater sur la séquence Lyon. Une fois le point de fuite principal estimé,

1. *outlier* pourrait être traduit par valeur extrême ou valeur aberrante.

cette étape est répétée à l'aide des droites restantes (qui sont éloignées du point de fuite principal). Ceci permet de ne pas écarter l'existence potentielle d'un second point de fuite.

Utilisation des trajectoires

Puisque les trajectoires convergent théoriquement vers le point de fuite, elles sont utilisées pour estimer ce dernier et valider les résultats obtenus à partir des bordures des voies. Toutes les trajectoires sont dans un premier temps approximées par des droites à l'aide d'un ajustement par la méthode des moindres carrés. Le point de fuite est ensuite estimé en analysant les intersections de l'ensemble des droites. Le point correspondant au maximum d'intersection est retenu comme étant le point de fuite principal de l'image (voir Figure 3.9).

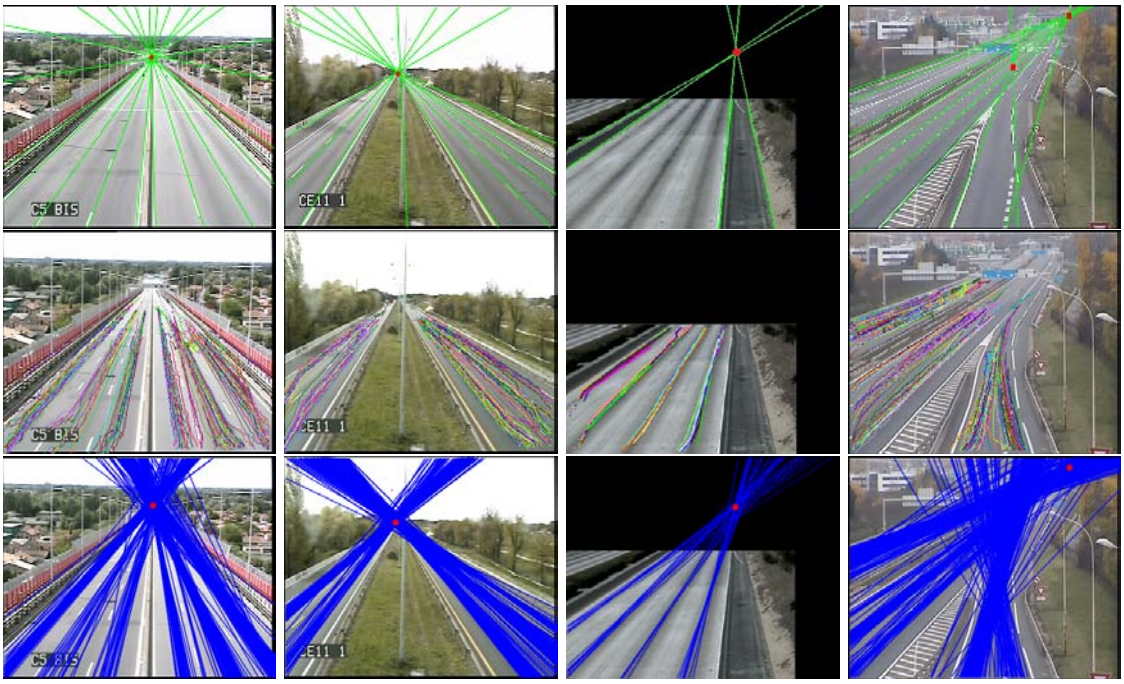


FIGURE 3.9: Estimation du point de fuite dans les séquences de test. (1ère ligne) Estimation du point de fuite à l'aide de l'espace de Hough. Seules les bordures des voies passant par le point de fuite sont conservées. (2ème ligne) Trajectoires des objets. (3ème ligne) Approximation des trajectoires à l'aide de droites et estimation du point de fuite à l'aide du maximum d'intersection.

Les résultats sont illustrés sur la Figure 3.9. Sur la première ligne sont représentées les estimations du point de fuite (en rouge) et des bordures des voies (en vert). Ces dernières sont obtenues en ne conservant que les droites passant aux alentours du point de fuite. Notons l'introduction d'un second point de fuite sur la séquence 4 causée par la présence de la voie d'insertion. Notons également l'échec de l'algorithme face à la séquence 2 qui ne comporte pas de marquage au sol pour délimiter les voies de circulation. Sur la seconde ligne sont représentées les trajectoires utilisées pour valider l'estimation du point de fuite. Le troisième ligne montre les résultats de l'estimation du point de fuite à l'aide des droites

approximant les trajectoires des objets. Ces résultats sont très proches de ceux résultants des bordures des voies. Un seul maximum d'intersection est autorisé dans notre implémentation, ce qui explique l'absence de détection du second point de fuite dans la séquence 4.

L'estimation des bordures des voies pour la séquence 2 est un échec du à l'absence de marquage au sol. Une méthode alternative consiste à exploiter les trajectoires des objets de la séquence. En supposant que les véhicules circulent au centre des voies de circulation et en considérant que la largeur des voies est identique, les délimitations sont estimées à partir du centre des voies. La carte des trajectoires est passée en entrée de l'algorithme et la transformée de Hough est directement appliquée sur cette carte. Ceci permet d'extraire le centre des voies à l'aide des trajectoires, les délimitations des voies sont ensuite estimées en considérant que leur largeur dans l'image reste constante.

FIGURE 3.10: Utilisation des trajectoires pour l'extraction des bordures des voies. **A gauche** - Trajectoires (couleurs aléatoires), **Au centre** - Trajectoires et approximation par une droite (en bleu), **A droite** - Estimation des bordures des voies (en rouge).

L'utilisation des trajectoires permet alors d'estimer les bordures des voies comme illustré sur la Figure 3.10, avec à gauche les trajectoires utilisées, au centre l'estimation des droites à l'aide de la transformée de Hough (à partir des coordonnées des points de trajectoires) et à droite l'estimation des bordures des voies en considérant une largeur constante entre les voies.

3.2.3 Estimation de la profondeur dans l'image

L'objectif de cette étape consiste à introduire une notion de profondeur dans l'image. Ceci va permettre de diviser chaque voie en sous-régions qui recouvrent théoriquement la même surface dans le monde réel. L'information de profondeur exploite la ligne d'horizon (définie par la position en y dans l'image) en utilisant l'approche proposée par C. Käs dans [Kas 2009].

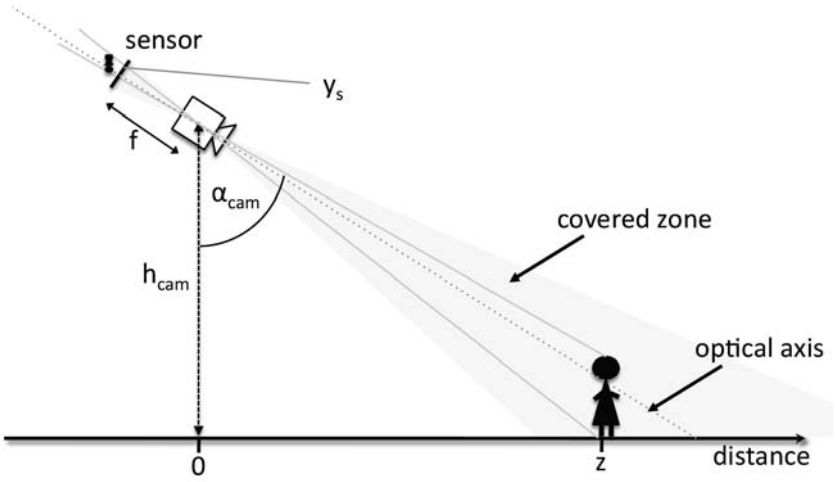


FIGURE 3.11: Configuration de la position de la caméra (modèle sténopé).

En supposant un modèle de caméra sténopé (Figure 3.11), la profondeur d'un point physique au sol z_{obj} par rapport à la caméra est directement liée à sa projection verticale sur l'image y_{obj} selon

$$z_{obj} = h_{cam}.\tan\left(\arctan\left(\frac{y_{obj} - d/2}{f}\right) + \alpha_{cam}\right) \tag{3.22}$$

avec d la dimension du capteur, f la focale de la caméra, h_{cam} la hauteur de la caméra et α_{cam} son orientation par rapport au sol supposé parfaitement horizontal. Un angle α_{cam} de $0°$ correspond à une vue du dessus tandis qu'un angle α_{cam} de $90°$ signifie que la caméra est parallèle au sol.

La première étape consiste à estimer l'orientation de la caméra. En supposant le sol horizontal, la position du point de fuite (z_{vp}) dans l'image correspond à une profondeur à l'infini $z_{vp} \to \infty$. Et puisque

$$\lim_{z\to\infty} \arctan(z/h_{cam}) = \frac{\pi}{2} \tag{3.23}$$

en inversant l'équation (3.22), on obtient :

$$\alpha_{cam} = \frac{\pi}{2} - \arctan\left(\frac{y_{vp} - d/2}{f}\right) \tag{3.24}$$

Puisque f et d sont inconnues, nous admettrons des valeurs standards pour une caméra classique 35mm avec $f = 35mm$ et $d = 24mm$. La hauteur de la caméra h_{cam} de l'équation (3.22) influe uniquement en tant que coefficient multiplicateur reliant la position au sol z_{obj} et la position sur le capteur y_{obj}, nous le fixons à une valeur standard pour les caméras utilisées à 7m.

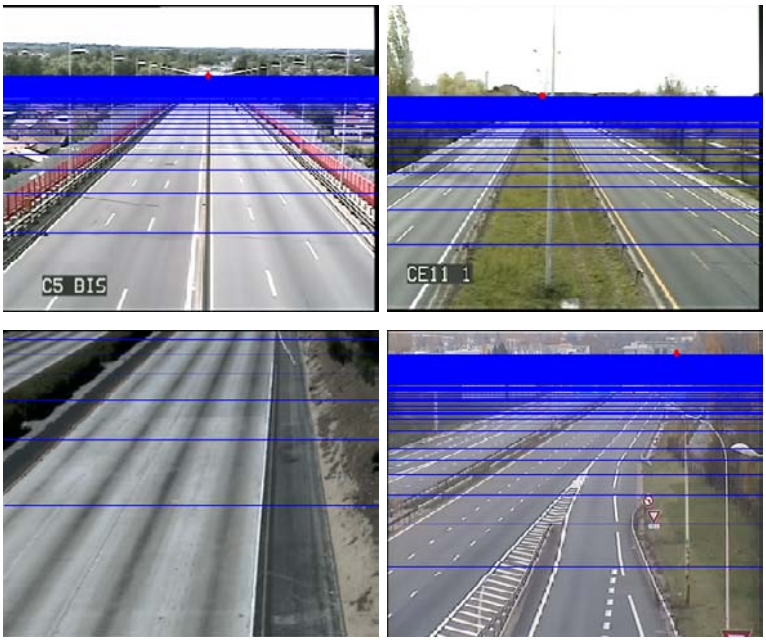


FIGURE 3.12: Estimation des lignes de profondeurs dans la scène.

Orientation	C5	CE 11	Highway	Lyon
α	79.74°	83.56°	58.43°	74.81°

TABLE 3.1: Estimation de l'orientation α des caméras pour les séquences C5, CE11, Highway et Lyon.

La Figure 3.12 montre les résultats de l'estimation de la profondeur dans l'image pour les quatre séquences de test. A partir de la position du point de fuite, les orientations des caméras (en supposant f, d et h fixés) sont dans un premier temps estimées (Tableau 3.1), puis utilisées pour déterminer les lignes de profondeur dans l'image (Equation 3.22). A partir des marquages au sol, on peut vérifier approximativement la validité des résultats.

3.2.4 Fusion des résultats et modèle final de la scène

A cette étape du traitement, nous avons à notre disposition d'une part des informations sur la structure de la scène, avec la connaissance de la position du point de fuite, des bordures des voies de circulation et une estimation approximative de la profondeur dans l'image, d'autre part des informations sur l'*apparence* de la scène, les objets et leurs comportements, avec une estimation de la couleur de l'arrière-plan de la scène, des trajectoires des objets et du sens de direction du trafic.

La fusion de l'ensemble de ces informations va permettre de construire un modèle de la scène sous surveillance. Le modèle consiste en un découpage spatial de la scène en cellules contenant des informations relatives aux comportements des objets. Cette fusion se déroule en deux étapes :

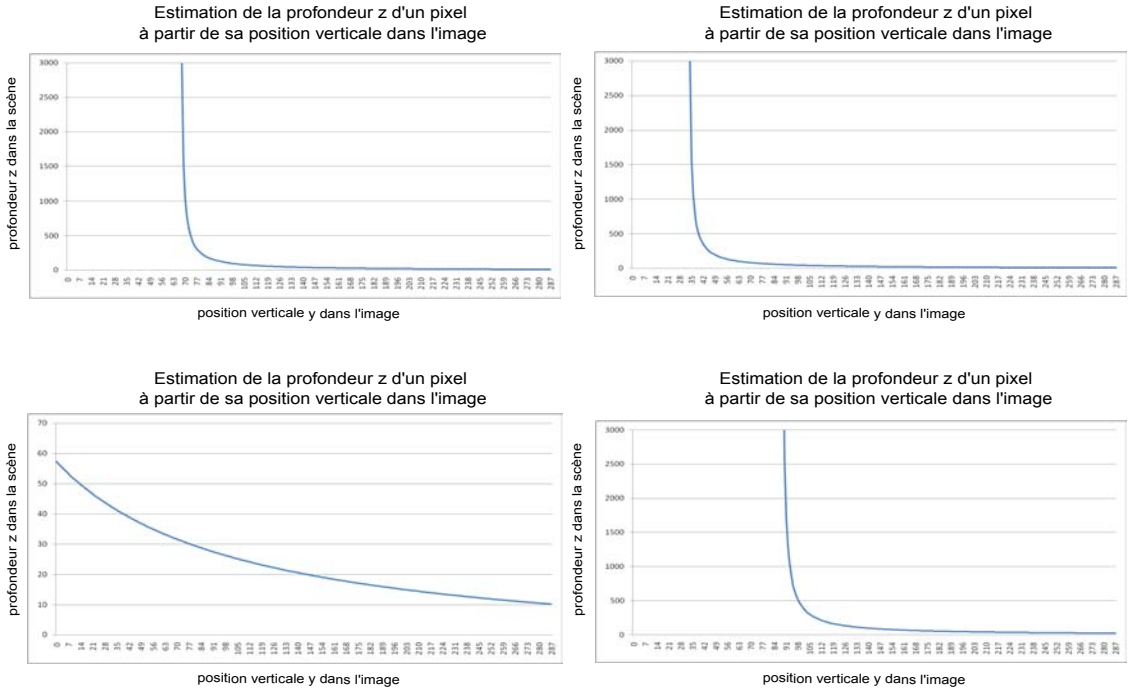


FIGURE 3.13: Tracés des profondeurs z dans la scène en fonction des positions verticales y dans l'image.

- Partitionnement spatial de la scène.
- Attribution des caractéristiques sémantiques des cellules.

Partitionnement spatial de la scène

Le partitionnement spatial consiste à effectuer un découpage de l'image à partir des informations sur la structure de la scène. La région active est définie comme étant l'ensemble des pixels contenus dans une voie de circulation. Cette zone active est découpée en cellules de tailles différentes sur l'image, mais recouvrant en théorie la même surface dans le monde réel. Ces cellules sont obtenues en partitionnant chaque voie de circulation à l'aide des lignes de profondeur comme illustré sur la Figure 3.14.

Ce découpage spatial permet de diviser la zone sous surveillance en sous-zones (ou cellules) qui théoriquement, couvrent la même surface dans la scène réelle. Toutes les informations relatives à la détection d'évènements (sens de circulation, zone d'entrée/sortie, ...) sont stockées à l'intérieur de chaque cellule.

Attribution des caractéristiques sémantiques des cellules

A chaque cellule est associé un ensemble de caractéristiques, contenant les informations telles que la vitesse moyenne (apparente) observée, la taille moyenne des véhicules, le sens de circulation, le type de zone (zone interdite, de circulation, d'entrée ou de sortie, ...).

Le type de zone est définie de la façon suivante. Par défaut, toute cellule contenue dans une voie de circulation est considérée comme étant une *zone de circulation*. Les cellules

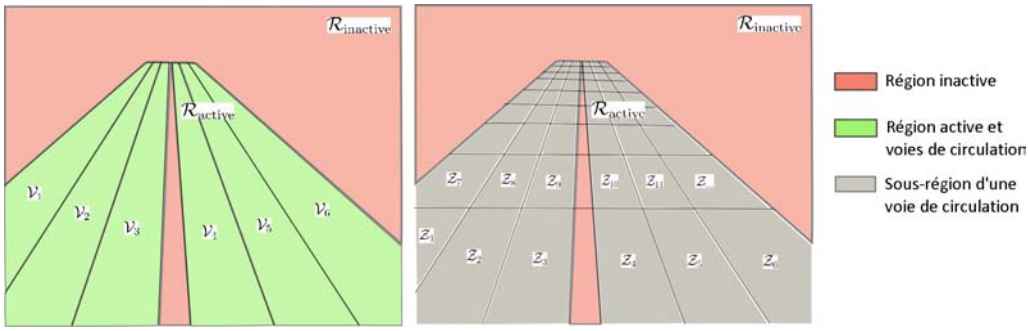


FIGURE 3.14: Illustration du modèle de la structure de la scène sous surveillance. L'image est découpée en sous-zones relatives aux voies de circulations auxquelles elles appartiennent et en fonction de la profondeur dans l'image.

positionnées aux extrémités des voies de circulation sont assignées en tant que *zone d'entrée* ou *zone de sortie* en fonction du sens de direction du trafic (une *zone d'entrée* correspond à la cellule à l'extrémité opposée au sens de direction).

Chaque cellule contient des statistiques sur la taille et la vitesse des objets. En supposant que la séquence d'apprentissage contient statistiquement plus de véhicules légers que de poids lourds ou de motos, les tailles et vitesses moyennes sont considérées comme étant caractéristiques des véhicules légers. Ces informations sont utilisées pour effectuer une classification approximative des objets circulant sur la route.

Enfin chaque cellule contient des informations relatives au sens de direction du trafic. Une moyenne spatiale de l'orientation de la première composante du modèle permet d'assigner pour chaque cellule la direction moyenne prise par les objets.

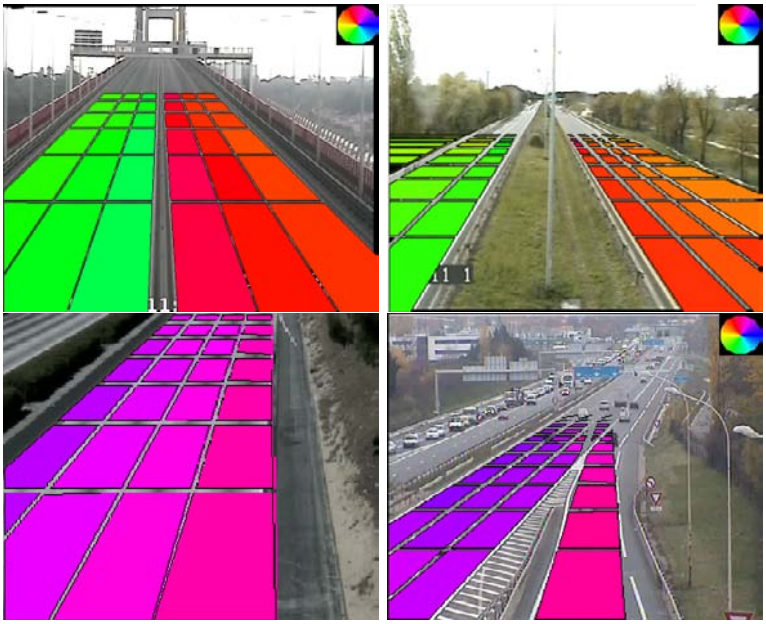


FIGURE 3.15: Résultats de la modélisation de la scène.

La Figure 3.15 montre les résultats de l'estimation et la construction du modèle de la scène. Cette estimation contient de nombreuses informations :

- Une estimation de l'arrière-plan, qui sera utilisée pour initialiser le modèle d'arrière-plan développé dans le Chapitre 4.
- Une estimation du sens de direction du trafic, qui sera utilisée pour détecter les évènements anormaux relatifs au modèle de mouvement appris.
- Une estimation de la structure de la scène, modélisée par un ensemble de cellules qui couvrent en théorie la même surface dans le monde réel.
- Une estimation des bordures des voies, qui sera utilisée pour aider à la segmentation des objets.
- Une estimation des zones d'entrée et de sortie, qui sera utilisée pour aider à la gestion du suivi des objets.

L'ensemble de ces informations a été correctement estimé pour les séquences de test. Notons cependant les limites de cette approche, qui nécessite des trajectoires rectilignes et un marquage au sol bien visible.

3.3 Conclusion

Nous avons vu dans ce chapitre la construction d'un modèle relatif à la structure de la scène. Un processus complet d'initialisation est proposée dans lequel une séquence d'apprentissage est analysée. Cette analyse a permi de segmenter les voies de circulation et d'estimer approximativement la profondeur dans l'image. Les délimitations des voies et les lignes de profondeur sont fusionnées pour partitionner la scène sous surveillance en cellules élémentaires caractérisant la même surface dans le monde réel. En parallèle, un modèle statistique des orientations des vecteurs mouvements est proposé pour caractériser le sens de direction du trafic.

Chapitre 4

Analyse spatio-temporelle : détection de mouvement

Sommaire

Introduction

La détection des régions en mouvement est une étape souvent essentielle dans les systèmes de vidéo-surveillance. Une segmentation robuste et précise simplifie le traitement des étapes suivantes de l'analyse. Obtenir une segmentation précise et peu coûteuse en temps de calcul est un problème ouvert et difficile, dû aux perturbations et à la dynamique des scènes extérieures tels que les changements de luminosité (éclairages, ombres, reflets), la présence d'arrière-plan dynamique (mouvements de branches d'arbres, panneaux à messages variables), les occlusions, ...

Ce chapitre présente le module de détection de mouvement utilisé par notre système, dont l'objectif est de fournir une carte binaire des objets en mouvement, notée $\mathcal{F}$. Notre approche consiste en une soustraction d'arrière-plan basée sur un modèle statistique de la caractéristique couleur des pixels. Les informations de contour et de mouvement sont utilisées en collaboration avec la soustraction d'arrière-plan pour rendre plus robuste la segmentation face aux changements de luminosité. Nous commençons ce chapitre en présentant la modélisation statistique utilisée et l'estimation des paramètres pour l'apprentissage du modèle (Section 4.1). Nous verrons dans la Section 4.2 l'application du modèle pour la soustraction d'arrière-plan et la classification des pixels. Dans la dernière section de ce chapitre (Section 4.3) seront présentés les résultats expérimentaux conduits sur les séquences de test.

4.1 Modélisation statistique des données

La détection de mouvement peut être vu comme un problème de classification des pixels en deux classes : les pixels d'arrière-plan et les pixels d'avant-plan. Pour effectuer une telle classification, un modèle statistique est utilisé pour caractériser la répartition des couleurs des pixels. D'un point de vu statistique, la valeur d'un pixel à un instant t est considérée comme étant un échantillon (ou une observation) issu d'un vecteur aléatoire caractérisant la présence de différentes surfaces (objets ou arrière-plan) dans l'image. En statistique, une variable aléatoire est entièrement décrite par sa densité de probabilité (graphiquement représentée par les valeurs possibles *vs* leurs fréquences relatives d'apparition).

Les techniques d'estimation de densité de probabilité peuvent être rangées en deux catégories : les méthodes *non paramétriques* et les méthodes *paramétriques*. Les méthodes *non paramétriques* n'imposent aucune hypothèse sur les données. Parmi ces méthodes on retrouve l'estimation à l'aide d'un histogramme des fréquences ou encore les estimateurs à noyaux (généralement gaussiens). La distribution est souvent représentée par les données elles-mêmes (ou un ensemble d'échantillons des données) et il est difficile d'en obtenir une représentation compacte. Quant aux méthodes *paramétriques*, elles supposent une forme prédéfinie de la distribution à l'aide d'une loi paramétrée. Ces méthodes fournissent généralement une forme compacte de la représentation des données puisque seuls les paramètres du modèle suffisent à le décrire entièrement.

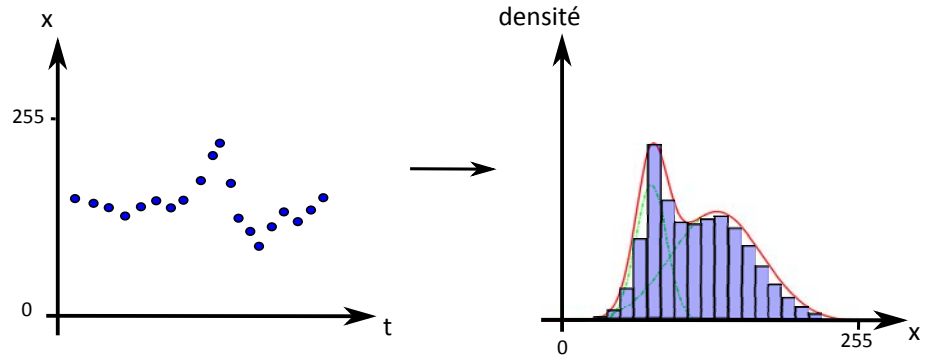


FIGURE 4.1: Illustration d'une modélisation d'un ensemble de données (à gauche) à l'aide de son histogramme (à droite en bleu) ou de deux gaussiennes (gaussiennes en vert à droite et combinaison linéaire en rouge)

Nous avons fait le choix d'une modélisation *paramétrique*, dans laquelle on associe à chaque pixel de l'image une densité de probabilité, notée p. L'utilisation d'un modèle paramétrique permet de n'avoir à stocker que les paramètres utilisés dans le modèle. Ces paramètres sont appris lors qu'une étape d'apprentissage non supervisée, i.e. aucune cible (ou classe) explicite n'est considérée dans la série de données d'entrainement. Il s'agit d'un problème d'estimation de densité dont l'objectif est d'obtenir une bonne représentation statistique des données (Figure 4.1). Dans le paragraphe suivant, nous rappelons le formalisme mathématique utilisé pour la modélisation.

4.1.1 Modèle de mélange de lois de probabilité

On considère un ensemble de données observées $\mathcal{X} = x_1, x_2, \ldots, x_n$ issues d'une variable aléatoire $\mathbf{x}$ décrite par une densité de probabilité notée p inconnue et que l'on cherche à estimer. On suppose que cette densité de probabilité p s'écrit comme une combinaison linéaire d'un nombre fini K de lois de probabilités paramétriques notées p_k et de paramètre ϕ_k. Sous cette forme, la densité p est appelée mélange de lois de probabilités et s'écrit

$$p(\mathbf{x}|\Phi) = \sum_{k=1}^{K} w_k p_k(\mathbf{x}|\phi_k), \qquad \text{avec} \sum_{k=1}^{K} w_k = 1 \tag{4.1}$$

Les lois p_k sont appelées les composantes du mélange, ϕ_k représente le vecteur des paramètres de la composante k et w_k les poids du mélange accordés aux k composantes tels que leur somme soit unitaire. Le vecteur paramètre du mélange à estimer est noté Φ et regroupe l'ensemble des poids w_k et des vecteurs paramètres ϕ_k de chaque composante tel que $\Phi = \{w_1, \ldots, w_K, \phi_1, \ldots, \phi_K\}$.

L'Equation 4.1 représente un modèle statistique à part entière et il est possible de générer de nouvelles données de la façon suivante. Dans un premier temps une distribution est choisie avec un probabilité donnée par les poids de mélange, puis la nouvelle valeur est générée selon la densité de la composante correspondante. Mathématiquement, en introduisant une variable aléatoire $\mathbf{z}$ qui exprime de quelle composante est issu l'échantillon,

ceci se traduit par

$$\begin{aligned} \mathbf{z} &\sim \text{Mult}\,(w_1, w_2, \ldots, w_K) \\ \mathbf{x}|\mathbf{z} &\sim p_k \end{aligned} \tag{4.2}$$

et l'équation 4.1 peut s'écrire

$$p(\mathbf{x}, \mathbf{z}|\Phi) = \sum_{k=1}^{K} p(\mathbf{z}|\mathbf{x}, \Phi) p_k(\mathbf{x}|\phi_k) \tag{4.3}$$

L'estimation des paramètres Φ du modèle permet de déterminer la densité p dont sont issues les observations dont nous disposons. Généralement, cette estimation est obtenue par la méthode du maximum de vraisemblance. Par définition, la fonction de vraisemblance ℓ traduit la probabilité d'observer le jeu d'échantillon $\mathcal{X}$ à partir de la densité p. Mathématiquement, elle s'écrit comme une fonction du paramètre Φ de la densité p, telle que

$$\ell(\Phi) = \log p(\mathbf{x}|\Phi) = \log \sum_{k=1}^{K} w_k p_k(x_i|\phi_k) \tag{4.4}$$

En supposant l'ensemble des observations $\mathcal{X} = x_1, x_2, \ldots, x_n$ indépendantes et identiquement distribuées (*iid*), la vraisemblance aux données $\ell(\Phi|\mathcal{X})$ s'écrit

$$\ell(\Phi|\mathcal{X}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} w_k p_k(x|\phi_k) \tag{4.5}$$

Maximiser la vraisemblance ℓ revient par sa définition à déterminer les paramètres optimaux $\widehat{\Phi}$ donnant la densité $\widehat{p}$ la plus proche de la densité inconnue p décrivant $\mathcal{X}$. Le vecteur paramètre optimal s'écrit

$$\widehat{\Phi} = argmax_{\Phi}\left(\ell(\Phi|\mathcal{X})\right) \tag{4.6}$$

Généralement, les paramètres recherchés sont obtenus en dérivant l'Equation 4.5 par rapport à chacun des paramètres. Dans le cas de lois *usuelles*, une solution analytique s'obtient après dérivation. Il est cependant difficile d'obtenir une forme analytique dans le cas d'un mélange de loi. En effet, en écrivant la dérivée de l'Equation 4.5 par rapport à un des paramètres ϕ_j on a

$$\begin{aligned} \frac{\partial \ell(\Phi|\mathcal{X})}{\partial \phi_j} &= \sum_{i=1}^{n} \frac{1}{\sum_{k=1}^{K} w_k p_k(x_i|\phi_k)} \frac{\partial p_j(x_i|\phi_j)}{\partial \phi_j} \\ &= \sum_{i=1}^{n} \frac{w_j p_j(x_i|\phi_j)}{\sum_{k=1}^{K} w_k p_k(x_i|\phi_k)} \frac{1}{p_j(x_i|\phi_j)} \frac{\partial p(x_i|\phi_j)}{\partial \phi_j} \\ &= \sum_{i=1}^{n} \frac{w_j p_j(x_i|\phi_j)}{\sum_{k=1}^{K} w_k p_k(x_i|\phi_k)} \frac{\partial \log p_j(x_i|\phi_j)}{\partial \phi_j} \\ &= \sum_{i=1}^{n} W_{i,j}(\Phi|x_i)\ \ell_p(\phi_j|x_i) \end{aligned} \tag{4.7}$$

L'Equation 4.7 possède une somme contenant un terme relatif aux composantes du mélange ℓ_p et un poids associé $W_{i,j}$. Dans le cas particulier où $K = 1$, le poids $W_{i,j} = 1$ et on est ramené à une estimation dans le cas d'une seule composante. Tandis que dans le cas d'une mélange de lois ($K > 1$), l'estimation des paramètres revient à maximiser une fonction de vraisemblance *pondérée* , dont les termes sont pondérés par un poids $W_{i,j}$ relatif aux échantillons x_i tels que

$$W_{i,j}(\Phi|x_i) = \frac{w_j p_j(x_i|\phi_j)}{\sum_{k=1}^{K} w_k p_k(x_i|\phi_k)} \equiv p(\mathbf{z} = j|\mathbf{x} = x_i, \Phi) \tag{4.8}$$

Estimer les paramètres d'un mélange revient à maximiser une fonction de vraisemblance *pondérée* (Equation 4.7), dont les poids correspondent aux probabilités conditionnelles de $\mathbf{z}$=$\mathbf{k}$ sachant x_i. Ces poids sont inconnus et dépendent eux-mêmes des paramètres que l'on cherche à estimer, ce qui rend l'estimation des paramètres à travers le maximum de vraisemblance difficile. L'estimation des paramètres d'un mélange nécessite donc une méthode d'optimisation, comme par exemple une descente de gradient sur la fonction de vraisemblance. Nous nous interressons dans la suite à une méthode alternative d'estimation du maximum de vraisemblance à l'aide de l'algorithme EM.

4.1.2 Estimation des paramètres à l'aide de l'algorithme EM

Principe de l'algorithme

L'algorithme EM (*Expectation-Maximization*) [Dempster 1977] est une méthode itérative d'optimisation permettant de maximiser la vraisemblance d'un jeu de données à un modèle en présence de données manquantes. Cet algorithme considère le jeu de données X comme étant incomplet et suppose l'existence d'une variable aléatoire cachée représentant un paramètre ou des valeurs d'échantillons inconnus. Ainsi, le jeu complet de données est défini par $\mathcal{X}_c = \{\mathcal{X}, \mathcal{Z}\}$, avec $\mathcal{X}$ et $\mathcal{Z}$ respectivement l'ensemble des données observées et l'ensemble des données cachées. On note alors $\mathbf{x}_c$ et $\mathbf{z}$, les variables aléatoires dont sont issues respectivement les données $\mathcal{X}_c$ et $\mathcal{Z}$. En introduisant la variable cachée $\mathbf{z}$ dans la fonction de log-vraisemblance ℓ, on peut écrire

$$\ell(\Phi|\mathcal{X}) = \log p(\mathbf{x}|\Phi) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\Phi) \tag{4.9}$$

où le terme de droite indique simplement que la log vraisemblance s'exprime en fonction de la variable cachée, sommés sur l'ensemble des données cachées (marginalisation de la densité jointe) et suivant un modèle statistique de la forme $p(\mathbf{x}, \mathbf{z}|\Phi)$. Notons que nous ne connaissons ni les paramètres du modèle Φ, ni les valeurs des données cachées $\mathbf{z}$.

Mathématiquement, la clé de l'algorithme EM consiste à maximiser la log vraisemblance à l'aide d'une borne inférieure qui s'en rapproche par itération. La convergence de l'algorithme vers un maximum local est assurée par la concavité de la fonction logarithme connue sous le nom d'inégalité de Jensen (la moyenne des logarithmes est inférieure au logarithme de la moyenne).

Notons $q(\mathbf{z})$ une distribution arbitraire sur les données cachées $\mathbf{z}$, la log vraisemblance

des données s'écrit

$$\ell(\Phi|\mathcal{X}) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\Phi) = \log \sum_{\mathbf{z}} q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z}|\Phi)}{q(\mathbf{z})} \tag{4.10}$$

et l'inégalité de Jensen permet d'écrire

$$\begin{aligned} \log \sum_{\mathbf{z}} q(\mathbf{z}) \frac{p(\mathbf{x}, \mathbf{z}|\Phi)}{q(\mathbf{z})} &\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\Phi)}{q(\mathbf{z})} \\ &= \sum_{\mathbf{z}} q(\mathbf{z}) \log p(\mathbf{x}, \mathbf{z}|\Phi) + \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{1}{q(\mathbf{z})} \\ &\equiv J(q, \Phi) \end{aligned} \tag{4.11}$$

Ainsi, plutôt que de maximiser la vraisemblance $\ell(\Phi|\mathcal{X})$, l'algorithme EM maximise l'expression $J(q, \Phi)$ qui correspond à une borne inférieure de la fonction de vraisemblance ℓ. Puisque nous ne connaissons pas la distribution de la variable cachée , l'algorithme EM consiste dans un premier temps à déterminer la densité q qui maximise J avec les paramètres Φ fixés. Une fois estimée, la densité q est fixée et l'algorithme consiste à déterminer les paramètres Φ qui maximisent l'expression de J. Le calcul du maximum de l'étape d'*Expectation* est obtenu lorsque $q(\mathbf{z})^{t+1} = p(\mathbf{z}|\mathbf{x}, \Phi^t)$. Lorsque cette égalité est atteinte, la fonction $J(q, \Phi) = \ell(\Phi|\mathcal{X})$. Quant à l'étape de *Maximization*, elle consiste à maximiser le premier terme de $J(q, \Phi)$ (puisque le second ne dépend pas des paramètres Φ) et peut s'écrire

$$\Phi^{t+1} = \arg\max_{\Phi} \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \Phi^t) \log p(\mathbf{x}, \mathbf{z}|\Phi^t) \tag{4.12}$$

En utilisant cette formulation, l'algorithme EM se traduit par les étapes suivantes

Algorithme 2: Algorithme EM

1. Initialisation de tous les paramètres du mélange $\Phi^0 = \{w_1, \ldots, w_K, \theta_1, \ldots, \theta_K\}$.
2. Répéter jusqu'à convergence
   (a) *Expectation* : $q^t = \arg\max_q J(q, \Phi^t)$
   (b) *Maximization* : $\Phi^{t+1} = \arg\max_\Phi J(q^t, \Phi)$
3. Renvoyer l'estimation finale Φ et q

Application à un mélange de lois

Dans le cadre d'une classification statistique à l'aide d'un mélange de lois, la variable cachée $\mathbf{z}$ est définie comme étant la variable désignant la composante k du mélange ayant générée la donnée $\mathbf{x}$. en supposant qu'une seule composante k ne peut générer une donnée x_i, l'expression de la log vraisemblance de l'Equation 4.5 devient

$$\ell(\Phi|\mathcal{X}, \mathcal{Z}) = \sum_{i=1}^{n} \log w_{k_i} p_{k_i}(x_i|\phi_{k_i}) \tag{4.13}$$

On cherche dans un premier temps à exprimer la densité conditionnelle de la variable cachée $\mathbf{z}$ connaissant les données observées $\mathbf{x}$ et le jeu de paramètres Φ^{t-1}. Son expression est donnée par la formule de Bayes et constitue l'étape d'*Expectation*

$$p(\mathbf{z} = k|\mathbf{x}, \Phi^t) = \frac{w_k p_k(\mathbf{x}|\phi_k^t)}{\sum_{j=1}^{K} w_j p_j(\mathbf{x}|\phi_j^t)} \tag{4.14}$$

Quant à l'étape de *Maximization*, elle consiste à différentier la quantité J par rapport aux paramètres du mélange et de les mettre à zéro. Notons qu'il n'est pas nécessaire de spécifier la forme paramétrique des distributions p pour déterminer les poids w. Ces deux étapes sont répétées jusqu'à convergence et chaque itération garantie une convergence vers un maximum local de la fonction de log-vraisemblance complétée [Krishnan 1997]. Cette approche peut être utilisée sur un jeu de données scalaires, mais également dans le cas d'une description de données vectorielles : si x est un vecteur de dimension d, alors la densité $p_k(x)$ est une distribution de dimension d. Pour une description plus approfondie de la méthode, le lecteur intéressé pourra se référer à [McLachlan 2008].

