Estimation du contexte par vision embarquée et schémas de commande pour l'automobile

Moez Ammar

▶ **To cite this version:**

Moez Ammar. Estimation du contexte par vision embarquée et schémas de commande pour l'automobile. Autre [cond-mat.other]. Université Paris Sud - Paris XI, 2012. Français. NNT : 2012PA112425 . tel-00811797

HAL Id: tel-00811797

https://theses.hal.science/tel-00811797

Submitted on 22 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE PARIS-SUD

ÉCOLE DOCTORALE : STITS
Institut d'Electronique Fondamentale

***DISCIPLINE** PHYSIQUE*

THÈSE DE DOCTORAT

soutenue le 21/12/2012

par

Moez AMMAR

Estimation du contexte par vision embarquée et schémas de commande pour l'automobile

Directeur de thèse :	Hugues MOUNIER	Professeur, Université Paris Sud
Co-directeur de thèse :	Sylvie LE HEGARAT	Professeur, Université Paris Sud

Composition du jury :

Président du jury :	Didier AUBERT	Directeur de recherche, INRETS
Rapporteurs :	Roland CHAPUIS	Professeur, Université Blaise Pascal
	Arben CELA	Maitre de Conférences HDR, Université Paris-Est
Examinateurs :	Roger REYNAUD	Professeur, Université Paris Sud
	Ahmed BEN HAMIDA	Professeur, Université de Sfax

Remerciements

Tout d'abord, je remercie les membres du jury : monsieur Didier AUBERT Directeur de recherche à l'INRETS pour avoir accepté de présider ma soutenance de thèse, monsieur Roland CHAPUIS, Professeur à l'Université Blaise Pascal, et monsieur Arben CELA, Maitre de Conférences HDR à l'Université Paris-Est, qui ont bien voulu rapporter sur cette thèse. Vos remarques et suggestions auront contribué à améliorer la qualité de mon manuscrit. Monsieur Roger REYNAUD, Professeur à l'Université Paris Sud, et monsieur Ahmed BEN HAMIDA, Professeur à l'Université de Sfax, qui ont accepté d'ètre des examinateurs.

C'est avec beaucoup de reconnaissance que je souhaite remercier ma directrice de thèse madame Sylvie LE HEGARAT, Professeur à l'Université Paris Sud, pour m'avoir permis de faire mes premiers pas dans le monde de la recherche et pour m'avoir incité à poursuivre cette voie, pour la formation scientifique de qualité qu'elle m'a apportée et surtout pour la confiance qu'elle m'a accordée tout au long de ma thèse. Votre grande rigueur, votre dynamisme et votre passion pour la recherche scientifique m'ont beaucoup appris. Je pars avec un bagage scientifique très enrichi. Un grand merci aussi pour vos encouragements et votre soutien pendant les moments de doute ou de désespoir. vous avez su toujours comment m'encourager et me motiver. Je vous remercie aussi pour le temps que vous m'avez consacré dans la lecture minutieuse de ce manuscrit. Travailler avec vous et apprendre à vos cotés auront été un plaisir mais surtout un honneur.

Une profonde gratitude envers mon directeur de thèse monsieur Hugues MOUNIER, Professeur à l'Université Paris Sud, pour sa disponibilité et ses constants encouragements tout au long de ces années. Je vous remercie également pour vos lectures, vos conseils avisés et les échanges toujours fructueux que nous avons eu. C'était un honneur de travailler avec vous et sous votre encadrement.

Je remercie également monsieur Abdelaziz KALLEL pour l'interet qu'il a porté à mon travail et pour ces conseils et son aide.

Mes remerciement très sincères vont à l'ensemble de mon équipe : Marius, Samir, Samia, François, Flavien, Cyrille, je vous remercie tous pour la bonne ambiance et pour votre aide aimable. Je tiens aussi à remercier des amis que j'ai rencontré à l'IEF et le L2S, Ahmed, Islam, Ali, Yasmine, Yasser.

Mon parcours n'aurait jamais pu rencontrer le succès sans le soutien à toute épreuve que mes parents m'ont apporté. Il n'existe aucun mot pour traduire ma reconnaissance, mon amour et mon affection. Ces années de thèse ont été très dures à gérer, mais vous étiez toujours là pour m'encourager et me motiver sans jamais se plaindre. Un grand merci à toi maman d'avoir toujours pensé à moi avant même de penser à toi. Un grand merci à toi mon cher papa, sans toi je n'aurais jamais pu arriver jusque-là. Je remercie mes frères, Mourad, Hichem et Wissem et ma soeur Sameh de m'avoir poussé et soutenu jusqu'au bout.

Un spécial merci à ma chère Takoua, qui a partagé avec moi des périodes difficiles mais joyeuses durant cette de thèse.

Je tiens sincèrement à remercier toutes les personnes qui ont contribué de près ou de loin à l'accomplissement de cette thèse.

Je dédie affectueusement cette thèse à ma "Tunisie" qui s'est liberée grâce à ses jeunes révolutionnaires....

Résumé

Les systèmes dotés d'autonomie doivent continument évaluer leur environnement, via des capteurs embarqués, afin de prendre des décisions pertinentes au regard de leur mission, mais aussi de l'endo-système et de l'exosystème. Dans le cas de véhicules dits 'intelligents', l'attention quant au contexte environnant se porte principalement d'une part sur des objets parfaitement normalisés, comme la signalisation routière verticale ou horizontale, et d'autre part sur des objets difficilement modélisables de par leur nombre et leur variété (piétons, cyclistes, autres véhicules, animaux, ballons, obstacles quelconques sur la chaussée, etc.). La décision a contrario offre un cadre formel, adapté à ce problème de détection d'objets variables, car modélisant le bruit plutôt qu'énumérant les objets à détecter.

La contribution principale de cette thèse est d'adapter des mesures probabilistes de type NFA (Nombre de Fausses Alarmes) au problème de la détection d'objets soit ayant un mouvement propre, soit saillants par rapport au plan de la route. Un point fort des algorithmes développés est qu'ils s'affranchissent de tout seuil de détection. Une première mesure NFA permet d'identifier le sous-domaine de l'image (pixels non nécessairement connexes) dont les valeurs de niveau de gris sont les plus étonnantes, sous hypothèse de bruit gaussien (modèle naïf). Une seconde mesure NFA permet ensuite d'identifier le sous-ensemble des fenêtres de significativité maximale, sous hypothèse de loi binômiale (modèle naïf). Nous montrons que ces mesures NFA peuvent également servir de critères d'optimisation de paramètres, qu'il s'agisse du mouvement 6D de la caméra embarquée, ou d'un seuil de binarisation sur les niveaux de gris. Enfin, nous montrons que les algorithmes proposés sont génériques au sens où ils s'appliquent à différents types d'images en entrée, radiométriques ou de disparité.

A l'opposé de l'approche *a contrario*, les modèles markoviens permettent d'injecter des connaissances a priori sur les objets recherchés. Nous les exploitons dans le cas de la classification de marquages routiers.

A partir de l'estimation du contexte (signalisation, détection d'objets 'inconnus'), la partie commande comporte premièrement une spécification des trajectoires possibles et deuxièmement des lois en boucle fermée assurant le suivi de la trajectoire sélectionnée. Les diverses trajectoires possibles sont regroupées en un faisceau, soit un ensemble de fonctions du temps où divers paramètres permettent de régler les invariants géométriques locaux (pente, courbure). Ces paramètres seront globalement fonction du contexte extérieur au véhicule (présence de vulnérables, d'obstacles fixes, de limitations de vitesse, etc.) et permettent de déterminer l'élément du faisceau choisi. Le suivi de la trajectoire choisie s'effectue alors en utilisant des techniques de type platitude différentielle, qui s'avèrent particulièrement bien adaptées aux problèmes de suivi de trajectoire. Un système différentiellement plat est en effet entièrement paramétré par ses sorties plates et leurs dérivées. Une autre propriété caractéristique de ce type de systèmes est d'être linéarisable de manière exacte (et donc globale) par bouclage dynamique endogène et transformation de coordonnées. Le suivi stabilisant est alors trivialement obtenu sur le système linéarisé.

Abstract

To take relevant decisions, autonomous systems have to continuously estimate their environment via embedded sensors. In the case of 'intelligent' vehicles, the estimation of the context focuses both on objects perfectly known such as road signs (vertical or horizontal), and on objects unknown or difficult to describe due to their number and variety (pedestrians, cyclists, other vehicles, animals, any obstacles on the road, etc.). Now, the a contrario modelling provides a formal framework adapted to the problem of detection of variable objects, by modeling the noise rather than the objects to detect.

Our main contribution in this PhD work was to adapt the probabilistic NFA (Number of False Alarms) measurements to the problem of detection of objects simply defined either as having an own motion, or salient to the road plane. A highlight of the proposed algorithms is that they are free from any detection parameter, in particular threshold. A first NFA criterion allows the identification of the sub-domain of the image (not necessarily connected pixels) whose gray level values are the most amazing under Gaussian noise assumption (naive model). A second NFA criterion allows then identifying the subset of maximum significant windows under binomial hypothesis (naive model). We prove that these measurements (NFA) can also be used for the estimation of intrinsec parameters, for instance either the 6D movement of the onboard camera, or a binarisation threshold. Finally, we prove that the proposed algorithms are generic and can be applied to different kinds of input images, for instance either radiometric images or disparity maps.

Conversely to the a contrario approach, the Markov models allow to inject a priori knowledge about the objects sought. We use it in the case of the road marking classification.

From the context estimation (road signs, detected objects), the control part includes firstly a specification of the possible trajectories and secondly the laws to achieve the selected path. The possible trajectories are grouped into a bundle, and various parameters are used to set the local geometric invariants (slope, curvature). These parameters depend on the vehicle context (presence of vulnerables, fixed obstacles, speed limits, etc...), and allows determining the selected the trajectory from the bundle. Differentially flat system is indeed fully parameterized by its flat outputs and their derivatives. Another feature of this kind of systems is to be accurately linearized by endogenous dynamics feed-back and coordinate transformation. Tracking stabilizer is then trivially obtained from the linearized system.

Table des matières

1 Introduction **3**

1.1 Contexte applicatif (Systèmes autonomes versus ADAS) 4

1.1.1 Système d'aide à la conduite (ADAS) 5

1.1.2 Système autonome 5

1.2 Position du problème 6

1.3 Contribution 7

I Traitement d'images **9**

2 Détection d'objets 'inattendus' **11**

2.1 Introduction 12

2.2 État de l'art 12

2.2.1 Détection de changement en vision monoculaire 12

2.2.1.1 Détection de changement 13

2.2.1.2 Suivi d'objets 14

2.2.2 Détection d'objets par stéréovision 15

2.2.2.1 Projection stéréoscopique 16

2.2.2.2 Détection d'objets *a priori* 18

2.2.2.3 Collaboration entre 3D et mouvement 20

2.2.3 Conclusion 21

2.3 Approche proposée 21

2.3.1 Historique : des *gestalts* au principe de Helmholtz 21

2.3.2 Méthode développée 27

2.3.2.1 Niveau 1 : Détection des points sur l'image 28

2.3.2.2 Niveau 2 : Détection des groupements de points 33

2.3.3 Application 47

2.3.3.1 Trois différentes déclinaisons de l'approche proposée 47

2.3.3.2 Premiers exemples de résultats 50

2.3.3.3 Classification des objets selon les labels *mobile* ou *statique* 52

2.3.3.4 Résultats 54

2.4 Conclusion 59

3 Signalisation routière **63**

3.1 Introduction 64

3.2 Détection des panneaux routiers 64

3.3 Détection des marquages au sol 65

3.3.1 Approche proposée . . . 68
3.3.2 Prétraitement des images . . . 69
3.3.3 Classification niveau pixel . . . 69
3.3.4 Classification niveau composante connexe . . . 71
3.3.5 Classification niveau objet . . . 72
3.3.6 Résultats et discussion . . . 73
3.3.7 Conclusion . . . 78
3.4 Conclusion . . . 79

II Automatique 81

4 Commande du véhicule 83
4.1 État de l'art . . . 84
4.1.1 Planification de trajectoire . . . 84
4.1.2 Faisceaux de trajectoires . . . 85
4.1.3 Platitude et suivi de trajectoire . . . 86
4.1.3.1 Suivi de trajectoire . . . 86
4.1.3.2 Platitude . . . 86
4.1.3.3 Paramétrisation . . . 89
4.1.3.4 Méthodologie . . . 89
4.1.3.5 Platitude et la linéarisation du bouclage . . . 90
4.1.3.6 Algorithme d'extension dynamique . . . 90
4.1.3.7 Suivi de trajectoire en boucle fermée . . . 91
4.2 Modélisation cinématique augmentée du véhicule . . . 91
4.2.1 Modèle cinématique augmenté du véhicule . . . 91
4.2.2 Platitude du modèle . . . 92
4.2.3 Suivi de trajectoire avec stabilité . . . 93
4.2.4 Simulation . . . 94
4.3 Conclusion . . . 100

5 Conclusion générale et perspectives 101

Chapitre 1

Introduction

Sommaire

1.1 Contexte applicatif (Systèmes autonomes versus ADAS) 4

1.1.1 Système d'aide à la conduite (ADAS) 5

1.1.2 Système autonome 5

1.2 Position du problème 6

1.3 Contribution 7

1.1 Contexte applicatif (Systèmes autonomes versus ADAS)

Les dernières décennies ont été témoin d'une croissance exponentielle dans le domaine du transport routier. Cette croissance s'observe au niveau de deux axes complémentaires. Le premier axe est le développement des moyens de transport (automobiles, trains, avions...). Le deuxième est celui des réseaux de transport.

Cette croissance a induit une augmentation du nombre d'utilisateurs du réseau routier, qui elle-même a induit une augmentation des accidents routiers. Dans le même temps, la sécurité routière est passée d'un problème généralement négligé par les citoyens sous prétexte que "cela n'arrive qu'aux autres", à un problème d'actualité.

Historiquement, les catastrophes aériennes et/ou maritimes étaient beaucoup plus médiatiques que les accidents routiers, peut-être parce que, pour une catastrophe aérienne ou maritime le bilan des morts et blessés est généralement très lourd. Mais la probabilité d'un accident routier est plus importante que celle d'une catastrophe aérienne, et le bilan global des accidents routiers est finalement plus important. En Europe, on dénombre des centaines de morts par jour, des blessés graves, souvent handicapés à vie et des milliers de blessés légers.

Par exemple dans le cas de la France, pour l'année 2001, le bilan des accidents routiers est de 8160 morts (soit 23 personnes par jour), dont 822 sont des piétons [1]. Le bureau d'enquête et d'analyse pour la sécurité de l'aviation civile, BEA [2], recense pour la même année 83 morts (53 par des avions et 30 par des Ultra-Léger Motorisés ULM [3]) et 62 blessés (26 par des avions et 36 par des ULM) pour 274 (175 par des avions et 99 par des ULM) accidents.

Ces quelques chiffres illustrent l'ampleur de la situation pour les accidents de la route, et ont incité le gouvernement à sensibiliser les citoyens et à déployer de nouvelles stratégies pour faire face à ces accidents, notamment via les NTIC [4].

Ainsi les progrès réalisés ces dernières années se situent en terme de confort (toujours très *marketing*) mais également de sécurité. Les avancées technologiques que nous citons par la suite ont commencé à porter leurs fruits : en 2011 (dernière statistique officielle), sur le transport routier, il y a eu 3963 morts, dont 519 piétons. Ces résultats ont été obtenus notamment grâce aux travaux des laboratoires publics et des industries dans le domaine des NTIC, qui fournissent des infrastructures routières et des véhicules intelligents pour améliorer la sécurité routière. Dans le cadre d'un grand projet national, Programme de REcherche et D'Innovation dans les Transports terrestres (PREDIT [5]), les entreprises et laboratoires regroupent leurs travaux sous la direction des ministères français. A l'échelle de l'Europe, plusieurs projets (PReVENT, ISMAEL, GST, HUMIST, AIDE ...) ont émergé dans ce domaine.

Notre travail s'intéresse à l'amélioration de la sécurité d'un véhicule automobile. Pour cela, nous ne bornons pas notre intérêt au véhicule lui-même (structure matérielle, actionneurs, capteurs, moteur, ...) mais nous considérons les différents éléments avec lesquels il interagit notamment son environnent (route, signalisation, piéton, voitures, etc.), qui évolue temporellement, et le conducteur. Dans la suite, nous distinguons des solutions ayant pour objectif d'assister le conducteur dans sa tâche de conduite (systèmes coopératifs ou Systèmes d'aide à la conduite), et des solutions ayant pour objectif de suppléer le conducteur (Systèmes autonomes) avec toute l'instrumentation et les systèmes de décision à bord du véhicule).

1. http ://ec.europa.eu/transport/road_safety/specialist/statistics/care_reports_graphics/index_en.htm
2. Bureau d'Enquête et d'analyse pour la sécurité de l'Aviation civile
3. Ultra-Léger Motorisé
4. Nouvelles Technologies de l'Information et de la Communication
5. Programme de REcherche et d'Innovation dans les Transports terrestres

1.1.1 Système d'aide à la conduite (ADAS)

Les systèmes d'aide à la conduite, ou *Advanced Driver Assistance Systems* (ADAS), ont pour objectif de fournir des informations précises au conducteur [104, 13, 70], comme la présence de piétons [61, 54, 58], la présence d'autres véhicules et leur distance [121], la détection de la signalisation routière (horizontale : marquages au sol, et verticale : panneaux), la détection de la route [6] et localisation du véhicule par rapport à la route. Mais contrairement aux systèmes autonomes, ces systèmes ne remplacent pas le conducteur. Leur rôle principal est d'assister le conducteur par exemple en le suppléant en cas de vigilance défaillante. Ces systèmes peuvent aller jusqu'à présenter des solutions de trajectoire possible au conducteur, qui en tiendra compte dans son choix de commande.

Notons finalement que ces dernières années ont vu l'émergence de systèmes de conduite téléopérée (par exemple le projet SIMCOD). Ces systèmes se situent à la frontière des deux précédents. En effet, pour les très courtes échelles temporelles (inférieures au temps de transmission en direction de l'opérateur distant, c'est-à-dire inférieures à la seconde) le système doit être autonome, tandis que pour des échelles temporelles plus grandes, le système offre une aide à la conduite pour l'opérateur distant. Soulignons que dans un système téléopéré la perception de l'environnement par l'opérateur se fait exclusivement par les capteurs du système (caméras, etc.) et qu'il ne s'agit plus de suppléer à une vigilance défaillante du conducteur, mais de lui fournir les informations sur le contexte nécessaires à la conduite.

1.1.2 Système autonome

Un véhicule autonome est un véhicule capable d'exécuter sa tâche sans opérateur humain, même à distance. Actuellement, pour des raisons de sécurité (et juridiques), il n'est pas encore envisageable de laisser un véhicule en autonomie complète circuler dans les rues (en France du moins). Certainement, dans le futur, ces véhicules verront le jour à l'instar du véhicule *GoogleCar* dans les états du Névada et de Californie.

De tels systèmes permettent de s'affranchir de l'intervention humaine en rendant la commande du véhicule totalement autonome. Actuellement, des systèmes automatiques existent pour les faibles vitesses (*Stop and Go* [126, 93]). Ils arrêtent et démarrent automatiquement les véhicules dans les embouteillages, leur objectif principal étant de relayer le conducteur dans des situations stressantes mais sans danger de mort.

Un autre exemple de système autonome est fourni par les véhicules guidés automatiquement dans des zones dédiées telles que les voies de bus. De tels véhicules autonomes permettent de circuler d'un point de départ à un point d'arrivée selon (au moins) un chemin prédéfini reliant ces deux points. Techniquement, il s'agit de planifier une trajectoire de référence et de la suivre en tenant compte du contexte intrinsèque au véhicule (notamment sa cinématique) et du contexte externe au véhicule (environnement, notamment route et obstacles).

Ainsi, une première étape est la définition d'un modèle géométrique de la route pour les différents types de route existantes (par exemple autoroute, urbain, non urbain), modèle qui permet au véhicule de se positionner par rapport à la route. Une autre étape est la détection des objets vulnérables, et leur localisation, afin notamment d'ajuster la trajectoire de référence pour éviter les obstacles. Finalement, le véhicule doit suivre la trajectoire de référence modifiée tout en tenant compte du modèle cinématique du véhicule.

PiCar, le prototype d'une voiture autonome L'objectif du projet PiCar (Plateforme Instrumentée pour la Conduite Automatique sur Route) est de développer les outils nécessaires pour la

conception de voiture autonome. Il s'agit, là encore, d'un système autonome évoluant dans un environnement dynamique : la route, avec les autres voitures et les piétons. Actuellement, PiCar est capable de suivre une route et de repérer des obstacles afin de les éviter. La voiture revendique alors le titre de voiture autonome, celle-ci pouvant avancer sans conducteur et sans risque pour les usagers de la route. Pour cela, la voiture embarque deux types de capteurs.

- Des capteurs proprioceptifs qui délivrent des informations comme la vitesse, l'accélération, la direction de la voiture, etc.
- Des capteurs extéroceptifs. Il s'agit des caméras, télémètres, et des capteurs à ultrasons. Ces données externes sont utilisées notamment pour détecter les obstacles.

Précisément, la voiture est équipée d'un système de vision stéréoscopique, relié à un ordinateur central chargé d'exécuter le traitement d'image. Un des objectifs de ces travaux est d'améliorer la détection d'objets.