4.1.3 Mélange de lois gaussiennes

La distribution normale (ou gaussienne) est une des principales distributions de probabilité dû à plusieurs facteurs : non seulement il s'agit d'une formulation mathématiquement élégante contenant peu de paramètres (moyenne et matrice de covariance) mais de plus, cette distribution apparaît naturellement en pratique dans de nombreuses situations réelles. On peut montrer que la distribution de la moyenne d'un ensemble suffisamment grand ($N > 30$) de variables aléatoires x indépendantes et identiquement distribuées suivant une loi quelconque d'espérance $\mathbb{E}[x]$ et de variance $\mathbb{V}[x]$ converge vers une loi de distribution normale. Cette affirmation est connue en mathématiques sous le nom de théorème central limite et montre le caractère universel et l'importance des lois gaussiennes. En traitement du signal, cette distribution trouve un intérêt particulier pour modéliser un bruit additif de mesure d'une observation. La densité de probabilité d'une distribution gaussienne s'écrit

$$p_k(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma_k^{-1}(\mathbf{x}-\mu_k)} \tag{4.15}$$

avec μ_k le vecteur moyenne, Σ_k la matrice de covariance et d la dimension d'un vecteur aléatoire x. La notation $p(|)$ fait référence à une probabilité conditionnelle et $|.|$ fait réference au déterminant d'une matrice. L' expression analytique de l'estimation des paramètres en appliquant l'algorithme EM dans le cas d'un mélange gaussien [Bilmes 1998] est donnéee par

Expectation :

$$p(\mathbf{z} = k|x_i, \Phi^t) = \frac{w_k p(x_i|\mu_k^t, \Sigma_k^t)}{\sum_{l=1}^{K} w_l p(x_i|\mu_l^t, \Sigma_l^t)} \tag{4.16}$$

Maximization :

$$\begin{cases} w_k^{t+1} = \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t) \\ \mu_k^{t+1} = \dfrac{\displaystyle\sum_{i=1}^{N} x_i p(\mathbf{z}=k|x_i,\Phi^t)}{\displaystyle\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)} \\ \Sigma_k^{t+1} = \dfrac{\displaystyle\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)(x_i-\mu_k^{t+1})(x_i-\mu_k^{t+1})^T}{\displaystyle\sum_{i=1}^{N} p(\mathbf{z}=k|x_i,\Phi^t)} \end{cases} \tag{4.17}$$

Ces équations permettent d'estimer de façon récursive les paramètres d'un mélange de distribution gaussiennes.

4.2 Application à la détection de mouvement

Le modèle de mélange présenté dans la section précédente est sans doute un des modèles les plus utilisés pour modéliser la couleur de l'arrière-plan dans les images [Bouwmans 2008]. Il s'agit d'un processus appliqué pour chaque pixel de l'image. Dans une séquence vidéo, chaque pixel est décrit par une distribution p caractérisant la répartition des valeurs de son historique récent. De plus, chaque pixel est considéré comme étant une variable aléatoire $\mathbf{x}$ caractérisée par sa densité p. Les différentes valeurs prises par $\mathbf{x}$ sont caractéristiques des différentes surfaces des objets de la scène. Les surfaces caractéristiques de l'arrière-plan sont supposées majoritairement représentées par rapport aux surfaces relatives aux objets. En indéxant chacune des surfaces à l'aide de l'indice $k \in [1,\ldots,K]$, et en supposant que ces surfaces sont décrites par des distributions gaussiennes, alors ces dernières correspondent aux composantes p_k du mélange gaussien. La densité de probabilité de $\mathbf{x}$ est dans ce cas entièrement décrit par un mélange de K lois gaussiennes supposées indépendantes, et dont chacune des composantes est représentative des différentes surfaces des objets ou de la structure de la scène.

Nous présentons dans cette section l'approche utilisée par notre système, qui consiste en une modélisation statistique de la couleur, et d'un enrichissement de cette information à l'aide des vecteurs mouvement issus du flot optique et de la différence de gradient pour aider à la segmentation. La détection des régions en mouvement est ainsi principalement basée sur la segmentation couleur. La caractéristique de contour est utilisée pour valider (ou invalider) le résultat de la segmentation couleur. Les caractéristiques mouvements issues du flot optique sont, quant à elles, initialement prévues pour la détection de contre-sens. Nous nous sommes finalement intéressés à combiner cette information pour valider les régions en mouvement à la sortie du module.

4.2.1 Modélisation de la couleur

La modélisation de la couleur présentée ici est basée sur les travaux de [Stauffer 1999]. L'approche consiste à modéliser la répartition des couleurs pour chaque pixel à l'aide d'un mélange de lois gaussiennes. Ainsi, chaque pixel de l'image est caractérisé par un vecteur couleur $x = \{r, g, b\}$ dans l'espace RGB et on définit la probabilité d'observer la valeur du vecteur x à l'instant t comme étant

$$P(x_t|\Phi) = \sum_{k=1}^{K} w_{k,t}.\mathcal{N}(x_t|\mu_{k,t}, \Sigma_{k,t})$$

avec K le nombre de distributions du modèle (généralement entre 3 et 5), w_k le poids accordé à la gaussienne k de moyenne $\mu_{k,t}$ et de covariance $\Sigma_{k,t}$. La densité $\mathcal{N}$ est une distribution gaussienne qui s'écrit :

$$\mathcal{N}(x_t|\mu, \Sigma) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x_t-\mu)^T\Sigma_k^{-1}(x_t-\mu)}$$

Pour réduire la complexité calculatoire, Stauffer et Grimson [Stauffer 1999] considèrent les composantes couleurs comme étant indépendantes et de variance identique σ_k telle que

$$\Sigma_{k,t} = \sigma_{k,t}.Id$$

Ainsi, chaque pixel est caractérisé par un mélange de K distributions gaussiennes. Ce modèle est entièrement décrit par son vecteur paramètres $\Phi_t = [w_1, \ldots, w_K, \mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K]$, qui est inconnu et que l'on cherche à estimer.

Classification de Stauffer et Grimson [Stauffer 1999]

Une comparaison au modèle est effectuée pour chaque nouvelle image de la vidéo pendant la période d'apprentissage. Dans un premier temps, les K distributions sont classées en utilisant comme critère le ratio poids-variance $w_{k,t}/\sigma_{k,t}$. Ce tri permet de mettre en avant les distributions ayant une forte probabilité d'occurrence (fort poids) et une faible variabilité dans sa construction (faible variance). Ceci suppose que la couleur d'arrière-plan est plus souvent présente que celle d'un objet et que sa valeur est relativement stable. Une fois triées, les B premières distributions sont considérées comme représentatives de l'arrière-plan, tel que

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{i=0}^{b} w_{i,t} > T \right) \tag{4.18}$$

avec T un seuil défini empiriquement. Les autres distributions sont considérées comme représentatives d'objets en mouvement. Pour chaque pixel d'une nouvelle image, un test de similarité entre le pixel et le modèle est effectué en calculant la distance de Mahalanobis pour toutes les gaussiennes du modèle, définie par :

$$d\left(I_{s,t}\right) = \sqrt{\left(I_{s,t} - \mu_{i,s,t}\right)^T \Sigma_{i,s,t}^{-1} \left(I_{s,t} - \mu_{i,s,t}\right)} \tag{4.19}$$

Maintenance du modèle

L'estimation des paramètres d'un mélange de distributions gaussiennes à l'aide de l'algorithme EM est présentée dans la section précédente (Equations 4.17). Dans le cadre d'une application en temps réel de surveillance d'une scène extérieure, le modèle d'arrière-plan doit être mis à jour pour prendre en compte les changements de luminosité. Il est donc nécessaire, régulièrement, de ré-estimer le vecteur paramètre Φ_i. Lorsque le nombre d'échantillons N devient grand, les Equations 4.17 deviennent inadaptées et nécessitent le stockage de l'ensemble des échantillons. Dans leurs travaux, Stauffer et Grimson proposent une version récursive de la mise à jour des paramètres, obtenue en évaluant les expressions pour chaque paramètre en $N+1$. La mise à jour des paramètres est effectuée à l'aide des équations suivantes :

$$\begin{cases} w_k^{N+1} &= (1-\alpha)w_k^N + \alpha p(k|x_{N+1}, \Phi^{t-1}) \\ \mu_k^{N+1} &= (1-\rho_k)\mu_k^N + \rho_k x_{N+1} \\ \Sigma_k^{N+1} &= (1-\rho_k)\Sigma_k^N + \rho_k (x_{N+1} - \mu_k^{N+1})(x_{N+1} - \mu_k^{N+1})^T \end{cases} \tag{4.20}$$

avec

$$\begin{aligned} \alpha &= \frac{1}{(N+1)} \\ \rho_k &= \frac{p(k|x_{N+1}, \Phi^{t-1})}{(N+1)w_k^{N+1}}. \end{aligned} \tag{4.21}$$

Le paramètre de mise à jour α est appelé *facteur d'oubli* (ou *taux d'apprentissage*) et permet d'accorder moins d'importance aux anciennes observations dans la mise à jour.

La modélisation de l'arrière-plan par cette approche fournit une bonne précision des résultats au détriment d'une complexité calculatoire. Dans [Zivkovic 2004], les auteurs proposent une version améliorée permettant de réduire la complexité calculatoire et la mémoire utilisée. Le nombre de gaussiennes pour chaque pixel est régulièrement mis à jour à l'aide d'une formulation bayésienne, se traduisant par l'ajout d'un terme dans l'équation de mise à jour du poids des gaussiennes. Ce terme conduit à la possibilité d'obtenir un poids négatif, signifiant que la gaussienne considérée n'est plus représentative de l'arrière-plan et peut être supprimée. Ainsi, l'équation de mise à jour utilisée s'écrit

$$w_k^{N+1} = (1-\alpha)w_k^N + \alpha p(k|x_{N+1}, \Phi^{t-1}) - \alpha.c_T \tag{4.22}$$

avec c_T un paramètre constant appelé *complexity prior* et fixée à 0.01. Lorsque le poids d'une gaussienne devient négative, elle est supprimée du modèle.

Initialisation du modèle

Le modèle est initialisé pendant la période d'apprentissage (Section 3.1.1), durant laquelle la couleur pour chaque pixel de l'arrière-plan est caractérisée par une gaussienne. Initialement, une seule composante pour chaque pixel compose le modèle de mélange (K=1). Leur moyenne et variance sont issues des paramètres des gaussiennes de la période d'apprentissage. Rappelons que le nombre de gaussiennes est régulièrement mis à jour à l'aide de l'introduction d'un terme dans l'équation de mise à jour des poids des gaussiennes (voir section précédente).

Algorithme 3: Mise à jour des paramètres Φ du modèle couleur d'un pixel

Input : Vecteur caractéristique x_t (nouvelle observation)
Data : Vecteur des paramètres du modèle Φ_{t-1}, masque foreground $\mathcal{F}_t$
Output : Version mise à jour du vecteur des paramètres Φ_t

---

for each *pixel* x_t **do**
  if $\mathcal{F}_t \neq 0$ **then**
    c=0, match=0
    { Récupération de la composante la plus proche }
    for k=*1 to* K **do**
      $d_k^2 = \sum_{d=1}^{D} \frac{(x_{t,d}-\mu_{k,d})^2}{\sigma_{k,d}^2}$
      if $d_k^2 \leq \lambda$ **then**
        if *match=0* **then**
          m=c
        else if $\frac{w_k}{\sigma_k^2} \geq \frac{w_m}{\sigma_m^2}$ **then**
          m=c
        match=1

    if *match=0* **then**
      { Remplacement de la dernière composante par une nouvelle }
      $j = \arg\min_k d_k^2$
      $w_j = \alpha$, $\mu_j = x_t$, $\Sigma = \sigma^2$ Id
    else
      { Mise à jour de tous les poids et du nombre de composantes}
      for k=*1 to* K **do**
        $w_k = (1-\alpha)w_k - \alpha c_T$
        if $w_k < 0$ **then**
          $w_j = 0$, $\mu_j = 0$, $\Sigma = 0$ Id

      { Mise à jour de la composante génératrice }
      $w_m = w_m + \alpha$
      $\mu_m = (1 - \frac{\alpha}{w_m})\mu_m + \frac{\alpha}{w_m} x$
      $\sigma_m^2 = \sigma_m^2 + \frac{\alpha}{w_m}((\mu_m - x)^2 - \sigma_m^2)$
      $\sigma_m^2 = \max(\min(\sigma_m^2, \sigma_{max}^2), \sigma_{min}^2)$

    { Tri et sélection des B distributions les plus représentatives }
    Supprimer les composantes telles que $w_k < 0$
    Trier les paramètres $\{w_k, \mu_k, \sigma_k^2\}$ *en fonction du rapport* w_k/σ_k^2
    $B = argmin_k(\sum_{i=1}^{k} w_i > T)$

4.2.2 Détection des ombres portées

Les ombres peuvent être classées en deux catégories : les ombres propres (*self-shadow*) et les ombres portées (*cast-shadow*). Les ombres portées provoquent régulièrement des fausses

détections dans les algorithmes de segmentation d'objets, et génèrent des fusions d'objets et la distorsion de leurs formes. L'utilisation d'un traitement particulier pour la détection d'ombre permet d'augmenter la qualité de la segmentation. Nous ne nous intéressons qu'aux ombres projetées par les objets vidéo. Celles incorporées dans l'arrière-plan ne sont pas analysées et sont considérées quasi-constantes dans le temps. L'algorithme de suppression d'ombre ne traite que les pixels appartenant au masque *foreground*. L'approche proposée est basée sur les travaux de [Horprasert 1999] et consiste en une analyse couleur du pixel par rapport à celle de l'arrière-plan. Cette analyse exploite la propriété suivante : la présence d'une ombre assombrit la surface sur laquelle elle est projetée, tout en conservant les mêmes propriétés colorimétriques. Une mesure de distorsion chromatique et de luminosité basée sur les travaux de [Horprasert 1999] est définie pour permettre la segmentation.

Les caractéristiques couleur d'une ombre portée peuvent être observées sur la Figure 4.2, dans laquelle la répartition des valeurs de l'image est reportée dans l'espace RGB. L'ensemble des points représentés semblent alignés selon une droite passant par l'origine du repère. Partant de cette constatation, l'approche proposée dans [Horprasert 1999] définit la ligne chromatique, la ligne passant par l'origine O du repère et le point B d'un pixel non exposé à une ombre. On considère alors l'hypothèse suivante : un pixel d'ombre a tendance à suivre la ligne chromatique dans l'espace RGB.

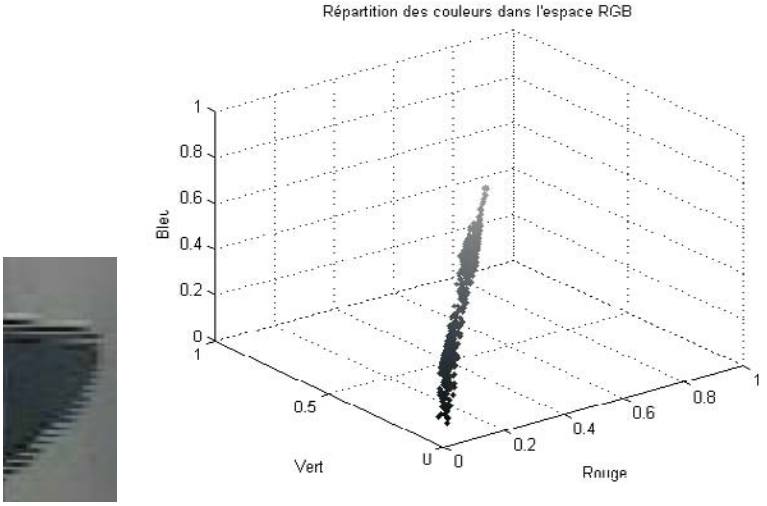


FIGURE 4.2: Répartition de la couleur dans l'espace RGB.

La classification consiste en une mesure de similarité entre une observation (que l'on cherche à classifier) et une valeur de référence (supposée sans ombre) obtenue généralement à l'aide d'une estimation de l'arrière-plan. Ainsi, la distance entre l'observation I et la valeur de référence B est décomposée en deux parties : une distorsion chromatique et une distorsion de luminosité (voir Figure 4.3).

La distorsion chromatique CD est définie comme étant la distance orthogonale entre la couleur de référence B et la couleur observée I. Il s'agit de la plus courte distance entre

l'observation I et la ligne chromatique, donnée par

$$c_s = \sqrt{\left(\frac{I_R - \alpha_s \mu_R}{\sigma_R}\right)^2 \left(\frac{I_G - \alpha_s \mu_G}{\sigma_G}\right)^2 \left(\frac{I_B - \alpha_s \mu_B}{\sigma_B}\right)^2} \tag{4.23}$$

La distorsion de luminosité est définie comme représentant la déviation entre le point B et le point projeté sur la ligne chromatique (Figure 4.3). Il s'agit d'une valeur indicatrice de la déviation

$$\alpha_s = \frac{\left(\frac{I_R \cdot \mu_R}{\sigma_R^2} + \frac{I_G \cdot \mu_G}{\sigma_G^2} + \frac{I_B \cdot \mu_B}{\sigma_B^2}\right)}{\left(\left[\frac{\mu_R}{\sigma_R}\right]^2 + \left[\frac{\mu_G}{\sigma_G}\right]^2 + \left[\frac{\mu_B}{\sigma_B}\right]^2\right)} \tag{4.24}$$

Si α_s est inférieur à 1, alors le pixel de l'image est plus sombre que celui de l'arrière-plan, tandis qu'une valeur supérieure à 1 indique que le pixel est plus clair que l'arrière-plan.

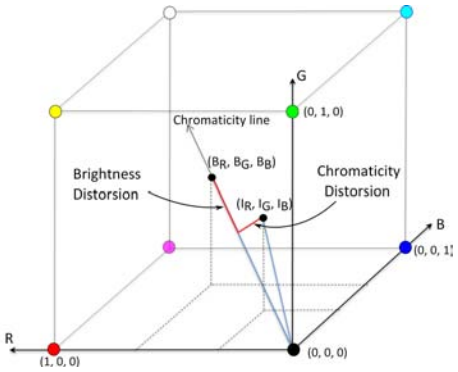


FIGURE 4.3: Représentation du modèle pour la détection d'ombre dans l'espace couleur RGB. La ligne chromatique est définie par la droite passant par l'origine du repère et le point issu de l'arrière-plan. Une mesure de distorsion chromatique et de distorsion de luminosité sont utilisées pour la classification d'un pixel en ombre ou en reflet.

En définissant un seuil sur la distorsion chromatique et deux seuils sur la distorsion de luminosité, les pixels d'ombres et de reflets sont estimés : un pixel est considéré comme étant une ombre s'il possède sensiblement la même couleur que l'arrière-plan (distorsion chromatique faible et inférieure à un seuil τ_{chrom}) mais une luminosité plus faible que l'arrière-plan (distorsion de luminosité inférieure à un seuil τ_{lum}^{low}). De façon similaire, un pixel est considéré comme étant un reflet s'il possède sensiblement la même couleur que l'arrière-plan (distorsion chromatique faible et inférieure à un seuil τ_{chrom}) mais une luminosité plus élevée que l'arrière-plan (distorsion de luminosité supérieure à un seuil τ_{lum}^{high}).

4.2.3 Estimation du flot optique

Le calcul du flot optique s'effectue au travers de l'estimation du déplacement des points d'intérêt dans l'image. Il est représenté sous la forme d'un champ de vecteurs contenant les normes et orientations des déplacements. La processus d'estimation de la méthode utilisée se décompose en deux étapes :

- Extraction des points caractéristiques.
- Estimation du vecteur déplacement associé.

Extraction de points caractéristiques

Il n'y a pas de définition universelle de ce que constitue un point caractéristique et sa définition exacte dépend du problème et du type d'application. Cependant, la majorité des détecteurs de points d'intérêt possède une approche commune, il s'agit d'analyser spatialement l'intensité lumineuse de l'entourage d'un pixel afin d'en extraire les caractéristiques et discontinuités dans de multiples directions. Parmi les méthodes d'extraction existantes, on retrouve les détecteurs de contours (Canny, Sobel), les détecteurs de coins (Morravec, Harris, Shi-Tomasi) ainsi que des détecteurs plus évolués (surf, fast) robustes aux changements d'échelle et d'intensité lumineuse. Nous utilisons dans cette étape le détecteur de coin proposé par Shi et Tomasi [Shi 1994], basé sur le détecteur de Harris qui fournit de très bons résultats sur les séquences utilisées. Le détecteur est appliqué dans les régions détectées par soustraction d'arrière-plan. Ceci permet de réduire le temps de calcul et de n'estimer le déplacement des points d'intérêt que pour les objets en mouvement. Le détecteur fournit en sortie un ensemble de n points caractéristiques.

Estimation du déplacement des points caractéristiques

Le suivi de points caractéristiques consiste à déterminer le déplacement pour chaque point considéré entre plusieurs images consécutives. Il s'agit d'une méthode de mise en correspondance permettant l'extraction d'un champ de vecteurs de déplacement dans lequel chaque vecteur est associé à un point caractéristique. Les stratégies de mise en correspondance sont nombreuses et dépendent du type de descripteur utilisé ainsi que de sa composition. Il s'agit généralement d'utiliser une métrique de distance entre descripteurs permettant d'associer deux points caractéristiques lorsque la distance est minimale.

Le KLT Tracker ([Tomasi 1991], [Shi 1994]) utilisé ici est une méthode différentielle qui minimise une fonction résiduelle basée sur la somme des différences quadratiques de l'intensité des pixels dans un voisinage du point considéré. La taille du voisinage a une influence directe sur la précision du résultat. Lorsque le voisinage est petit, la précision est augmentée puisque l'on ne considère que le voisinage proche du pixel.

Dans l'objectif de pouvoir traiter des déplacements plus grands, une version pyramidale est adoptée ([Bouguet 2001]). Le résultat de l'algorithme fournit un champ de vecteurs caractérisant le déplacement des objets dans la scène.

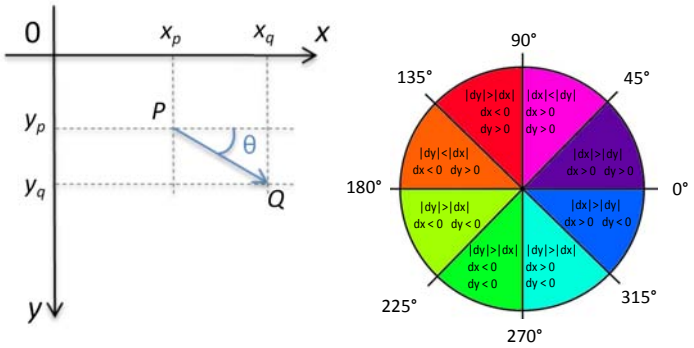


FIGURE 4.4: Représentation d'un vecteur déplacement 2D et estimation de l'orientation parmi huit orientations possibles selon les valeurs prises de $|d_x|$ et $|d_y|$.

La Figure 4.4 montre la représentation utilisée pour un vecteur de déplacement 2D issu de l'algorithme d'estimation de flot optique. En notant $d_x = x_q - x_p$ et $d_y = y_q - y_p$, la norme du vecteur $\vec{PQ}$, notée ρ est donnée par la distance euclidienne définie par

$$\rho_i = \sqrt{d_x^2 + d_y^2} \tag{4.25}$$

L'orientation du vecteur est obtenue à l'aide de la fonction arctan2, définie par

$$\theta = \arctan 2(y, x) = \begin{cases} \arctan(y/x) & \text{si} \quad x > 0 \\ \arctan(y/x) + \pi & \text{si} \quad y \geq 0, x < 0 \\ \arctan(y/x) - \pi & \text{si} \quad y < 0, x < 0 \\ \pi/2 & \text{si} \quad y > 0, x = 0 \\ -\pi/2 & \text{si} \quad y < 0, x = 0 \\ \text{indéfini} & \text{si} \quad y = 0, x = 0 \end{cases} \tag{4.26}$$

La Figure 4.5 illustre quelques exemples de résultats obtenus.

FIGURE 4.5: Exemple d'estimation du déplacement des vecteurs mouvements pour quelques séquences.

4.2.4 Différence temporelle de gradient

L'information de gradient est estimée en utilisant un masque de Sobel dans les directions x et y. La détection de mouvement basé sur le gradient consiste en une simple différence

des normes des gradients entre deux images. La segmentation des régions en mouvement s'effectue de la façon suivante. Soit x_t et x_{t-1} les valeurs en niveaux de gris d'un pixel à l'instant t et $t-1$, et $(g_x, g_y)_t$, $(g_x, g_y)_{t+1}$ respectivement les dérivées spatiales dans les directions x et y aux instant t et $t-1$. La norme pour chaque pixel est alors définie par $\rho_t = \sqrt{g_x^2 + g_y^2}$, et la variance globale moyenne pour toute l'image, notée $\overline{\sigma}$, est calculée. Si $\sqrt{(g_{x_t} - g_{x_{t-1}})^2} > 3\overline{\sigma}$, alors le pixel est considéré comme étant en mouvement.

4.2.5 Classification des pixels

La classification des pixels s'effectue en combinant les masques obtenus à l'aide des caractéristiques couleurs, de gradient et de flot optique. La différence de gradient est découpée en blocs de taille 4x4 et pour chaque bloc, si la somme des différences est non nulle, alors celui-ci est combiné au champ de vecteurs issu du flot optique. De cette façon, le masque mouvement est obtenu en effectuant une opération ET binaire entre le masque gradient et le champ de vecteurs issu du flot optique. Ce masque est combiné avec le masque *foreground* $\mathcal{F}$ obtenu par soustraction d'arrière-plan et filtré par l'opération de suppression d'ombres et de reflets. Ces opérations sont illustrées sur la Figure 4.6.

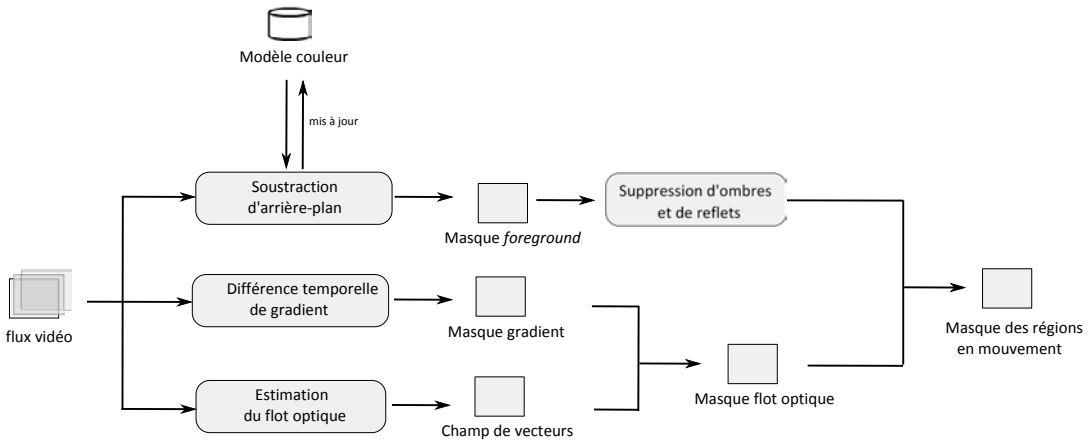


FIGURE 4.6: Classification des pixels en quatre classes possibles (*background*, *foreground*, *shadow*, *highlight*) à l'aide des caractéristiques couleur, de gradient et de flot optique.

4.3 Résultats expérimentaux

Les tests sont conduits sur un ensemble de vidéos de scènes autoroutières parmi celles présentées dans la Section 2.3.4. L'analyse des résultats porte sur environ 42 minutes de vidéo sur l'ensemble des séquences choisies. Plus de 350 images (prises en moyenne toutes les 8-10 secondes) ont été annotée manuellement pour fournir la vérité-terrain de chacune des séquences. Chaque objet de la vérité-terrain est représenté sous la forme d'une liste de points (polygone) délimitant l'objet d'intérêt et à l'aide de sa boite englobante. L'ensemble des polygones permettent la création d'un masque binaire $\mathcal{M}_{gt}$ qui sera comparé au masque des objets en mouvement ($\mathcal{F} = \mathcal{M}_{ar}$). L'imprécision de la sélection des objets par l'utilisateur introduit un biais dans l'évaluation des performances, particulièrement important lors d'une évaluation par pixel dans l'image. La comparaison de notre approche avec quelques

algorithmes existants permet néanmoins de situer nos résultats. Pour compléter l'analyse, une évaluation au niveau des objets est également proposée, permettant de n'évaluer que l'extraction finale des objets.

4.3.1 Métriques d'évaluation

Les performances de l'algorithme sont mesurées en termes de vrais positifs (TP, présence d'un objet correctement détecté), de faux positifs (FP, détection non présente dans la vérité-terrain) et de faux négatifs (FN, présence d'un objet non détecté). Nous différencions deux niveaux d'analyse (voir Section 2.3) : une analyse au niveau du pixel (chaque pixel est considéré indépendamment des autres dans l'image), et une analyse au niveau objet (les pixels sont regroupés et analysés en tant qu'objet). Cette dernière nécessite d'extraire du masque *foreground* les groupes de pixels en mouvement. Cette extraction s'effectue à l'aide d'une analyse en composantes connexes basée sur l'algorithme décrit dans [Suzuki 1985]. Pour chaque sortie d'algorithme, un filtre median spatial (taille 3x3) est appliqué au masque des objets en mouvement.

Les performances du système sont évaluées pour chaque image au niveau pixel à l'aide des mesures de précision et de rappel. Aucune considération sur l'objet ni sur le résultat de l'étiquetage n'est prise en compte et chaque pixel est considéré indépendamment des autres. Une mesure de F-score et de FAR (*False Alarm Rate*) sont ajoutés aux mesures de Précision et de Rappel.

$$\text{Det}_{pixel} = \begin{cases} \text{TPR} & = \dfrac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{FAR} & = \dfrac{\text{FP}}{\text{TP} + \text{FP}} \\ \text{F-score} & = \dfrac{2.\text{Precision}.\text{TPR}}{\text{Precision} + \text{TPR}} \end{cases} \tag{4.27}$$

4.3.2 Configuration

Pour cette analyse, les séquences de test utilisées sont C5, C21, Lyon et C15. Il s'agit de séquences de scènes autoroutières comportant tout type d'objets motorisés tels que des véhicules légers, des motos ou des poids lourds. La séquence C5 est particulièrement intéressante puisqu'elle contient de nombreux changements de luminosité causés par le passage réguliers de nuages, provoquant un changement de luminosité important. Dans l'objectif de situer nos résultats, le système est comparé à quelques algorithmes issus de l'état de l'art. Quatre algorithmes ont été sélectionnés : le filtrage médian adaptatif (AMF, *Adaptive Median Filtering*) [McFarlane 1995], le modèle gaussien (Wren) [Wren 1997], le mélange de gaussiennes (Grimson) [Stauffer 1999] et le mélange de gaussiennes (Zivkovic) [Zivkovic 2004]. Les paramètres utilisés pour l'analyse sont identiques pour les séquences de test utilisées et sont rassemblés dans la Table 4.1.

Méthode	**Paramètres**
Adaptive Median Filter	$c = 1$
Gaussienne (Wren)	$\alpha = 0.01$
Mélange de gaussiennes (Grimson)	$\alpha = 0.01$, $T = 0.8$
Mélange de gaussiennes (Zivkovic)	$\alpha = 0.01$, $T = 0.8$, $c_T = 0.05$, $\tau_{shadow} = 0.45$
Système proposé	$\alpha = 0.01$, $T = 0.8$, $c_T = 0.05$, $\tau_{shadow} = 0.45$, $\tau_{highlight} = 1.25$, $\rho_{min} = 1$

TABLE 4.1: Paramètres utilisés par les algorithmes pour l'évaluation de la détection de mouvement.

4.3.3 Résultats

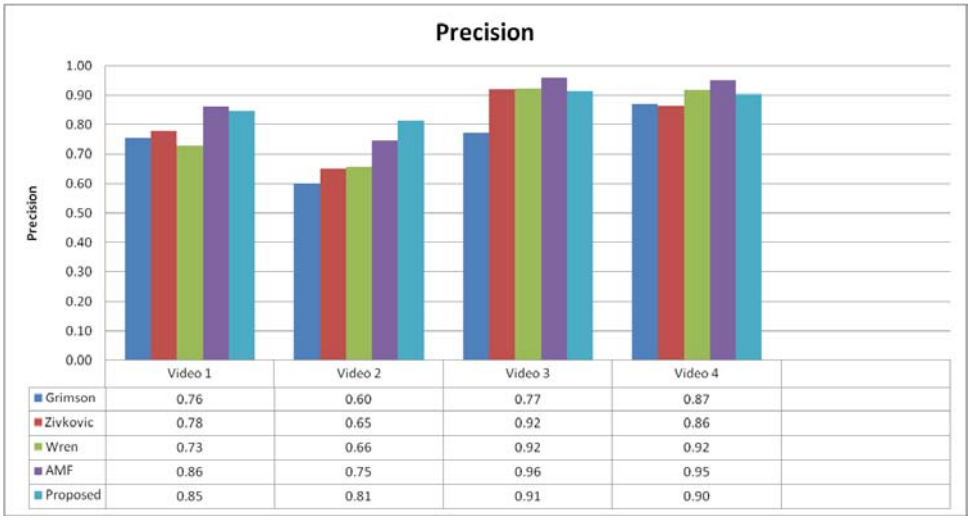
Les Figures 4.7a, 4.7b et 4.7c présentent les résultats pour les quatre séquences de test (C5, C21, Lyon et C15) à l'aide des mesures de Precision, de Rappel (ou TPR, (*True Positive Rate*)) et de F-score. La mesure de Rappel (TPR) permet de rendre compte du nombre de pixels correctement classés parmi l'ensemble des pixels de la vérité-terrain, tandis que la mesure de Precision fournit le pourcentage de bonne classification parmi l'ensemble des résultats retournés. Quant à la mesure de F-score, elle est une combinaison entre la mesure de précision et la mesure de Rappel (TPR). Elle permet de fournir une note, même approximative, sous la forme d'une seule valeur scalaire. Cette valeur doit être aussi élevée que possible.