1.2 Position du problème

Comme dans la plupart des problèmes où l'on s'intéresse à un système, différentes disciplines ou domaines scientifiques collaborent. Notre objectif était de définir les principales briques en vue de la construction d'un système doté d'autonomie, c'est-à-dire capable d'évaluer le contexte, d'en déduire la trajectoire à suivre en fonction du contexte, et la commande à appliquer pour suivre cette trajectoire. Notre problème est donc initialement un problème de planification de trajectoire. Or nous défendons l'idée que cette planification est contextuelle au sens où elle dépend du contexte extérieur et de la dynamique du véhicule (qui donne le contexte d'applicabilité de la commande). Aussi nos contributions principales se situent-elles dans l'estimation du contexte extérieur ou environnement du véhicule, et du contexte lié à la commande du véhicule.

La première question que nous nous sommes posée est : "Qu'est ce qui dans l'environnement extérieur va influencer le système ADAS ?" Classiquement, nous citons :

- le contexte général de l'environnement extérieur, notamment les conditions météorologiques (par exemple, présence de brouillard) et les conditions de circulation (par exemple, densité du trafic) ;
- les objets énumérables liés au contexte routier, notamment la route et la signalisation routière ;
- les objets non énumérables, comme les bâtiments (en l'absence de carte détaillée), les véhicules, les piétons.

Pour les objets énumérables, le problème est celui de la reconnaissance de ces derniers, tandis que pour les objets non énumérables, le problème est celui de la détection et éventuellement de l'estimation de leur densité.

Après la définition du contexte externe, l'élaboration d'un schéma de conduite s'avère nécessaire pour évoluer dans ce contexte. La planification de trajectoire a pour objectif de trouver le meilleur chemin qui relie la position de départ P_i d'un mobile (véhicule, robot) à la position finale P_f, en respectant un certain nombre de contraintes et de critères. Ces derniers dépendent généralement des caractéristiques du véhicule (sa géométrie, sa cinématique et sa dynamique) et de l'environnement (non collision avec des obstacles fixes et prise en compte d'interactions de contact avec le véhicule). Explicitons quelques termes de base :

- Une **configuration** est l'ensemble des paramètres caractérisant de manière unique le véhicule dans son environnement ;
- L'ensemble des configurations, que l'on nomme **espace des configurations** sera noté $\mathscr{C}$. On munit généralement $\mathscr{C}$ d'une structure de variété différentiable, typiquement sur $\mathbb{R}^n$, où n est la dimension de $\mathscr{C}$;

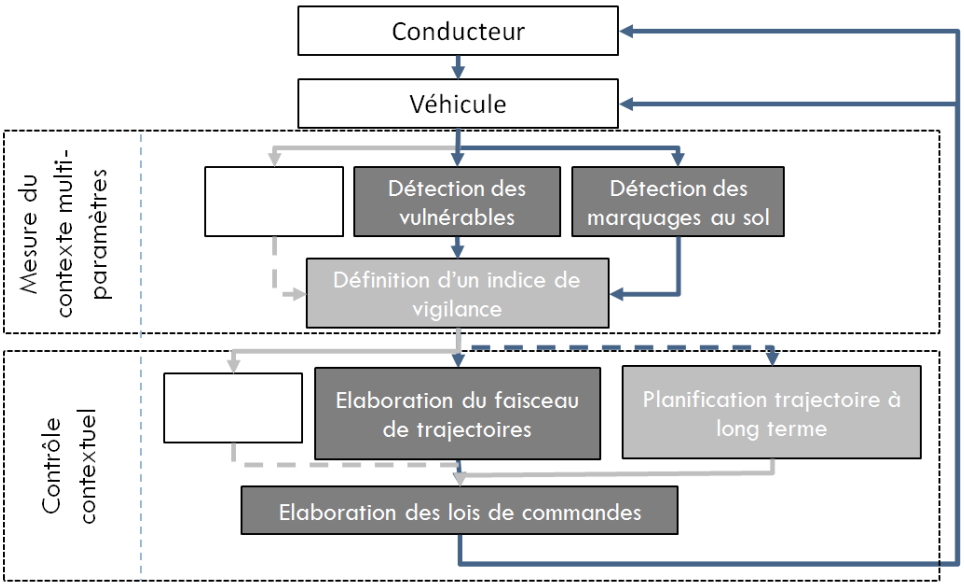


FIGURE 1.1: *Illustration des différents modules du mesure du contexte et du contrôle du véhicule.*

- Une **trajectoire** est une fonction continue d'un intervalle (éventuellement semi ouvert de type $[t_0, +\infty[$) de $\mathbb{R}$ dans $\mathscr{C}$;
- Une trajectoire est dite **admissible** si elle satisfait aux équations différentielles du modèle du véhicule ;
- La **planification de trajectoires** est le calcul d'une trajectoire admissible est sans collision du véhicule entre une configuration de départ et une configuration d'arrivée.

Dans la section 4.1.1 du chapitre 4 nous présentons certaines méthodes de planification en expliquant brièvement leur principe ainsi que leurs avantages et inconvénients.

1.3 Contribution

Dans ce chapitre, nous avons vu que les systèmes autonomes nécessitent une connaissance *a priori* de l'environnement et des obstacles présents dans cet environnement afin de définir la trajectoire à suivre.

Le rôle des systèmes d'aide à la conduite est d'alerter le conducteur sur les dangers potentiels ainsi que de lui proposer diverses trajectoires optimales (en termes de consommation du carburant, de trajet le plus court, de vitesse pour prévenir les collisions avec des obstacles...).

Nous défendons l'idée d'un système intermédiaire, i.e. partiellement doté d'autonomie, mais surtout capable d'évaluer la fiabilité et la pertinence de ses décisions versus celles du conducteur. Concrètement un tel système serait capable de prendre la main soit en cas d'urgence (par exemple, obstacle apparu très soudainement et à faible distance), soit en cas de défaillance du conducteur (par exemple malaise, endormissement). Dans un tel système le conducteur fait donc partie des éléments du contexte. Il fait également partie des éléments décideurs (sauf en cas de défaillance). Ainsi, le système peut présenter au conducteur un ensemble de trajectoires (appelé faisceau) parmi lesquelles choisir, défini à partir de différents critères (de vitesse, de confort, de sécurité, d'évitement de vulnérables).

Notre contribution se situe en amont de la définition de tels systèmes. Nous avons dit qu'une des premières tâches de ce type de systèmes est d'évaluer le contexte extérieur (hors conducteur) pour décider d'un ensemble de solutions (faisceau de trajectoires), et, d'une part en présence de plusieurs

solutions (i.e. en l'absence d'arrêt immédiat), et d'autre part en l'absence de défaillance du conducteur, d'interroger le conducteur (dans le véhicule ou à distance en cas de système téléopéré).

Notre contribution concerne l'estimation du contexte extérieur et la définition du faisceau de trajectoires incluant la faisabilité de la commande induite. La figure 1.1 illustre la décomposition de la solution adoptée. La première structure la relation entre le conducteur et le véhicule, la deuxième partie et la mesure du contexte et la troisième consiste au commande du véhicule. Donc nous définissons le contexte du milieu extérieur (urbain, non urbain autoroute, ..) et nous estimons la dangerosité de la situation et la densité des objets vulnérable. Bien évidemment d'autres mesures peuvent être introduites comme le confort des passagers. Dans les cas classique, une solution de trajectoire est proposée suivie d'un processus de suivie de trajectoire. Or tous ces paramètres sont non stationnaires temporairement ce qui peut causer un inconvénient majeur pour ce type de solutions. Ainsi nous avons choisi d'utiliser un faisceau de trajectoires.

La suite de ce manuscrit comprend alors deux parties. La première concerne le traitement d'images et la seconde l'automatique.

Première partie

Traitement d'images

Chapitre 2

Détection d'objets 'inattendus'

Sommaire

2.1 Introduction 12

2.2 État de l'art 12

2.2.1 Détection de changement en vision monoculaire 12

2.2.2 Détection d'objets par stéréovision 15

2.2.3 Conclusion 21

2.3 Approche proposée 21

2.3.1 Historique : des *gestalts* au principe de Helmholtz 21

2.3.2 Méthode développée 27

2.3.3 Application 47

2.4 Conclusion 59

2.1 Introduction

La détection des objets vulnérables et plus précisément les piétons est l'une des fonctionnalités nécessaires pour les systèmes autonomes et les systèmes d'aide à la conduite. Dans cette partie, nous nous intéressons à la détection des objets dans une séquence vidéo acquise par une caméra embarquée dans un véhicule en mouvement. Dans le cadre de l'application "mesure du contexte routier", deux types d'objets vont principalement susciter notre intérêt :

– Les objets en mouvement (e.g. voitures, vulnérables : piétons, cyclistes, etc.),

– Les objets saillants par rapport à la route (e.g. obstacles au sol, voitures, piétons arrêtés, etc.).

Par ailleurs, n'oublions pas que la caméra étant elle-même embarquée, elle est en mouvement, ce qui rend le problème autrement plus complexe que dans le cas d'une caméra statique. En particulier, contrairement au cas de la vidéosurveillance, il n'est pas possible d'acquérir de façon préalable des données représentant l'image en l'absence d'objet d'intérêt (images dites de *background*), et il faut compenser l'égomouvement de la caméra si l'on veut comparer des images acquises à deux instants différents correspondant à deux positions différentes de la caméra.

La suite de ce chapitre présente l'état de l'art sur la détection d'objets 'inattendus' via différentes approches de traitement d'images en vision mono ou stéréoscopique. Ce terme, 'inattendu', fait référence à quelque chose qui s'écarte de notre modèle. Par exemple, il s'agira d'une apparition (ou disparition) d'objet ou du déplacement d'un objet par rapport à une scène connue, et en termes de traitement d'images on cherchera des changements par rapport à une image de fond (*background*, selon la terminologie anglaise). Pour éviter les ambiguités par rapport à des méthodes qui traiteraient de la détection/reconnaissance/identification d'objets connus dans une scène inconnue, on parlera de détection de changement, les objets sont détectés par rapport à un modèle à t, qui peut être une image de fond ou une image précédente dans la séquence.

Dans la deuxième partie du chapitre, nous nous focalisons sur les modélisations dites *a contrario* et l'approche proposée est discutée. Enfin, sa pertinence est évaluée au travers des résultats obtenus sur des données simulées et sur des données réelles.

2.2 État de l'art

Dans la détection d'objets par un système de vision, la littérature distingue principalement : la vision monoculaire, c'est à dire utilisant une seule caméra, et la vision stéréoscopique, c'est-à-dire utilisant deux caméras pour observer la même scène. Dans cet état de l'art nous suivons cette distinction.