Sur la séquence C5, la Precision (Figure 4.7a) de notre algorithme est supérieure à trois des quatre algorithmes sélectionnés pour la comparaison avec une Precision de 85% contre une Precision allant de 73% à 78% pour l'algorithme Grimson, Zivkovic et Wren. Seul l'algorithme AMF possède une Precision similaire à 86% mais au détriment d'un Rappel beaucoup plus faible (Figure 4.7b). Le Rappel pour l'algorithme proposé est supérieur à tous les autres algorithmes avec une valeur à 79% contre des valeurs de Rappel allant de 62% pour l'algorithme AMF à 68% pour l'algorithme Zivkovic. Nous obtenons ainsi une valeur de F-score supérieur à l'ensemble des algorithmes avec une valeur à 82% contre des valeurs comprises entre 69% et 76%. Ces résultats sont illustrés sur les Figures 4.8 et 4.9, traduisant respectivement l'évolution temporelle du taux de fausses alarmes (FAR, *False Alarm rate*, égale à (1−Precision)) et l'évolution temporelle du taux de vrai positifs (TPR, *True Positive Rate*, égal au Rappel). Le changement de luminosité est visible et compris entre les images 7400 et 8000 de la séquence. La chute du Rappel aux images 1600, 4600, 7750 et 8600 est dûe à la difficulté de notre algorithme à faire face aux poids-lourds et aux problèmes de camouflage, comme illustré sur la Figure 4.10 sur la colonne centrale (image 4600). Cette figure contient sur la première ligne l'image originale, et sur les lignes 2 à 5 on retrouve respectivement l'algorithme Grimson, l'algorithme Wren, l'algorithme AMF et l'algorithme proposé. On remarque la difficulté rencontrés par les algorithmes Grimson, AMF et Wren pour traiter le passage du nuage.

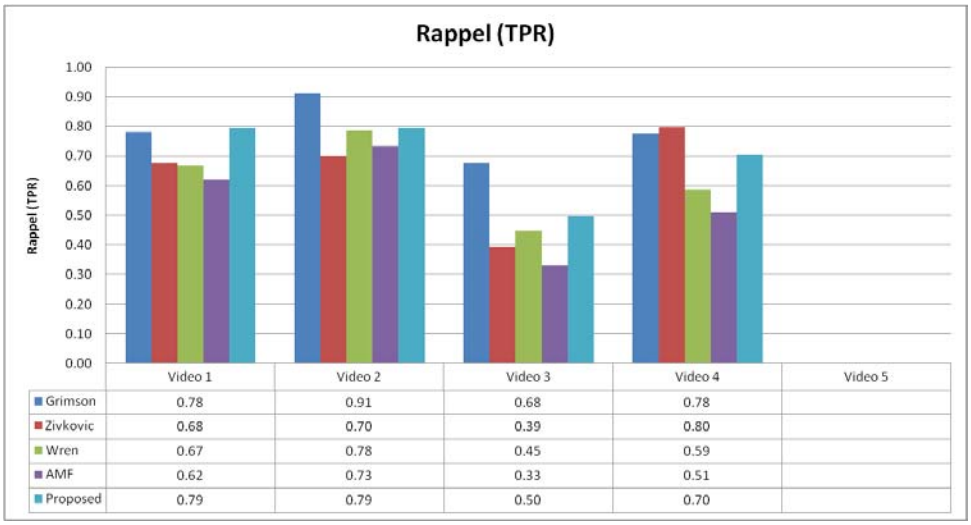
Sur la séquence C21, notre algorithme présente un meilleur taux de Precision que l'ensemble des autres algorithmes, avec une valeur de 81% contre 60% à 75%. Le rappel est

cependant similaire aux autres algorithmes sauf pour l'algorithme Grimson qui dépassent avec une valeur de 91%, contre 79% pour notre algorithme. Sur la Figure 4.11 sont présentés quelques résulats expliquant ces valeurs.

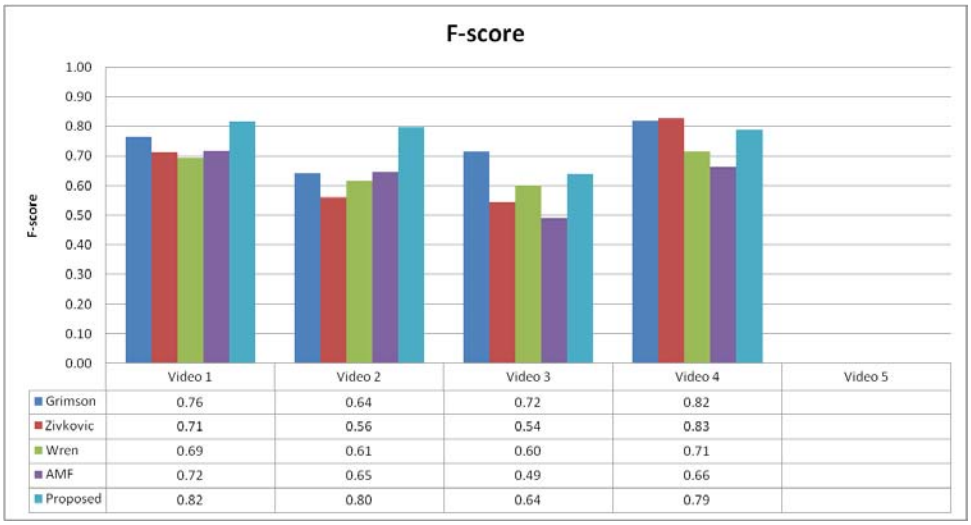
Sur les séquences Lyon et C15, les résultats sont relativement similaires, avec une Precision de 90% et 91%. Cette Precision est obtenue au détriment d'une bonne valeur de Rappel avec 50% pour notre algorithme, et 68% pour l'algorithme Grimson qui obtient la meilleure valeur de F-score. Sur les Figures 4.12 et 4.13 sont présentés quelques résultats des algorithmes sur les séquences Lyon et C15.



(a) Precision



(b) Rappel



(c) C5

FIGURE 4.7: Résultats de l'évaluation des performances des algorithmes Grimson, l'algorithme Wren, l'algorithme Zivkovic, l'algorithme AMF et l'algorithme proposé pour les quatre séquences de tests en termes de Precision, de Rappel et de F-Score.

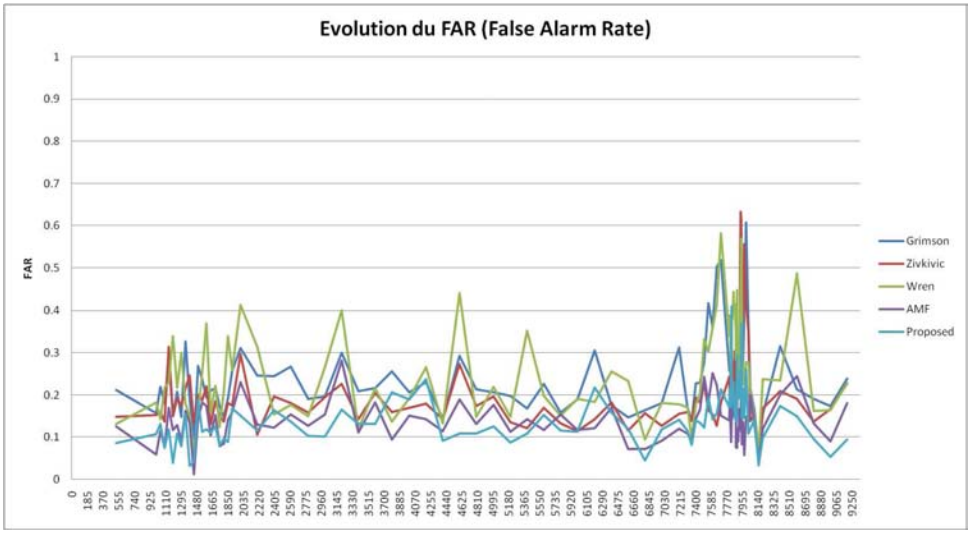


FIGURE 4.8: Résultats de l'évaluation de la détection de mouvement à l'aide du TPR (*True Positive Rate*) et du FAR (*False Alarm Rate*). Les séquences de test utilisées sont C5, Lyon, C9 et CE21. L'approche proposée est comparée avec différentes approches à l'aide d'une vérité-terrain obtenue manuellement.

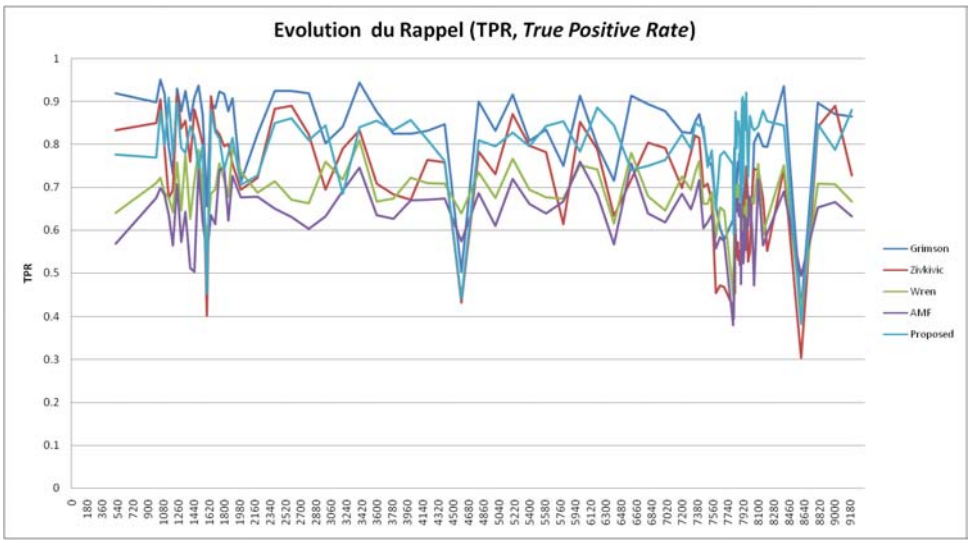


FIGURE 4.9: Résultats de l'évaluation de la détection de mouvement à l'aide du TPR (*True Positive Rate*) et du FAR (*False Alarm Rate*). Les séquences de test utilisées sont C5, Lyon, C9 et CE21. L'approche proposée est comparée avec différentes approches à l'aide d'une vérité-terrain obtenue manuellement.

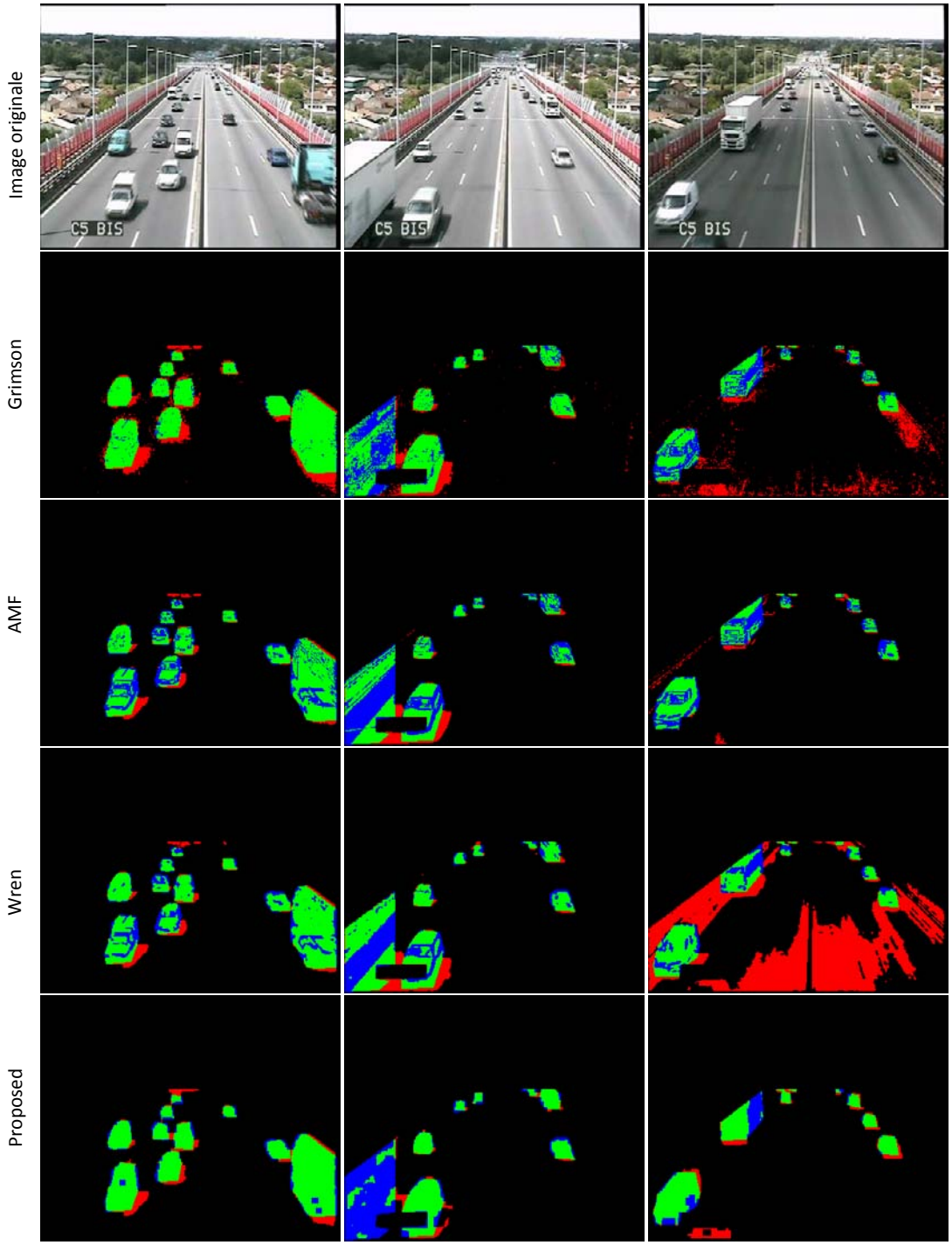


FIGURE 4.10: Résultats de la segmentation des régions en mouvement sur la séquence C5 pour les images 1050, 4600 et 7700. Les pixels *True Positive* (TP) sont représentés en vert, les pixels *False Positive* (FP) sont représentés en rouge et les pixels *True Negative* (TN) sont représentés en bleu. La première ligne correspond à l'image originale, la seconde ligne correspond à l'algorithme Grimson, la troisième ligne à l'algorithme Wren, la quatrième ligne à l'algorithme AMF et la dernière ligne à l'algorithme proposé.

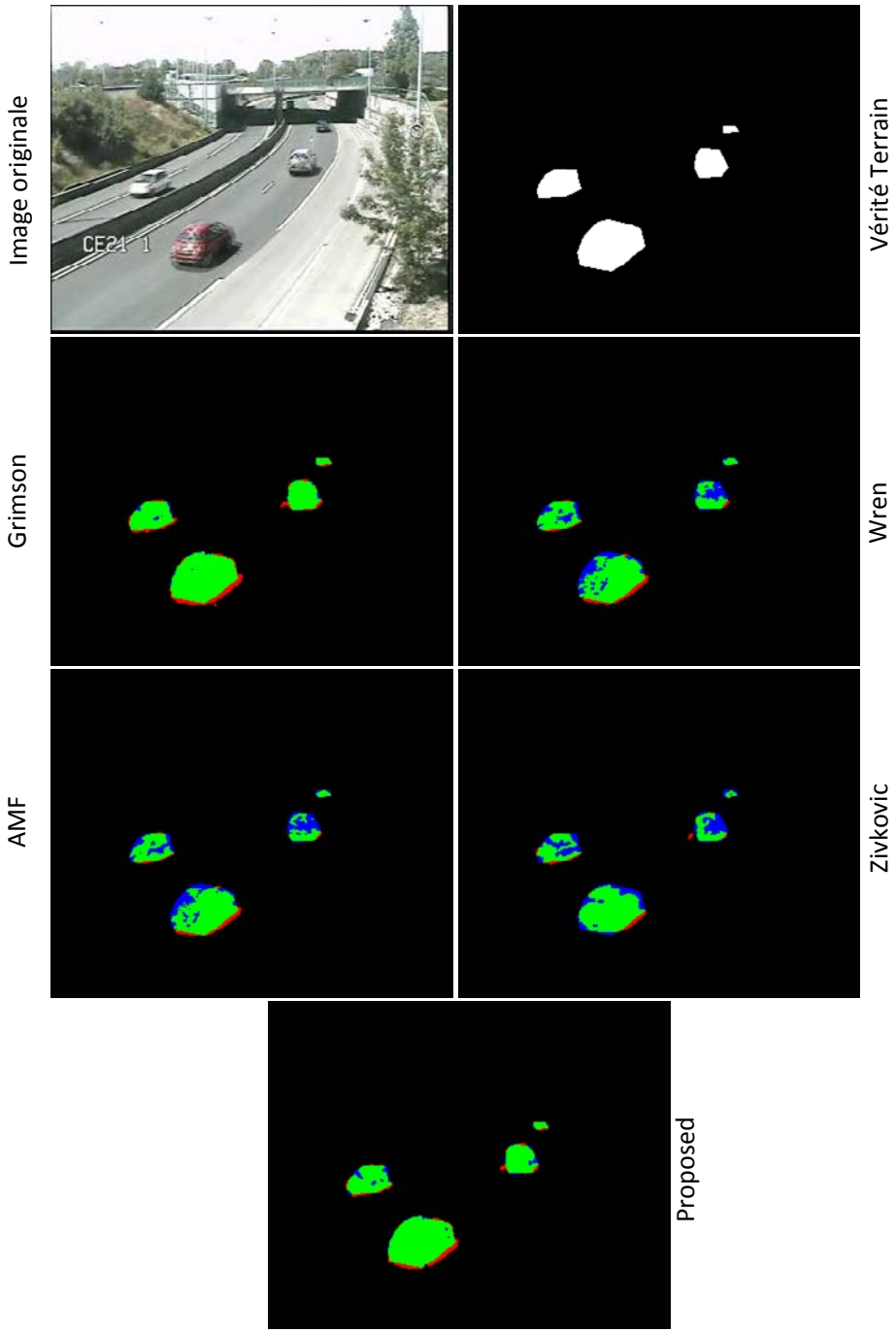


FIGURE 4.11: Résultats de la segmentation des régions en mouvement sur la séquence C21 (image 2600). Les pixels *True Positive* (TP) sont représentés en vert, les pixels *False Positive* (FP) sont représentés en rouge et les pixels *False Negative* (FN) sont représentés en bleu. Les algorithmes testés sont l'algorithme Grimson, l'algorithme Wren, l'algorithme Zivkovic, l'algorithme AMF.

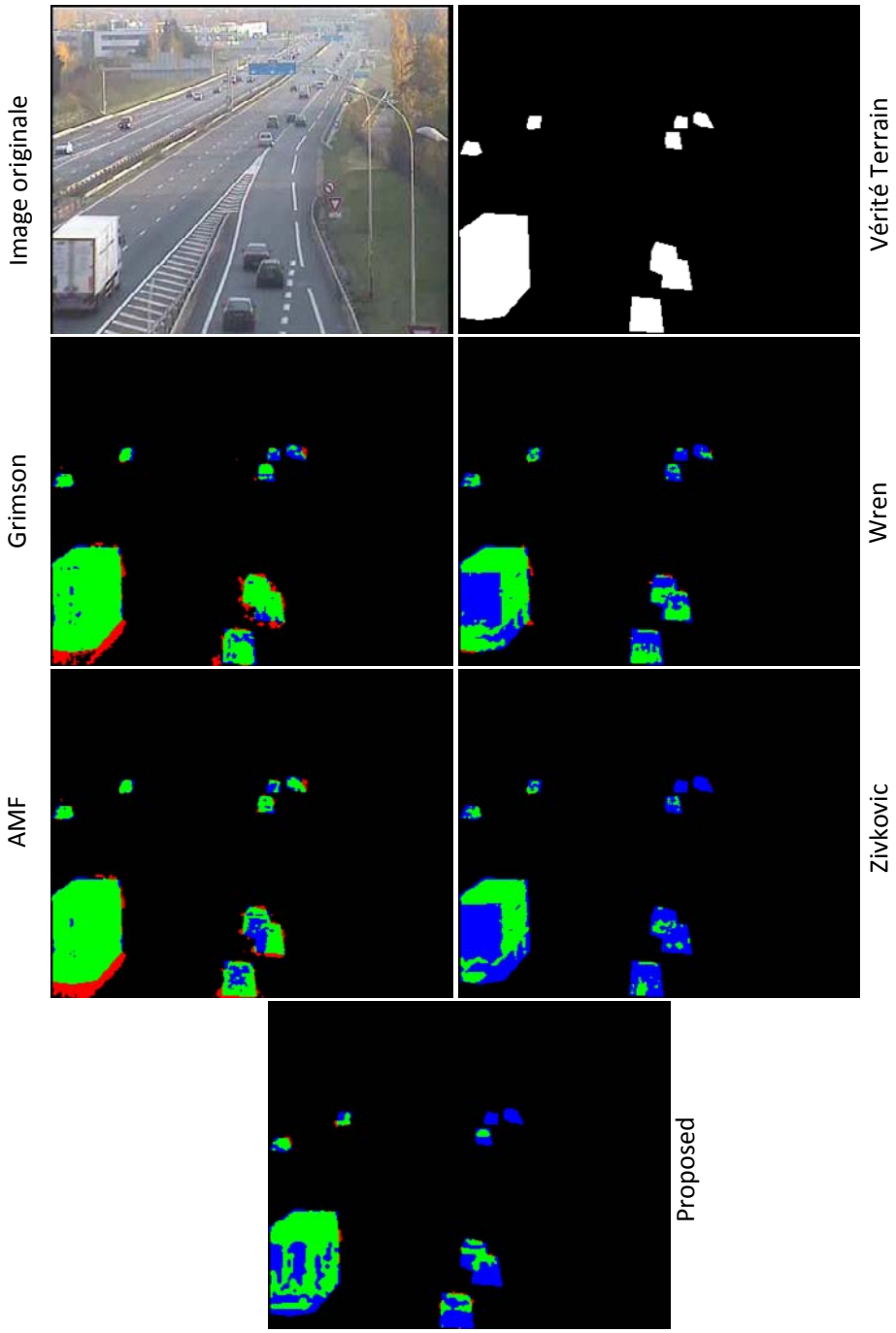


FIGURE 4.12: Résultats de la segmentation des régions en mouvement sur la séquence Lyon (image 13100). Les pixels *True Positive* (TP) sont représentés en vert, les pixels *False Positive* (FP) sont représentés en rouge et les pixels *False Negative* (FN) sont représentés en bleu. Les algorithmes testés sont l'algorithme Grimson, l'algorithme Wren, l'algorithme Zivkovic, l'algorithme AMF.

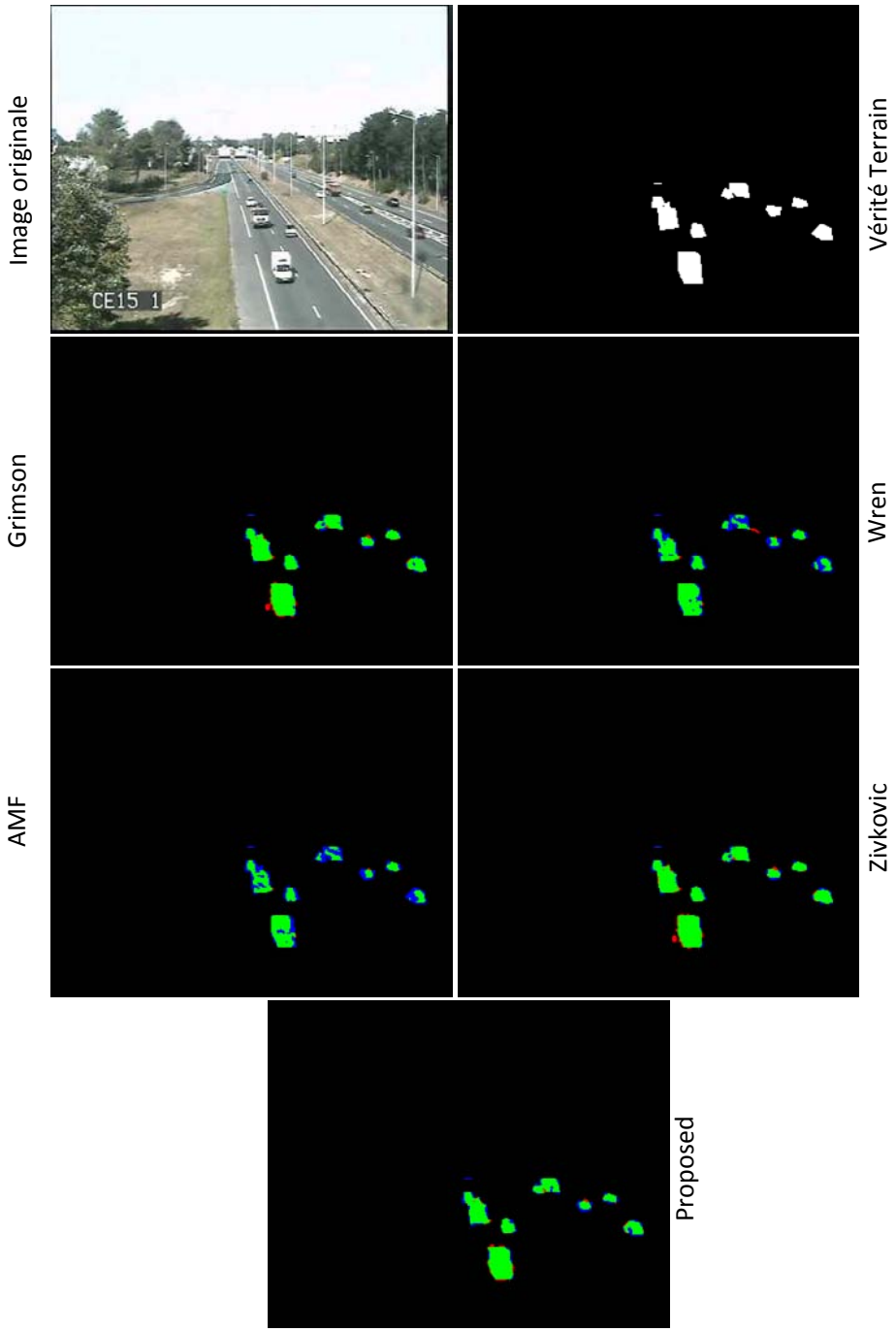


FIGURE 4.13: Résultats de la segmentation des régions en mouvement sur la séquence C15 (image 1900). Les pixels *True Positive* (TP) sont représentés en vert, les pixels *False Positive* (FP) sont représentés en rouge et les pixels *False Negative* (FN) sont représentés en bleu. Les algorithmes testés sont l'algorithme Grimson, l'algorithme Wren, l'algorithme Zivkovic, l'algorithme AMF.

4.4 Conclusion

Nous avons vu dans ce chapitre la construction d'un modèle d'arrière-plan destiné à détecter les objets en mouvement dans la scène. Ce modèle est basé sur un mélange de distributions paramétriques représentant pour chaque pixel la distribution couleur qui lui est associée. Pour une nouvelle image, tout pixel qui s'écarte de ce modèle est considéré comme potentiellement en mouvement. Dans l'objectif de s'adapter aux changements de la scène, le modèle est régulièrement mis à jour en utilisant une démarche d'*Expectation-Maximization* dans laquelle on estime dans un premier temps la vraisemblance de la donnée par rapport au modèle (étape d'*Expectation*), avant de mettre à jour les paramètres afin de maximiser cette vraisemblance (étape de *Maximization*). Cette approche permet d'obtenir une représentation statistique (basée sur un mélange de lois de probabilité) de l'arrière-plan couleur de la scène.

Pour prendre en compte les variations de lumière, un modèle colorimétrique [Horprasert 1999] centré la couleur de l'arrière-plan est utilisé. Ainsi, tout pixel ayant des propriétés chromatiques proches de celles de l'arrière-plan mais une intensité lumineuse plus faible ou plus élevée, dans un certaine mesure, est considéré comme étant une ombre ou un reflet. Il s'agit d'une restriction très forte qui accentue considérablement le problème de camouflage (diminue la précision), mais permet de conserver un bon taux de rappel lors d'un changement de luminosité.

Pour rehausser la précision de la segmentation, une information sur le gradient spatial est ajoutée basé sur la différence de gradient obtenue par le filtre de Sobel. Le masque de différence permet ainsi de valider la présence d'un objet en mouvement.

Cet algorithme a été testé sur un ensemble de vidéos et montre des résultats encourageants. L'algorithme a su être robuste face au passage d'un nuage sur la séquence C5 et conserve de bonnes performances sur les autres séquences. Notons cependant une faiblesse de l'algorithme causée par le problème de camouflage lorsque la couleur d'un objet est proche de celle de l'arrière-plan. Par exemple, en présence d'un poids lourd possédant une remorque uniforme en couleur et proche de celle de la route.

Chapitre 5

Analyse des caractéristiques intrinsèques des objets : extraction et suivi

Sommaire

Introduction

Le suivi d'objets est une étape fondamentale dans les systèmes de vidéo-surveillance puisqu'il est la base de l'analyse de trajectoires, de comportements et de reconnaissances d'activités. Le fondement des problèmes de suivi consiste à suivre et associer correctement les objets détectés. Historiquement, il s'agit d'un problème d'association de données, dont les premières applications étaient essentiellement l'analyse de données fournies par les radars ou les sonars. Les *cibles* représentent les objets d'intérêts, tandis que les *mesures* (ou *observations*) représentent les régions en mouvement obtenues en sortie du module de détection de mouvement.

Le chapitre précédent décrit la procédure de soustraction d'arrière-plan appliquée à la séquence vidéo pour extraire les régions en mouvement. Le résultat est un masque binaire noté $\mathcal{F}$ (*foreground*) dans lequel les pixels de valeurs nulles sont représentatifs de l'arrière-plan, tandis que les pixels de valeurs non nulles sont représentatifs d'un ou plusieurs objets en mouvement. Ce chapitre présente l'étape suivante du traitement qui consiste à extraire, classer et suivre les objets de la scène. Il s'agit de construire et maintenir une liste d'objets vidéo, caractéristiques des objets réels de la vidéo. La construction de cette liste se décompose en deux grandes étapes (Figure 5.1) : une étape d'extraction des objets et de leurs caractéristiques à partir du masque *foreground*, et une étape de suivi des objets extraits. L'étape de suivi repose sur un filtrage prédictif basé sur un modèle de mouvement à vitesse constante.

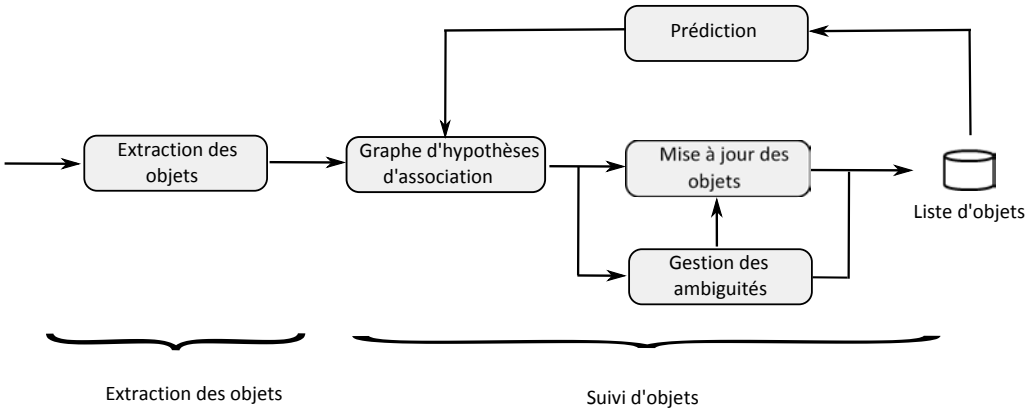


FIGURE 5.1: Les deux étapes de l'analyse des caractéristiques intrinsèques des objets. L'étape d'extraction des objets permet de construire une liste d'objets (ou groupes d'objets) détectés dans l'image courante et le processus de suivi d'objets permet de maintenir leur identité au cours du temps.

La première section de ce chapitre (Section 5.1) présente les fondements théoriques de l'estimation bayésienne et le cas particulier du filtrage de Kalman sur lequel se base l'algorithme de suivi. Nous verrons également l'utilisation d'un graphe pour l'extension à un suivi multi-cible. La Section 5.2 présente l'algorithme complet de suivi d'objets utilisé par notre système. Celui-ci se décompose en quatre étapes : une étape d'extraction des objets, une étape de prédiction, une étape de génération d'hypothèses et une étape de mise à jour des objets et de résolution des ambiguités. Dans la dernière section de ce chapitre

(Section 5.3), nous présenterons les résultats obtenus sur un ensemble de séquences vidéos.

5.1 Principe du filtrage bayésien

Les erreurs de segmentation des régions en mouvement altérent les performances de l'algorithme de suivi. Par exemple, lorsqu'un objet n'a pas été détecté ou lorsqu'il est occulté, il est nécessaire de pouvoir effectuer une prédiction de sa position ou de sa configuration. Cette prédiction s'appuie sur un modèle statistique de déplacement de l'objet et sur l'historique des positions précédentes. Chaque objet est soumis à un modèle de mouvement caractérisant une information *a priori* connue sur le problème à traiter. Ce modèle statistique est représenté sous la forme d'un modèle d'état dans lequel la configuration d'un objet (son état) n'est accessible qu'à travers une mesure bruitée (observation bruitée).

Le problème de suivi est résolu à l'aide d'une démarche statistique bayésienne. Cette méthode s'appuie sur l'estimation de l'état à partir de connaissances *a priori* sur le modèle physique associé au problème ainsi que sur les propriétés statistiques des signaux perturbateurs. Ainsi, l'objet est conceptualisé comme ayant un état propre interne évoluant au cours du temps. Le problème de suivi revient donc à estimer la configuration des objets à partir de mesures ou d'observations bruitées.

5.1.1 Estimation bayésienne récursive

Représentation probabiliste dans l'espace d'état

La représentation d'un problème sous la forme d'un système d'état est illustrée schématiquement sur la Figure 5.2. L'état (ou la configuration) d'un objet est représenté sous la forme d'un vecteur, appelé vecteur d'état et composé d'un ensemble de variables supposées aléatoires, appelées variables d'état. Dans une forme très simple, le vecteur d'état correspond par exemple à un vecteur position d'un objet en mouvement dans la scène, mais celui-ci peut être enrichi par tout autre caractéristique (vitesse, taille, ...).

Notons $x_t \in \mathcal{R}^{n_x}$ le vecteur d'état à l'instant t et de dimension n_x. Ce vecteur d'état est inconnu et on cherche à l'estimer à partir d'un vecteur d'observations, noté $z_t \in \mathcal{R}^{n_z}$ et de dimension n_z. Notons également $z_{t_1:t_2} = \{z_{t_1}, ..., z_{t_2-1}, z_{t_2}\}$ l'ensemble formé par les observations entre les instants t_1 et t_2 inclus. L'état initial noté x_0 est distribué selon $p(x_0)$. En pratique, ce que l'on observe est toujours moins informatif que ce qu'on cherche à estimer, on suppose donc que $n_z < n_x$. L'évolution temporelle du vecteur d'état est soumise à une loi d'évolution f, et la relation entre l'observation et le vecteur d'état est décrite par une loi d'observation notée h.

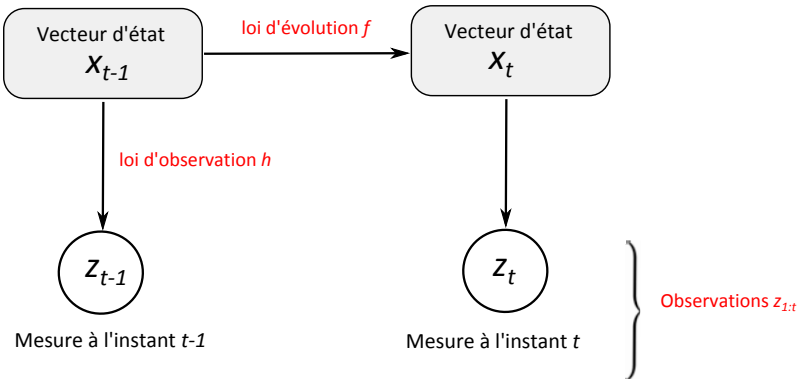


FIGURE 5.2: L'estimation de l'état d'un système nécessite sa description complète à travers la connaissance de trois types d'informations : les observations $z_{1:t}$, la loi d'observation et la loi d'évolution du système.