2.2.1 Détection de changement en vision monoculaire

Classiquement en détection de changements, on distingue entre les méthodes qui visent à suivre un objet et les méthodes de détection à proprement parler des changements entre deux images. Dans le premier cas, l'objet considéré est déjà présent dans l'image précédente, et toute information sur sa position précédente (forme, etc.) peut être utilisée pour le détecter dans l'image courante. Dans le deuxième cas, la détection de changement, la seule information disponible (hors de l'image analysée) est l'image de *background* (ou le *modèle à t*).

2.2.1.1 Détection de changement

Comme dit précédemment, le changement est défini par rapport à l'image *background* (ou le *modèle à t*). Une façon de formaliser cette idée est de supposer que l'image observée est la somme de deux images, respectivement appelées *foreground* et *background*. Ainsi un prétraitement extrêmement classique consiste à soustraire ce *background* à l'image étudiée. L'image *background* peut être choisie parmi une liste d'images de la séquence qui ne contiennent aucun objet à détecter : scène vide d'objet d'intérêt.

Dans la pratique, les scènes réelles contiennent des structures et des configurations du paysages qui varient que ce soit dû un changement de position des objets statiques (on rappelle qu'ici on ne s'intéresse pas au suivi d'objets) ou bien dû à un changement des conditions d'acquisition (luminosité, nuages, ...). Aussi, une partie des travaux de recherche se sont-ils intéressés au développement de techniques robustes de mise à jour de l'image *background* utilisée (ou image de référence). Sans prétendre être exhaustif, citons quelques techniques proposées dans ce contexte pour estimer l'image *background*.

Tout d'abord, lorsque la scène est sujette à des variations non récurrentes ou périodiques (changement de luminosité, modification des objets constituant le *background*, e.g. voitures en stationnement présentes ou non), l'image *background* doit intégrer ces différentes modifications afin de ne pas les percevoir comme des changements (i.e. n'appartenant pas au *background*). Les premières approches consistent bien sûr à faire un filtrage sur une fenêtre temporelle glissante (par exemple moyenne sur les n images précédentes), sachant que le choix du filtre doit être pertinent. Très rapidement est apparue l'idée que le bruit sur l'image *background* était non stationnaire (son amplitude de même que celle des variations dépend du pixel considéré). Ainsi, a-t-on considéré des images *background* avec deux couches : une qui représente la valeur moyenne du fond, m_s (en chaque pixel s) et une qui représente l'écart-type σ_s (en chaque pixel), l'idée étant ensuite de prendre en compte σ_s dans la décision de détection de changement. Ainsi, pour le modèle gaussien adaptatif [132], les paramètres (m_s, σ_s), en chaque pixel s, sont calculés sur une séquence sans changement (i.e. sans objet en mouvement), et sont mis à jour à partir de l'observation $y_{s,t}$ à t en ce pixel selon les relations : $m_s \leftarrow (1-\rho)\, m_s + \rho.y_{s,t}$ et $\sigma_s^2 \leftarrow (1-\rho)\, \sigma_s^2 + \rho\,(y_{s,t} - m_s)^2$, où ρ est le paramètre d'apprentissage. De façon similaire, le filtre $\Sigma-\Delta$ (Sigma-Delta [90]) permet une actualisation du *background*, extrêmement simple et relativement robuste. En chaque nouvelle image disponible à t, l'actualisation de (m_s, σ_s), en chaque pixel s, se fait simplement à partir de la valeur $y_{s,t}$ observée à t (une version simple est que si $|y_{s,t} - m_s| > \delta$, alors $m_s \leftarrow m_s + signe\,(y_{s,t} - m_s)\,\delta$, et si $|y_{s,t} - m_s| > 0$, alors $\sigma_s \leftarrow \sigma_s + signe\,(|y_{s,t} - m_s| - \sigma_s)$, i.e. pour m_s stable, σ_s est la médiane du bruit autour de m_s). Un inconvénient de cette méthode de mise à jour de l'image *background* est la rapidité d'assimilation des changements : par exemple, pour le cas classique $\delta = 1$, un objet présentant une différence de radiométrie de 50 sera assimilé (donc non détecté dans les images suivantes) après 50 images, soit deux secondes pour une caméra à $25Hz$.

Ensuite, lorsque la scène est sujette à des variations pseudo-périodiques (e.g. mouvements des branches dans les arbres dus au vent), l'image *background* doit intégrer ces variations afin de ne pas les percevoir comme des changements (i.e. n'appartenant pas au *background*). Nous avons vu qu'une manière de prendre en compte ces variations est d'avoir un écart-type associé à ces pixels très importante, et par suite dans la décision de prendre en compte cet écart-type (e.g. le seuil de décision appliqué sur l'image différence absolue, entre l'image courante et l'image *background*, est proportionnel à l'écart-type au pixel considéré). L'inconvénient est que dans les régions sujettes à des variations pseudo-périodiques tout changement d'amplitude inférieure à ces variations ne sera pas perceptible. Kyungnam et al. ont alors proposé [77, 78] une représentation de l'image *background* à partir d'un

dictionnaire, appelé *CodeBook*. Chaque pixel s de l'image *background* est alors représenté par plusieurs (mettons k ici) triplets $(m_s, \sigma_s, c_s)_i, i \in \{1, ..., k\}$, ainsi qu'un triplet $(m_s, \sigma_s, c_s)_0$ pour l'apprentissage. La mise à jour par l'observation $y_{s,t}$ est faite comme suit : soit $j = \underset{i \in \{1,...,k\}}{argmin} (|m_{s,i} - y_{s,t}|)$ (j est l'indice du triplet contenant la moyenne la plus proche de l'observation) ; si $|m_{s,j} - y_{s,t}| < \gamma.\sigma_{s,j}$, alors mise à jour du triplet d'indice j (par exemple par le filtre sigma-delta) et $c_{s,j} = t$ (le paramètre c_s stocke le dernière date où le triplet a été rencontré), sinon, soit le triplet d'apprentissage satisfait la condition ($|m_{s,0} - y_{s,t}| < \gamma.\sigma_{s,0}$) et alors le nombre de fois où il a été rencontré est incrémenté ($c_{s,0} \leftarrow c_{s,0}+1$) et ses paramètres $(m_s, \sigma_s)_0$ sont mis à jour, soit le triplet d'apprentissage est réinitialisé $(m_s, \sigma_s, c_s)_0 \leftarrow (y_{s,t}, \sigma_{init}, 1)$. Le nombre de mots du dictionnaire associé à un pixel est géré en ajoutant le triplet d'apprentissage après qu'il ait été sélectionné un certain nombre de fois ($c_{s,0} > c_{min}$) et inversement un triplet est éliminé s'il n'a pas été sélectionné depuis longtemps ($t - c_{s,i} > c_{max}$). Moyennant un choix des trois paramètres $(\gamma, c_{min}, c_{max})$ de l'algorithme *ad hoc* et un cout supplémentaire en temps de calcul, ce modèle de *background* permet de gérer des images dans lesquelles certaines zones ne peuvent pas être modélisées par une seule valeur.

Nous gardons le terme de *background* même pour des images peut être multicouches, avec des interprétations pour chaque couche, spécifique à la formation de ce *background*, par exemple couche des moyennes et couche des écart-types pour chaque mot du dictionnaire). Ayant alors obtenu une image de *background*, un test est effectué pour déterminer si la valeur observée s'éloigne de manière significative de la moyenne (la plus proche en cas de *CodeBook*). Ce test est généralement de nature probabiliste (mélange de gaussiennes [31, 107, 120]), même si d'autres approches ont été proposées ([64], [45]). Par exemple, dans le cadre de l'approche par intervalles, l'image *background* contient trois couches : les valeurs minimale, maximale représentant l'intervalle des valeurs admissibles comme *background*, et la variation temporelle maximale entre images successives au pixel considéré, et un changement est décidé si la valeur observée sort de l'intervalle ou si la dynamique observée est supérieure à celle maximale [64].

Notons finalement que les méthodes décrites précédemment l'ont été au niveau pixelique (et donc sont relativement sujettes à des erreurs au niveau pixel), et qu'une façon de les rendre plus robustes est de les étendre au niveau fenêtre. Par exemple, si on suppose sue, la valeur d'un pixel sans changement est une variable aléatoire qui suit une distribution normale de moyenne et d'écart type donné, la somme des différences quadratiques sur une fenêtre suivra alors une loi χ^2, qui si la taille de la fenêtre est suffisamment grande tend rapidement vers la loi normale. Il est alors possible de prendre une décision à l'échelle de la fenêtre dès lors que la loi peut être supposée et calculée.

Pour conclure, la limite majeure de ces approches reste l'existence de paramètres (parfois nombreux) pour lesquels un apprentissage doit être décidé (choix d'une séquence d'apprentissage etc.) et l'existence d'un seuil final de décision (ou de paramètres reliant ce dernier aux paramètres du *background*, e.g. seuillage à n fois l'écart-type).

2.2.1.2 Suivi d'objets

Nous présentons à présent succinctement quelques techniques de suivi d'objet sachant que notre contribution dans cette thèse se situera en amont de l'étape de suivi. Dans le cas du suivi, l'objet est déjà présent dans l'image précédente et les informations sur cet objet (forme, texture, taille, position, vitesse...) sont utilisées pour le retrouver dans l'image courante. Parmi les méthodes de suivi les plus classiques, citons le filtre de Kalman [17], les filtres particulaires [16], les filtres markoviens, et les méthodes de suivi non-paramétriques tels que le mean-shift ([52],[29, 107]). Ces techniques peuvent s'appliquer sur des caractéristiques 2D (ou 3D) de l'objet. Ces caractéristiques sont généralement des

caractéristiques calculées au niveau objet (ou fenêtre), e.g. distribution de caractéristiques niveau pixel (e.g. la couleur) que l'on cherche à retrouver d'une image à la suivante. Notons que d'une part quand le suivi concerne un type d'objet spécifique (véhicules, piétons), l'*a priori* sur cet objet peut guider le choix des caractéristiques (e.g. [19]). Notons également que la recherche de l'objet dans l'image suivante peut être guidée par un *a priori* sur son mouvement (modèle de déplacement comme dans les filtres de Kalman, particulaires etc.).

Citons également dans ce paragraphe les contours actifs [1, 2, 89, 102, 103]. Ces derniers ont moins pour objectif de suivre un objet en déplacement (même s'ils peuvent être utilisés pour) que de suivre les déformations d'un objet. Ils ne travaillent donc pas à l'échelle de l'objet mais du pixel, même si la fonctionnelle qu'ils utilisent est globale. Classiquement, ces méthodes obtiennent la solution par minimisation d'une fonctionnelle incluant différents termes correspondant aux différents critères sur l'objet. Par exemple, dans [12], la fonctionnelle s'écrit $\alpha \int_{\Omega} d\sigma + \int_{D\setminus\Omega} (B-I)^2 \, d\sigma$ avec Ω la région correspondant à l'objet recherché, B l'image de *background*, I l'image courante de domaine D, et α un paramètre de pondération des termes. Cette écriture revient alors à chercher l'objet de surface minimale, qui contienne les points où I est le plus différent de B. D'autres fonctionnelles ont été proposées de façon à intégrer également des critères sur le contour de l'objet. Ces méthodes restent lourdes algorithmiquement et dépendantes de l'initialisation du contour de l'objet (dans le cas du suivi, on se sert du résultat de la détection sur l'image précédente).