Supposons maintenant que l'évolution temporelle du vecteur d'état est soumis à l'hypothèse et aux propriétés statistiques des processus de Markov à l'ordre un : les propriétés statistiques de l'état x_t à l'instant t ne dépendent que du passé d'ordre un, c'est à dire de l'état à l'instant précédent x_{t-1}. Puisque seules les observations z_t sont accessibles, on parle de modèle de Markov caché. Il s'agit, dans le cas discret, d'un processus défini par les couples $\{x_t, z_t\}$ (Figure 5.3). Généralement, on fait l'hypothèse que les mesures sont indépendantes conditionnellement à l'état, ce qui se traduit par :

$$p(z_t|x_t, z_{1:t-1}) = p(z_t|x_t) \tag{5.1}$$

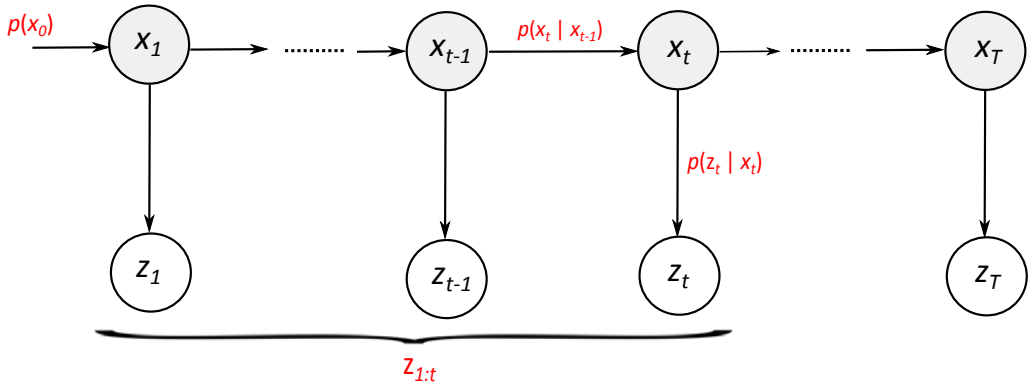


FIGURE 5.3: Illustration d'un modèle de markov caché (HMM). L'état d'un objet à l'instant t est donné par x_t et les observations associées sont notées z_i et sous l'hypothèse markovienne à l'ordre 1, l'état x_t ne dépend que de son état précédent x_{t-1}. Les flèches indiquent les dépendances entre les variables.

Les relations du graphe présentées sur la Figure 5.3 peuvent être modélisées par un système discret comportant deux équations. La première décrit l'évolution temporelle du vecteur d'état, elle est décrite par une **densité de transition** notée $p(x_t|x_{t-1})$ et s'exprime

à travers l'équation d'état donnée par

$$x_t = f(x_{t-1}, q_t, t) \tag{5.2}$$

avec

- x_t le vecteur d'état de dimension n_x du système à l'instant t.
- q_t est un vecteur aléatoire de même dimension que le vecteur d'état, appelé bruit d'état et correspond à l'erreur du modèle.
- f une fonction potentiellement dépendante du temps t de $\mathcal{R}^{n_x}$ dans $\mathcal{R}^{n_x}$.

La deuxième équation caractérisant le système est l'équation d'observation, qui traduit le fait que les grandeurs mesurées ne soient pas directement les variables d'état, mais une fonction de ces variables. La relation entre la mesure z_t et l'état x_t est décrite par la densité notée $p(z_t|x_t)$ appelée **vraisemblance de l'observation** (ou de la mesure). Elle traduit l'équation d'observation donnée par

$$z_t = h(x_t, r_t, t) \tag{5.3}$$

avec

- z_t est le vecteur d'observation de dimension n_z du système à l'instant t.
- r_t est un vecteur aléatoire de même dimension que le vecteur d'observation, appelé bruit d'observation et correspond à l'erreur d'observation.
- h un opérateur d'observation potentiellement dépendant du temps t de $\mathcal{R}^{n_x}$ dans $\mathcal{R}^{n_z}$. Cet opérateur relie l'observation à l'instant z_t avec l'état caché x_t.

L'objectif consiste, à partir des connaissances dont on dispose sur $p(x_0)$, $p(x_t|x_{t-1})$, $p(z_t|x_t)$ et sur les mesures $z_{1:t}$, à estimer la densité *a posteriori* $p(x_{0:t}|z_{1:t})$ pour tout $t \geq 1$. On parle de filtrage statistique bayésien ou d'inférence bayésienne. L'inférence dans les modèles à espace d'état couvre en général trois problèmes principaux : la prédiction lorsque l'on cherche à estimer $p(x_{0:t}|z_{1:T})$ avec $T < t$, le filtrage lorsque l'on cherche $p(x_{0:t}|z_{1:T})$ avec $T = t$ et le lissage lorsque l'on cherche $p(x_{0:t}|z_{1:T})$, avec $T > t$.

Filtrage optimal bayésien

Le principe général du filtrage optimal bayésien consiste à estimer la vraisemblance d'observer l'état courant d'un système x_t *conditionnellement* à l'ensemble des observations disponibles $z_{1:t} = \{z_1, ..., z_t\}$. Cette vraisemblance est exprimée sous la forme d'une densité de probabilité, notée $p(x_t|z_{1:t})$ et est appelée distribution *a posteriori* (voir paragraphe précédent). Si l'on suppose connue la densité de probabilité à l'instant initial $t = 0$, $p(x_0)$, la description du modèle d'état à travers $p(x_t|x_{t-1})$ et $p(z_t|x_t)$ et la séquence de mesure $z_{1:t}$, alors la probabilité $p(x_t|z_{1:t})$ peut être estimée par une approche récursive composée de deux étapes : une étape de prédiction et une étape de correction (Figure 5.4).

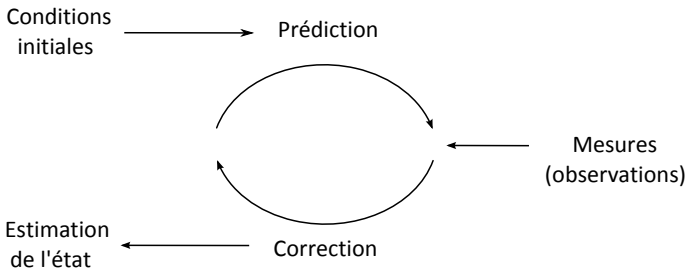


FIGURE 5.4: Principe récursif du filtrage bayésien basé sur une étape de prédiction et une étape de correction à l'aide des observations disponibles.

L'**étape de prédiction** est dictée par l'équation d'état (Equation 5.2) et consiste à fournir une estimation de la densité de l'état $p(x_t|z_{1:t-1})$ à l'instant t à partir de l'estimation précédente de la densité $p(x_{t-1}|z_{1:t-1})$. En supposant connue l'estimation de la densité *a posteriori* à l'instant précédent $p(x_{t-1}|z_{1:t-1})$ et sous l'hypothèse markovienne, la distribution jointe s'écrit

$$\begin{aligned} p(x_t, x_{t-1}|z_{1:t-1}) &= p(x_t|x_{t-1}, z_{1:t-1})p(x_{t-1}|z_{1:t-1}) \\ &= p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1}) \end{aligned}$$

En intégrant sur l'espace d'état dans lequel est décrit x_{t-1}, on obtient l'équation de Chapman-Kolmogorov qui définie l'étape de prédiction du filtre optimal :

$$p(x_t|z_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1} \tag{5.4}$$

L'**étape de correction** consiste à corriger l'estimation de la distribution *a posteriori*. La vraisemblance de la mesure $p(z_t|x_t)$ et la distribution *a priori* issue de l'équation de Chapman-Kolmogorov sont utilisées lorsqu'une nouvelle mesure z_t est disponible pour actualiser $p(x_t|z_t)$ à l'aide de la règle de Bayes. Cette étape constitue l'étape de correction du filtre optimal et s'écrit

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{\int p(z_t|x_t)p(x_t|z_{1:t-1})dx_t} \tag{5.5}$$

où le dénominateur $p(z_t|z_{1:t-1})$ est un facteur de normalisation dépendant de $p(z_t|x_t)$ qui est défini par l'équation d'observation (Equation 5.3). Ainsi, en injectant l'équation (5.4) dans (5.5) et en remplaçant le terme de normalisation par une constante $C = p(z_t|z_{1:t-1})$ (puisqu'indépendante de la variable x_t) on obtient l'équation générale du filtre de Bayes qui traduit la proportionnalité entre l'estimation *a posteriori* de la densité $p(x_t|z_t)$ avec la vraisemblance de l'observation $p(z_t|x_t)$, la densité de transition du modèle dynamique $p(x_t|x_{t-1})$ et la densité *a posteriori* $p(x_{t-1}|z_{t-1})$ estimées à l'étape précédente :

$$\underbrace{p(x_t|z_{1:t})}_{\substack{\text{densité } a \text{ posteriori} \\ \text{à l'instant } t}} \propto \underbrace{p(z_t|x_t)}_{\substack{\text{vraisemblance de} \\ \text{l'observation}}} \int \underbrace{p(x_t|x_{t-1})}_{\substack{\text{densité de} \\ \text{transition}}} \underbrace{p(x_{t-1}|z_{1:t-1})}_{\substack{\text{densité } a \text{ posteriori} \\ \text{à l'instant } t-1}} \tag{5.6}$$

Le cycle récursif du filtrage bayésien est présenté sur la Figure 5.5 et permet en théorie d'estimer de façon exacte la densité $p(x_t|z_{1:t})$ à chaque intervalle de temps. Cependant en pratique, les Equations 5.4 et 5.5 admettent une solution analytique que dans certaines conditions particulières. C'est le cas du filtrage de Kalman qui fournit une solution optimale lorsque les modèles dynamiques sont linéaires et que l'ensemble des processus mis en jeu est gaussien.

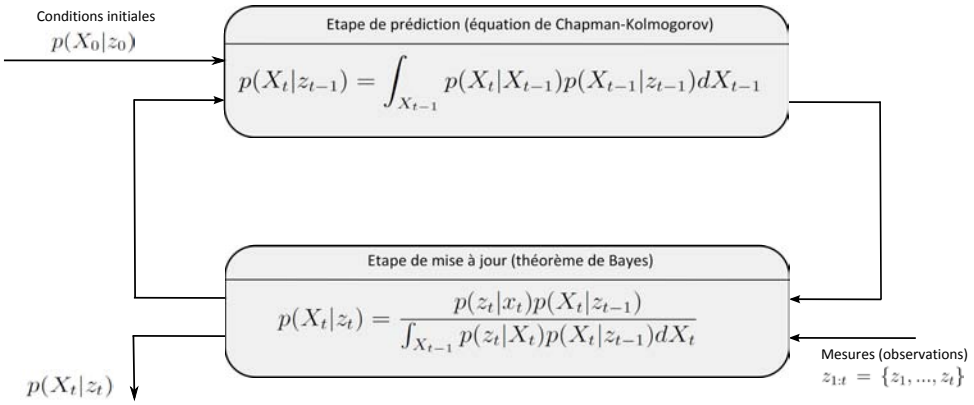


FIGURE 5.5: Illustration du principe de l'estimation récursive de Bayes.

5.1.2 Filtrage de Kalman

Le filtre de Kalman [Kalman 1960] est un filtre bayésien particulier qui fournit une solution optimale et exacte lorsque le problème est décrit par un système dynamique linéaire perturbé par des bruits additifs considérés gaussiens. Autrement dit, l'utilisation du filtre de Kalman nécessite la description du modèle sous la forme d'un système linéaire, et les perturbations appliquées au système sont de natures gaussiennes. Ceci se traduit par les équations suivantes :

$$\begin{aligned} x_t &= F_t x_{t-1} + q_t \\ z_t &= H_t x_t + r_t \end{aligned} \tag{5.7}$$

avec

$$\begin{aligned} q_t &\sim \mathcal{N}(0, Q_t) \\ r_t &\sim \mathcal{N}(0, R_t) \end{aligned} \tag{5.8}$$

et

- F est appelée matrice de transition du système, est connue et peut éventuellement évoluer au cours du temps. Elle traduit la relation markovienne de l'évolution de l'état entre l'instant précédent x_{t-1} et l'instant courant x_t.
- q_t est un vecteur aléatoire de même dimension que x_t, inconnu et inaccessible à la mesure, et est appelé bruit d'état (ou bruit du modèle). Ce vecteur est supposé gaussien, centré en zéro et de matrice de covariance Q et est supposé décorrélé de r_t.
- H_t est appelée matrice d'observation et traduit la relation entre le vecteur d'état (inconnu *a priori*) et l'observation.

– r_t est un vecteur aléatoire de même dimension que z_t également inconnu et appelé bruit de mesure. Ce vecteur est supposé gaussien, centré en zéro et de matrice de covariance R et est supposé décorrélé de q_t.

D'un point de vue statistique, le modèle est défini par sa densité de transition $p(x_t|x_{t-1})$, la vraisemblance des observations $p(z_t|x_t)$ et la densité initiale $p(x_0)$ toutes supposées gaussiennes et définies par :

$$\begin{aligned} p(x_t|x_{t-1}) &= \mathcal{N}(x_t|F_t x_{t-1}, Q_t) \\ p(z_t|x_t) &= \mathcal{N}(z_t|H_t x_t, R_t) \\ p(x_0) &= \mathcal{N}(x_0|m_0, P_0) \end{aligned} \tag{5.9}$$

Ainsi, puisque la loi d'évolution f est supposée être une fonction linéaire de x_t et de w_t, puisque la loi d'observation h est également supposée fonction linéaire de x_t et de v et puisque les bruits q_t et r_t sont des processus gaussiens, alors par linéarité des propriétés gaussiennes, on montre que la densité recherchée $p(x_t|z_{1:t})$ est de nature gaussienne, paramétrée par un moyenne m et une matrice de covariance P. Le filtre de Kalman est un filtre linéaire bayésien qui fournit l'estimation du vecteur d'état en deux étapes : une étape de prédiction (équation de Chapman-Kolmogorov 5.4) et une étape de correction (règle de Bayes 5.5).

L'**étape de prédiction** fournit une solution sous la forme

$$p(x_t|z_{1:t-1}) = \mathcal{N}(x_t|m_{t|t-1}, P_{t|t-1}) \tag{5.10}$$

$$p(x_t|z_{1:t}) = \mathcal{N}(x_t|m_{t|t}, P_{t|t}) \tag{5.11}$$

où $m_{t|t-1}$ et $P_{t|t-1} = \mathbb{E}[m_{t|t-1} m_{t|t-1}^T]$ sont respectivement la prédiction sur la moyenne et la matrice de covariance de la densité d'état à l'instant t à partir des échantillons jusqu'à l'instant $t-1$. La prédiction sur le vecteur d'état $m_{t|t-1}$ et la matrice de covariance $P_{t|t-1}$ sont données par

$$\begin{cases} m_{t|t-1} = F_t m_{t-1|t-1} \\ P_{t|t-1} = Q_{t-1} + F_t P_{t-1|t-1} F_t^T \end{cases} \tag{5.12}$$

L'**étape de correction** prend en compte la nouvelle mesure disponible z_t afin de calculer l'erreur de prédiction (appelée *innovation*) caractérisée par sa moyenne e_t et sa covariance S_t :

$$\begin{cases} e_t = z_t - H_t m_{t|t-1} \\ S_t = H_t P_{t|t-1} H_t^T + R_t \end{cases} \tag{5.13}$$

L'erreur de prédiction est ensuite réinjectée dans le système pour corriger et actualiser l'estimation de l'état à travers une matrice appelée gain de Kalman notée K_t

$$K_t = P_{t/t-1} H_t^T S_t^{-1} \tag{5.14}$$

$$\begin{cases} m_{t/t} = m_{t/t-1} + K_t e_t \\ P_{t/t} = (I - K_t H_t) P_{t/t-1} \end{cases} \tag{5.15}$$

Cette procédure est répétée de façon itérative pour chaque nouvelle mesure et permet d'estimer la densité du vecteur d'état $p(x_t|z_{1:t})$ par correction entre chaque nouvelle mesure z_t et la prédiction de l'état à l'instant t, $m_{t|t-1}$. Cette densité, supposée gaussienne, est entièrement décrite par l'estimation de sa moyenne $m_{t|t}$ et de sa covariance $P_{t|t}$ fournies à la sortie du filtre.

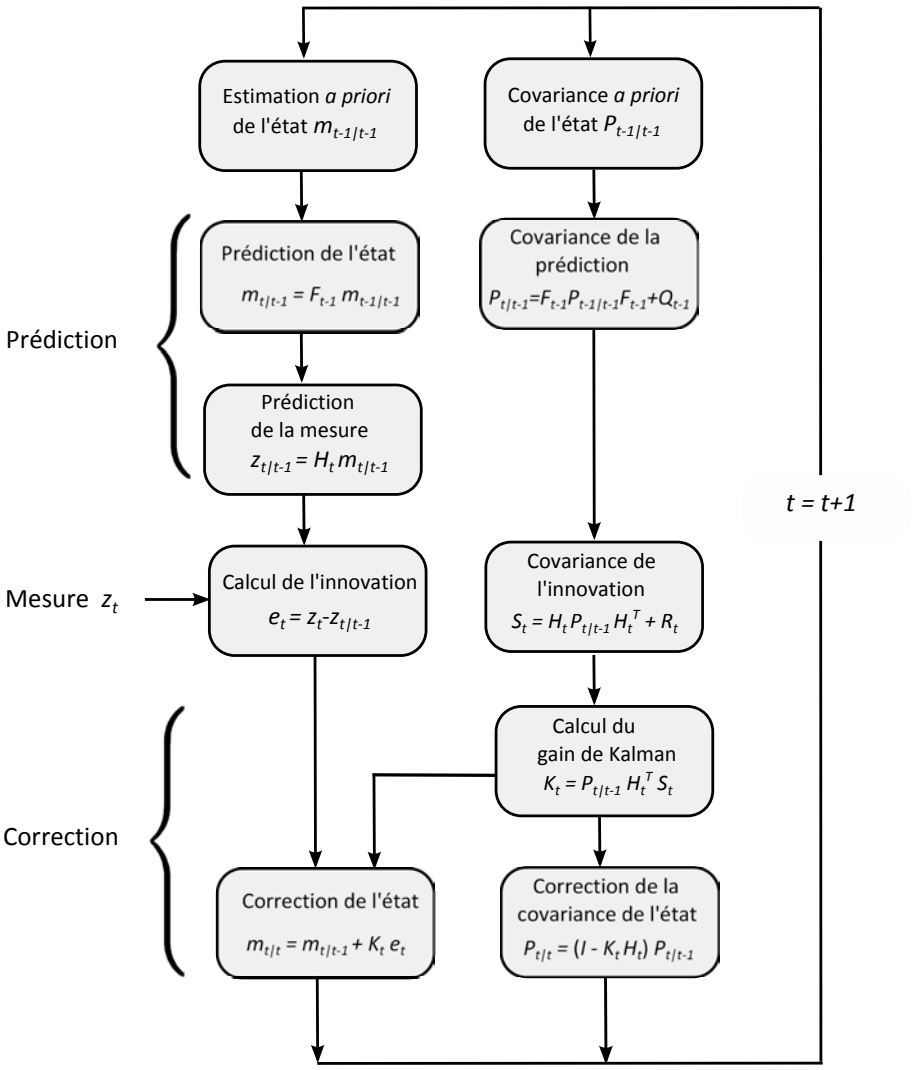


FIGURE 5.6: Illustration du principe du filtrage récursif de Kalman.

Algorithme de Kalman

L'algorithme de Kalman (Algorithme 4) comporte les étapes suivantes
– Initialisation du vecteur d'état X et de sa matrice de covariance P.

$$\begin{aligned} m_{0/0} &= m_0 \\ P_{0/0} &= P_0 \end{aligned} \tag{5.16}$$

– Calcul de l'estimation de l'état du système $\hat{X}_t$ à l'instant t à partir des mesures à l'instant précédent $t-1$.

$$m_{t/t-1} = F_t m_{t-1/t-1} \tag{5.17}$$

– Mise à jour intermédiaire de la matrice de covariance de l'état $P_{t/t-1}$.

$$P_{t/t-1} = F_t P_{t-1/t-1} F_t^T + Q_t \tag{5.18}$$

– Calcul du gain du filtre de Kalman K_t. Ce gain ne dépend pas des données mesurées et tient compte uniquement des caractéristiques statistiques du bruit de mesure.

$$K_t = P_{t/t-1} H_t^T \left(H_t P_{t/t-1} H_t^T + R_t\right)^{-1} \tag{5.19}$$

– Mise à jour de la matrice de covariance de l'état.

$$P_{t/t} = \left(I - K_t H_t\right) P_{t/t-1} \tag{5.20}$$

– Correction de l'estimation de l'état.

$$m_{t/t} = m_{t/t-1} + K_t \left(Z_t - H_t m_{t/t-1}\right) \tag{5.21}$$

Le processus complet est résumé dans l'algorithme 4.

Algorithme 4: Algorithme de Kalman

Input : Mesure (observation) z_t d'un objet à l'instant t
Data : Les matrices F, H, W, P, R sont fixes et initialisées.
Output : Estimation du vecteur d'état

Initialisation des matrices $m_{0/0}$, $P_{0/0}$.
$m_{0/0} = m_0$
$P_{0/0} = P_0$
for each *instant* t **do**

– Prédiction de $P_{t/t-1}$ et de $m_{t/t-1}$ en utilisant les équations de propagation.

$$m_{t/t-1} = F_t m_{t-1/t-1} \qquad \text{(Equ. 5.17)}$$
$$P_{t/t-1} = F_t P_{t-1/t-1} F_t^T + Q_t \qquad \text{(Equ. 5.18)}$$

– Calcul du gain du filtre K_t, correction de l'état estimé et de son incertitude.

$$K_t = P_{t/t-1} H_t^T \left(H_t P_{t/t-1} H_t^T + R\right)^{-1} \qquad \text{(Equ. 5.19)}$$
$$P_{t/t} = \left(I - K_t H_t\right) P_{k/k-1} \qquad \text{(Equ. 5.20)}$$
$$m_{t/t} = m_{t/t-1} + K_t \left(Z_t - H_t m_{t/t-1}\right) \qquad \text{(Equ. 5.21)}$$

– Fournir l'état estimé $m_{t/t}$ en sortie du filtre.

5.1.3 Suivi multi-cible

Le filtrage présenté dans les sections précédentes permet de suivre une cible à partir de mesures bruitées et d'une estimation basée sur un modèle (linéaire dans le cas du filtre de Kalman) et connaissant la densité à priori de l'état à travers les conditions initiales. Dans le cas d'un suivi multi-cibles, plusieurs mesures sont disponibles et doivent être associées avant de mettre à jour les objets suivis. Il s'agit d'un problème d'association de données, qui consiste à mettre en correspondance les objets détectés entre deux images consécutives (ou proches temporellement) à l'aide d'une mesure de similarité entre descripteurs. Durant cette étape, un objet peut être assigné à plusieurs objets candidats dans l'image suivante et vice-versa. Par conséquent, l'étape de mise en correspondance fournit un ensemble d'hypothèses contenant potentiellement des ambiguités d'associations, résultant par exemple des intéractions entre objets, ou des entrées et sorties du champ de vision de la caméra. Pour prendre en compte ces ambiguités, le problème de suivi multi-objets est modélisé à l'aide d'un graphe, dont les nœuds forment les objets et les arêtes caractérisent la similarité entre objets suivis et objets candidats.

Processus d'association et génération d'hypothèses

Le mécanisme d'association des objets suivis avec les objets nouvellement détectés est une étape importante des algorithmes de suivi d'objets. Il s'agit de mettre en correspondance les objets suivis avec les mesures disponibles en tenant compte d'éventuelles fragmentations de mesures (plusieurs mesures pour le même objet) ou d'absence de mesure. Soit $\mathcal{O}_n^{t-1} = \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$ une liste des objets suivis et $\mathcal{C}_m^t = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ la liste des objets candidats, i.e les objets détectés dans l'image courante et fournis par le module d'extraction d'objets. Le problème d'association de données consiste à déterminer la correspondance entre les n objets suivis $\mathcal{O}_n^{t-1}$ et les m objets candidats $\mathcal{C}_m^t$. Dans notre système, les objets candidats (mesures) sont obtenus à partir du module d'extraction d'objets (5.2.2). Le processus d'association est effectué pour chaque nouvelle image à l'instant t, et consiste à associer les $n_{\mathcal{O}}$ objets suivis avec les $n_{\mathcal{C}}$ *blobs* candidats observés dans l'image courante. Dans la cas le plus simple, la relation est bijective et chaque objet i suivi $\mathcal{C}_{(i)}$ est associé à un et un seul objet j candidat $\mathcal{O}_{(j)}$. Cependant, dans les scènes réelles, les objets entrent et sortent de façon dynamique dans la scène, peuvent être occultés par d'autres objets ou par l'arrière-plan, peuvent se regrouper ou fusionner, ... Par conséquent, le nombre d'objets suivis et candidats est différent ($\mathcal{C}_{(j)} \neq \mathcal{O}$), plusieurs objets candidats peuvent être associés à un seul objet suivi et *vice-versa*.

Le problème d'association est modélisé, à chaque instant t, sous la forme d'un graphe noté $G_t = \{V, E\}$ où V et E sont respectivement les nœuds et les arêtes du graphe. L'ensemble des nœuds V peuvent être séparés en deux partitions : la partition des objets suivis $\mathcal{O}_i$ et la partition des objets candidats $\mathcal{C}_m$. Cette partition des nœuds en deux sous-ensemble classe le graphe parmi les graphes bipartis, comme représenté sur la Figure 5.7.

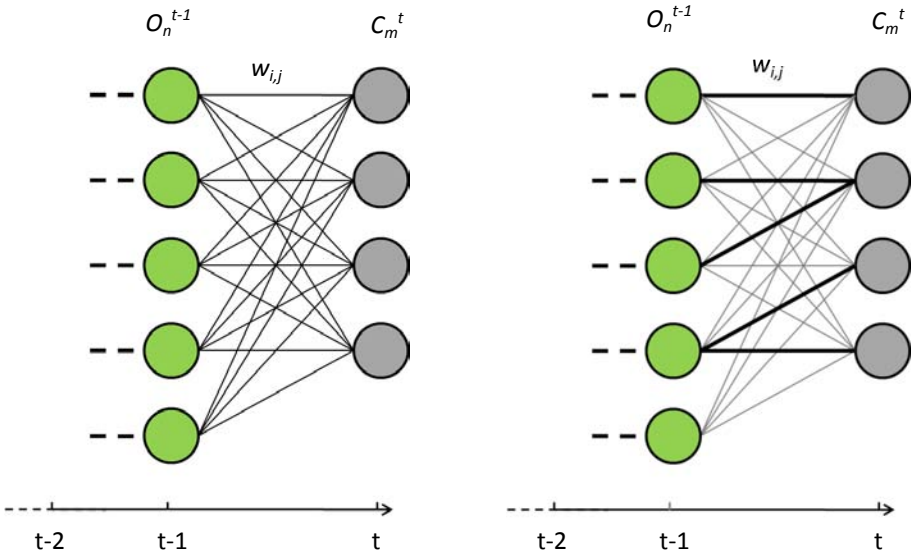


FIGURE 5.7: Illustration d'un graphe biparti pour le problème d'association de données. (À gauche) À chaque arête du graphe est associé un poids $w_{i,j}$ caractérisant la similarité entre le nœud Objet $\mathcal{O}_i$ et le nœud candidat $\mathcal{C}_j$. (À droite) Seules les arêtes dont le poids est supérieur à un seuil sont conservées, les autres arêtes (grisées) sont supprimées.

Il s'agit d'un graphe pondéré, autrement dit, à chaque arête est associé un poids caractérisant la similarité entre les nœuds aux extrémités (objet suivi $\mathcal{O}_i$ et objet candidat $\mathcal{C}_j$). A chaque nouvelle image, le processus d'association consiste à calculer la pondération de toutes les arêtes du graphe. Cette pondération est déterminée à l'aide d'une mesure de similarité entre le vecteur caractéristique de l'objet suivi et le vecteur caractéristique de l'objet candidat. Soit $\xi = (x_1, x_2, \ldots, x_p)$ un vecteur caractéristique de p variables aléatoires et $\xi^{\mathcal{O}} = \left(x_1^{\mathcal{O}}, x_2^{\mathcal{O}}, \ldots, x_p^{\mathcal{O}}\right)$, $\xi^{\mathcal{C}} = \left(x_1^{\mathcal{C}}, x_2^{\mathcal{C}}, \ldots, x_p^{\mathcal{C}}\right)$ respectivement les vecteurs caractéristiques de l'objet suivi $\mathcal{O}$ et de l'objet candidat $\mathcal{C}$. La pondération des arêtes du graphe est déterminée par la distance euclidienne normalisée, notée $D_{L2}(\mathcal{O}_i, \mathcal{C}_j)$ et donnée par

$$D_{L2}(\mathcal{O}_i, \mathcal{C}_j) = \sqrt{\sum_{k=1}^{p} \frac{\left(x_k^{\mathcal{O}} - x_k^{\mathcal{C}}\right)^2}{\sigma_k^2}} \tag{5.22}$$

avec σ_k^2 l'incertitude (variance) de la caractéristique x_k.

Une fois calculée, seules les arêtes dont les poids sont supérieurs à un seuil sont conservées. Il s'agit d'une étape de présélection des associations, permettant de limiter l'espace de recherche dans la mise en correspondance en imposant une proximité (mesure de similarité) entre un objet suivi et un candidat. Une fois les arêtes à faible poids supprimées, il peut subsister des ambiguités d'association, par exemple lorsque plusieurs objets candidats sont associés au même objet suivi ou vice-versa. Ainsi, des hypothèses d'associations sont générées selon les cinq cas possibles (voir Figure 5.8) :

– *Perfect match* (*One to One*) - Il s'agit d'une correspondance parfaite sans ambiguité. Un objet candidat (et un seul) est associé à un objet suivi.

- *Lost* (*One to None*) - Il n'y a pas de mesure associée à la cible, i.e. aucun objet candidat ne correspond à l'objet suivi.
- *New* (*None to One*) - Inversement au cas précédent, il n'y a pas de cible associée à la mesure, i.e. aucun objet suivi ne correspond à l'objet candidat.
- *Split* (*One to Many*) - Plusieurs candidats sont associés à une même cible, il s'agit d'une division d'objets.
- *Merge* (*Many to One*) - Plusieurs cibles sont associées à un même candidat, il s'agit d'une fusion d'objets.

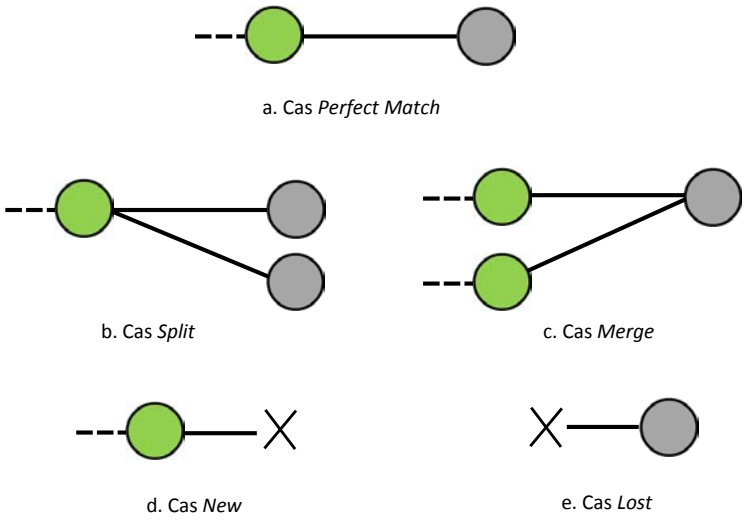


FIGURE 5.8: Représentation des cinq types d'hypothèses d'association possibles entre les objets suivis $\mathcal{O}$ et les objets candidats $\mathcal{C}$: le cas *Perfect Match* (*One to One*), le cas *Split* (*One to Many*), le cas Merge (*Many to One*), le cas *New* (*One to None*) et le cas *Lost* (*None to One*).

5.2 Algorithme de suivi d'objets

La section précédente a permi d'introduire le principe du filtrage prédictif et son application au suivi d'objets. Cette section décrit l'algorithme de suivi utilisé qui combine la procédure de mise en correspondance présentée dans la Section 5.1.3 associée à un filtrage de Kalman (section 5.1.2).

5.2.1 Vue générale de l'approche

La procédure complète est illustrée sur la Figure 5.9. Les *blobs* détectés dans l'image courante sont extraits et leurs caractéristiques sont estimées. L'ensemble des *blobs* forme la liste des candidats $\mathcal{C}_t$ à l'instant t. Cette liste $\mathcal{C}_t$ est comparée à la liste des objets suivis notée $\mathcal{O}_t$ à l'aide du processus d'appariement décrit dans la Section 5.1.3.

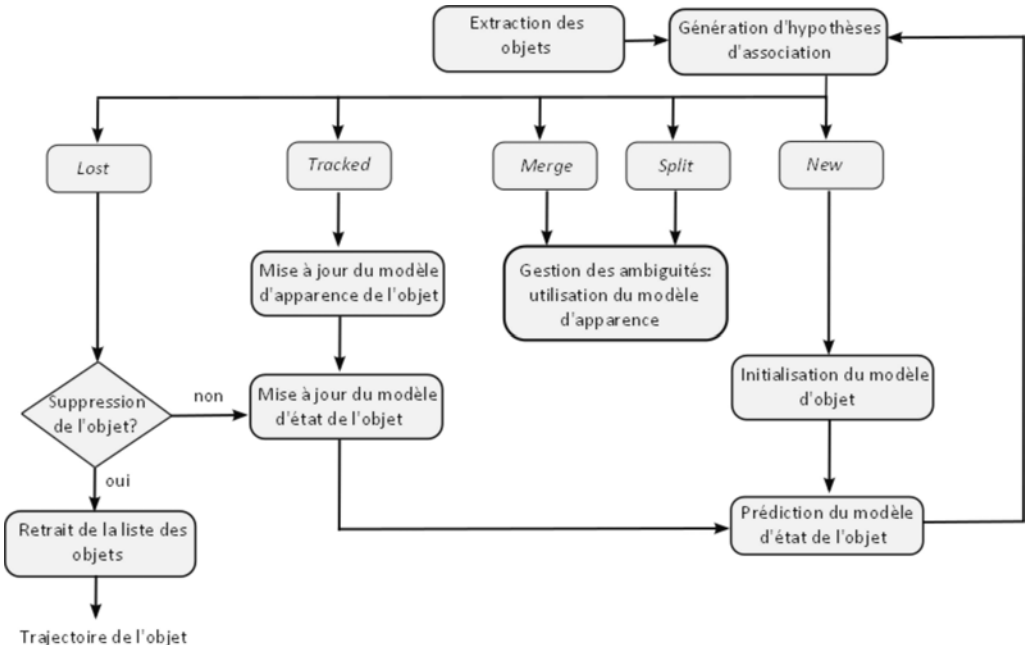


FIGURE 5.9: Représentation globale du système de suivi d'objets.

D'un point de vue algorithmique, le processus de suivi d'objets est cyclique et peut se décomposer en plusieurs étapes :

- L'*extraction* des objets et de leurs caractéristiques permet de construire une liste d'objets candidats de l'image courante.
- La *prédiction* détermine la position la plus probable de l'objet suivi dans l'image courante. La prédiction nécessite la connaissance des états de l'objet dans les images précédentes, ou d'un état initial fournit par le processus de détection d'objet. La prédiction est réalisée à l'aide d'un modèle de mouvement prédéfini, dans lequel est incorporé un modèle d'incertitude.
- La *mesure* consiste à détecter dans un alentour proche de l'objet suivi les candidats potentiels. Cette mesure permet de limiter le nombre d'associations possibles.
- La *mise en correspondance* consiste à comparer la position prédite d'un objet avec les observations afin d'identifier les correspondances. Seules les associations d'objets dont les caractéristiques sont proches sont conservées.
- La *mise à jour et l'estimation de l'état* de l'objet terminent le cycle en fournissant une estimation de l'état de l'objet (éventuellement les incertitudes associées). Cette mise à jour tient compte du modèle de mouvement choisi (prédiction) et des observations réalisées. Cette étape doit également prendre en compte les ambiguités présentes et les résoudre dans le processus d'association (division, fusion, nouvelle observation, objet perdu, ...).

5.2.2 Extraction des objets

L'étape d'extraction des objets consiste en une analyse du masque *foreground* obtenu par la segmentation de mouvement. Ce processus se décompose en plusieurs étapes comme

illustré sur la Figure 5.10.

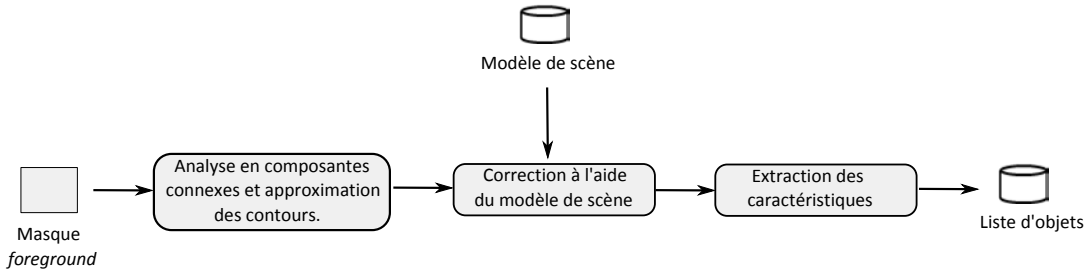


FIGURE 5.10: Les différentes étapes de l'extraction des objets de la carte *foreground* contenant une étape d'analyse en composantes connexes, une étape de simplification des contours extraits, une étape de correction à l'aide du modèle de scène et une étape d'extraction de caractéristiques des objets.