Citons également les approches par flot optique. L'objectif est alors d'estimer le champ des vitesses des pixels de l'image entre deux images. Notons que les vitesses des pixels ne sont qu'indirectement (par projection de la scène 3D sur le plan image) reliées au déplacement des objets (de même d'ailleurs que les méthodes citées précédemment qui estimaient un déplacement de l'objet dans le plan image). Il existe deux grandes classes d'approches pour calculer le flot optique : les approches locales dont la plus célèbre est celle proposée par Lucas et Kanade [88], et les approches globales dont la plus célèbre est celle proposée par Horn et Schunck [66]. Depuis ces deux articles [88] et [66], de nombreuses autres méthodes ont été proposées : soit en vue d'obtenir des résultats plus performants : soit en tenant compte des valeurs du flot calculées dans les images précédentes (e.g. [99, 101, 129]), soit en introduisant des processus de type 'bord' pour respecter les frontières des objets (e.g. [14]), soit en développant des approches multiéchelles afin d'estimer les grands déplacements (supérieurs à plusieurs pixels ; e.g. [94, 4]) par exemple, soit enfin en combinant les deux approches locale et globale afin de bénéficier des avantages de chacune [21], soit en vue d'accélérer les calculs (e.g. l'algorithme FOLKI [82] permet d'estimer rapidement un flot optique dense par une technique de recalage de fenêtre de type Lucas-Kanade itératif et multirésolution).

En conclusion les méthodes de suivi sont nombreuses. Dans notre application il y aurait certainement un intérêt à les exploiter pour éliminer certaines fausses détections. Cependant notre travail est resté en amont du suivi des objets 'inattendus' et le suivi fait partie des perspectives.

2.2.2 Détection d'objets par stéréovision

Une partie des difficultés des techniques en monovision provenait du fait que les objets n'étant perçus que via leur projection sur le plan image, ils étaient moins distincts que dans l'espace 3D (ambiguités dues à la perte de l'information de coordonnée en profondeur). Aussi, la stéréovision a-t-elle pour objet de restaurer cette information 3D grâce à deux perceptions de la même scène selon deux angles de vue ou géométrie d'acquisition légèrement différentes, à la façon du système visuel. La stéréovision s'appuie sur le formalisme de la géométrie épi-polaire afin d'obtenir la coordonnée profondeur des points 3D : il s'agir d'apparier les points 2D visibles sur les images du système stéréo

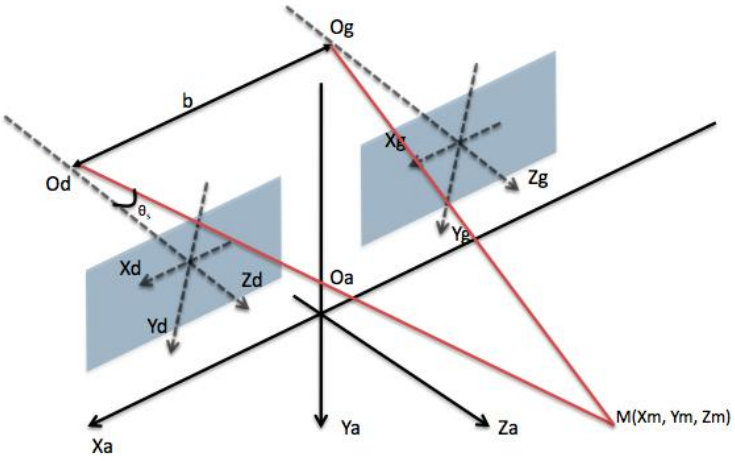


FIGURE 2.1: *Modélisation d'un capteur de stéréovision ([8]).*

(généralement deux images), de façon à obtenir une carte de disparité, qui peut être traduite en carte de profondeur, si les paramètres du système de stéréovision sont connus (e.g. grâce à un calibrage préalable). La figure 2.1 présente un schéma modélisant un capteur de stéréovision dans le cas où les images sont dites rectifiées, c'est-à-dire qu'il n'y a plus de distorsion radiale ou tangentielle et que les lignes épipolaires correspondent aux lignes de l'image, i.e. où deux pixels correspondant à un même point 3D se trouvent sur la même ligne. Dans la suite on suppose que les images stéréo ont été rectifiées.

Dans la suite de cette section, nous rappelons ici brièvement quelques notations et techniques de mise en correspondance, puis nous citons un exemple de détection d'objets à partir de données stéréoscopiques, enfin nous donnons un exemple de collaboration entre analyse structurelle et analyse du mouvement.

2.2.2.1 Projection stéréoscopique

Il existe de nombreux ouvrages développant les équations de la géométrie d'acquisition des images (en monovision comme en stéréovision). Nous donnons donc juste les résultats (e.g.[8]) de façon à introduire les notations utiles pour la suite. Sous l'hypothèse d'un modèle sténopé (*pin-hole* en anglais, [65]), dans le repère lié à la caméra (d'origine le centre optique), un point 3D de coordonnées $M = (X_M, Y_M, Z_M)^t$ est projeté, par conjugaison optique, sur le plan image en $m = \left(f' \frac{X_M}{Z_M}, f' \frac{Y_M}{Z_M}\right)^t$ avec f' la focale (en mm). Ensuite, la matrice photosensible va intégrer et échantillonner le signal sous forme de pixels, et opérer un changement de repère de sorte que les coordonnées de m dans l'image, avec une origine généralement en haut à gauche de l'image, soient

$$\begin{pmatrix} u_m \\ v_m \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_u & s_{uv} & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_m \\ y_m \\ 1 \end{pmatrix},$$

où $1/\alpha_u$ et $1/\alpha_v$ sont les dimensions d'un pixel selon les axes u et v (α_u etα_v en *pixels/mm*), s_{uv} le coefficient de non orthogonalité du capteur (supposé nul dans la suite), et (u_0, v_0) les coordonnées du centre optique de la caméra dans le repère de l'image; soit finalement, en posant $f_u = f'\alpha_u$ et

$f_v = f'\alpha_v$:

$$\begin{aligned} u_m &= u_0 + f_u \frac{X_M}{Z_M}, \\ v_m &= v_0 + f_v \frac{Y_M}{Z_M}. \end{aligned} \tag{2.1}$$

Dans le cas de deux caméras de stéréovision identiques (mêmes paramètres intrinsèques (f_u, f_v, u_0, v_0), on suppose également $f = f_u = f_v$), après rectification, les plans images sont supposés parallèles et alignés. Notons $(X_G, Y_G, Z_G)^t$ l'origine du centre optique de la caméra gauche et $(X_D, Y_D, Z_D)^t$ celui de la caméra droite, et soit b la distance entre les deux origines des repères des caméras (b est généralement appelée *baseline*, selon la terminologie anglaise). Ainsi, $(X_G, Y_G, Z_G)^t = (X_D + b, Y_D, Z_D)^t$. Le point 3D M se projette selon m_g sur le plan image gauche et selon m_d sur le plan image droite avec :

$$\begin{aligned} u_{m_d} &= u_0 + f\frac{X_M - X_D}{Z_M - Z_D} & \qquad u_{m_g} &= u_0 + f\frac{X_M - X_D - b}{Z_M - Z_D} \\ v_{m_d} &= v_0 + f\frac{Y_M - Y_D}{Z_M - Z_D} & \qquad v_{m_g} &= v_0 + f\frac{Y_M - Y_D}{Z_M - Z_D} \end{aligned} \tag{2.2}$$

En prenant comme référence la caméra droite, $(X_D, Y_D, Z_D) = (0, 0, 0)$, et en introduisant la différence de coordonnée colonne d'un même point 3D respectivement projeté sur chacune des images rectifiées de la paire stéréo, notée δ_m et appelée disparité, les équations reliant les coordonnées 3D du point M au coordonnées sur l'image (u_m, v_m) et à δ_m sont :

$$\begin{aligned} u_m &= u_0 + f\frac{X_M}{Z_M} \\ v_m &= v_0 + f\frac{Y_M}{Z_M} \\ \delta_m &= f\frac{b_s}{Z_M} \end{aligned} \tag{2.3}$$

La disparité δ_m est inversement proportionnelle à la coordonnée 3D en profondeur (Z_M).

L'appariement des points consiste à reconnaître, dans la paire d'images, les pixels représentant un même point 3D. La performance de l'appariement, aussi appelé mise en correspondance des pixels 2D, est donc cruciale puisqu'elle détermine la densité de la carte de disparité. Citons quelques techniques usuelles de mise en correspondance, locales ou globales.

Les méthodes locales consistent à, pour chaque pixel s de la première image I_1 de la paire stéréo, chercher son correspondant s' dans la seconde image I_2 dans un voisinage V_s autour du pixel de même coordonnées image que s (dans le cas d'images rectifiées, la recherche se fait uniquement selon les colonnes, V_s est donc un voisinage horizontal). L'hypothèse est donc $\forall s = p_{I_1}(M)\,,\ s' = p_{I_2}(M) \in V_s$,où M est un point 3D et $p_I(M)$ sa projection sur l'image I. Le choix de s' parmi les pixels de s peut se faire selon différents critères, généralement basés sur la minimisation d'une fonction de coût. Par exemple, pour le *Bloc-Matching* le coût à minimiser est la somme des différences absolues (notée ensuite $SAD(|W|, s, s')$) entre les pixels de fenêtres de taille $|W|$ respectivement centrées sur s pour l'image I_1 et sur s' pour l'image I_2. En outre, de façon à robustifier la décision, la mise en correspondance entre s et s' n'est validée que s'il y a accord mutuel, i.e. $s' = \underset{t \in V_s}{argmin}\,(SAD(|W|, s, t))$ et $s = \underset{t \in V_{s'}}{argmin}\,(SAD(|W|, t, s'))$, et si le coût minimal obtenu est inférieur à un seuil donné. Le choix de la taille de la fenêtre est alors fonction de l'écart maximal supposé entre les positions des pixels se correspondant dans les deux images, qui lui-même dépend de l'écart physique des deux caméras (*baseline*). Une fenêtre de grande taille (e.g. 15×15) convient pour traiter des régions contenant des objets (très) proches de la caméra et/ou faiblement texturées, mais il s'en suit une faible précision de localisation du pixel apparié. Une fenêtre de petite taille (e.g. 3×3) permet une bonne localisation du

pixel à apparier à condition qu'il soit effectivement très proche du pixel initial (cas d'objets éloignés de la caméra) et que le bruit soit faible.