Analyse en composantes connexes et approximation des contours

Cette première étape consiste à extraire du masque *foreground* la connexité des régions à l'aide d'une analyse en composantes connexes [Suzuki 1985]. Cette opération (appelée aussi étiquetage) consiste à analyser l'image binaire d'entrée afin de fournir une image segmentée en sortie dans laquelle chaque région (*blob*) est identifiée. L'approche proposée dans [Suzuki 1985] fournit une liste de contours (liste de points) des régions connexes de l'image binaire.

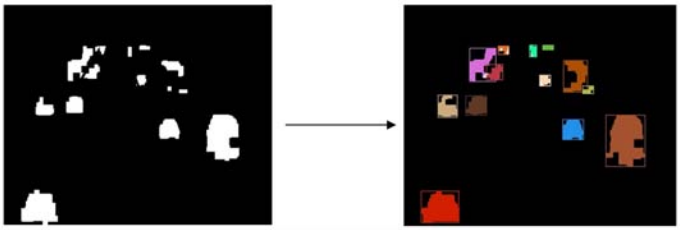


FIGURE 5.11: Extraction de *blobs* à l'aide d'un algorithme d'étiquetage basé sur l'extraction de contours d'une carte binaire [Suzuki 1985]. Une couleur aléatoire a été attribuée pour chaque région étiquetée et les boites englobantes des objets sont représentées.

La complexité de chaque contour (nombre de points) est ensuite réduite à l'aide de l'algorithme de Douglas-Peucker ([Douglas 1973], [Hershberger 1992]). Il s'agit d'une méthode de simplification d'un ensemble de points sur le principe de diviser pour régner (*divide and conquer algorithm*). Dans une première étape, le premier point et le dernier point de la chaîne sont reliés par un segment. L'ensemble des points entre le premier et le dernier de la chaîne sont analysés et les distances entre les points et ce segment sont estimées. Si une distance dépasse un seuil prédéfini ϵ, le point à la plus grande distance de la ligne est ajouté à la chaîne et devient à la fois le point de départ et le point d'arrivée du second et du premier nouveau segment créé. L'algorithme poursuit ces étapes pour chaque nouveau segment ajouté et se termine lorsque toutes les distances calculées sont inférieures au seuil

ϵ (fixé dans notre implémentation à 3% du périmètre initial). La Figure 5.12 illustre le principe de ces étapes sur un exemple simple.

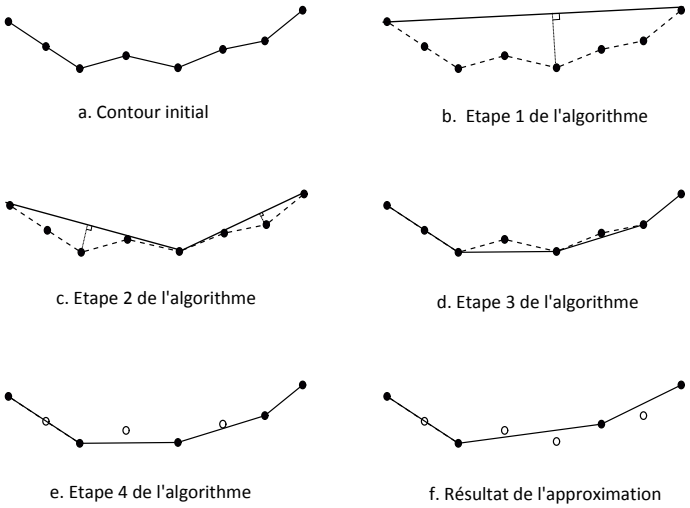


FIGURE 5.12: Illustration de l'algorithme de Douglas-Peucker pour l'approximation d'un contour.

Utilisation du modèle de scène

L'information sur la structure de la scène et sa sémantique associée sont utilisées pendant l'étape d'extraction des objets. Elles permettent d'associer à chaque objet la voie de circulation sur laquelle ils circulent, mais également d'aider à la segmentation de groupes d'objets. Lorsque les objets extraits sont de petites tailles (typiquement les véhicules légers et motos correctement segmentés) et que la caméra est en face de la scène sous surveillance, la majorité des pixels est généralement contenue à l'intérieur d'une voie de circulation du modèle de scène. Tandis que pour des régions de plus grandes tailles (poids lourds ou groupe de véhicules), ils occupent généralement plusieurs voies de circulation. Les pixels de chaque région extraite sont classés en tant que *correct* ou *ambigu*, où un pixel est considéré comme *ambigu* lorsqu'il n'appartient pas à la voie de circulation précédemment associée. Le modèle contenant la structure de la scène est utilisé pour séparer les éventuels groupes de véhicules issus de la segmentation. Les groupes de véhicules sont identifiés à l'aide de trois critères :

- Ratio de l'aire entre les pixels *ambigus* et l'aire totale de la région considérée est supérieur à un seuil.
- La largeur de l'objet est supérieure à la largeur de la voie de circulation.
- La différence des distributions couleur entre les pixels *ambigus* et ceux *corrects* est supérieure à un seuil.

Ces critères sont évalués pour chaque région détectée afin d'aider à la segmentation des objets. Si ces trois critères sont satisfaits (seuils définis expérimentalement), alors la région est découpée selon la bordure de voie contenue dans le modèle de scène. Quelques résultats de segmentation et de corrections sont illustrés sur la Figure 5.13.

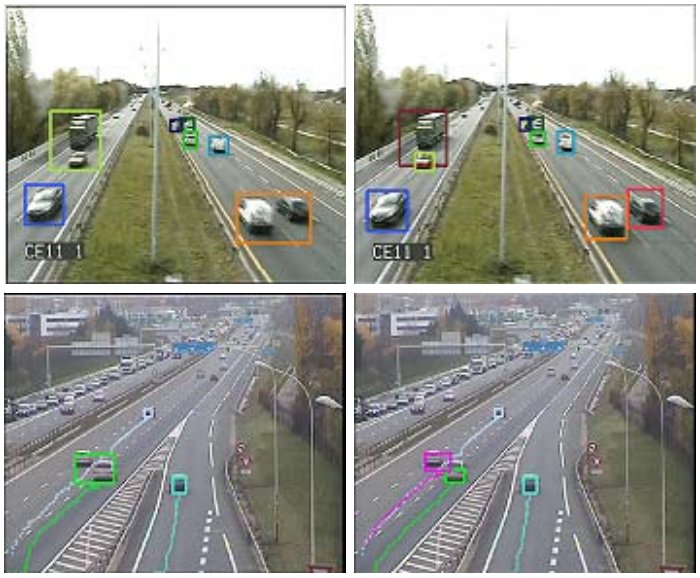


FIGURE 5.13: Exemple de correction apportée à la segmentation des objets en utilisant la structure du modèle de la scène. Les régions en mouvement sont correctement segmentées lorsqu'ils contiennent plusieurs objets

Extraction des caractéristiques

Une fois les contours extraits, les *blobs* sont analysés et filtrés pour constituer une liste d'objets. Dans l'objectif de caractériser la forme et le contenu colorimétrique des objets, un ensemble de caractéristiques est défini. Ces caractéristiques peuvent être regroupées en trois catégories :

- **Caractéristiques de formes**. Les descripteurs de formes fournissent une représentation 2D de l'objet et de sa forme telle qu'elle est perçue par la caméra. L'objectif de ces descripteurs est de fournir une description structurale de l'objet permettant de refléter les différences d'apparence visuelle et de structure. Sans doute la représentation la plus simple d'un objet est sa boite englobante. Cette boite englobante fournit la dimension spatiale et la position d'un objet en utilisant trois caractéristiques : le centre, la largeur et la hauteur de la boite englobante. Le ratio $ratio = hauteur/largeur$ ou encore la surface $aire = hauteur * largeur$ permet de fournir une caractéristique simple sur la taille de la boite englobante. Lorsque le contour de l'objet est connu, l'objet peut être représenté à l'aide d'une liste de points, fournissant une délimitation plus précise de l'objet que par sa simple boite englobante. Cette délimitation permet d'obtenir la densité d'un objet, définit comme étant le rapport entre le nombre de pixels appartenant à l'objet sur le nombre total de pixels de la boite englobante.

- **Caractéristiques d'apparences**. Les descripteurs d'apparences ont pour objectif de caractériser l'apparence de l'objet, généralement à l'aide de sa couleur ou de sa texture.

– **Caractéristiques sémantiques**. Les descripteurs sémantiques utilisent l'information contextuelle fournie par le modèle de la scène. Il s'agit d'associer, pour chaque objet, la position relative au modèle de scène. Le numéro de la voie et de la zone associée sont conservés dans un vecteur.

Les caractéristiques de formes, d'apparences et sémantiques utilisées sont rapportées dans les tableaux 5.2, 5.3 et 5.4. Quant aux caractéristiques relatives au vecteur d'état, elles sont rapportées dans le tableau 5.1. Pour chaque *blob* extrait du masque *foreground*, un vecteur contenant l'ensemble de ces caractéristiques lui est associé et sera utilisé dans les étapes suivantes de l'algorithme.

Caractéristiques du vecteur d'état	Description
(x, y)	Position du centre de la boite englobante
$(\dot{x}, \dot{y})$	Vecteur de déplacement du centre de la boite englobante
width	Largeur de la boite englobante
height	Hauteur de la boite englobante

TABLE 5.1: Caractéristiques contenues dans le vecteur d'état.

Caractéristiques de formes	Description
ratio	Rapport hauteur/largeur de la boite englobante (*width*/*height*).
area	Aire de la forme contenue dans la boite englobante
dispersedness	Rapport entre le carré du périmètre et l'aire de la forme
moments	Moments géométriques

TABLE 5.2: Caractéristiques de formes utilisées.

Caractéristiques d'apparences	Description
μ_R, μ_G, μ_B	Valeurs moyennes des composantes couleurs
σ_R, σ_G, σ_B	Variances des composantes couleurs
$\overline{Sg_x}$, $\overline{Sg_y}$	Sommes des magnitudes des gradients dans les directions x et y

TABLE 5.3: Caractéristiques d'apparences utilisées.

Caractéristiques sémantiques	Description
N_{lane}	Numéro de voie sur laquelle le véhicule circule
ID	Numéro identifiant (unique) de l'objet
C	Classe du véhicule

TABLE 5.4: Caractéristiques sémantiques utilisées.

5.2.3 Filtrage prédictif

Cette section présente la mise en place du filtrage prédictif. La configuration d'un objet est représentée par son vecteur d'état contenant sa position (x, y) et son vecteur vitesse (v_x, v_y). A l'instant t, le vecteur d'état $X_t^{(i)}$ pour un objet i s'écrit $X_t^{(i)} = \begin{bmatrix} x_t^{(i)} & y_t^{(i)} & \dot{x}_t^{(i)} & \dot{y}_t^{(i)} \end{bmatrix}^T$. Pour faciliter la lecture, nous omettrons de préciser l'indice $^{(i)}$ lorsqu'aucune ambiguité ne se présente. A l'aide des relations de Newton, il est possible d'écrire les équations (non linéaires) d'évolution du vecteur d'état telles que

$$\begin{cases} x_t = x_{t-1} + \dot{x}_{t-1}\Delta t + \frac{1}{2}\ddot{x}_{t-1}\Delta t^2 \\ y_t = y_{t-1} + \dot{y}_{t-1}\Delta t + \frac{1}{2}\ddot{y}_{t-1}\Delta t^2 \\ \dot{x}_t = \dot{x}_{t-1} + \ddot{x}_{t-1}\Delta t \\ \dot{y}_t = \dot{y}_{t-1} + \ddot{y}_{t-1}\Delta t \end{cases} \tag{5.23}$$

ou encore, écrit sous forme matricielle

$$\underbrace{\begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix}}_{x_t} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{F_{t-1}} \cdot \underbrace{\begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{bmatrix}}_{x_{t-1}} + \underbrace{\Delta t \cdot \begin{bmatrix} \frac{1}{2}\ddot{x}_{t-1}\Delta t \\ \frac{1}{2}\ddot{y}_{t-1}\Delta t \\ \ddot{x}_{t-1} \\ \ddot{y}_{t-1} \end{bmatrix}}_{w_t} \tag{5.24}$$

avec (x, y) la position prédite de l'objet, $(\dot{x}, \dot{y})$ le vecteur vitesse des objets, $(\ddot{x}, \ddot{y})$ le vecteur accélération des objets et Δt représente l'intervalle de temps entre deux prédictions.

Equations d'évolution

Le filtre de Kalman nécessite la description de l'évolution du vecteur d'état à l'aide un système dynamique linéaire. Dans le cadre du suivi d'objets, nous considérons un modèle de mouvement défini à l'aide d'un modèle autorégressif à l'ordre un. L'état estimé d'un objet est une extrapolation linéaire de l'état précédent à laquelle est ajouté un bruit gaussien représentant l'incertitude du modèle. Ceci revient à considérer le vecteur accélération ($\ddot{x}_t$, $\ddot{y}_t$) comme étant un bruit blanc gaussien noté w_t et centré en zéro. Cette hypothèse permet d'écrire la relation linéaire appelée équations d'évolution telles que

$$x_t = F_{t-1}x_{t-1} + w_t \tag{5.25}$$

Equations de mesures

Les équations de mesures traduisent la relation entre l'observation et le vecteur d'état. Le vecteur d'observation (ou de mesure) z est composé des positions x et y auxquelles est ajouté un bruit de mesure u supposé gaussien.

$$z_t = x_t + u_t \tag{5.26}$$

avec z_t la mesure à l'instant t de la position en x et en y d'un objet, et u_t un bruit de mesure supposé gaussien centré en zéro. Sous forme matricielle, les équations de mesure s'écrivent

$$\begin{bmatrix} z_{xt} \\ z_{yt} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{bmatrix} + \begin{bmatrix} u_{xt} \\ u_{yt} \end{bmatrix} \tag{5.27}$$
$$z_t = Hx_t + u_t$$

5.2.4 Génération d'hypothèses d'association

Le problème d'association des objets suivis avec les objets candidats est modélisé à l'aide d'un graphe pondéré noté G dans lequel chaque nœud représente un objet vidéo. Pour deux images consécutives I_{t-1} et I_t, le graphe est réduit à un graphe biparti : chaque nœud O_i^{t-1} du graphe est relié à un objet candidat C_j^t, avec i représentant le nombre d'objets suivis à l'instant $t-1$ et j le nombre d'objets détectés à l'instant t dans la nouvelle image. Un poids $w_{i,j}$ est associé à chaque arête du graphe et caractérise la similarité entre les objets des noeuds reliés. Le processus d'association consiste à relier chaque objet à l'instant $t-1$ avec ceux obtenus à l'instant t.

Le processus de mise en correspondance entre les objets candidats et les objets suivis comporte plusieurs étapes (Figure 5.14) :

1. Etape de prédiction. La position de l'objet est prédite en fonction de la vitesse et la direction qui lui sont associées. Si l'objet est nouveau, sa vitesse et son orientation sont initialisées à l'aide de la carte de mouvement obtenue lors de l'initialisation.
2. Mesure de similarité. Une mesure de similarité est effectuée entre les vecteurs caractéristiques des objets. Cette mesure fournit le poids accordé aux arêtes, les arêtes dont les poids sont inferieurs à un seuil T_w sont supprimés. Durant cette étape, des hypothèses d'association sont faites selon les cinq cas possibles : Correspondance parfaite (*Perfect Match*), Perdu (*Lost*), Nouveau (*New*), Divisé (*Split*) et Fusionné (*Merge*).
3. Mise en correspondance. La mise en correspondance consiste à associé les objets selon les hypothèses faites à l'étape précédente.
4. Mise à jour des objets. L'ensemble des caractéristiques des objets sont mis à jour.

Statut du suivi d'un objet

Nous définissons cinq états pour décrire la configuration d'un objet durant le processus de suivi. L'ensemble des états possibles sont listés dans la Table 5.5.

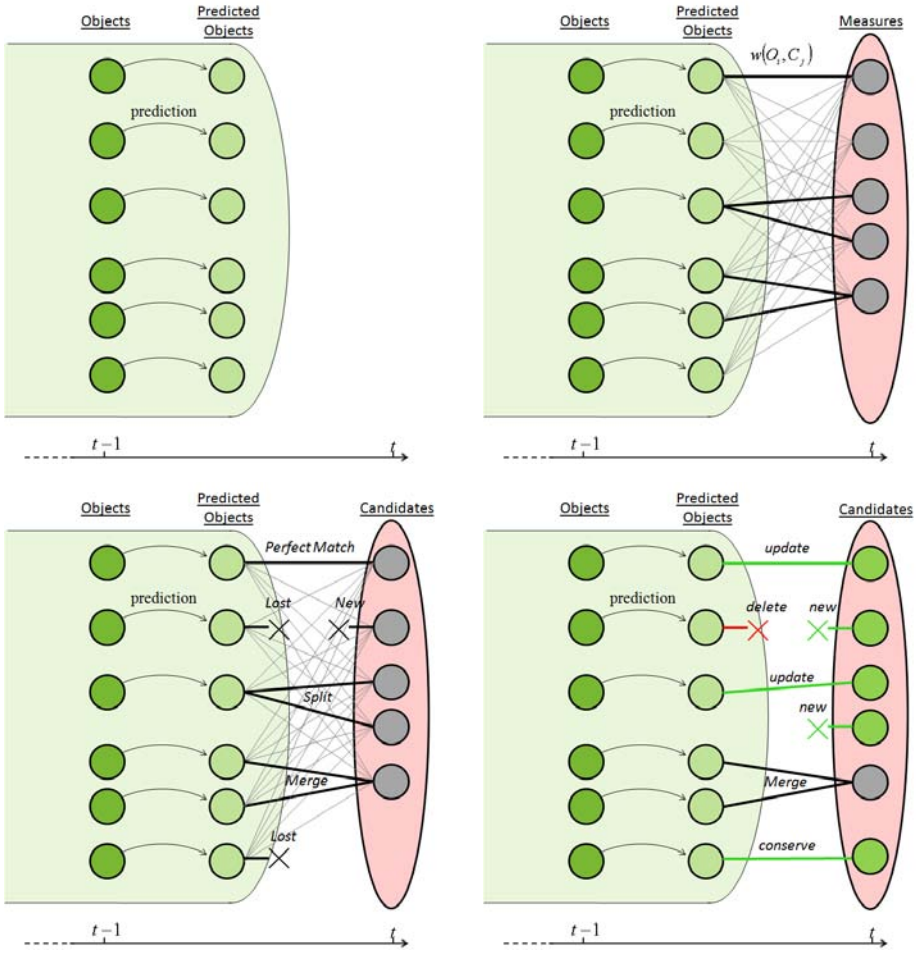


FIGURE 5.14: Les quatre étapes de la procédure de suivi d'objets : Prédiction, Mesure de similarité, Mise en correspondance et Mise à jour des objets.

Etat sémantique	Description
New	Un objet est assigné à l'état *New* lorsqu'il apparaît pour la première fois dans la scène. Il peut s'agir d'un véritable objet en mouvement ou d'un faux positif, nous introduisons une mesure de confiance permettant de mesurer la fiabilité de l'objet.
Tracked	L'objet est suivi et est considéré comme étant un véritable objet. Il peut arriver qu'un objet suivi ne soit pas détecté à l'image suivante (dû par exemple à une occlusion, la fusion avec d'autres objets ou sa sortie du champ de vision de la caméra, ...). Dans ce cas, un test est effectué pour déterminer s'il a fusionné (*Merge*) ou *Lost*.
Splitted	Un objet suivi s'est séparé en plusieurs objets. Il peut s'agir d'une fragmentation d'un objet, ou encore de la séparation d'un groupe d'objets.
Merged	Deux objets suivis ont fusionnés pour ne faire qu'un objet.
Lost	Un objet suivi n'a pas été détecté dans la nouvelle image.

TABLE 5.5: Description des états sémantiques pour le suivi d'objets parmi les cinq états possibles : *New* l'objet est nouveau dans la liste, *Tracked* l'objet est suivi et enregistré dans la liste, *Splitted* l'objet a été divisé, *Merged* l'objet a été fusionné, et *Lost* l'objet ne possède pas de candidat dans l'image.

Pré-selection des candidats

Une présélection des candidats (*gating process*) est effectuée dans l'objectif d'éliminer les paires Objet-Candidat peu probables. Un fenêtrage spatial est utilisé dans l'espace d'état et est généralement de forme rectangulaire, circulaire ou ellipsoïdale. Lorsqu'une mesure (observation) est incluse dans la fenêtre de recherche, elle devient candidate pour l'association avec l'objet suivi.

Mise en correspondance

Le processus de mise en correspondance consiste à associer à l'objet suivi le candidat le plus probable dans la liste d'objets détectés. Cette procédure rencontre une des situations suivantes :

- Une seule observation est contenue dans la fenêtre de validation. Il s'agit alors d'une correspondance parfaite.
- Plusieurs observations sont contenues dans la fenêtre de validation d'un objet suivi. Il existe une ambiguité dans l'association, on parle de division dans l'association.
- Une observation est contenue dans plusieurs fenêtres de validation, il existe également une ambiguité dans l'association, on parle de fusion.
- Une fenêtre d'observation ne contient aucune observation, dans ce cas, l'objet n'a pas été détecté, on parle d'objet perdu.
- Une observation n'appartient à aucune fenêtre de validation. Dans ce cas, l'observation est utilisée pour initialiser un nouvel objet à suivre.

Une matrice d'association, notée A de taille mxn est créée dans laquelle chaque élément $M[i;j]$ est égal à 1 si l'objet candidat $\mathcal{C}^{(i)}$ est associé à l'objet $\mathcal{O}^{(j)}$, 0 sinon.

Mise à jour des objets

Lors d'une correspondance parfaite (*Perfect Match*), l'objet est mis à jour avec le candidat correspondant. Un objet perdu (*Lost*) est supprimé seulement s'il est perdu pendant suffisamment de temps. Sinon, l'objet est conservé, mis à jour avec l'état obtenu après prédiction et on continue à le suivre. Les nouveaux objets (*New*) sont directement enregistrés en tant que nouveau nœud dans le graphe. Dans le cas d'une division ou d'une fusion d'objets (*Split* ou *Merge*), la résolution s'effectue à l'aide d'une validation temporelle décrite dans la section suivante.

5.2.5 Résolution des ambiguités

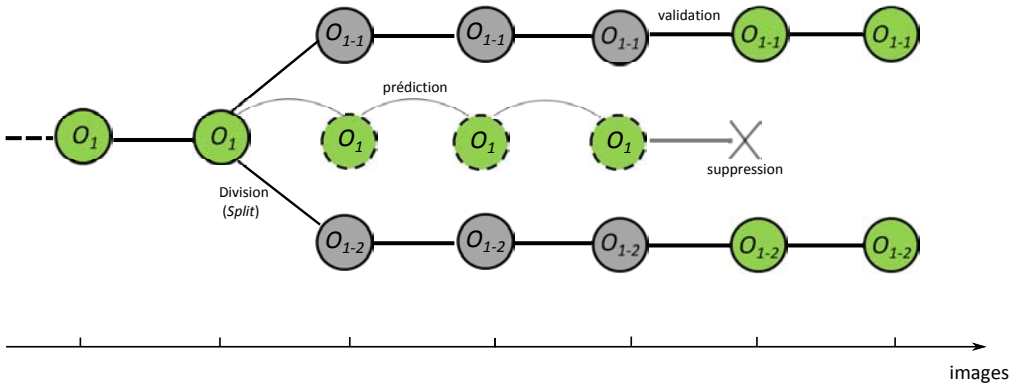
Les scénarii de division et fusion (*Split* et *Merge*) sont résolus en utilisant une information temporelle plus importante (suivi des objets à plus long terme). Dans les deux cas, les objets originaux sont conservés et mis à jour avec leurs états prédits. Dans un même temps, des objets fusionnés ou divisés sont créés et leurs existences restent en suspend. Si un objet a été divisé et re-fusionne plus tard, seul le résultat du suivi de l'objet original est conservé. Au contraire, si un objet a été fusionné et qu'il se divise plus tard, seul le résultat du suivi des objets originaux est conservé. Ceci permet de conserver une cohérence spatiale et temporelle de l'évolution des objets.

Division d'objets

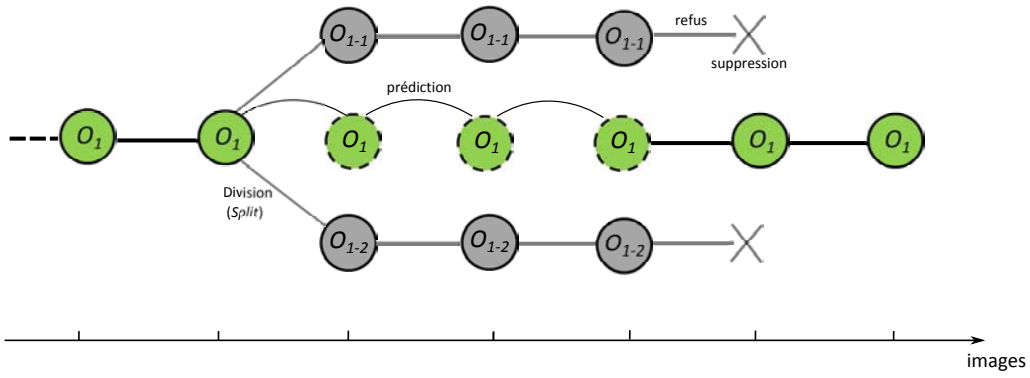
Lorsqu'une situation de division se produit, l'algorithme de suivi procède de la façon suivante. De nouveaux objets correspondants aux fractions d'objets sont créés et sont suivis à l'aide de leurs caractéristiques d'apparence. Quant à l'objet original est conservé, il est suivi à l'aide du filtrage prédictif et son existence reste en suspend. La division est validée dans plusieurs cas :

- Les objets résultants prennent des directions différentes.
- Les objets résultants sont correctement suivis pendant suffisamment longtemps dans les images suivantes (typiquement pendant $\tau_{div} = 5$images).

Si l'un des cas précédemment exposés se produit, alors la division est validée et l'objet original est supprimé. Les objets résultants voient leurs trajectoires enrichies par celles de l'objet original, comme illustré sur la Figure 5.15.



a. Validation de la division d'objets



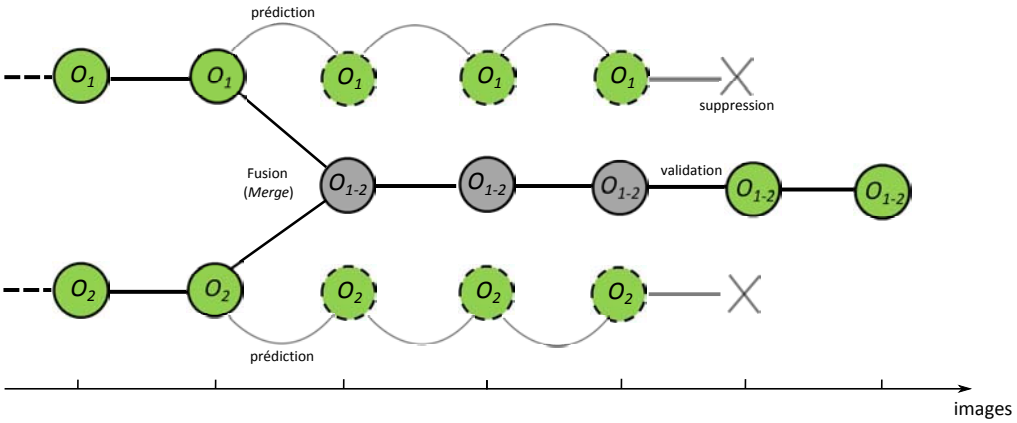
b. Refus de la division d'objets

FIGURE 5.15: Traitement de la division d'objets, exemple de validation (a) et de refus (b) pour la division d'un objet O_1 en deux objets O_{1-1} et O_{1-2}.

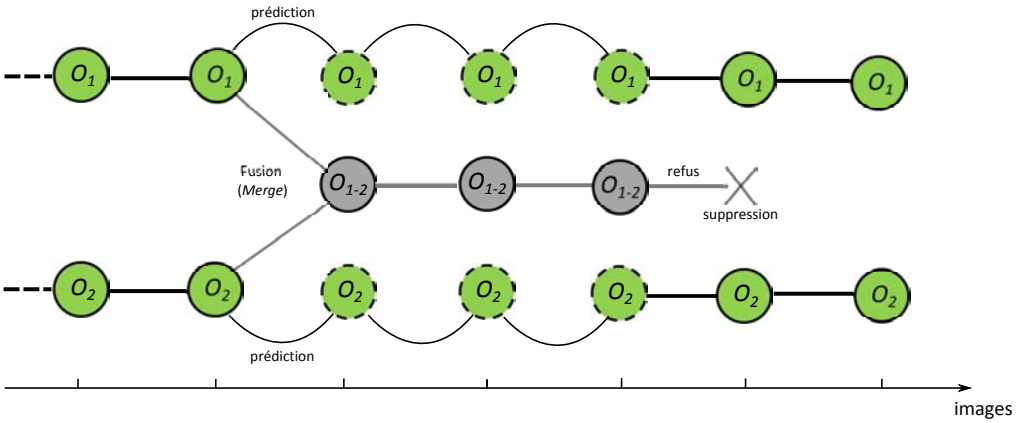
Fusion d'objet

Inversement, lorsqu'une occlusion est détectée les objets impliqués basculent dans l'état *Merged*. Pendant la durée de l'occlusion, un nouvel objet est créé correspondant à la fusion

des objets. Celui-ci est suivi normalement jusqu'à la fin de l'occlusion. Les trajectoires des objets sont ensuite corrigées à l'aide du suivi indépendant des objets. Cette procédure est illustrée sur la Figure 5.16.



a. Validation de la fusion d'objets



b. Refus de la fusion d'objets

FIGURE 5.16: Traitement de la fusion des objets, exemple de validation (a) et de refus (b) pour la fusion de deux objets O_1 et O_2.

Suivi des objets temporaires

Le processus de mise en correspondance basé sur l'apparence des objets consiste à calculer le coefficient de corrélation entre l'image de l'objet candidat et celle de l'objet suivi. Le coefficient de corrélation $r_{\mathcal{O}\_\mathcal{C}}$ est calculé pour toutes les positions possibles dans un voisinage proche de l'objet suivi. En notant F et T respectivement la région de l'image d'un objet candidat de taille w x h (issu du masque *foreground*) et la région de l'image de

l'objet suivi de taille W x H (*template*), le coefficient de corrélation est donné par

$$Corr(i,j) = \frac{\sum_i \sum_j \left[F(x,y) - \overline{F}\right] . \left[T(x-i, y-j) - \overline{T}\right]}{\sqrt{\sum_i \sum_j \left[F(x,y) - \overline{F}\right]^2 . \sum_i \sum_j \left[T(x,y) - \overline{T}\right]^2}} \tag{5.28}$$

avec

$$\begin{aligned} \overline{F} &= \frac{1}{wh} \sum_{x=1}^{w} \sum_{y=1}^{h} F(x,y) \\ \overline{T} &= \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} T(x,y) \end{aligned} \tag{5.29}$$

Ce coefficient traduit la similarité entre l'image F et le template T dans laquelle chaque image est ramenée à sa moyenne et normalisée. Ce coefficient est maximum et vaut 1 lorsque l'image F et le template I sont parfaitement identiques. Le déplacement du template est estimé en déterminant le déplacement donnant un coefficient de corrélation maximum, tel que

$$(dx, dy) = \arg\max_{(i,j)} Corr(i,j) \tag{5.30}$$

Utilisation du modèle de scène

Les paragraphes précédents décrivent la procédure utilisée pour traiter les ambiguités dans le processus d'association. Pour aider à la validation des objets en suspend (objets originaux dans le cas de fusion et objet original dans le cas de division), le modèle de scène est utilisé de la façon suivante :

Lorsqu'une ambiguité est détectée dans une zone d'entrée z_{in}, alors elle est automatiquement validée. Puisque les bordures des voies sont utilisées pour l'extraction des objets (voir Section 5.2.2), le cas d'un objet fusionné sur plusieurs voies ne peut pas se produire.

Inversement, lorsqu'une ambiguité est détectée dans une zone de sortie z_{out}, alors elle est refusée. Autrement dit, aucun nouvel objet n'est créé, et les objets originaux sont conservés et suivis à l'aide de la prédiction faite par le filtre prédictif.

Enfin, lorsqu'une ambiguité est détectée dans la zone de circulation z_{circ}, la procédure décrite précédemment est appliquée, et de nouveaux objets en suspend sont créés. Lorsque ceux-ci atteignent une zone de sortie, alors l'étape de validation (ou le refus de validation) s'effectue et les trajectoires sont corrigées.

L'utilisation du modèle de scène permet ainsi de faciliter la gestion des ambiguités. La Figure 5.17 illustre un exemple de correction apporté en utilisant cette approche.

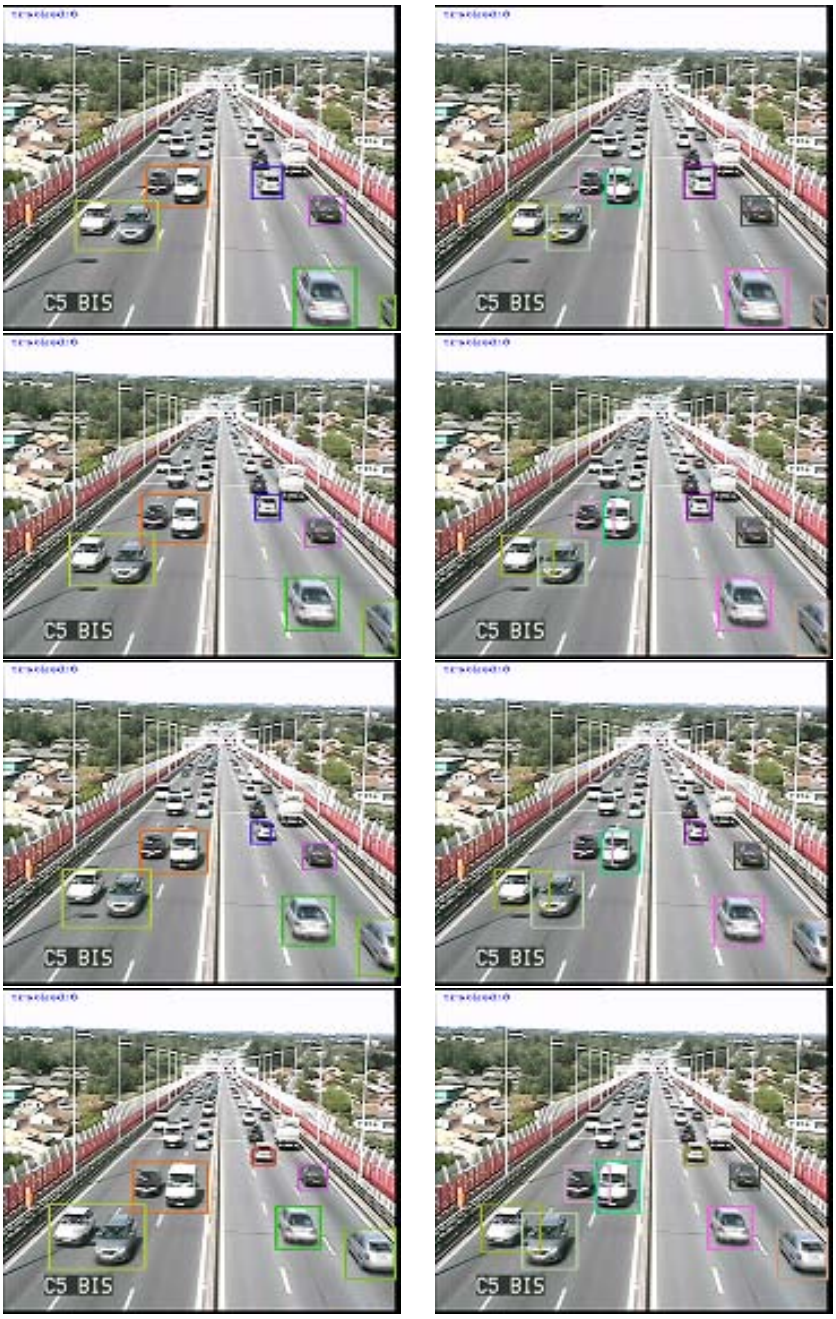


FIGURE 5.17: Exemple de traitement des ambiguités en utilisant l'approche proposée. (A gauche) Sans traitement des ambiguités. (A droite) Avec traitement des ambiguités.

5.3 Résultats expérimentaux

Les tests de performances sont conduits sur les vidéos C5, Lyon, C11, HighwayII en analysant l'extraction des objets détectés ainsi que leurs trajectoires. Pour cette analyse, la vérité-terrain a été annotée manuellement pour de courtes séquences (entre 800 images pour la vidéo HighwayII et 7500 images pour la vidéo C5) à l'aide de l'outil *viper-toolkit*[1]. Les trajectoires pour chaque objet sont construites à partir du centre des boites englobantes sélectionnées manuellement.

Les seuils utilisés pour segmenter les groupes d'objets à l'aide des bordures des voies ont été déterminé expérimentalement. Dans le cadre de cette évaluation, le ratio de l'aire entre les pixels *ambigus* et l'aide totale de la région considérée est fixé à $\tau_{area} = 0.65$, quant à la différence des distributions couleur, elle est obtenue par différence entre les couleurs moyenne, à l'aide d'une distance euclidienne. Le seuil de différence utilisé est égal à $\lambda_{color} = 30$ et la différence maximale des composantes couleurs est celle retenue pour la comparaison.

Dans notre implémentation, les paramètres Q et R sont fixes, l'ensemble des autres variables est estimé dynamiquement. Le paramètre de covariance de l'état P est mis à jour automatiquement à chaque itération de l'algorithme. Les valeurs des paramètres utilisées par l'algorithme sont :

$$F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \tag{5.31}$$

5.3.1 Métriques d'évaluation

L'évaluation de l'extraction des objets est effectuée de la façon suivante. Le masque des objets en mouvement est étiquetté à l'aide d'une analyse en composantes connexes, chaque groupe de pixels est ensuite mis en association avec les objets de la vérité-terrain de la façon suivante : si l'aire de recouvrement entre le résultat de l'algorithme et la vérité-terrain est supérieure à un seuil (fixé à 85%) alors l'objet est considéré comme associé à celui de la vérité-terrain (noté $AR_{\mapsto GT}$). Un objet de la vérité-terrain ne peut être associé qu'à un seul objet de l'algorithme. Si deux objets issus de l'algorithme sont associés au même objet de la vérité-terrain, alors celui possédant l'aire de recouvrement la plus grande lui est associé et le second objet est considéré comme étant un faux positif. Cette mise en association permet d'estimer les performances en termes de précision, de rappel auxquels

1. http ://viper-toolkit.sourceforge.net/

est ajouté une mesure de F-score.

$$\text{Det}_{objet} = \begin{cases} \text{Precision} & = \dfrac{\#AR_{\mapsto GT}}{\max(\#AR, 1)} \\ \text{Rappel} & = \dfrac{\#AR_{\mapsto GT}}{\max(\#GT, 1)} \\ \text{F-score} & = \dfrac{2.\text{Precision.Rappel}}{\text{Precision} + \text{Rappel}} \end{cases} \tag{5.32}$$

Quant à l'évaluation du processus de suivi des objets, elle est obtenue en comparant l'aire de recouvrement entre les boites englobantes des objets détectés et celles des objets de la vérité-terrain : si cette aire de recouvrement est supérieure à un seuil (fixé à 85% de l'aire minimum dans notre implémentation), alors l'objet est associé à la vérité-terrain. Chaque trajectoire est comparée aux trajectoires des objets de la vérité terrain de la façon suivante : si un objet de l'algorithme est associé au même objet de la vérité-terrain pendant la majorité de sa durée de vie (fixée à 85% de la durée de vie totale de l'objet dans la vérité-terrain), alors il est considéré comme correctement suivi (TP, *True Positive*). Si deux objets sont associés au même objet de la vérité-terrain, alors celui possédant le plus grand recouvrement spatial est conservé et le second objet est considéré comme étant un faux positif (FP, *False Positive*). Ces considérations permettent d'estimer les performances du suivi d'objets en terme de Precision, Rappel et de F-score, donnés par

$$\text{Sui1} = \begin{cases} \text{Precision} & = \dfrac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Rappel} & = \dfrac{\text{TP}}{\text{TP} + \text{FN}} \\ \text{F-score} & = \dfrac{2.\text{Precision.Rappel}}{\text{Precision} + \text{Rappel}} \end{cases} \tag{5.33}$$

5.3.2 Résultats

L'ensemble des résultats sont présentés dans les Tables 5.6 et 5.7 et illustrés sous la forme d'un graphique sur les Figures 5.18 à 5.22.

Extraction des objets	C5	Lyon	CE 11	Highway
Nombre d'images	830	11250	7490	800
Nombre d'objets	51	83	100	32
Precision	0.96	0.96	0.93	0.90
Rappel	0.88	0.71	0.78	0.90
F-score	0.91	0.82	0.84	0.90

TABLE 5.6: Résultats de l'évaluation des performances du processus d'extraction d'objets sur quatre séquences de test : C5, Lyon, CE 11 et Highway. Les résultats sont présentés en termes de Précision, Rappel et F-score sur l'ensemble des trajectoires analysées avec utilisation du modèle de scène.

Suivi des objets	C5	Lyon	CE 11	Highway
Nombre d'images	830	11250	7490	800
Nombre d'objets	51	83	100	32
Precision	0.93	0.88	0.89	0.94
Rappel	0.78	0.89	0.90	0.91
F-score	0.85	0.88	0.89	0.92

TABLE 5.7: Résultats de l'évaluation des performances de l'algorithme de suivi d'objets sur quatre séquences de test : C5, Lyon, CE 11 et Highway. Les résultats sont présentés en termes de Précision, Rappel et F-score sur l'ensemble des trajectoires analysées.

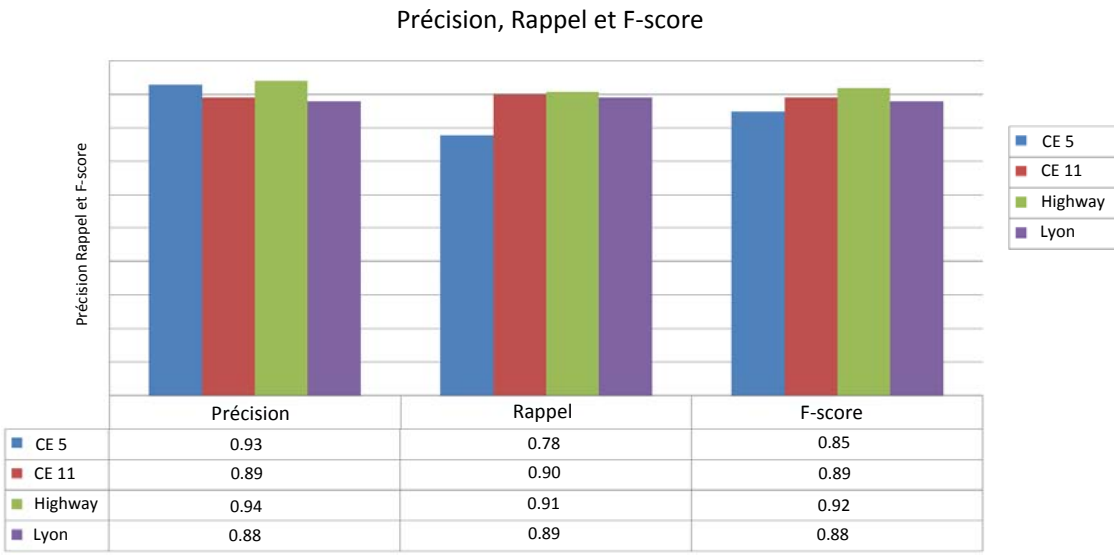


FIGURE 5.18: Représentation graphique des résultats de l'analyse des performances du suivi d'objets.

Les résultats montrent des performances de suivi d'objets avec une valeur de Precision comprise entre 0.88% et 0.94% et un Rappel compris entre 78% et 91% dépendant de la complexité de la vidéo.

La vidéo C5 est une séquence courte (830 images) mais contenant de nombreux objets (51 objets). Le faible taux de précision à 78% est du aux nombreuses fusions d'objets causées par la proximité des objets dans la scène. Malgré ce faible taux, la Precision conserve une très bonne valeur égale à 93% ce qui fournit une F-score de 85%. La figure 5.19 montre quelques résultats de suivi des objets de la scène.

La séquence Lyon est un séquence longue (11250 images) contenant relativement peu d'objets (83 objets), mais ceux-ci sont particulièrement proches sur la voie d'insertion ce qui provoque des erreurs de suivis d'objets avec de nombreuses situations de fusions d'objets. Les résultats obtenus fournissent une Precision de 88% et un Rappel de 89%, soit un F-score de 88%. La Figure 5.20 illustre les résultats du suivi d'objets pour quelques images de la séquence.

La séquence C11 est une séquence plus longue que la première (7490 images) représentant un trafic également beaucoup plus fluide (100 objets). Les véhicules sont relativement bien espacés ce qui permet d'obtenir de bons résultats aussi bien en Precision (89%) qu'en Rappel (90%) ce qui fournit un F-score de 89%. Quelques résultats de suivi d'objets sont présentés sur la Figure 5.21.

La séquence Highway, quant à elle est une séquence courte (800 images) contenant peu d'objets (33 objets). Les objets sont également bien séparés permettant de nouveau d'obtenir de bons résultats, avec 94% en Precision, 91% en Rappel ce qui donne 92% en F-score. La Figure 5.22 fournit quelques résultats du suivi d'objets de la séquence.

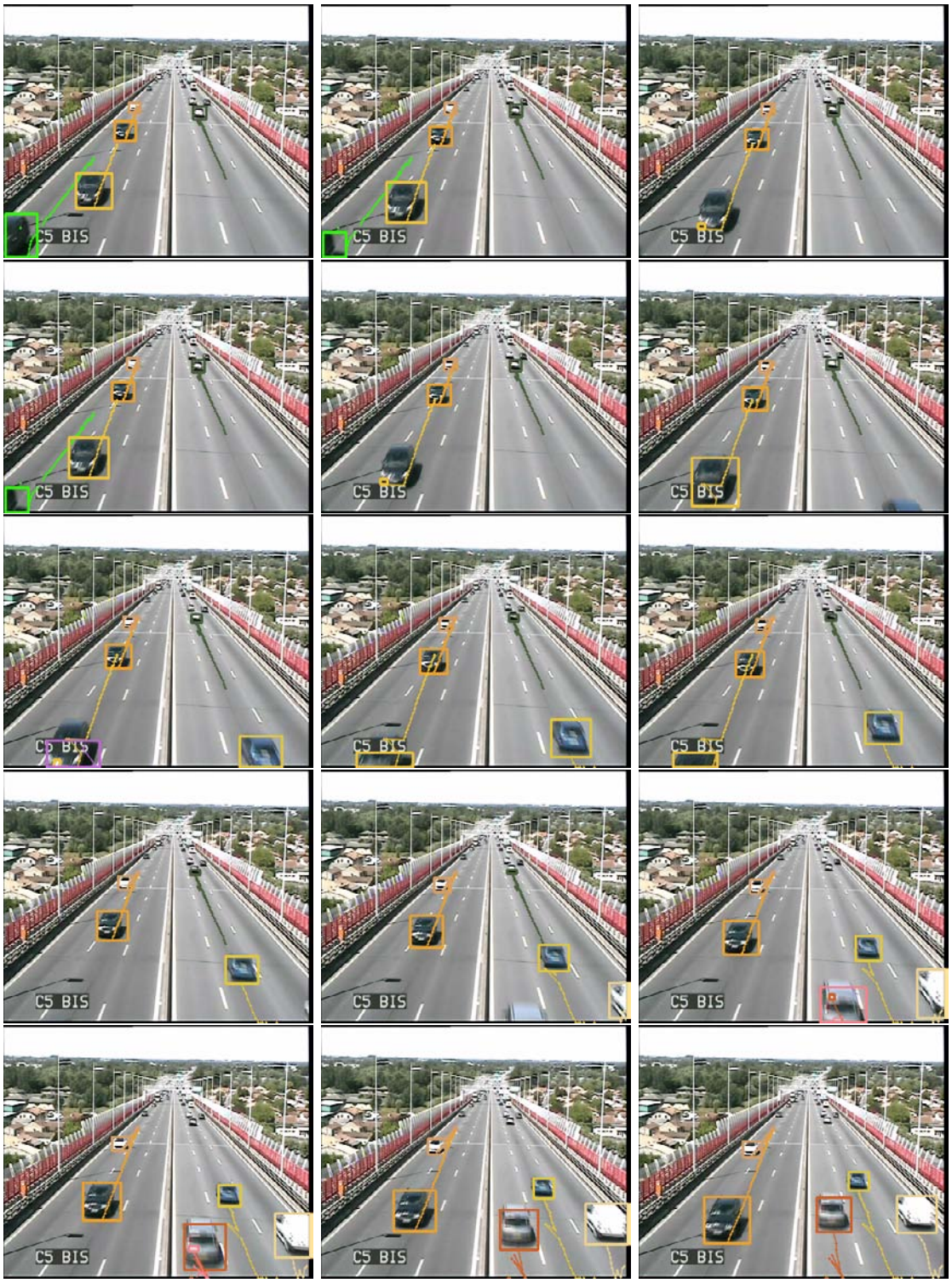


FIGURE 5.19: Exemple de trajectoires issues du processus de suivi d'objets pour la séquence C5.

FIGURE 5.20: Exemple de trajectoires issues du processus de suivi d'objets pour la séquence Lyon.

FIGURE 5.21: Exemple de trajectoires issues du processus de suivi d'objets pour la séquence C11.

FIGURE 5.22: Exemple de trajectoires issues du processus de suivi d'objets pour la séquence Highway.

5.4 Conclusion

Nous avons vu dans ce chapitre le processus de suivi d'objet utilisé par notre système. Celui-ci se décompose en quatre étapes : l'extraction des objets, le filtrage prédictif, la génération d'hypothèses d'association et la mise à jour et la gestion des ambiguités.

Nous avons dans un premier temps développé le principe du filtrage bayésien avant de présenter le cas particulier du filtrage de Kalman dans lequel les densités de probabilité sont toutes considérées comme gaussiennes. Ainsi, le suivi d'objet se décompose en une étape de prédiction basée sur les configurations précédentes, une étape de mesure dans laquelle la configuration de l'observation est extraite et une étape de correction permettant d'ajuster la configuration d'un objet ainsi que les incertitudes associées.

L'utilisation du modèle de la structure de la scène a été utilisée pour aider à la segmentation des objets, notamment lorsque des objets sont proches et considérés comme un groupe de véhicules. Les bordures des voies sont utilisées pour aider à la séparation des groupes de véhicules. Lorsqu'une région en mouvement occupe plusieurs voies de circulation, alors un test est effectué en comparant le nombre et les caractéristiques des pixels de la voie de circulation principale de l'objet avec ceux contenus dans les voies de circulation voisines.

Pour rendre le suivi d'objet multi-cible, un graphe bi-parti est utilisé et permet de modéliser et d'émettre des hypothèses d'association. Chaque objet détecté est mis en correspondance avec les objets suivis (enregistrés dans une liste d'objets) et des hypothèses d'association sont générées parmi les cinq cas possibles : *Perfect Match*, *Lost*, *New*, *Merged*, *Split*. Les configurations des objets *Perfect Match*, *Lost* et *New* sont ensuite mises à jour. Quant aux cas de *Merged* et *Split*, ils sont traités à l'aide d'une information temporelle plus grande.

Dans le cas d'associations ambigues telles que la division ou la fusion d'objets, un traitement particulier est appliqué à l'aide d'un suivi à plus long terme. En cas de division ou de fusion, les objets originaux sont conservés et de nouveaux objets (divisés ou fusionnés) sont créés. Une validation temporelle permet de valider ou de refuser la division ou la fusion.

Chapitre 6

Analyse comportementale des objets : détection d'évènements

Sommaire

6.1 Introduction

Ce chapitre présente les méthodes utilisées pour l'analyse de haut-niveau sémantique du contenu vidéo. La détection d'évènements dans les vidéos constitue la dernière étape de la chaîne de traitement. Il s'agit d'une étape essentielle durant laquelle les informations issues des analyses de bas-niveau et de niveau intermédiaire sont interprétées en une description sémantique de haut niveau. La première section de ce chapitre traite de l'analyse de comportement des objets. Il s'agit principalement d'une analyse des trajectoires permettant d'extraire des statistiques sur l'état du trafic et de détecter les changements de voies des véhicules. La seconde section est consacrée à la détection d'incidents dans les scènes autoroutières. Deux types d'incidents sont présentés : la détection de véhicules en contre-sens et la détection de véhicules à l'arrêt. Dans la troisième section, nous présentons une approche différente basée sur la génération de carte spatio-temporelles (carte 1d+t). Cette approche permettra d'extraire des statistiques sur l'état du trafic, de détecter la présence d'un bouchon ou d'un véhicule à l'arrêt.

6.2 Analyse de comportement

L'analyse de comportement est basée sur une analyse des résultats de la détection de mouvement et du suivi des objets. Trois types d'évènements sont considérés : la détection de véhicules en contre-sens, la détection de changements de voies de véhicules et la détection d'arrêt. La détection de contre-sens compare les vecteurs mouvements de l'image avec le modèle de sens de direction du trafic. Les vecteurs s'écartant du modèle sont extraits afin de former les véhicules en contre-sens. Un filtrage spatial et temporel permet de supprimer les bruits et fausses détections. Quant à la détection de changements de voie, elle est basée sur l'analyse des trajectoires des objets vis-à-vis des positions des voies. Une machine à états finis permet d'attribuer un état aux véhicules en fonction de leur position dans la scène. Finalement, la détection d'arrêt s'appuie sur la détection des objets dans la scène. Un pixel est considéré comme appartenant à un objet à l'arrêt si sa couleur s'écarte du modèle couleur (pixel *foreground*) pendant suffisamment longtemps et que son intensité lumineuse ne varie pas dans le temps.

6.2.1 Détection de véhicules en contre-sens

La détection de contre-sens se base sur l'information de mouvement obtenue par l'estimation du flot optique (voir Section 4.2.3). La détection se déroule en deux étapes : une étape d'estimation des vecteurs mouvements et une étape de comparaison au modèle statistique appris lors de l'initialisation du système.

Estimation des vecteurs mouvements

L'estimation des vecteurs mouvements est la première étape du module de détection de contre-sens. Notons $\mathcal{M}$ la carte des vecteurs mouvements telle que $M(i,j) = (d_x, d_y)$, avec d_x et d_y respectivement les déplacements en x et en y du pixel $(i,\ j)$ considéré. Notons ρ

la norme et θ l'orientation du vecteur $\vec{d} = (d_x, d_y)$ telles que

$$\begin{aligned} \rho &= \sqrt{d_x^2 + d_y^2} \\ \theta &= \arctan \frac{d_y}{d_x} \end{aligned} \tag{6.1}$$

La carte des vecteurs $\mathcal{M}$ est obtenue lors de l'estimation du flot optique (voir Section 4.2.3). Elle est dans un premier temps filtrée en ne conservant les vecteurs mouvements que si leur norme est supérieure à un seuil τ_ρ (fixé à 1 pixel dans notre implémentation). Ceci permet de supprimer les vecteurs de faibles amplitudes caractéristiques des bruits de mesure et d'acquisition.

Comparaison au modèle

Une fois estimés, les vecteurs sont comparés au modèle statistique appris lors de l'étape d'apprentissage (voir Section 3.1.5). Rappelons que le modèle est représenté sous la forme d'un mélange de lois de type von-Mises s'écrivant sous la forme

$$p(\mathbf{x}|\mu, \kappa) = \sum_{k=1}^{K} w_k p_{vM}(\mathbf{x}|\mu_k, \kappa_k)$$

avec

$$p_{vM}(\mathbf{x}|\mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \exp^{(\kappa cos(\mathbf{x}-\mu))}$$

Durant l'apprentissage, le nombre de composantes K du mélange a été fixé et les paramètres du modèle $\Phi = \{w_1, \ldots, w_K, \mu_1, \ldots, \mu_K, \kappa_1, \ldots, \kappa_K\}$ pour chaque pixel ont été estimés à l'aide d'une séquence d'apprentissage (voir Section 3.1.6). Ainsi, la différence d'angle est calculée pour chaque pixel contenant un vecteur déplacement de norme $\rho > \tau_\rho$ et d'orientation θ telle que

$$d\theta = \arg\min_k(\theta - \mu_k, 2\pi - (\theta - \mu_k)) \tag{6.2}$$

Si la différence d'angle $d\theta$ est supérieure à un seuil prédéfini τ_θ, le pixel est considéré comme appartenant à un objet en contre-sens. Les pixels détectés comme étant *anormaux* sont ensuite étiquetés à l'aide d'une analyse en composantes connexes et les contours de petites tailles sont supprimés. Un filtrage temporel médian (taille 3) est également appliqué pour éliminer les faux positifs dûs aux erreurs d'estimation du flot optique.

Résultats de l'analyse

Cette méthode a été appliquée sur une séquence de test modifiée manuellement pour simuler le déplacement des véhicules en contre-sens. La voie de circulation de droite a été manuellement sélectionnée et *inversée* temporellement simulant une marche arrière des véhicules pendant un temps prédéfini.

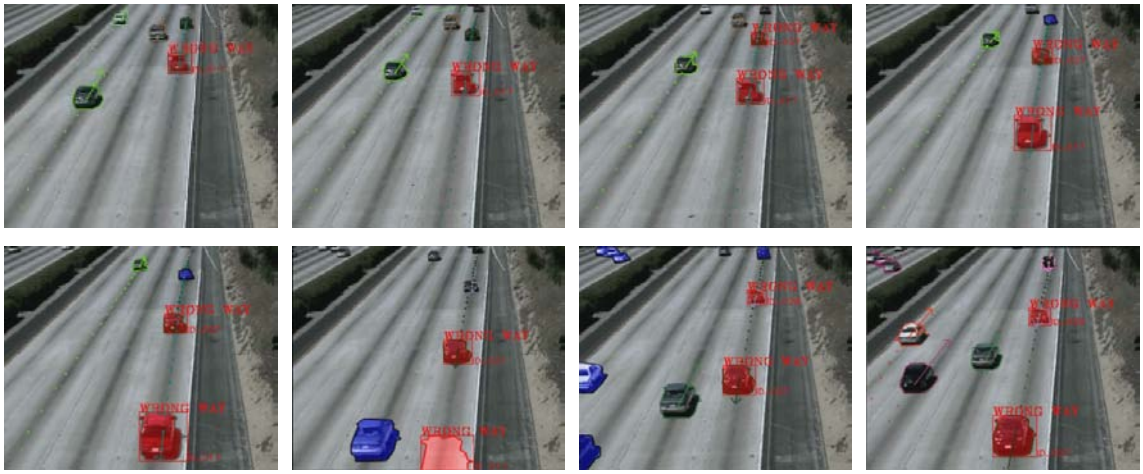


FIGURE 6.1: Illustration de la détection de véhicules en contre-sens sur une séquence vidéo modifiée. La voie de circulation de droite a été sélectionnée et mise à l'envers pour simuler le contre-sens. La détection de contre-sens est illustrée à l'aide de boites englobantes rouges sur les véhicules détectés.

La Figure 6.1 illustre les résultats obtenus par la méthode présentée. Dans le cadre de ce test, une soustraction d'arrière-plan permet d'extraire les objets en mouvement Pour chaque objet extrait, les vecteurs mouvements sont estimés et comparés au modèle. L'évènement de contre-sens a été correctement détecté pour cette séquence modifiée. Cependant, la séquence utilisée ne reflète pas de façon significative une situation réelle et d'autres tests doivent être menés sur des séquences réelles.

6.2.2 Détection de changements de voies

Nous nous intéressons maintenant aux changements de voie effectués par les véhicules. Cette information peut être utile pour caractériser le dynamisme du comportement des véhicules, mais peut également être utilisée pour valider la présence d'un véhicule arrêté qui fait office d'obstacle aux véhicules qui le précèdent. L'approche utilisée est une analyse de l'historique des positions (trajectoires) des véhicules sur les voies. Comme illustré sur la Figure 6.3, chaque voie de circulation est divisée en deux parties : une zone centrale et une zone de transition. Ces zones sont définies par la moitié et le quart de la largeur de chacune des voies. Durant le processus de suivi, une machine à états finis est utilisée pour définir la position des véhicules relative aux voies de circulation (Figure 6.2). Dans notre cas, la position d'un véhicule est définie par le centre du segment du bas de sa boite englobante.

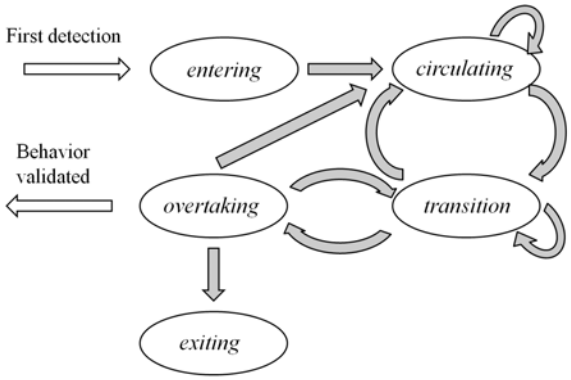


FIGURE 6.2: Machine à états utilisée pour la détection de changements de voies des véhicules.

La machine à états finis procède de la façon suivante. La première fois qu'un véhicule est détecté dans la scène, l'état *entering* lui est assigné. Puis, tant qu'il circule dans la zone centrale d'une voie de circulation, le véhicule possède l'état *circulating* associé au numéro de voie sur laquelle il circule. S'il passe la limite de la zone de transition, alors l'objet passe dans l'état *transition* et tant que sa position n'a pas entièrement changé de voie, le changement de voie n'est pas validé. De cette façon, en dehors de l'entrée ou la sortie de l'objet dans la scène, il ne peut être que dans trois états possibles : *circulating*, *transition* (l'état de changement de voie n'a pas encore été validé) ou *overtaken*. Notons que si un objet reste dans l'état *transition* pendant une longue période de temps, ceci peut être considéré comme une situation anormale.

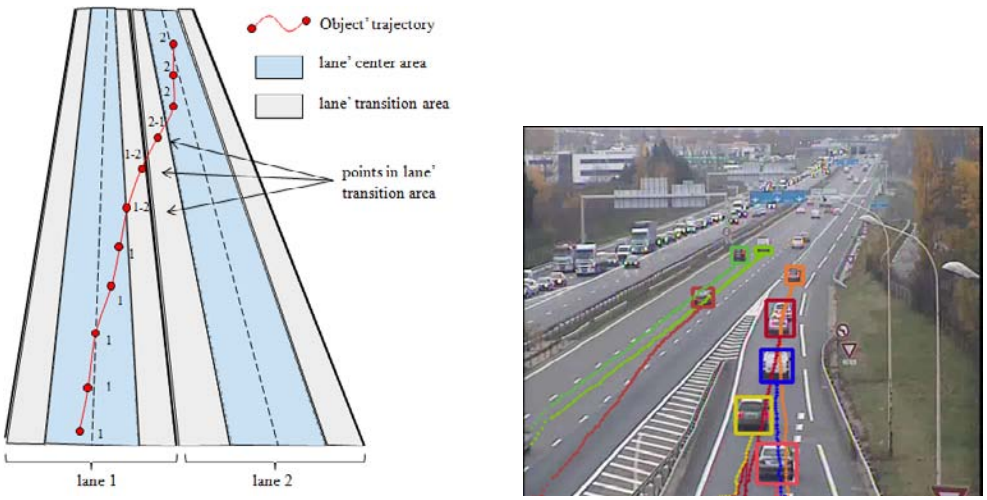


FIGURE 6.3: (**À gauche**) - Exemple théorique de trajectoire d'un véhicule qui change de voie. Sa trajectoire passe par la zone de transition avant d'atteindre la zone centrale de la voie adjacente. (**À droite**) - Exemple réel d'un changement de voie d'un véhicule (trajectoire rouge).

Les tests sont conduits sur trois séquences de test sur lesquelles le nombre de changements de voie a été comptabilisé manuellement. Le Tableau 6.1 montre les résultats obtenus

pour la détection de changements de voie. Les changements de voie des véhicules ont été correctement détectés pour les séquences de test avec un taux de bonne détection de plus de 90%.

Changements de voie	C5	CE 11	Lyon
Nombre d'images	800	7490	11250
Nombre d'objets	51	100	83
Nombre d'évènements (changements de voie)	5	30	20
Nombre d'évènements détectés	5	27	18
Taux de bonne détection	1.0	0.90	0.90

TABLE 6.1: Résultats du comptage du nombre de changements de voie pour les séquences C5, CE 11 et Lyon.

6.2.3 Détection de véhicules à l'arrêt

La détection d'objets abandonnés a fait l'objet de nombreuses recherches pour la surveillance d'espaces publics. Appliquées à la surveillance d'une scène autoroutière, ces méthodes peuvent détecter et identifier un véhicule à l'arrêt. Dans une situation normale (trafic fluide sans embouteillage), les véhicules circulent continuellement sur la route, ce qui se traduit par un masque d'objets en mouvement (*foreground*) continuellement en mouvement et les objets se déplacent dans le sens de direction du trafic. Par conséquent, un objet à l'arrêt est détecté comme un *blob* spatialement immobile et considéré comme appartenant à l'avant-plan pendant un certain temps. La détection d'objets à l'arrêt présenté dans cette section contient trois étapes (Figure 6.4) :

1. Identification des pixels *statiques*
2. Détection des pixels ne changeant pas de couleur
3. Validation spatio-temporelle

L'identification des pixels *statiques* s'effectue à travers l'analyse du masque *foreground* des objets en mouvement fourni par le processus de soustraction d'arrière-plan. L'objectif consiste à détecter l'ensemble des pixels classés en tant qu'avant-plan pendant suffisamment longtemps. La détection des pixels ne changeant pas de couleur permet d'identifier, parmi les pixels précédemment détectés, ceux qui conservent la même couleur. Finalement, une étape de validation permet d'éliminer le bruit et les fausses alarmes en regroupant les pixels ne changeant pas de couleur pendant suffisamment longtemps. Les petits groupes de pixels sont supprimés et seuls ceux conservant approximativement la même taille sont conservés.

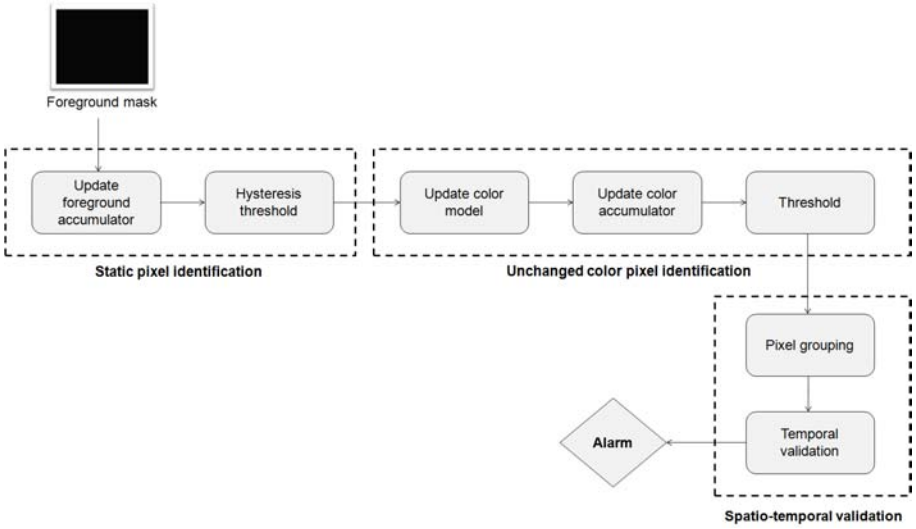


FIGURE 6.4: Processus de détection d'objets à l'arrêt.

Identification des pixels *statiques*

Les pixels *statiques* sont identifiés en analysant le masque d'avant-plan. Il s'agit d'une analyse par pixel utilisant une image accumulateur de même taille que l'image d'entrée. Chaque pixel du *foreground* est analysé : les valeurs de l'accumulateur sont incrémentées (respectivement décrémentées) en fonction de leur appartenance à l'avant-plan (respectivement l'arrière-plan). Lorsque les positions des pixels *foreground* ont été accumulées, un seuillage par hystérésis est utilisé pour valider les pixels *statiques*. Si la valeur correspondante de l'accumulateur dépasse le seuil haut T_{high}, alors le pixel est considéré comme étant *statique*. Celui-ci doit alors retrouver une valeur inférieure au seuil bas T_{low} pour être de nouveau considéré comme étant en mouvement.

Détection des pixels conservant la même couleur dans le temps

Une fois les pixels *statiques* identifiés, l'étape suivante consiste à modéliser leur couleur pour évaluer ceux conservant la même information colorimétrique. Chaque pixel *statique* est modélisé par une distribution gaussienne, paramétrée par sa valeur moyenne μ et son écart-type σ estimés par la même procédure que celle décrite dans la Section 4.2. Pour comptabiliser la durée de conservation de la couleur du pixel, un second accumulateur est utilisé et une procédure analogue à celle employée pour détecter les pixels *statiques* remplti cet accumulateur. Cependant, plutôt que d'utiliser un seuillage par hystérésis, un simple seuillage à l'aide d'une valeur seuil T_{color} est employé.

Validation spatio-temporelle

Cette dernière étape consiste à valider et supprimer les bruits impulsionnels présents dans l'accumulateur. Les pixels considérés comme étant *statiques* et ayant conservés la même couleur sont regroupés à l'aide d'une analyse en composantes connexes. Les pixels isolés et le bruit sont supprimés à l'aide d'un critère spatio-temporel : les petits objets sont

supprimés et seuls ceux conservant approximativement la même taille pendant un temps prédéfini sont conservés.

Résultats de l'analyse

Cette approche a été testée sur trois séquences de test. La première a été modifiée manuellement pour simuler l'arrêt de deux véhivules sur la chaussée de droite, tandis que les deux séquences suivantes correspondent à des situations réelles. Il s'agit de séquences publiques dans lesquelles des véhicules s'arrêtent sur le bord de la route. Les résultats obtenus sont illustrés sur les Figures 6.5, 6.6 et 6.7.

La Figure 6.5 montre la détection avec succès de l'arrêt des véhicules sur la chaussée de droite. Les véhicules ont été correctement détectés sur cette séquence modifiée qui, comme dans le cas de la détection de contre-sens ne reflète pas exactement une situation réelle d'arrêt (déviation des véhicules précédents par exemple), mais permet cependant de valider l'approche sur une séquence simple.

Sur les Figures 6.6 et 6.7 sont illustrées les détections d'arrêts sur les séquences I-LIDS-Medium et I-LIDS-Easy [1] dans lesquelles une camionette et un véhicule léger s'arrêtent sur le bord de la route. L'arrêt d'un véhicule est représenté par l'affichage de sa boite englobante sur la vidéo. Les véhicules arrêtés on été correctement détectés pour les séquences, avec également la détection d'arrêt du véhicule situé au stop sur la séquence I-LIDS-Easy (Figure 6.6) qui ne s'engage pas immédiatement sur la route.

FIGURE 6.5: Illustration de la détection d'arrêt de véhicules sur la séquence de test C5. La séquence a été manuellement modifiée pour simuler l'arrêt de véhicules sur la chaussée. La détection d'arrêt est affichée à l'aide de boites englobantes rouges sur les véhicules détectés.

1. http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

FIGURE 6.6: Illustration de la détection d'arrêt de véhicules sur la séquence de test publique I-LIDS-Easy. La détection d'arrêt est affichée à l'aide de boites englobantes rouges sur les véhicules détectés.

FIGURE 6.7: Illustration de la détection d'arrêt de véhicules sur la séquence de test publique I-LIDS-Medium. La détection d'arrêt est affichée à l'aide de boites englobantes rouges sur les véhicules détectés.

6.3 Cartes spatio-temporelles pour l'analyse du trafic

En supposant la représentation d'une vidéo sous la forme d'un volume d'images de dimensions $(x\,;y\,;t)$, avec respectivement les dimensions $(x\,;y)$ de l'image et la dimension temporelle t. Une carte spatio-temporelle est une représentation (1d+t) issue d'une coupe dans le volume d'images d'une séquence vidéo. Cette coupe est définie selon un segment AB (appelé *scanline*) de l'image et selon l'axe temporel t de la séquence. Sur la Figure 6.8 sont illustrées les constructions des coupes horizontales et verticales définies selon le segment AB de l'image. Notons que cette coupe se généralise en une coupe transversale et n'est pas restreinte au cas horizontal ou vertical.