Ces méthodes locales ne sont généralement fiables que sur des zones texturées de l'image, et elles fournissent des cartes éparses (absence de valeur en cas d'échec de l'appariement, en particulier sur les régions homogènes où l'appariement est ambigu). Néanmoins, ces méthodes restent les plus rapides et largement utilisées pour des applications embarquées. A l'inverse, les méthodes globales estiment la carte de disparité globalement et de façon dense. Elles utilisent des approches de type variationnel, i.e. avec minimisation d'une fonction appelée énergie, à l'instar de l'approche d'Horn et Schunk [66] pour le calcul du flot optique. Ces méthodes ont pour elles de fournir une carte de de disparité dense et robuste aux problèmes d'occultations, de faible densité de texture ou de motifs périodiques. Cependant, ces méthodes sont lourdes et nécessitent d'importantes ressources de calcul. La programmation dynamique est un exemple d'algorithme utilisé pour résoudre ces problèmes d'optimisation globale.

Dans la suite, nous notons Δ l'image disparité (dans la géométrie de l'image acquise par la caméra droite).

2.2.2.2 Détection d'objets *a priori*

Plusieurs méthodes de détection d'objets exploitent l'information 3D déduite du système de stéréovision, e.g. [20, 22, 122]. Il est bien entendu possible d'étendre les différentes techniques monovisions qui utilisent des caractéristiques 2D des groupes de pixels à des caractéristiques points 3D (notamment la compacité et la connexité spatiale). Dans cette section, nous limitons notre état de l'art à une méthode à la fois originale et très efficace, à savoir la uv-disparité [79, 67]. Cette méthode, initialement proposée par Labayrade [79] est basée sur une approche cumulative, i.e. type transformée de Hough. Il s'agit de trouver un espace où des objets paramétriques se projettent avec un degré de liberté en moins (au moins) de façon à pouvoir s'accumuler. Dans le cas de la transformée de Hough pour des droites, une droite du plan image se projette en un point (de coordonnées (a, b) ou (r_0, θ) selon le choix de l'espace de projection dicté par la représentation paramétrique de la droite, cartésienne ou polaire). Dans le cas de la v-disparité, l'espace de projection est choisi de sorte qu'un plan parallèle à la direction l'axe X (colonnes de l'image) et non vertical, notamment plan horizontal, se projette selon une droite.

Soit un plan d'équation

$$Y_M = h_r - Z_M \sin\theta, \tag{2.4}$$

où h_r est la distance à laquelle le plan intersecte l'axe Y et θ l'angle entre ce plan et le plan horizontal. Alors, la disparité pour les points de ce plan est :

$$\delta_m = \frac{b_s}{h_r}\left[(v_m - v_0) + f\sin\theta\right] \text{ si } v_m \geq v_0 - f\sin\theta, \tag{2.5}$$

où v est la coordonnée en ligne de m projeté sur l'image de M appartenant au plan considéré. Donc, pour une ligne donnée, la disparité associée est constante (pour les pixels représentant des points du plan considéré), et lorsque la ligne varie la disparité varie linéairement (toujours pour les pixels représentant des points du plan considéré). Aussi, [79] a-t-il proposé de détecter le plan comme un segment de droite dans l'espace $v\Delta$ où, pour chaque ligne de l'image, la ligne de l'espace $v\Delta$ représente l'histogramme de la disparité sur cette ligne du plan image. Pour un plan correspondant à un nombre suffisant de pixels, la disparité en une ligne donnée apparaitra comme le ou un mode (maximum) de l'histogramme, et l'équation du segment de droite passant par ces valeurs maximales dans l'histo-

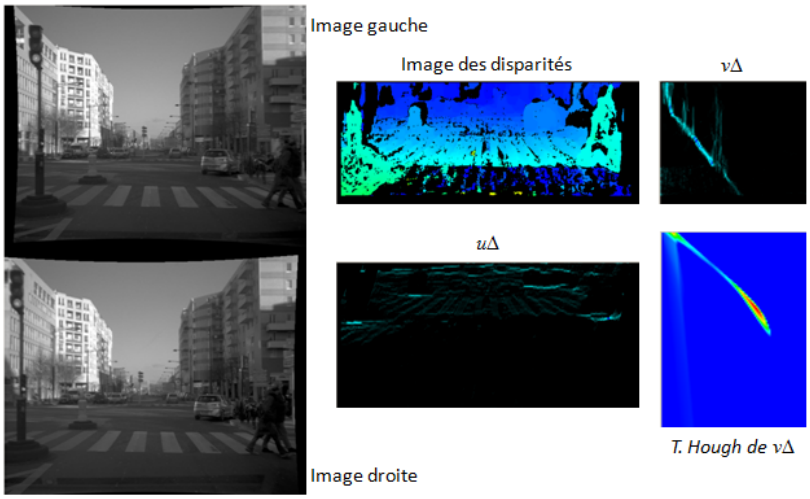


FIGURE 2.2: *UV-disparité.*
L'image de disparité (en fausses couleurs avec en noir les valeurs pour lesquelles la disparité n'a pu être calculée) est calculée en appariant les pixels de l'image gauche et de l'image droite (restreintes à la moitié inférieure), le plan de la route apparait comme une droite dans l'image $v\Delta$, détectable par transformée de Hough (axe x : coordonnée de l'intersection avec la ligne du bas de l'image, axe y : angle $\in \left[0, \frac{\pi}{4}\right]$), les obstacles sont détectables sur l'image $v\Delta$ (projection selon la dimension des lignes) ou sur l'image $u\Delta$ (projection selon la dimension des colonnes).

gramme des valeurs de disparité permet de retrouver les paramètres du plan : la pente du segment, détectée par exemple par transformée de Hough, permet d'obtenir h_r et les extrémités du segment, $(\delta_m, v_m) = (0, v_0 - f \sin\theta)$, et $\left(\frac{b_s}{h_r}\left[(v_{max} - v_0) + f \sin\theta\right], v_{max}\right)$, dans l'espace $v\Delta$, permettent de retrouver θ.

En résumé, par projection dans l'espace $v\Delta$ il est possible de détecter la route si cette dernière est modélisable comme une succession de morceaux de plans horizontaux et obliques par rapport à Z. Une fois détecté le plan de la route, la détection des objets saillants par rapport à ce plan peut se faire par exemple en détectant les pixels qui se projettent dans $v\Delta$ au-dessus de la droite représentant le plan de la route cf. figure 2.2 (en effet un objet assimilable à un plan vertical a une disparité constante lorsque la ligne varie et se projette donc comme un segment vertical dans l'espace $v\Delta$).

On peut aussi détecter des plans parallèles à l'axe Y, de manière similaire à l'approche précédente pour détecter des plans parallèles à X, en changeant l'espace de projection. Maintenant, on considère l'espace $u\Delta$, où pour chaque colonne de l'image, la colonne de l'espace $u\Delta$ représente l'histogramme de la disparité pour cette colonne de l'image. Maintenant, pour un objet assimilé à un plan vertical parallèle à Y, qui est à peu près le cas des objets existant dans une scène (piétons, voitures, bâtiments, etc.), la disparité de cet objet est à peu près linéaire selon les lignes, voire constante en ligne dans le cas d'objets parallèles au plan image.

L'étape suivante a pour but de séparer les différents objets en les associant à des composantes connexes obtenues par seuillage à hystérésis dans l'image $u\Delta$. On en déduit finalement la boite englobante (dans l'image originale) à partir des limites des composantes connexes (valeurs min et max) en colonnes dans l'espace $u\Delta$, et en lignes dans l'espace $v\Delta$ en supposant que l'objet est en contact avec le plan de la route. Il est également possible d'ajouter un filtrage sur la taille et la position des objets afin de supprimer certaines fausses alarmes.

2.2.2.3 Collaboration entre 3D et mouvement

La méthode présentée au paragraphe précédent visait à détecter des objets de forme paramétrique connue *a priori*. Citons à présent pour clore ce bref tour d'horizon, l'existence de techniques de détection 3D des objets en mouvement, qui combinent les informations 3D issues du système de vision stéréo et des informations sur le mouvement issues de l'analyse d'images successives dans la séquence (e.g. [8]).

Rappelons ici les équations régissant le déplacement des points 3D statiques de la scène de part le mouvement de la caméra puisque nous en aurons également besoin pour compenser l'égomouvement.

Soit un repère absolu $\mathcal{R}_a$ où la caméra évolue (par exemple, embarquée sur un véhicule), et un repère relatif $\mathcal{R}_r$ lié à la caméra. Nous supposons que dans le repère absolu le mouvement de la caméra entre deux instants t_0 et t_1 est composé d'une composante translationnelle $\mathbf{T} = \begin{pmatrix} T_X & T_Y & T_Z \end{pmatrix}^t$, et d'une composante rotationnelle $\mathbf{\Omega} = \begin{pmatrix} \Omega_X & \Omega_Y & \Omega_Z \end{pmatrix}^t$ faible de sorte que l'approximation des 'petits angles' soit valide.

Soit $M = \begin{pmatrix} M_X & M_Y & M_Z \end{pmatrix}^t$ un point statique de la scène. A l'instant $t = t_0$, nous considérons que les deux repères $\mathcal{R}_a$ et $\mathcal{R}_r$ sont confondus.

Les coordonnées de M à l'instant t_1 sont alors :

$$\begin{aligned} M(t_1) &= \begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix}_{\mathcal{R}_a} \\ &= \begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix}_{\mathcal{R}_r} + \begin{pmatrix} M_X \\ M_Y \\ M_Z \end{pmatrix}_{\mathcal{R}_r} \wedge \begin{pmatrix} \Omega_X \\ \Omega_Y \\ \Omega_Z \end{pmatrix} - \begin{pmatrix} T_X \\ T_Y \\ T_Z \end{pmatrix} \\ &= \begin{pmatrix} M_X + \Omega_Z M_Y - \Omega_Y M_Z - T_X \\ M_Y + \Omega_Z M_X - \Omega_X M_Z - T_Y \\ M_Z + \Omega_Y M_X - \Omega_X M_Y - T_Z \end{pmatrix} \end{aligned} \tag{2.6}$$

On en déduit alors les coordonnées du pixel m projeté sur l'image du point 3D M aux instants t_0 et t_1 :

$$m(t_0) = \begin{pmatrix} x_m \\ y_m \end{pmatrix} = \begin{pmatrix} f\frac{X_M}{Z_M} \\ f\frac{Y_M}{Z_M} \end{pmatrix} \tag{2.7}$$

$$m(t_1) = \begin{pmatrix} x'_m \\ y'_m \end{pmatrix} = \begin{pmatrix} \frac{x_m + y_m\Omega_Z - f\Omega_Y - f\frac{T_X}{Z_M}}{\frac{x_m}{f}\Omega_Y - \frac{y_m}{f}\Omega_X - \frac{T_Z}{Z_M} + 1} \\ \frac{y_m - x_m\Omega_Z + f\Omega_X - f\frac{T_Y}{Z_M}}{\frac{x_m}{f}\Omega_Y - \frac{y_m}{f}\Omega_X - \frac{T_Z}{Z_M} + 1} \end{pmatrix} \tag{2.8}$$

Ici x_m et y_m sont les coordonnées (colonne,ligne) dans l'image par rapport au centre optique : $x_m = u_m - u_0$ et $y_m = v_m - v_0$. Nous avons choisi dans cette partie bibliographique de garder les notations usuelles du domaine. L'équation est difficile à manipuler de part le terme Z_M qui est inconnu, sauf si l'on dispose d'un système stéréovision permet de déduire Z_M de la disparité δ_m : $Z_M = \frac{fb_s}{\delta_m}$.