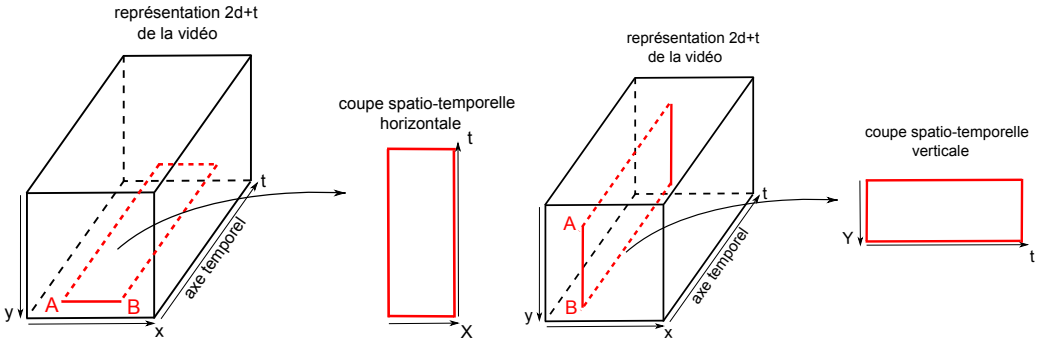


FIGURE 6.8: Illustration de la construction de cartes spatio-temporelles dans le cas particulier d'une coupe horizontale (à gauche) et verticale (à droite).

Dans le cadre de cette étude, les segments AB sont définis le long des voies de circulation comme les exemples illustrés sur les Figures 6.9 et 6.10. La carte spatio-temporelle est composée de discontinuités de couleur et de texture qui reflète les passages des véhicules le long du segment.

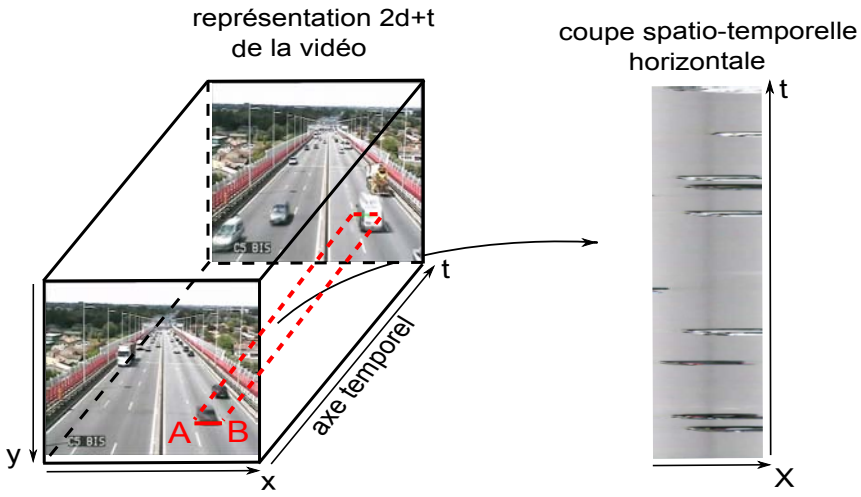


FIGURE 6.9: Construction d'une carte spatio-temporelle horizontale sur la largeur d'une voie de circulation de la séquence C5.

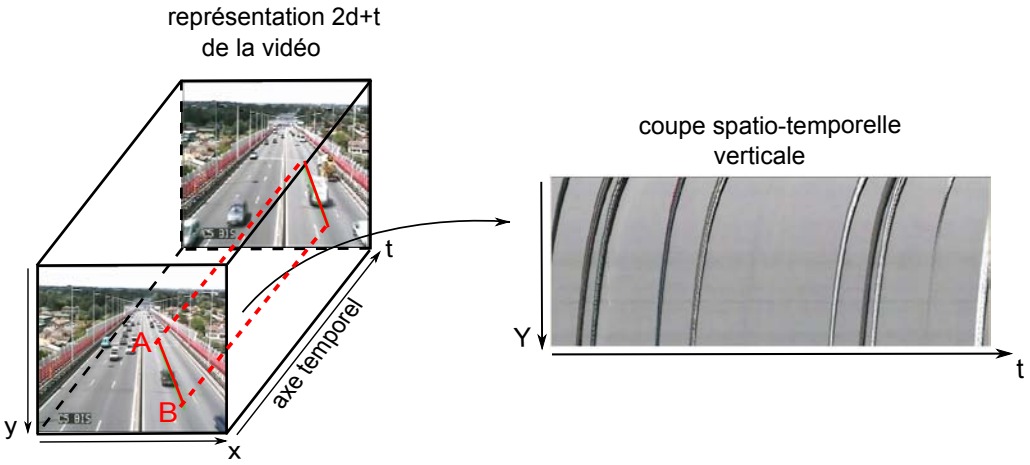


FIGURE 6.10: Construction d'une carte spatio-temporelle verticale le long d'une voie de circulation de la séquence C5.

L'analyse du trafic routier à l'aide des cartes spatio-temporelles est possible grâce, notamment, aux déplacements structurés des véhicules sur les voies de circulation (trajectoires quasi-linéaires sur l'ensemble du déplacement). Lorsque les véhicules sont suffisemment espacés (distance inter-véhicule élevée), les cartes spatio-temporelles sont caractérisées par un ensemble de *batônnets* traduisant le passage d'un véhicule (voir Figure 6.13). Ainsi, le débit du trafic Q_j de la voie de circulation j peut être estimé en comptabilisant le nombre de *bâtonnets* de la carte spatio-temporelle et en le divisant par le temps (en secondes) de l'accumulation de la carte. La comptabilisation des véhicules sur la carte spatio-temporelle peut se révéler difficile lorsque le débit du trafic devient élevé. En effet, plus les véhicules sont espacés spatialement sur la voie de circulation, plus les *bâtonnets* sont espacés dans le temps sur la carte. Inversement, lorsque le trafic devient saturé ou lorsque les distances de sécurité ne sont pas respectées, il devient difficile de bien segmenter les véhicules. De plus, la forme et l'orientation des *bâtonnets* sont caractéristiques du type de véhicule et de la vitesse à laquelle il circule. En effet, plus la largeur du *bâtonnets* est élevée, plus la taille du véhicule est grande. Et plus la pente du *bâtonnet* est élevée, plus sa vitesse est grande. Notons que le sens de direction du trafic est directement donné par le signe de la pente du *bâtonnet*.

Ainsi, la représentation à travers une carte spatio-temporelle permet d'obtenir de nombreuses informations telles que le nombre de véhicules, leur taille, vitesse et sens de direction. Dans le cadre de l'analyse du trafic, l'utilisation d'une telle représentation (1d+t) a été utilisée avec succès dans le système VISATRAM [Zhu 2000] et présentée également dans les travaux de [Malinovskiy 2009]. Dans leurs travaux, les auteurs ajoutent une étape de correction perspective permettant de corriger la forme courbée des *bâtonnets* causée par le ralentissement apparant des véhicules lorsqu'ils s'éloignent de la caméra. Le segment AB (*scanline*) est alors défini sur l'image corrigée. Le processus complet est décrit dans la section suivante.

6.3.1 Génération d'une carte spatio-temporelle

Basé sur les travaux de [Malinovskiy 2009], l'algorithme proposé se décompose en plusieurs étapes (Figure 6.11) :

- **Initialisation des *scanlines***. L'utilisateur sélectionne les *scanlines* et la zone de détection. Cette zone de détection est utilisée pour calculer la matrice de transformation perspective permettant d'obtenir une vue synthétique du dessus.
- **Construction des cartes spatio-temporelles**. Les cartes spatio-temporelles sont construites par accumulation des profils d'intensité lumineuse le long des *scanlines* précédemment définies.
- **Analyse des cartes spatio-temporelles**. Il s'agit d'une étape d'extraction des profils de véhicules (comptage des profils, estimation de la largeur et hauteur, ...).

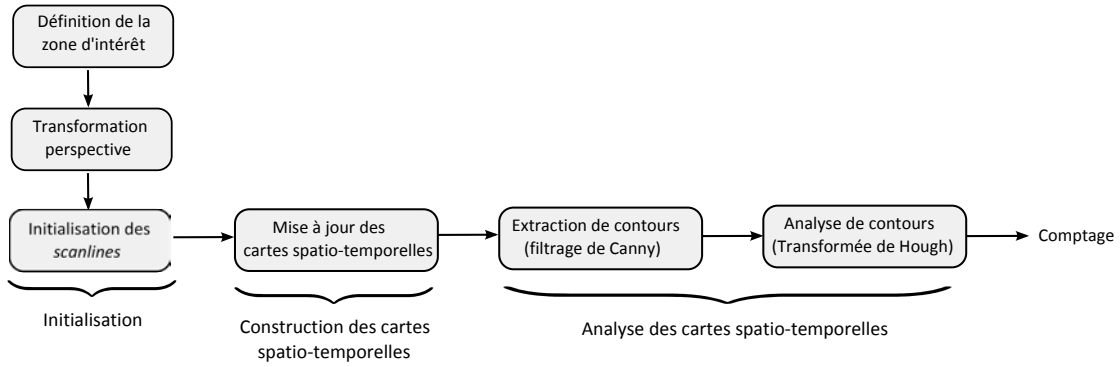


FIGURE 6.11: Utilisation des *scanlines* pour le comptage de véhicules et l'extraction de caractéristiques du trafic routier.

Initialisation des *scanlines*

L'étape d'initialisation consiste à sélectionner le segment AB pour la construction des cartes spatio-temporelles. Dans l'objectif de corriger la distorsion provoquée par la perspective dans l'image, une transformation perspective est appliquée nécessitant la sélection de quatre points dans l'image. La sélection du segment AB est sélectionnée sur l'image corrigée. Ces quatre points correspondent à la zone de détection délimitée par les bordures des voies. Dans le modèle de scène présenté dans le Chapitre 3, la zone de détection correspond aux coins supérieurs de gauche et de droite des sous-bandes les plus éloignées de la caméra et des coins inférieurs gauches et droits des sous-bandes les plus proches de la caméra. Une fois le polygone défini, la matrice de transformation perspective est estimée. Une transformation perspective est en une projection d'un plan S_1 vers un autre plan S_2. Soit $M = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ un point homogène du plan S_1 (plan d'origine) et $m = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ un point homogène du plan S_2 (plan d'arrivée). La forme générale d'un projection perspective est définie à l'aide des équations suivantes [Heckbert 1989] :

$$\begin{aligned} x &= \frac{au + bv + c}{gu + hv + 1} \\ y &= \frac{du + ev + f}{gu + hv + 1} \end{aligned} \tag{6.3}$$

Ecrites sous forme matricielle, ces équations deviennent :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} u & v & 1 \end{bmatrix} . \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \tag{6.4}$$

avec $a-h$ les huit coefficients de la matrice de transformation à déterminer. Ces coefficients sont estimés à l'aide de quatre points M_1 - M_4 de coordonnées connues et dont les positions correspondantes sur le plan d'arrivée m_1 - m_4 sont connues également. On obtient ainsi un système de 8 équations à résoudre (possédant 8 inconnues).

Cette étape est réalisée de façon automatique à l'aide du modèle de scène. Les segments sont automatiquement définis à partir des centres des voies obtenus soit à l'aide des trajectoires des objets (voir Section 3.2.2), soit après estimation des bordures des voies (voir Section 3.2.1).

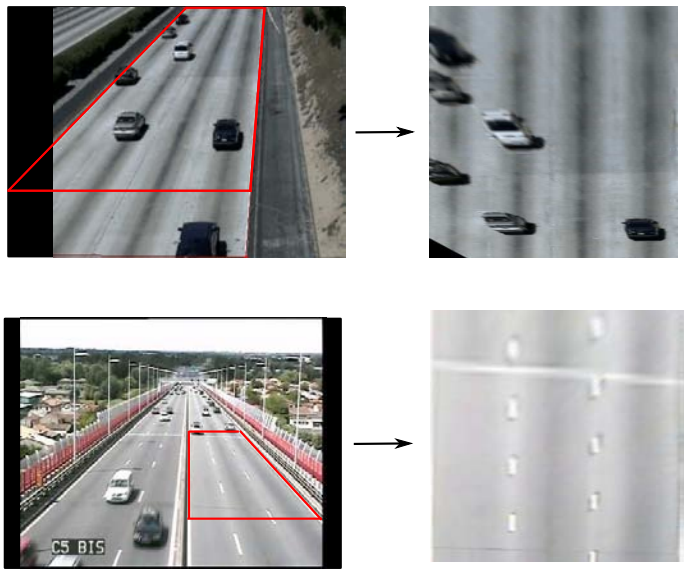


FIGURE 6.12: Illustration de la transformation perspective de l'image originale (à gauche) vers l'image transformée (à droite) pour la séquence Highway.

Construction des cartes spatio-temporelles

La construction des cartes spatio-temporelles consiste à accumuler les lignes de profil d'intensité lumineuse au cours du temps, avec une durée d'accumulation τ_{acc} fixe et définie par l'utilisateur. La taille d'une carte dépend donc de la durée d'accumulation (largeur de la carte) et de la longueur de la *scanline* (hauteur de la carte).

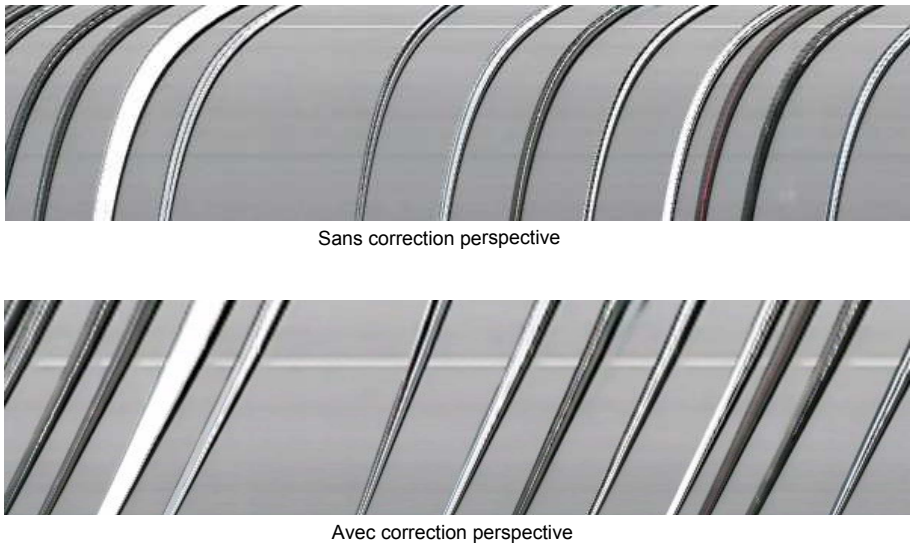


FIGURE 6.13: Illustration de la correction apportée par la transformation perspective sur la séquence de test C5.

La Figure 6.13 montre l'exemple d'une carte spatio-temporelle pour la voie de circulation de la chaussée de gauche de la séquence CE11. La transformation perspective a été appliquée et la *scanline* sélectionée manuellement. Pour une bonne visibilité et dans un soucis d'affichage, la carte est représentée avec une durée d'accumulation $\tau_{acc} = 24$ secondes. La transformation perpective permet de réduire les effets de perspective lorsque les véhicules s'éloignent de la caméra. L'analyse d'une telle représentation permet d'obtenir de nombreuses informations sur l'état du trafic.

Analyse des cartes spatio-temporelles

L'analyse des cartes spatio-temporelles a pour objectif d'extraire les informations relatives à l'état du trafic : statistiques sur le nombre d'objets, leur vitesse et leur taille par exemple. L'analyse s'effectue à travers la segmentation de la carte à l'aide des informations de couleurs ou de contours.

L'utilisation de la couleur est basée sur le modèle statistique présenté au Chapitre 4. Cette approche est identique à la soustraction d'arrière-plan présentée, mais restreint aux pixels contenus sur les lignes de profil (*scanline*). Ainsi, pour chaque pixel de la *scanline*, un mélange de lois gaussiennes est utilisé pour caractériser la distribution des couleurs du pixel considéré. Les équations de mises à jour utilisées sont identiques à celles présentées dans la Section 4.2. Une fois appris, le modèle est utilisé pour comparer chaque nouvelle valeur au modèle et effectuer la soustraction d'arrière-plan. Les pixels s'écartant du modèle sont ensuite étiquetés et comptabilisés.

L'analyse des cartes à travers les contours consiste à extraire les discontinuités représentatives du passage des objets. Dans un premier temps, les contours sont estimés à l'aide d'un filtrage de Canny. La carte de contours est ensuite analysée à l'aide de la transformée de Hough afin d'extraire les droites présentes sur les *bâtonnets* de la carte. Cette détection de ligne, basée sur l'équation paramétrique d'une droite et de ses paramètres ρ et θ, permet d'obtenir les segments de droites relatifs aux *bâtonnets* de la carte. Les segments sont

ensuite regroupés pour former les objets.

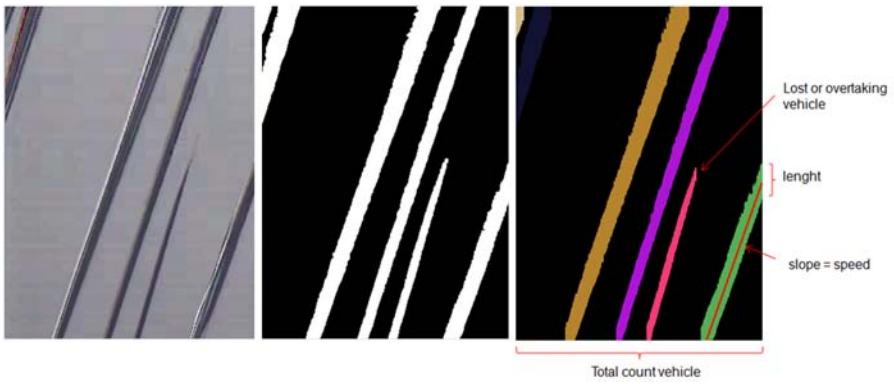


FIGURE 6.14: Exemple d'analyse couleur d'une carte spatio-temporelle ; la carte spatio-temporelle (à gauche) subit une soustraction d'arrière-plan (au centre) permettant d'extraire les objets (à droite).

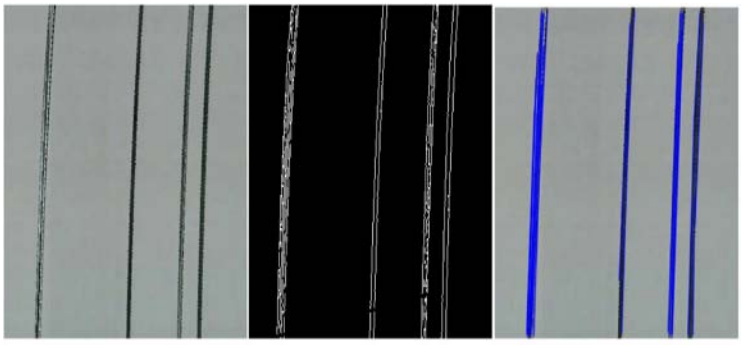


FIGURE 6.15: Exemple d'analyse des contours d'une carte spatio-temporelle ; la carte spatio-temporelle (à gauche) subit une détection de contours à l'aide de l'opérateur de Canny (au centre). La carte de contours est ensuite analysée à l'aide de la transformée de Hough pour extraire les droites représentatives des objets (à droite).

6.3.2 Application au comptage de véhicules

Le comptage de véhicules est une fonctionnalité largement répandue dans les systèmes de vidéo-surveillance de trafic routiers ; cette information permet de prévenir un trafic dense ou encore d'améliorer l'aménagement et la logistique du réseau routier. L'utilisation d'un système de vision permet de s'affranchir de l'utilisation de boucles inductives nécessitant des travaux sur la chaussée et des coûts d'entretien. L'utilisation des informations issues du processus de suivi permet d'obtenir de façon immédiate cette information, nous proposons cependant, dans notre système, l'utilisation de cartes spatio-temporelles [Malinovskiy 2009] pour effectuer cette tâche.

Le comptage de véhicules est effectué sur les séquences de test C5, CE11, HighwayII, Lyon à l'aide d'une segmentation couleur basée sur le modèle statistique présenté au Chapitre 4. La période d'apprentissage utilisée est de 600 images (600 échantillons pour chaque

pixel de la *scanline*). Une fois appris, le modèle est mis à jour régulièrement à l'aide des équations présentées dans la Section 4.2.1 (Equations 4.20 et 4.22). Les résultats obtenus en utilisant la procédure précédemment décrite sont présentés dans la Table 6.2. Notons que pour cette application, les segments ont été placés au centre des voies de circulation de la chaussée face à la caméra, et correspondent à des segments de petites tailles comme illustré sur la Figure 6.16.

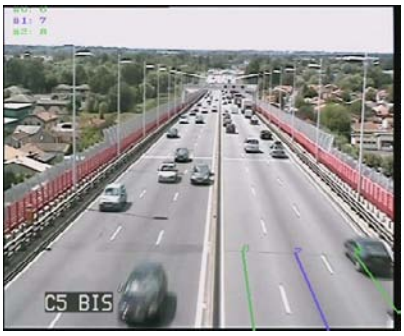


FIGURE 6.16: Illustration du placement des *scanlines* pour le comptage de véhicules sur la séquence C5. Il s'agit de segments de petites tailles placés dans les zones d'entrée de la scène, au centre des voies de circulation.

Comptage des objets	C5	Lyon	CE 11	Highway
Vérité-terrain	51	83	100	32
Algorithme	42	63	89	28
Précision	0.97	0.95	0.98	0.87
Rappel	0.84	0.78	0.85	1.00
F-score	0.90	0.85	0.91	0.93

TABLE 6.2: Résultats de l'évaluation des performances de l'algorithme de comptage d'objets sur quatre séquences de test : C5, Lyon, CE 11 et Highway. Les résultats sont présentés en termes de Précision, Rappel et F-score sur l'ensemble des *scanlines*.

6.3.3 Application à la détection d'arrêt

L'utilisation des cartes spatio-temporelles a également été appliquée à la détection d'arrêt de véhicules. En effet, l'arrêt d'un véhicule sur la voie de circulation (et par conséquent le long de la *scanline*) se traduit par une orientation horizontale du *bâtonnet* caractérisant l'objet en question. En se basant sur cette particularité, le processus de détection d'arrêt consiste à extraire les bordures des *bâtonnets* de la carte spatio-temporelle afin d'en estimer leurs orientations.

La séquence utilisée est caractérisée par l'arrêt d'un véhicule sur la voie de droite pendant une durée très courte. Ce véhicule est suivi par un poids lourd qui est contraint de changer de voie pour éviter la collision avec le véhicule arrêté. Sur la Figure 6.17 est illustré le scénario précédemment décrit en utilisant la *scanline* du centre de la voie de droite. Quelques secondes avant l'arrêt du véhicule, un poids lourd circule normalement. Notons

qu'aucune transformation perspective n'a été appliquée ce qui provoque une distorsion du *bâtonnet* relatif au poids lourd. Puis le véhicule entre en scène et s'arrête rapidement sur la voie. Le camion qui le précède effectue alors un dépassement pour éviter le véhicule léger. Celui-ci redémarre alors lentement avant de s'éloigner de la caméra.

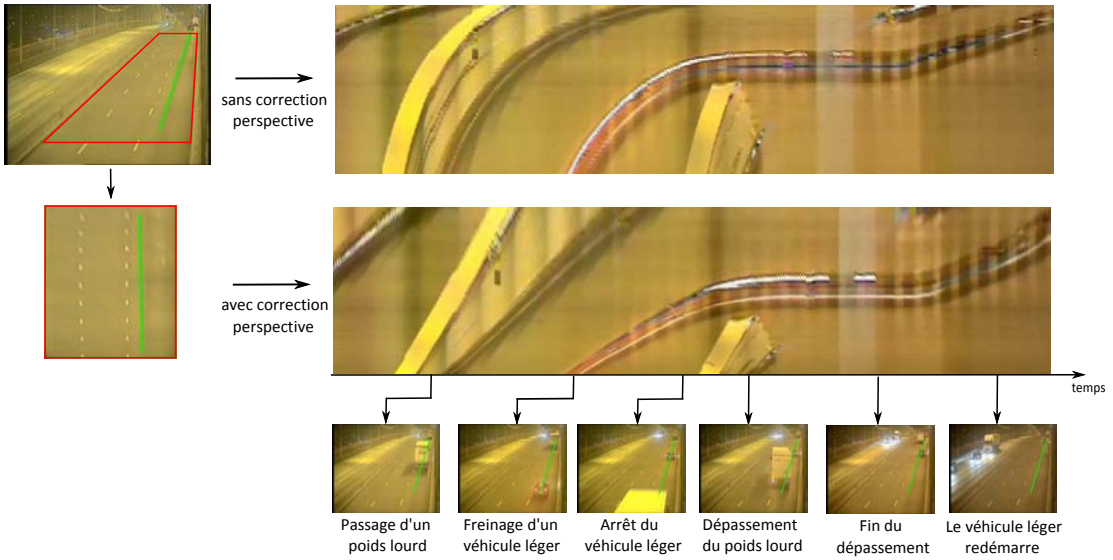


FIGURE 6.17: Illustration du scénario d'arrêt de séquence de nuit à l'aide du profil d'intensité lumineuse le long de la voie de droite. Pour une meilleure lisibilité, la carte spatio-temporelle a été affichée sur une durée de 24 secondes.

Ainsi, la première étape de l'analyse consiste à extraire les contours des cartes spatio-temporelles. Une fois les contours extraits, une transformation de Hough (basée sur la forme paramétrique d'une droite) permet d'extraire les droites contenues dans la carte des contours. Finalement un test sur chacune des droites détectées permet de valider l'éventuelle présence de droites horizontales et donc de valider celle d'un véhicule à l'arrêt. Dans le cadre du test effectué, chacune des voies possède trois *scanline* définies par le centre et les quarts de la largeur des voies. Les résultats de la détection d'arrêt sur la séquence de nuit (séquence C10) sont illustrés sur la Figure 6.18. L'arrêt du véhicule est correctement détecté sur cette séquence.

FIGURE 6.18: Résultats de la détection d'arrêt à l'aide des lignes de profil d'intensité lumineuse (*scanline*).

6.3.4 Application à la détection de bouchon

La densité du trafic routier est une information importante pour caractériser l'état du trafic. Il est notamment intéressant de détecter l'état saturé du trafic ou la formation d'un bouchon. Pour caractériser la densité du trafic, nous définissons trois états : un état *très fluide*, un état *fluide* et un état *embouteillage*. Dans l'état *très fluide* les véhicules circulent normalement, à vitesse supposée approximativement constante avec une distance inter-véhicule importante. L'état *fluide* est un état intermédiaire dans lequel il existe un nombre important de véhicules qui roulent plus lentement. Généralement, la distance de sécurité n'est pas respectée et la distance inter-véhicule est faible. Enfin, l'état *embouteillage* correspond à un état saturé du trafic. Les véhicules sont régulièrement à l'arrêt et roulent à très faible vitesse. Ces définitions permettent de dégager deux caractéristiques importantes pour estimer l'état du trafic routier : la vitesse de déplacement des véhicules et le taux d'occupation sur les voies.

La détection de bouchon proposée s'appuie sur l'analyse des cartes spatio-temporelles. Il s'agit d'une représentation 1d+t sur laquelle est reporté au cours du temps le profil de l'intensité lumineuse le long d'une droite, appelée *scanline*. L'image résultante combine les caractéristiques spatiales de l'intensité le long de la ligne et les caractéristiques temporelles au cours du temps. Chaque *scanline* génère un carte spatio-temporelle relative à la voie de circulation sur laquelle elle se situe et contient les informations nécessaires pour estimer de nombreuses informations, telles que la taille des véhicules, leurs vitesses, ... (voir Section 6.3.1).

Dans le cadre de la détection de bouchons, nous nous intéressons particulièrement à la pente des droites contenues sur la carte spatio-temporelle. Lorsque le trafic est fluide, la

pente des droites est relativement élevée, tandis qu'en présence d'un trafic encombré les objets sont à l'arrêt et la pente des droites de la carte est proche de zéro. Cette caractéristique est exploitée pour détecter la formation d'un bouchon. La Figure 6.19 montre un exemple de trafic encombré ainsi que la carte spatio-temporelle correspondante.

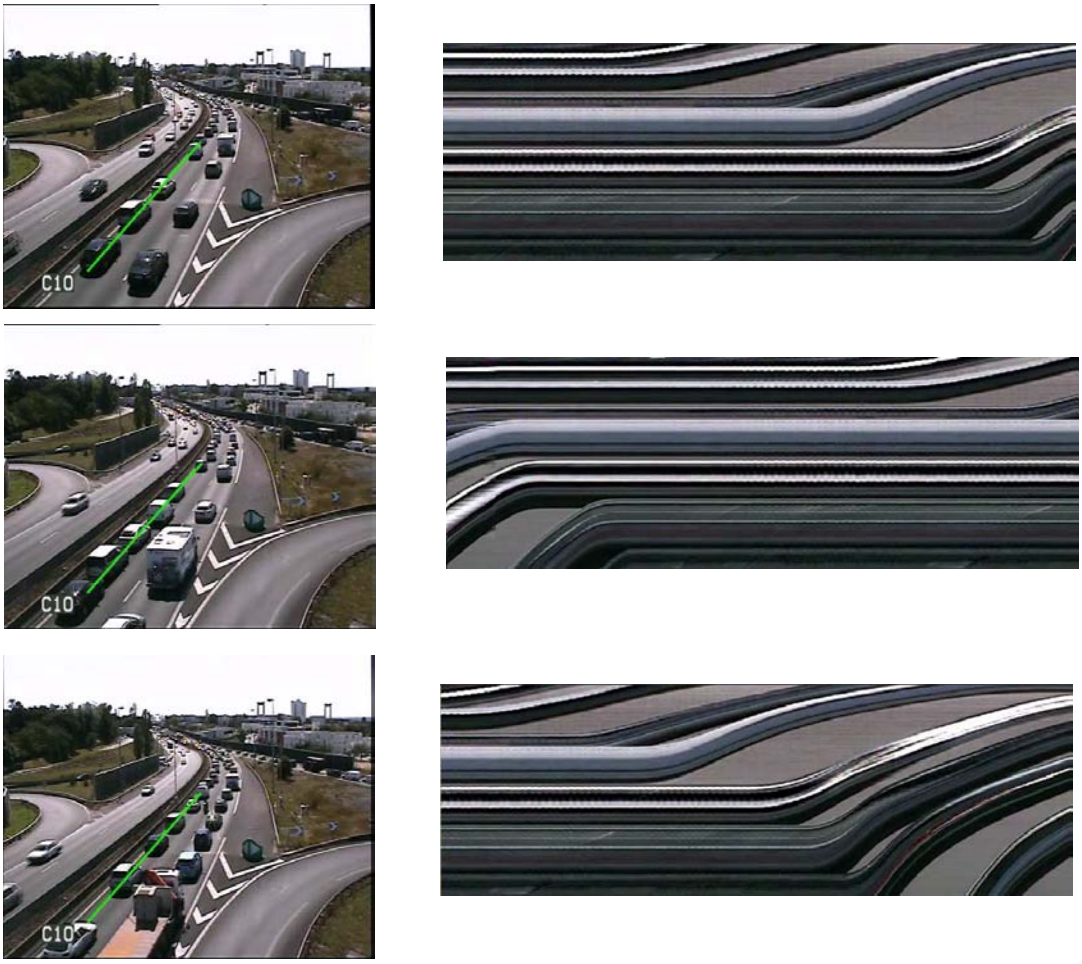


FIGURE 6.19: Illustration de la détection de bouchons à l'aide d'une carte spatio-temporelle. L'arrêt des véhicules est détecté par la présence de lignes horizontales sur la carte spatio-temporelle.

6.4 Conclusion

Nous avons vu dans ce chapitre deux approches pour l'analyse de comportements et la détection d'évènements. La première approche se base sur les résultats obtenus par les traitements de bas-niveau et de niveau intermédiaire (détection, extraction et suivi d'objets). La persistence temporelle des régions en mouvement et l'analyse des trajectoires a permis d'extraire quelques statistiques du trafic et d'effectuer une détection des évènements tels que l'arrêt d'un véhicule. En comparant les vecteurs mouvement issus du processus d'estimation du flot optique avec le modèle du sens de direction du trafic, les véhicules en contre-sens sont également détectés. Quant à la seconde approche, elle est basée sur l'étude d'une coupe longitudinale dans le volume 2d+t de la vidéo. Cette approche permet de s'affranchir de l'utilisation des trajectoires sensibles aux erreurs de segmentation et permet d'extraire des informations sur les objets (nombre, taille, vitesse, ...) mais également de détecter des évènements tels que l'arrêt d'un véhicule ou la formation d'un bouchon.

Conclusion et perspectives

Les travaux présentés dans ce manuscrit décrivent un système de vidéo-surveillance destiné à une analyse automatique et autonome de la scène. L'application à travers l'analyse du trafic routier nous a permis d'aborder plusieurs domaines : la modélisation de la scène, la détection de mouvement, l'extraction et le suivi d'objets, et l'analyse de comportement. Cette conclusion tente de dégager nos contributions et de proposer quelques perspectives envisagées.

Initialisation du système

Dans l'objectif d'aider à la segmentation et au suivi des objets, une étape d'initialisation est proposée durant laquelle une séquence d'apprentissage est analysée. Cette initialisation a été conçue de façon à être la plus générique et la moins supervisée possible afin d'être déployable sur un maximum de sites. Ainsi, un processus d'apprentissage non supervisé permet de construire un modèle complet de la scène comprenant des informations sur sa structure, les objets y évoluant et leurs comportements. Le modèle est estimé à travers l'analyse des caractéristiques de bas-niveau de la séquence d'apprentissage (couleur, contour, vecteurs mouvements).

Une modélisation de l'orientation du mouvement à travers un modèle statistique est adoptée. Ce modèle consiste en un mélange de lois de type von-Mises (adaptées aux données circulaires) dont les paramètres sont estimés récursivement à l'aide de la séquence d'apprentissage. Sa construction ne nécessite aucune intervention de l'utilisateur et permet d'obtenir une estimation du sens de direction du trafic. Parallèlement, l'image d'arrière-plan est construite et permet, à partir de ses contours, d'estimer les bordures des voies. Un algorithme de soustraction d'arrière-plan et de mise en association permet d'obtenir la zone d'intérêt et les trajectoires des objets évoluant dans la scène. En ne conservant que les trajectoires considérées *fiables* (pas d'ambiguité d'association), les trajectoires moyennes sont également estimées. Enfin, une estimation du point de fuite dans l'image à partir des trajectoires ou des bordures des voies permet de fournir une estimation approximative de la profondeur dans l'image. L'ensemble de ces informations est fusionné pour segmenter la zone d'intérêt en cellules qui couvrent théoriquement la même surface dans le monde réel. Chaque cellule joue alors le rôle d'*agent* élémentaire et contient l'ensemble des caractéristiques relatives aux comportement des objets.

Cette approche ne nécessite aucune intervention de l'utilisateur, mais nécessite certaines contraintes. Tout d'abord, la séquence de test doit représenter un trafic fluide et non encombré. Ceci permet de s'assurer d'avoir la couleur d'arrière-plan suffisamment représentée. De plus, la caméra doit être positionnée en face des voies de circulation qui elles-même doivent être rectilignes (ou de faibles courbures). En effet, les bordures des voies ou les trajectoires sont représentées sous la forme de droites. Une solution envisageable serait une généralisation de cette représentation sous la forme d'un polynôme. Enfin, l'estimation de la profondeur dans la scène est approximative et mériterait d'être améliorée par des techniques de calibration de caméras plus approfondies.

Détection de mouvement

L'approche proposée au Chapitre 4 combine de façon complémentaire les informations issues du modèle couleur, de la différence de gradient et des vecteurs mouvement après estimation du flot optique. Les modèles statistiques de couleur et de l'orientation des vecteurs

mouvements ont été décrits de manière approfondie, de l'étude théorique d'estimation de densité d'un mélange de lois à la description algorithmique et son implémentation.

Un modèle couleur d'ombre et de reflets centré sur la couleur d'arrière-plan permet de prendre en compte les changements de luminosité, en considérant en tant qu'ombre ou reflet tout pixel ayant des propriétés chromatiques proches de l'arrière-plan mais avec une intensité lumineuse plus faible ou plus forte (dans une certaine mesure). Pour aider à la validation des objets en mouvement, les informations de gradient et de mouvement sont utilisées. Malgré les résultats encourageants obtenus, l'algorithme proposé reste sensible au problème de camouflage particulièrement présent dans la vidéo lors du passage de poids lourds. En effet, la nature uniforme en couleur des remorques, par exemple, ne permet pas d'obtenir d'informations de gradients ou de mouvements, et si la couleur de la remorque est proche de celle de l'arrière-plan, le problème de camouflage n'est pas résolu. Une solution envisageable serait d'utiliser un modèle d'objet ou encore le résultat du suivi d'objet afin de conserver la forme de l'objet au cours du suivi.

De nombreuses perspectives sont envisagées ; tout d'abord l'information de gradient peut être modélisée également à l'aide d'un mélange de lois. L'idée consisterait en une modélisation statistique d'un vecteur combinant la couleur et le gradient à l'aide d'un mélange de lois gaussiennes de dimension 5 (trois composantes couleurs et deux composantes gradients). En parallèle, l'utilisation d'un modèle d'orientation du gradient à l'aide de lois von-Mises peut être intéressante pour caractériser la structure de la scène. L'avantage de l'utilisation de l'orientation du gradient est qu'elle est peu sensible aux changements de luminosité. Une étude des différents espaces couleurs est également à mener, ainsi que l'utilisation conjointe d'invariants colorimétriques dans l'objectif d'aider à la détection des ombres. Le principe consisterait à décomposer la couleur en 2 composantes : les composantes d'intensité et les composantes chromatiques, afin de n'avoir à se soucier que des composantes couleurs et en s'affanchissant des variations de lumières contenues dans la composante d'intensité. Enfin, une incorporation du modèle de scène et plus particulièrement de la profondeur dans l'image est envisagée afin de mettre à jour différemment les pixels proches et ceux éloignés de la caméra.

Extraction et suivi des objets

Le Chapitre 5 présente notre approche pour l'extraction et le suivi des objets. Une étude du modèle statistique bayésien sur lequel repose le filtrage prédictif a été présentée. Basé sur un modèle de mouvement linéaire au premier ordre, un fitrage de Kalman (cas particulier d'un modèle *tout gaussien*) est appliqué et permet d'estimer la position des objets suivis en présence de mesures bruitées. Pour étendre le problème de suivi mono-cible au problème de suivi multi-cibles, l'association des objets suivis avec ceux nouvellement détectés est effectuée à l'aide d'un graphe pondéré. Des hypothèses d'association sont émises et une étape de validation de la mise en correspondance dans le cas d'ambiguités est proposée en utilisant une information temporelle plus importante. Lorsque les objets sont proches de la caméra, les bordures des voies sont utilisées pour séparer les éventuels groupes de véhicules détectés. Dans le cas d'ambiguités relevées dans les zones de circulation, les caractéristiques d'apparence sont utilisées. ainsi, l'approche utilisée est une combinaison d'un algorithme de suivi basé sur un filtrage prédictif avec un algorithme de mise en correspondance permettant la génération d'hypothèses d'association.

Les perspectives envisagées sont les suivantes. si l'étude théorique a été décrite dans ce chapitre, aucun étude concernant le choix des caractéristiques n'a été mis en place. Il serait nécessaire d'étudier l'influence du choix des caratéristiques afin de ne conserver que celles les plus discriminantes pour l'application de suivi. L'incorporation du modèle de scène et de la profondeur dans l'image peut être intégré au filtrage prédictif. La prédiction de la position d'un objet serait alors dépendante de la distance à la caméra et reflèterait le ralentissement apparent des véhicules lorsqu'ils s'éloignent de la caméra. L'intégration d'informations sémantiques dans le vecteur d'état est également envisagé, avec par exemple l'utilisation de la classe des objets (véhicule léger, poids lourd, ...) pour aider au suivi. Il est également envisagé de mettre à jour dynamiquement les matrices de covariance Q (traduisant l'erreur de prédiction) et R (traduisant l'erreur de mesure). En utilisant par exemple la mesure de similarité objet-candidat, ces matrices de covariance (supposées diagonales) seraient alors ré-évaluées pour traduire la confiance plus ou moins grande que l'on a de la mesure et, par conséquent accorder plus ou moins d'importance à la prédiction. Enfin l'extension du filtrage de Kalman vers un filtrage particulaire est également dans nos perspectives. L'objectif étant d'exploiter le caractère multi-modale des distributions modélisées à l'aide des particules, afin de prendre en compte plusieurs configurations possibles des objets (les plus probables) et d'aider à la gestion des divisions et fusions dans le processus de suivi.

Analyse de comportement

L'analyse de comportement présentée dans le Chapitre 6 permet de détecter les véhicules en contre-sens, les véhicules à l'arrêt, les changements de voie et d'estimer les statistiques de trafic de la scène. Les véhicules en contre-sens sont détectés à l'aide du modèle statistique du sens de direction du trafic. Une comparaison de l'orientation des objets avec le modèle est proposée afin de détecter les éventuels véhicules circulant en contre-sens. Le changement de voie est détecté en étudiant l'historique des positions des objets suivis (trajectoires). Une machine à états finis permet d'attribuer à chaque objet un status de circulation en fonction de sa position sur les voies de circulation. Quant à la détection d'arrêt, elle repose sur l'extraction des objets et le masque des objets en mouvement obtenu. Lorsqu'un pixel appartient suffisamment longtemps à un objet, un modèle d'apparence est créé afin de vérifier qu'il ne change pas de couleur. Si un pixel est considéré comme appartenant à un objet et s'il ne change pas de couleur pendant suffisamment longtemps, alors il est considéré comme appartennt à un pixel à l'arrêt. Finalement, les statistiques du trafic sont évaluées à travers la construction de cartes spatio-temporelles initialisées au centre des voies. Ces cartes permettent d'extraire de nombreuses informations telles que le nombre de véhicules, leur vitesse et leur taille. Une détection d'arrêts ou de contre-sens ainsi qu'une détection de bouchons est également possible en analysant la pente des droites qui composent les *bâtonnets* représentatifs des véhicules.

Les perspectives envisagées sont les suivantes : ajouter une corrélation entre les cartes spatio-temporelles des différentes voies de circulation sous surveillance. En effet, la projection perspective ne corrige pas entièrement les effets de perspectives qui entraînent par exemple le comptage double d'un véhicule lorsque celui-ci déborde sur la voie adjacente. Avec le caractère répétitif des cartes spatio-temporelles, une analyse fréquentielle pourrait dégager et caractériser les répétitions de passage des véhicules.

Publications de l'auteur

P. Simonetto, P.Y. Koenig, F. Zaidi, et al. - Solving the traffic and flitter challenges with tulip, *Visual Analytics Science and Technology (VAST)*, p. 247-248, 2009.

C. Kaes, M. Brulin, H. Nicolas, C. Maillet - Compressed domain aided analysis of traffic surveillance videos, *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*, p.1-8, 2009.

M. Brulin, H. Nicolas, C. Maillet - Utilisation de la géométrie de la scène pour l'analyse du trafic routier, *Compression et Représentation des Signaux Audiovisuels (CORESA)*, p. 219-224, 2010.

M. Brulin, H. Nicolas, C. Maillet - Video surveillance analysis using scene geometry, *Image and Video Technology, 2010 Fourth Pacific-Rim Symposium on (PSIVT)*, p. 450-455, 2010.

M. Brulin, H. Nicolas, C. Maillet - Analyse d'un trafic routier dans un contexte de vidéo Surveillance - *Sciences of Electronics, Technologies of Information and Telecommunications Conference (SETIT)*, 2012.

Bibliographie

[Adam 2008] A. Adam, E. Rivlin, I. Shimshoni et D. Reinitz. *Robust real-time unusual event detection using multiple fixed-location monitors.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 3, pages 555–560, 2008.

[Ahmad 2011] KA Ahmad, Z. Saad, N. Abdullah, Z. Hussain et M.H.M. Noor. *Moving Vehicle Segmentation in a Dynamic Background using Self-adaptive Kalman Background Method.* IEEE International Colloquium on Signal Processing and its Applications, 2011.

[Albiol 2011] A. Albiol, L. Sanchis et J.M. Mossi. *Detection of Parked Vehicles Using Spatiotemporal Maps.* Intelligent Transportation Systems, IEEE Transactions on, no. 99, pages 1–15, 2011.

[Alin 2011] A. Alin, M.V. Butz et J. Fritsch. *Tracking moving vehicles using an advanced grid-based Bayesian filter approach.* In Intelligent Vehicles Symposium (IV), 2011 IEEE, pages 466–472. IEEE, 2011.

[Allen 2004] J.G. Allen, R.Y.D. Xu et J.S. Jin. *Object tracking using camshift algorithm and multiple quantized feature spaces.* In Proceedings of the Pan-Sydney area workshop on Visual information processing, pages 3–7. Australian Computer Society, Inc., 2004.

[Andrade 2006a] E.L. Andrade, S. Blunsden et R.B. Fisher. *Hidden markov models for optical flow analysis in crowds.* In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, volume 1, pages 460–463. IEEE, 2006.

[Andrade 2006b] E.L. Andrade, S. Blunsden et R.B. Fisher. *Modelling crowd scenes for event detection.* In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, volume 1, pages 175–178. IEEE, 2006.

[Aziz 2011] K.E. Aziz, D. Merad, B. Fertil et N. Thome. *Pedestrian head detection and tracking using skeleton graph for people counting in crowded environments.* Evaluation, pages 516–519, 2011.

[Banerjee 2006] A. Banerjee, I.S. Dhillon, J. Ghosh et S. Sra. *Clustering on the unit hypersphere using von Mises-Fisher distributions.* Journal of Machine Learning Research, vol. 6, no. 2, page 1345, 2006.

[Bay 2006] H. Bay, T. Tuytelaars et L. Van Gool. *Surf : Speeded up robust features.* Computer Vision–ECCV 2006, pages 404–417, 2006.

[Bay 2008] H. Bay, A. Ess, T. Tuytelaars et L. Van Gool. *Speeded-up robust features (SURF).* Computer Vision and Image Understanding, vol. 110, no. 3, pages 346–359, 2008.

[Benabbas 2011] Y. Benabbas, N. Ihaddadene et C. Djeraba. *Motion pattern extraction and event detection for automatic visual surveillance.* Journal on Image and Video Processing, vol. 2011, page 7, 2011.

[Bilmes 1998] J.A. Bilmes. *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models.* International Computer Science Institute, vol. 4, page 126, 1998.

[Birchfield 1998] S. Birchfield. *Elliptical head tracking using intensity gradients and color histograms.* In Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pages 232–237. IEEE, 1998.

[Bissacco 2004] A. Bissacco, P. Saisan et S. Soatto. *Gait recognition using dynamic affine invariants.* In International Symposium on Mathematical Theory of Networks and Systems, 2004.

[Bodor 2003] R. Bodor, B. Jackson et N. Papanikolopoulos. *Vision-based human tracking and activity recognition.* In Proc. of the 11th Mediterranean Conf. on Control and Automation, volume 1. Citeseer, 2003.

[Bouguet 2001] J.Y. Bouguet. *Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm.* Rapport technique, Technical report). Intel Corporation, 2001.

[Boukouvalas 1999] C. Boukouvalas, J. Kittler, R. Marik et M. Petrou. *Color grading of randomly textured ceramic tiles using color histograms.* Industrial Electronics, IEEE Transactions on, vol. 46, no. 1, pages 219–226, 1999.

[Bouwmans 2008] T. Bouwmans, F. El Baf, B. Vachon*et al.* *Background modeling using mixture of gaussians for foreground detection-a survey.* 2008.

[Bouwmans 2010] T. Bouwmans et F.E. Baf. *Statistical background modeling for foreground detection : A survey.* Handbook of Pattern Recognition and Computer, 2010.

[Brox 2010] T. Brox, B. Rosenhahn, J. Gall et D. Cremers. *Combined region and motion-based 3d tracking of rigid and articulated objects.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 3, pages 402–415, 2010.

[Buch 2011] N. Buch, S.A. Velastin et J. Orwell. *A review of computer vision techniques for the analysis of urban traffic.* Intelligent Transportation Systems, IEEE Transactions on, vol. 12, no. 3, pages 920–939, 2011.

[Candamo 2010] J. Candamo, M. Shreve, D.B. Goldgof, D.B. Sapper et R. Kasturi. *Understanding transit scenes : a survey on human behavior-recognition algorithms.* Intelligent Transportation Systems, IEEE Transactions on, vol. 11, no. 1, pages 206–224, 2010.

[Carta 2008] J.A. Carta, C. Bueno et P. Ramírez. *Statistical modelling of directional wind speeds using mixtures of von Mises distributions : Case study.* Energy conversion and management, vol. 49, no. 5, pages 897–907, 2008.

[Cohen 1991] L.D. Cohen. *On active contour models and balloons.* CVGIP : Image understanding, vol. 53, no. 2, pages 211–218, 1991.

[Cohen 2000] S. Cohen. *Exploitation et télématique routière(éléments d'évaluation socio-économique).* Rapport INRETS(Arcueil), 2000.

[Collins 2003] R.T. Collins. *Mean-shift blob tracking through scale space.* In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, pages II–234. IEEE, 2003.

[Comaniciu 2002] D. Comaniciu et P. Meer. *Mean shift : a robust approach toward feature space analysis.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pages 603–619, 2002.

[Cox 1996] I.J. Cox et S.L. Hingorani. *An efficient implementation of Reid s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 18, no. 2, pages 138–150, 1996.

[Dempster 1977] A.P. Dempster, N.M. Laird et D.B. Rubin. *Maximum likelihood from incomplete data via the EM algorithm.* Journal of the Royal Statistical Society. Series B (Methodological), pages 1–38, 1977.

[Douglas 1973] D.H. Douglas et T.K. Peucker. *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature.* Cartographica : The International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, pages 112–122, 1973.

[Du 2009] Y. Du et F. Yuan. *Real-time vehicle tracking by Kalman filtering and Gabor decomposition.* In Information Science and Engineering (ICISE), 2009 1st International Conference on, pages 1386–1390. IEEE, 2009.

[Elgammal 2000] A. Elgammal, D. Harwood et L. Davis. *Non-parametric model for background subtraction.* Computer Vision-ECCV 2000, pages 751–767, 2000.

[Fang 2011] H. Fang, J.W. Kim et J.W. Jang. *A Fast Snake Algorithm for Tracking Multiple Objects.* 2011.

[Foroughi 2008] H. Foroughi, B.S. Aski et H. Pourreza. *Intelligent video surveillance for monitoring fall detection of elderly in home environments.* In Computer and Information Technology, 2008. ICCIT 2008. 11th International Conference on, pages 219–224. IEEE, 2008.

[François 1999] A.R.J. François et G.G. Medioni. *Adaptive color background modeling for real-time segmentation of video streams.* In Proceedings of the International Conference on Imaging Science, Systems, and Technology, pages 227–232, 1999.

[Frank 2005] A. Frank. *On Kuhn's Hungarian method-a tribute from Hungary.* Naval Research Logistics (NRL), vol. 52, no. 1, pages 2–5, 2005.

[Fukunaga 1975] K. Fukunaga et L. Hostetler. *The estimation of the gradient of a density function, with applications in pattern recognition.* Information Theory, IEEE Transactions on, vol. 21, no. 1, pages 32–40, 1975.

[Gao 2001] D. Gao et J. Zhou. *Adaptive background estimation for real-time traffic monitoring.* In Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, pages 330–333. IEEE, 2001.

[Gevers 2004] T. Gevers et H. Stokman. *Robust histogram construction from color invariants for object recognition.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 1, pages 113–118, 2004.

[Gouaillier 2009] V. Gouaillier et A.E. Fleurant. *Intelligent video surveillance : Promises and challenges.* Technological and Commercial Intelligence Report, 2009.

[Haralick 1973] R.M. Haralick, K. Shanmugam et I.H. Dinstein. *Textural features for image classification.* Systems, Man and Cybernetics, IEEE Transactions on, vol. 3, no. 6, pages 610–621, 1973.

[Haritaoglu 2000] I. Haritaoglu, D. Harwood et L.S. Davis. *W4 : Real-time surveillance of people and their activities.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pages 809–830, 2000.

[Harris 1988] C. Harris et M. Stephens. *A combined corner and edge detector.* In Alvey vision conference, volume 15, page 50. Manchester, UK, 1988.

[Heckbert 1989] P. Heckbert. *Projective Mappings for Image Warping.* In in Fundamentals of Texture Mapping and Image Warping (Paul Heckbert, Masters Thesis), UC Berkeley. Citeseer, 1989.

[Herda 2001] L. Herda, P. Fua, R. Plankers, R. Boulic et D. Thalmann. *Using skeleton-based tracking to increase the reliability of optical motion capture.* Human movement science, vol. 20, no. 3, pages 313–341, 2001.

[Hershberger 1992] J.E. Hershberger et J. Snoeyink. Speeding up the douglas-peucker line-simplification algorithm. University of British Columbia, Dept. of Computer Science, 1992.

[Hill 1981] G.W. Hill. *Evaluation and Inversion of the Ratios of Modified Bessel Functions, I 1 (x)/I 0 (x) and I 1.5 (x)/I 0.5 (x).* ACM Transactions on Mathematical Software (TOMS), vol. 7, no. 2, pages 199–208, 1981.

[Horprasert 1999] T. Horprasert, D. Harwood et L.S. Davis. *A statistical approach for real-time robust background subtraction and shadow detection.* In IEEE ICCV, volume 99, pages 256–261. Citeseer, 1999.

[Hough 1962] P.V.C. Hough. *Method and means for recognizing complex patterns*, 1962. US Patent 3,069,654.

[Hu 2004a] W. Hu, T. Tan, L. Wang et S. Maybank. *A survey on visual surveillance of object motion and behaviors.* Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on, vol. 34, no. 3, pages 334–352, 2004.

[Hu 2004b] W. Hu, D. Xie et T. Tan. *A hierarchical self-organizing approach for learning the patterns of motion trajectories.* Neural Networks, IEEE Transactions on, vol. 15, no. 1, pages 135–144, 2004.

[Huang 2007] Z.Q. Huang et Z. Jiang. *Visual tracking based on color kernel densities of spatial awareness.* In Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on, pages 607–613. IEEE, 2007.

[Huang 2009] H. Huang, Z. Cai, S. Shi, X. Ma et Y. Zhu. *Automatic Detection of Vehicle Activities Based on Particle Filter Tracking.* A A, vol. 2, no. 2, page 2, 2009.

[Illingworth 1988] J. Illingworth et J. Kittler. *A survey of the Hough transform.* Computer vision, graphics, and image processing, vol. 44, no. 1, pages 87–116, 1988.

[Isard 1998] M. Isard et A. Blake. *Condensation, conditional density propagation for visual tracking.* International journal of computer vision, vol. 29, no. 1, pages 5–28, 1998.

[Ivanov 1999] Y. Ivanov, C. Stauffer, A. Bobick et WEL Grimson. *Video surveillance of interactions.* In Visual Surveillance, 1999. Second IEEE Workshop on,(VS'99), pages 82–89. IEEE, 1999.

[Jammalamadaka 2001] S.R. Jammalamadaka et A. Sengupta. Topics in circular statistics, volume 5. World Scientific Pub Co Inc, 2001.

[Johnson 1996] N. Johnson et D. Hogg. *Learning the distribution of object trajectories for event recognition.* Image and Vision Computing, vol. 14, no. 8, pages 609–615, 1996.

[Kalman 1960] R.E. Kalman. *A new approach to linear filtering and prediction problems.* Journal of basic Engineering, vol. 82, no. Series D, pages 35–45, 1960.

[Kameda 1996] Y. Kameda et M. Minoh. *A human motion estimation method using 3-successive video frames.* In International Conference on Virtual Systems and Multimedia, pages 135–140, 1996.

[Karaman 2010] S. Karaman, J. Benois-Pineau, R. Mégret, V. Dovgalecs, J.F. Dartigues et Y. Gaëstel. *Human daily activities indexing in videos from wearable cameras for monitoring of patients with dementia diseases.* In Pattern Recognition (ICPR), 2010 20th International Conference on, pages 4113–4116. IEEE, 2010.

[Karaman 2011] S. Karaman, J. Benois-Pineau, V. Dovgalecs, R. Mégret, J. Pinquier, R. André-Obrecht, Y. Gaëstel et J.F. Dartigues. *Hierarchical Hidden Markov Model in Detecting Activities of Daily Living in Wearable Videos for Studies of Dementia.* Arxiv preprint arXiv :1111.1817, 2011.

[Kas 2009] C. Kas et H. Nicolas. *Rough compressed domain camera pose estimation through object motion.* In Image Processing (ICIP), 2009 16th IEEE International Conference on, pages 3481–3484. IEEE, 2009.

[Kastrinaki 2003] V. Kastrinaki, M. Zervakis et K. Kalaitzakis. *A survey of video processing techniques for traffic applications.* Image and Vision Computing, vol. 21, no. 4, pages 359–381, 2003.

[Kim 2004] K. Kim, T.H. Chalidabhongse, D. Harwood et L. Davis. *Background modeling and subtraction by codebook construction.* In Image Processing, 2004. ICIP'04. 2004 International Conference on, volume 5, pages 3061–3064. IEEE, 2004.

[Kra]

[Krausz 2010] B. Krausz et R. Herpers. *MetroSurv : detecting events in subway stations.* Multimedia Tools and Applications, vol. 50, no. 1, pages 123–147, 2010.

[Krishnan 1997] T. Krishnan et GJ McLachlan. *The EM algorithm and extensions*, 1997.

[Kuhn 1955] H.W. Kuhn. *The Hungarian method for the assignment problem.* Naval research logistics quarterly, vol. 2, no. 1-2, pages 83–97, 1955.

[Lavee 2009] G. Lavee, E. Rivlin et M. Rudzsky. *Understanding video events : a survey of methods for automatic interpretation of semantic occurrences in video.* Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on, vol. 39, no. 5, pages 489–504, 2009.

[Legland 2003] F. Legland. *Filtrage particulaire.* In Proceedings 19eme Colloque GRETSI sur le Traitement du Signal et des Images, volume 1, pages 1–8, 2003.

[Lei 2010] F. Lei et X. Zhao. *Adaptive background estimation of underwater using Kalman-Filtering.* In Image and Signal Processing (CISP), 2010 3rd International Congress on, volume 1, pages 64–67. IEEE, 2010.

[Li 2003] Y. Li, L.Q. Xu, J. Morphett et R. Jacobs. *An integrated algorithm of incremental and robust pca.* In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, volume 1, pages I–245. IEEE, 2003.

[Li 2006] Y. Li, F. Chen, W. Xu et Y. Du. *Gaussian-Based Codebook Model for Video Background Subtraction.* Lecture notes in computer science, 2006.

[Lou 2002] J. Lou, Q. Liu, T. Tan et W. Hu. *Semantic interpretation of object activities in a surveillance system.* In Pattern Recognition, 2002. Proceedings. 16th International Conference on, volume 3, pages 777–780. IEEE, 2002.

[Lowe 1999] D.G. Lowe. *Object recognition from local scale-invariant features.* In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 2, pages 1150–1157. Ieee, 1999.

[Lowe 2004] D.G. Lowe. *Distinctive image features from scale-invariant keypoints.* International journal of computer vision, vol. 60, no. 2, pages 91–110, 2004.

[Lu 2001] W. Lu et Y.P. Tan. *A color histogram based people tracking system.* In Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, volume 2, pages 137–140. IEEE, 2001.

[Luvison 2012] B. Luvison, T. Chateau, J.T. Lapreste, P. Sayd et Q.C. Pham. *Automatic Detection of Unexpected Events in Dense Areas for Videosurveillance Applications.* 2012.

[Makhoul 1975] J. Makhoul. *Linear prediction : A tutorial review.* Proceedings of the IEEE, vol. 63, no. 4, pages 561–580, 1975.

[Makris 2005] D. Makris et T. Ellis. *Learning semantic scene models from observing activity in visual surveillance.* Systems, Man, and Cybernetics, Part B : Cybernetics, IEEE Transactions on, vol. 35, no. 3, pages 397–408, 2005.

[Malinovskiy 2009] Y. Malinovskiy, Y. Wang et Y.J. Wu. *Video-based vehicle detection and tracking using spatio-temporal maps.* Proceedings of 88th Annual Transportation Research Board Meeting, 2009. DVD-ROM, Washington, D.C.

[Mardia 2000] K.V. Mardia et P.E. Jupp. Directional statistics. John Wiley & Sons Inc, 2000.

[Matessi 1999] A. Matessi et L. Lombardi. *Vanishing point detection in the hough transform space.* Euro-Par99 Parallel Processing, pages 987–994, 1999.

[McFarlane 1995] N.J.B. McFarlane et C.P. Schofield. *Segmentation and tracking of piglets in images.* Machine Vision and Applications, vol. 8, no. 3, pages 187–193, 1995.

[McIvor 2000] A.M. McIvor. *Background subtraction techniques.* Proc. of Image and Vision Computing, vol. 1, no. 3, pages 155–163, 2000.

[McKenna 1999] S.J. McKenna, Y. Raja et S. Gong. *Tracking colour objects using adaptive mixture models.* Image and vision computing, vol. 17, no. 3-4, pages 225–231, 1999.

[McKenna 2000] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld et H. Wechsler. *Tracking groups of people.* Computer Vision and Image Understanding, vol. 80, no. 1, pages 42–56, 2000.

[McLachlan 2008] G.J. McLachlan et T. Krishnan. The em algorithm and extensions, volume 382. LibreDigital, 2008.

[Melli 2005] R. Melli, A. Prati, R. Cucchiara, L. de Cock et NV Traficon. *Predictive and probabilistic tracking to detect stopped vehicles.* In Application of Computer Vision, 2005. WACV/MOTIONSŠ05 Volume 1. Seventh IEEE Workshops on, volume 1, pages 388–393, 2005.

[Migliore 2006] D.A. Migliore, M. Matteucci et M. Naccari. *A revaluation of frame difference in fast and robust motion detection.* In Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks, pages 215–218. ACM, 2006.

[Mikolajczyk 2005] K. Mikolajczyk et C. Schmid. *A performance evaluation of local descriptors.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, no. 10, pages 1615–1630, 2005.

[Moeslund 2006] T.B. Moeslund, A. Hilton et V. Kruger. *A survey of advances in vision-based human motion capture and analysis.* Computer vision and image understanding, vol. 104, no. 2-3, pages 90–126, 2006.

[Monteiro 2007] G. Monteiro, M. Ribeiro, J. Marcos et J. Batista. *Wrongway drivers detection based on optical flow.* In Image Processing, 2007. ICIP 2007. IEEE International Conference on, volume 5, pages V–141. IEEE, 2007.

[Moravec 1980] H.P. Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover.* tech report CMURITR8003 Robotics Institute Carnegie Mellon University doctoral dissertation Stanford University, 1980.

[Nascimento 2005] J.C. Nascimento, M.A.T. Figueiredo et JS Marques. *Segmentation and classification of human activities.* In Proceedings of International Workshop on Human Activity Recognition and Modelling, 2005.

[Nghiem 2007] A.T. Nghiem, F. Bremond, M. Thonnat et V. Valentin. *ETISEO, performance evaluation for video surveillance systems.* In Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on, pages 476–481. IEEE, 2007.

[Oliver 2000] N.M. Oliver, B. Rosario et A.P. Pentland. *A Bayesian computer vision system for modeling human interactions.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pages 831–843, 2000.

[Paschos 2001] G. Paschos. *Perceptually uniform color spaces for color texture analysis : an empirical evaluation.* Image Processing, IEEE Transactions on, vol. 10, no. 6, pages 932–937, 2001.

[Perez 2002] P. Perez, C. Hue, J. Vermaak et M. Gangnet. *Color-based probabilistic tracking.* Computer Vision-ECCV 2002, pages 661–675, 2002.

[Piccardi 2004] M. Piccardi. *Background subtraction techniques : a review.* In Systems, Man and Cybernetics, 2004 IEEE International Conference on, volume 4, pages 3099–3104. Ieee, 2004.

[Porikli 2007] F. Porikli. *Detection of temporarily static regions by processing video at different frame rates.* In Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on, pages 236–241. IEEE, 2007.

[Power 2002] P.W. Power et J.A. Schoonees. *Understanding background mixture models for foreground segmentation.* In Proceedings Image and Vision Computing New Zealand, volume 2002, 2002.

[Pressigout 2005] M. Pressigout et E. Marchand. *Suivi temps-réel d'objet plan : approche hybride contour/texture.* 2005.

[Pucher 2010] M. Pucher, D. Schabus, P. Schallauer, Y. Lypetskyy, F. Graf, H. Rainer, M. Stadtschnitzer, S. Sternig, J. Birchbauer, W. Schneider*et al. Multimodal highway monitoring for robust incident detection.* In 13th International IEEE Conference on Intelligent Transportation Systems, pages 837–842, 2010.

[Pulford 2005] GW Pulford. *Taxonomy of multiple target tracking methods.* In Radar, Sonar and Navigation, IEE Proceedings-, volume 152, pages 291–304. IET, 2005.

[Reid 1979] D. Reid. *An algorithm for tracking multiple targets.* Automatic Control, IEEE Transactions on, vol. 24, no. 6, pages 843–854, 1979.

[Ribeiro 2004] M.I. Ribeiro. *Kalman and extended kalman filters : Concept, derivation and properties.* Institute for Systems and Robotics, page 43, 2004.

[Ribeiro 2005] P.C. Ribeiro et J. Santos-Victor. *Human activity recognition from video : Modeling, feature selection and classification architecture.* In Proceedings of International Workshop on Human Activity Recognition and Modelling. Citeseer, 2005.

[Ridder 1995] Christof Ridder, Olaf Munkelt et Harald Kirchner. *Adaptive Background Estimation and Foreground Detection using Kalman-Filtering.* pages 193–199, 1995.

[Rosenhahn 2005] B. Rosenhahn, U. Kersting, S. Andrew, T. Brox, R. Klette et H.P. Seidel. *A silhouette based human motion tracking system.* Rapport technique, CITR, The University of Auckland, New Zealand, 2005.

[Roy 2008] A. Roy, S. Kumar Parui, A. Paul et U. Roy. *A color based image segmentation and its application to text segmentation.* In Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on, pages 313–319. IEEE, 2008.

[Seibert 2006] M. Seibert, B.J. Rhodes, N.A. Bomberger, P.O. Beane, J.J. Sroka, W. Kogel, W. Kreamer, C. Stauffer, L. Kirschner, E. Chalome*t al. SeeCoast port surveillance.* In Proceedings of SPIE, volume 6204, page 62040B. Spie, 2006.

[Shah 1997] M. Shah. Motion-based recognition, volume 9. Springer, 1997.

[Shah 2011] M. Shah, J. Deng et B. Woodford. *Enhanced Codebook Model for Real-Time Background Subtraction.* In Neural Information Processing, pages 449–458. Springer, 2011.

[Shi 1994] J. Shi et C. Tomasi. *Good features to track.* In Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on, pages 593–600. IEEE, 1994.

[Sicre 2010] R. Sicre et H. Nicolas. *Human behaviour analysis and event recognition at a point of sale.* In Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on, pages 127–132. IEEE, 2010.

[Spagnolo 2006] P. Spagnolo, T.D. Orazio, M. Leo et A. Distante. *Moving object segmentation by background subtraction and temporal analysis.* Image and Vision Computing, vol. 24, no. 5, pages 411–423, 2006.

[Stauffer 1999] C. Stauffer et W.E.L. Grimson. *Adaptive background mixture models for real-time tracking.* In Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on., volume 2. IEEE, 1999.

[Stauffer 2000] C. Stauffer et W.E.L. Grimson. *Learning patterns of activity using real-time tracking.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pages 747–757, 2000.

[Stauffer 2003] C. Stauffer. *Estimating tracking sources and sinks.* In Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on, volume 4, pages 35–35. IEEE, 2003.

[Sun 2010] X. Sun, H. Yao, S. Zhang et B. Zhong. *On-Line Discriminative Appearance Modeling for Robust Object Tracking.* In Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on, pages 78–81. IEEE, 2010.

[Suzuki 1985] S. Suzuki*et al.* *Topological structural analysis of digitized binary images by border following.* Computer Vision, Graphics, and Image Processing, vol. 30, no. 1, pages 32–46, 1985.

[Takeo 1991] C.T. Takeo et T. Kanade. *Detection and tracking of point features*, 1991.

[Terzopoulos 1988] D. Terzopoulos, A. Witkin et M. Kass. *Constraints on deformable models : Recovering 3D shape and nonrigid motion.* Artificial Intelligence, vol. 36, no. 1, pages 91–123, 1988.

[Tian 2011] B. Tian, Q. Yao, Y. Gu, K. Wang et Y. Li. *Video processing techniques for traffic flow monitoring : A survey.* In Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, pages 1103–1108. IEEE, 2011.

[Tomasi 1991] C. Tomasi et T. Kanade. Detection and tracking of point features. School of Computer Science, Carnegie Mellon Univ., 1991.

[Torkan 2010] S. Torkan et A. Behrad. *A new contour based tracking algorithm using improved greedy snake.* In Electrical Engineering (ICEE), 2010 18th Iranian Conference on, pages 150–155. IEEE, 2010.

[Toyama 1999] K. Toyama, J. Krumm, B. Brumitt et B. Meyers. *Wallflower : Principles and practice of background maintenance.* In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 1, pages 255–261. Ieee, 1999.

[Trinh 2011] H. Trinh, Q. Fan, P. Jiyan, P. Gabbur, S. Miyazawa et S. Pankanti. *Detecting human activities in retail surveillance using hierarchical finite state machine.* In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 1337–1340. IEEE, 2011.

[Turaga 2008] P. Turaga, R. Chellappa, V.S. Subrahmanian et O. Udrea. *Machine recognition of human activities : A survey.* Circuits and Systems for Video Technology, IEEE Transactions on, vol. 18, no. 11, pages 1473–1488, 2008.

[Van de Sande 2008] K. Van de Sande, T. Gevers et C. Snoek. *Evaluation of color descriptors for object and scene recognition.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. Ieee, 2008.

[Verbeke 2007] N. Verbeke et N. Vincent. *A PCA-based technique to detect moving objects.* Image Analysis, pages 641–650, 2007.

[Vilnrotter 1986] F.M. Vilnrotter, R. Nevatia et K.E. Price. *Structural analysis of natural textures.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, no. 1, pages 76–89, 1986.

[Wang 2004] Y. Wang, E.K. Teoh et D. Shen. *Lane detection and tracking using B-Snake.* Image and Vision computing, vol. 22, no. 4, pages 269–280, 2004.

[Wang 2009] X. Wang, X. Ma et W.E.L. Grimson. *Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 3, pages 539–555, 2009.

[Wang 2010] X. Wang et X. Wang. *Effecient online appearance models for object tracking.* In Proceedings of the 10th WSEAS international conference on Multimedia systems & signal processing, pages 217–221. World Scientific and Engineering Academy and Society (WSEAS), 2010.

[Wang 2011a] L.F. Wang et C.H. Pan. *Effective multi-resolution background subtraction.* In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 909–912. IEEE, 2011.

[Wang 2011b] S.C. Wang, T.F. Su et S.H. Lai. *Detecting moving objects from dynamic background with shadow removal.* In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 925–928. IEEE, 2011.

[Wren 1997] C.R. Wren, A. Azarbayejani, T. Darrell et A.P. Pentland. *Pfinder : Real-time tracking of the human body.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 19, no. 7, pages 780–785, 1997.

[Xie 2010] X. Xie. *A review of recent advances in surface defect detection using texture analysis techniques.* ELCVIA : electronic letters on computer vision and image analysis, vol. 7, no. 3, pages 1–22, 2010.

[Xie 2011] S. Xie et J. Pan. *Hand Detection Using Robust Color Correction and Gaussian Mixture Model.* In Image and Graphics (ICIG), 2011 Sixth International Conference on, pages 553–557. IEEE, 2011.

[Yang 2011] M.H. Yang et J. Ho. *Toward Robust Online Visual Tracking.* Distributed Video Sensor Networks, pages 119–136, 2011.

[Yilmaz 2004] A. Yilmaz, X. Li et M. Shah. *Contour-based object tracking with occlusion handling in video acquired using mobile cameras.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 26, no. 11, pages 1531–1536, 2004.

[Yilmaz 2006] A. Yilmaz, O. Javed et M. Shah. *Object tracking : A survey.* Acm Computing Surveys (CSUR), vol. 38, no. 4, page 13, 2006.

[Zelniker 2008] E.E. Zelniker, S. Gong, T. Xiang*et al. Global abnormal behaviour detection using a network of CCTV cameras.* 2008.

[Zhang 2009] Z.H. Zhang, R.Q. Chen, H.Q. Lu, Y.K. Yan et H.Q. Cui. *Moving Foreground Detection Based on Modified Codebook.* In Image and Signal Processing, 2009. CISP'09. 2nd International Congress on, pages 1–5. IEEE, 2009.

[Zhong 2003] J. Zhong et S. Sclaroff. *Segmenting foreground objects from a dynamic textured background via a robust kalman filter*. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, pages 44–50. IEEE, 2003.

[Zhu 2000] Z. Zhu, G. Xu, B. Yang, D. Shi et X. Lin. *VISATRAM : A real-time vision system for automatic traffic monitoring*. Image and Vision Computing, vol. 18, no. 10, pages 781–794, 2000.

[Zivkovic 2004] Z. Zivkovic. *Improved adaptive Gaussian mixture model for background subtraction*. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 2, pages 28–31. Ieee, 2004.

[Zou 2007] X. Zou, D. Li et J. Liu. *Real-time vehicles tracking based on Kalman filter in an ITS*. In International Symposium on Photoelectronic Detection and Imaging : Technology and Applications 2007, pages 662306–662306. International Society for Optics and Photonics, 2007.