Dans [8], l'auteur présente un formalisme élégant qui exprime les coordonnées de la projection du point 3D M statique à t_1 à partir de celles du même point projeté à t_0 et du mouvement *a priori* de

la caméra ($\mathbf{\Omega}$,$\mathbf{T}$) : si $m\left(t_0\right) = \left(x_m, y_m, \delta_m\right)^t$,

$$m\left(t_1\right) = \begin{pmatrix} x_m^{'} \\ y_m^{'} \\ \delta_m^{'} \end{pmatrix} = \begin{pmatrix} \frac{x_m + y_m\Omega_Z - f\Omega_Y - \frac{T_X\delta_m}{b_s}}{\frac{x_m}{f}\Omega_Y - \frac{y_m}{f}\Omega_X - \frac{T_Z\delta_m}{f.b_s} + 1} \\ \frac{y_m - x_m\Omega_Z + f\Omega_X - \frac{T_Y\delta_m}{b_s}}{\frac{x_m}{f}\Omega_Y - \frac{y_m}{f}\Omega_X - \frac{T_Z\delta_m}{f.b_s} + 1} \\ \frac{\delta_m}{\frac{x_m}{f}\Omega_Y - \frac{y_m}{f}\Omega_X - \frac{T_Z\delta_m}{f.b_s} + 1} \end{pmatrix} \tag{2.9}$$

Dans [8], Bak propose de faire collaborer les informations de mouvement et structurelles de la façon suivante : à partir de points appariés sur deux images successives, il construit un système linéaire en ($\mathbf{\Omega}$,$\mathbf{T}$) , dont la solution $\left(\hat{\mathbf{\Omega}},\hat{\mathbf{T}}\right)$ (obtenue par minimisation du système linéaire présenté à l'équation (2.9)) correspond à l'égo-mouvement recherché. A partir de cette solution $\left(\hat{\mathbf{\Omega}},\hat{\mathbf{T}}\right)$, l'auteur propose de détecter les objets mobiles par mesure du mouvement résiduel après compensation de l'influence de l'égo-mouvement. Notons que la combinaison des informations de mouvement et structurelles est donc séquentielle et au bon sens du terme *ad hoc*.

2.2.3 Conclusion

En guise de conclusion, citons ce qui dans les travaux précédents est à la base de notre approche.

Tout d'abord, précisons que nous nous sommes limités au cas d'un système de stéréovision. Nous avons également présupposé le calcul des cartes de disparité sans discuter de la façon dont ces dernières étaient obtenues [125]. Nous nous sommes ensuite appuyé sur le formalisme développé dans [8], en particulier l'équation (2.9) nous servira à estimer l'image à t_1 connaissant l'image à t_0 et supposant le mouvement 6D de la caméra et supposant les objets statiques.

Nous verrons que notre approche théoriquement peut estimer de façon conjointe l'égomouvement (mouvement 6D de la caméra) et les changements par rapport à l'image précédente (objets ayant un mouvement propre. Malheureusement le temps de calcul est prohibitif aussi nous restreindrons nous à affiner l'égomouvement à partir d'un mouvement présupposé (toujours conjointement à l'estimation des objets en mouvement dans la scène). Toujours est-il que contrairement à [8], notre estimation (en l'absence de contraintes de temps de calcul) serait non séquentielle mais simultanée.

Enfin, nous verrons également que nous détectons des objets par rapport à un modèle, qui est une généralisation de la notion d'image de fond (*background*). Bien que nous ne le présentions pas, les travaux visant à généraliser cette image de fond, e.g. via un modèle avec dictionnaire, pourraient être exploités.

2.3 Approche proposée

2.3.1 Historique : des *gestalts* au principe de Helmholtz

La modélisation a contrario en traitement d'images est issue de la rencontre de deux principes : la notion de regroupement issue de la théorie des *gestalts* et le principe de Helmholtz. Nous rappelons très brièvement ces deux principes.

Dès la fin du $XIX^{\text{ème}}$ siècle, Helmholtz et Hering s'intéressent à la perception visuelle et s'interrogent sur l'interprétation à donner aux illusions visuelles, notamment optico-géométriques : quels sont les mécanismes qui conduisent à ces illusions et faut-il remettre en cause la fiabilité de la perception ?

Par exemple, dans le cas de l'illusion de Hering (cf. figure 2.3(a)), la convergence des lignes non verticales simule une perspective et une fausse impression de profondeur, qui font apparaitre les lignes

(a) (b)

FIGURE 2.3: *Exemples d'illusions visuelles : (a) illusion de Hering, (b) triangle impossible.*

verticales non parallèles alors qu'elles le sont. Précisément l'illusion provient d'un conflit entre deux lois de regroupement conduisant à deux interprétations différentes et incompatibles.

Les Gestaltiens (dont le nom remonte aux premiers articles [130, 131]) posent alors comme conjecture qu'il existe "un ensemble fini de lois de groupement gouvernant la synthèse perceptuelle des objets" et s'intéressent à l'identification de ces lois de regroupement : similarité de couleur/forme/texture, proximité, symétrie, parallélisme, bonne continuation/fermeture, perspective, complétion, etc. Ils montrent que ces lois sont si fortes qu'elles l'emportent sur le bon sens physique, conduisant ainsi à des illusions visuelles [1] (cf. figure 2.3(b)).

Une autre conclusion très importante pour le développement de mesures *a contrario* en traitement des images est que "une loi de groupement n'est active dans une image que si elle ne crée pas un trop grand nombre de détections." Cette propriété conduit à la formulation du principe de Helmholtz : "Dans le bruit, on ne voit rien" (Attneave, D.Lowe), ou autrement dit : "Toute structure trop régulière pour apparaitre 'par hasard' attire l'attention et devient une perception."

Illustrons le principe de Helmholtz sur l'exemple des dates d'anniversaire, tiré de [41], à savoir : "Faut-il s'étonner que au moins deux personnes aient la même dates d'anniversaire parmi N personnes ?"

Les dates d'anniversaire sont modélisées par des variables aléatoires indépendantes et identiquement distribuées (i.i.d.) selon la distribution uniforme sur $[1, 365]$, 365 est le nombre de jours dans une année en négligeant les années bissextiles. Soit X la variable aléatoire représentant le nombre de personnes de même date anniversaire. La probabilité qu'au moins deux personnes aient la même date d'anniversaire est : 1 moins la probabilité que toutes aient une date d'anniversaire différente, soit :

$$P(X \geq 2) = 1 - \left(1 - \frac{0}{365}\right) \times \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times .... \times \left(1 - \frac{N-1}{365}\right) = 1 - \prod_{k=0}^{N-1} \left(1 - \frac{k}{365}\right).$$

Numériquement pour $N = 3$, $P(X \geq 2) = 0.0082$, pour $N = 25$, $P(X \geq 2) = 0.5687$, et pour $N = 50$, $P(X \geq 2) = 0.9704$. On voit bien que, très rapidement quand N croît, "il ne faut plus s'étonner". L'analogue en image est l'exemple d'un alignement de points qui n'est étonnant que si ces points ne sont pas 'noyés' dans une image entièrement remplie de points (exemple repris quantitativement plus loin).

Dans ce chapitre nous rappelons les fondements de la modélisation *a contrario* proposée par Desolneux et al. [36] pour la détection d'évènements significatifs dans les images, puis la mesure de

1. Le site http ://www.charlatans.info/illusions&paradoxes.shtml donne d'autres exemples.

significativité proposée, et nous présentons l'adaptation faite à notre problème.

Principe de la détection *a contrario* Les approches *a contrario* ont été appliquées à différents problèmes de détection de structures dans les images [39], citons de façon non exhaustive : les alignements de points [36], les contours [37], les points de fuite [3], les segments de droites [127],la mise en correspondance de points et mesure de géométrie épipolaire [97], l'appariement de points suivant ces descripteurs locaux, entre les images [109], la reconnaissance de forme [23], la détection des changements (en images de télédétection [112], ou dans des scènes vidéo [42]).

L'approche *a contrario* exploite un modèle, appelé modèle naïf, qui représente l'absence de structures dans les données, au lieu de préciser le modèle *a priori* des structures, phénomènes ou objets recherchés. En effet, il est généralement beaucoup plus simple de modéliser le bruit (dans les données non-structurées) que d'énumérer les phénomènes ou objets recherchés et les modéliser sous leurs différentes formes. La détection est alors dite *a contrario* car un phénomène (par exemple un alignement, un changement, des points de fuite, etc.) sera détecté par contradiction du modèle naïf, au sens où la réalisation est très improbable sous l'hypothèse du modèle naïf (principe de Helmholtz : on détecte ce qui est 'trop structuré' pour se produire sous l'hypothèse du modèle naïf).

En résumé, la détection *a contrario* s'appuie sur un modèle naïf (à contredire), et une mesure de significativité des observations sous hypothèse du modèle naïf. Son atout majeur est donc qu'elle ne requiert donc que peu d'informations *a priori* par rapport à des détections à partir de modèles des objets recherchés. Précisons à présent la mesure de significativité proposée dans [39].

Mesure en terme de Nombre de Fausses Alarmes En 2000, Desolneux et al. [36] introduisent une mesure afin de quantifier la significativité d'un groupement. La mesure de significativité est directement reliée au nombre d'occurrences associé à un groupement. Ainsi, si ce nombre est trop important, i.e. si la significativité du groupement est trop faible, ce dernier n'a pas lieu d'être (selon l'observation qu' "une loi de groupement n'est active que si elle ne crée pas un trop grand nombre de détections"). Reprenons l'exemple des dates d'anniversaire de [41]. Le modèle naïf, déjà cité, est que les dates d'anniversaire sont i.i.d. uniformes sur l'année. Afin d'illustrer l'intérêt d'une mesure en nombre de fausses alarmes, raisonnons en probabilités, calculons la probabilité que n personnes aient la même date d'anniversaire. Pour $n = 3$, (le cas $n = 2$ ayant déjà été présenté) cette probabilité vaut 1 moins la probabilité que toutes les personnes aient une date d'anniversaire différente moins la probabilité que seuls des couples aient la même date d'anniversaire. Cette dernière probabilité est loin d'être triviale à calculer : il faut faire varier le nombre de couples de même date anniversaire de 1 à $N/2$, et pour un nombre de couples i, la probabilité (d'avoir i couples de même date anniversaire) est donnée par la probabilité que l'on ait à la fois N-i personnes nées à une date différente et i personnes nées à une date donnée (la même que celle de la personne du même couple), soit

$$\sum_{i=1}^{N/2} \frac{\prod_{k=1}^{i} \binom{N-2(k-1)}{2}}{i!} \prod_{k=1}^{i} \left(\frac{1}{365}\right) \prod_{k=1}^{N-i} \left(1 - \frac{k}{365}\right),$$

donc

$$P(X \geq 3) = P(X \geq 2) - \left(\frac{1}{365^N}\right) \sum_{i=1}^{N/2} \frac{\prod_{k=0}^{i-1} \binom{N-2k}{2}}{i!} \prod_{k=1}^{N-i} (365-k).$$

La probabilité que n personnes parmi les N aient la même date anniversaire se calcule selon le même principe, avec des calculs de moins en moins triviaux. Raisonnons maintenant en terme d'espérance du nombre d'occurrences. Cette espérance fournit une borne supérieure de la probabilité recherchée. Soit l'évènement "au moins n personnes parmi les N ont la même date d'anniversaire", son espérance est $\left(\frac{1}{365}\right)^{n-1}$ (la date du dit anniversaire est imposée à n-1 personnes) multipliée par le nombre de n-uplets parmi les N personnes, soit $\begin{pmatrix} N \\ n \end{pmatrix}$. Par exemple pour $n = 3$, on a $P(X \geq 3) \leq \frac{1}{365^2} \frac{N(N-1)(N-2)}{6}$, soit pour $N = 25, P(X \geq 3) = 0.0164 \lesssim 0.0173$, et pour $N = 50, P(X \geq 3) = 0.1264 \lesssim 0.1471$. Cet exemple illustre donc la simplification du calcul lorsque l'on raisonne en espérance du nombre d'occurrences, plutôt qu'en probabilités.

Dans le cas général, soient n objets, $O_1, O_2, ..., O_n$, présents dans une image dont k objets, $O_1, O_2, ..., O_k$, parmi eux ont un caractère en commun (e.g. couleur, position orientation, etc...). Alors on se trouve devant la question suivante : est-ce que ce caractère commun est obtenu par le hasard ou bien est-il significatif et suffisant pour grouper ces k objets [41]. La réponse est apportée par la mesure de l'espérance de cette occurrence sous l'hypothèse naïve. Selon cette dernière, les mesures ne seraient que du bruit (modèle de fond) et raisonnablement on peut supposer que les observations sont des variables aléatoires indépendantes identiquement distribuées [37].

La formalisation *a contrario* reformule alors le problème sous la question suivante : le groupement formé par les k objets parmi n contre dit-il le modèle naïf précédent au sens où sa présence est très improbable sous ce modèle naïf et donc il existe certainement une cause objective qui explique l'observation du groupe. Maintenant, si on décide de détecter un groupement et que le modèle naïf était vrai, la détection est une fausse alarme. Dans la suite, on appelle Nombre de Fausses Alarmes (NFA), l'estimation du nombre d'occurrences du groupement observé sous hypothèse du modèle naïf. D'après [37], la significativité peut être mesurée en terme de Nombre de Fausses Alarmes (NFA) : une réalisation est d'autant plus contradictoire du modèle naïf que le NFA associé est faible.

Définition 1. Soit E l'ensemble des évènements potentiellement détectables, et $\mathbb{P}$ la probabilité sur E d'apparition des évènements. A tout évènement $e \in E$ est associé son nombre de fausses alarmes, défini par

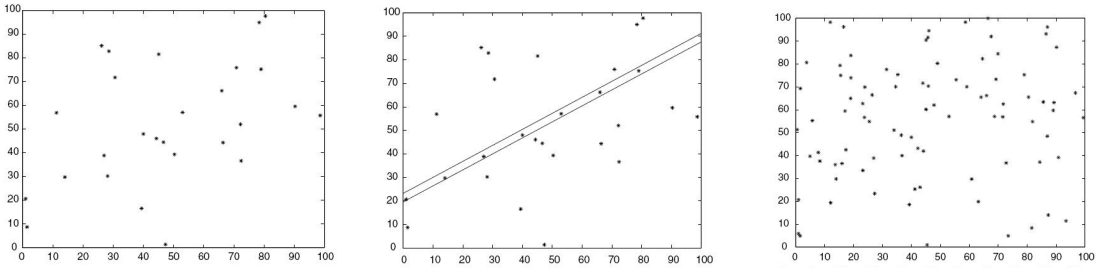
$$NFA(e) = |E| \mathbb{P}(e), \tag{2.10}$$

et pour tout réel $\varepsilon > 0$, e est dit $\varepsilon - significatif$ si est seulement si

$$NFA(e) \leq \varepsilon.$$

Cette définition ne peut être utilisée qu'après avoir défini le modèle naïf, i.e. un modèle permettant de donner un sens à la probabilité $\mathbb{P}(e)$. Elle garantit que l'espérance du nombre de fausses alarmes soit inférieure à ε . Dans [62], Grosjean et Moisan ont proposé une définition plus générale du nombre de fausses alarmes. Dans ce travail nous utiliserons la formulation classique du *NFA*.

Pour illustrer ces propos, nous reprenons l'exemple de la détection d'alignements proposé par [39]. Dans la figure 2.4 les alignements éventuels sont automatiquement détectés comme 'trop improbables' (principe de Helmholtz) par rapport au modèle naïf, qui est ici la distribution uniforme. Dans (b) l'alignement est significatif au vu de la densité des points sur l'ensemble de l'image, contrairement au cas présenté en (c), où pour qu'il soit détecté, un alignement devra contenir au moins 11 points [39]. En effet le calcul est le suivant : si le modèle naïf est la distribution i.i.d. uniforme des points, la probabilité d'observer au moins k points (ici groupés selon un alignement) parmi n est donnée par la



(a) (b) (c)

FIGURE 2.4: *Illustration du principe de Helmholtz pour la détection d'alignement ([38]).* (a) 20 points uniformément distribuées au hasard, et 7 alignés ajoutés, (b) l'alignement est détecté comme un grand écart, (c) même image que (a) en ajoutant 80 points aléatoires .

queue de la loi binomiale :

$$B(p,n,k) = \sum_{i=k}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) p^i \left(1-p\right)^{n-i},$$

où p est la probabilité d'apparition d'un point sur l'image selon le modèle naïf. Nous avons vu que l'on peut garantir en moyenne moins de ε fausses alarmes en ne détectant que les alignements de k points tels que $B(p,n,k) \times N_{tests} \leq \varepsilon$. Les alignements ont une certaine imprécision ne serait-ce que de part la discrétisation des points sur l'image. Soit R le diamètre du domaine image et a la largeur minimale d'une bande à l'intérieur de laquelle on teste la présence d'un alignement. Le nombre de tests est $N_{tests} = 2\pi \left(\frac{R}{a}\right)^2$ (i.e. $2\pi \left(\frac{R}{a}\right)$ directions à tester si l'on échantillonne la circonférence du cercle circonscrit à l'image par pas de $\left(\frac{R}{a}\right)$, et, pour une direction donnée, $\left(\frac{R}{a}\right)$ origines possibles pour les bandes perpendiculaires à cette direction). p est l'aire relative de la bande. A partir des valeurs numériques fournies dans [38], si l'on suppose en outre que $p = 0.03$ et $\left(\frac{R}{a}\right) = 30.$, on obtient des significativités $-log\left(B(p,n,k)N_{tests}\right)$ respectivement égales à 2.74 pour $k = 7, n = 27$, -4.50 pour $k = 7, n = 87$, et 0.66 pour $k = 11, n = 91$, i.e. 7 points alignés parmi 27 (ou 11 parmi 91) c'est significatif, mais pas 7 parmi 87. Il est intéressant de noter également significativité d'une bande (en terme de récipient d'un alignement) dépend de sa 'résolution' via le nombre de tests. Ainsi, si l'on considère des bandes deux fois plus larges, on a maintenant $p = 0.06$, et $\left(\frac{R}{a}\right) = 15$ dans notre exemple numérique, soit une significativité égale à -0.19 pour $k = 7, n = 27$, i.e. la bande n'est plus considérée comme significative. Ainsi l'intérêt de la mesure proposée est bien de permettre de comparer les regroupements de points, de manière à détecter les plus significatifs notamment en cas de recouvrement partiel entre deux regroupements (comme dans le cas des deux bandes supposées inclure les mêmes points mais de largeurs différentes (la bande la plus large est plus significative que celle étroite à partir de $k = 9$).

En résumé, la détection *a contrario* présente une certaine généricité dans la mesure où elle utilise très peu d'information *a priori*. L'*a priori* est en effet principalement le modèle naïf, qui de part sa définition même est relativement générique. Ainsi, dans le cas de points, il s'agit généralement de la loi binomiale, et dans le cas de valeurs numériques (e.g. niveaux de gris), il s'agit généralement de la loi gaussienne. L'approche *a contrario* permet alors de rassembler tous les paramètres de la détection (seuils, etc.) sous un seul critère : l'espérance du nombre de fausses alarmes. Ce dernier point est essentiel puisque, en pratique, il rend la détection entièrement automatique en fixant le nombre attendu de fausses alarmes à 1 par exemple.

Dans le paragraphe suivant, nous résumons deux utilisations précédentes de l'approche *a contrario*

pour la détection de changement.

Approches *a contrario* en détection de changement sur images à niveau de gris

Dans ce paragraphe, nous spécifions deux approches utilisant la détection *a contrario* pour détecter des changements par rapport à un état antérieur (cartographie pour le premier cas et image de fond pour le second).

Les changements sont par définition difficilement énumérables (en terme de forme). Aussi l'approche *a contrario* qui modélise le bruit préférentiellement aux objets d'intérêt parait-elle particulièrement pertinente. Maintenant, nous avons vu que, dans [40], Desolneux et al. proposent des mesures de significativité de certains groupements d'objets identifiés comme ayant une caractéristiques communes (alignements, *clusters*, etc.). Dans le cas des images, ces objets sont des points, ainsi les mesures précédentes s'appliquent sur des images de points. Afin de les utiliser pour la détection, il s'agit donc de définir une labelisation [2] préalable des pixels de l'image, en termes de pixels de fond (ou susceptibles de l'être), de pixels d'objet (ou susceptibles de l'être), et éventuellement pixels de label inconnu.

Une première solution est celle de Dibos et al. qui consiste simplement à seuiller l'image à niveau de gris pour obtenir une image sur laquelle la détection de points peut être effectuée [42]. Plus précisément, leur cadre applicatif est celui de la détection de changement sur des images à niveaux de gris acquises par une caméra fixe (typiquement un système de vidéosurveillance). Les auteurs comparent ensuite les valeurs des gradients (norme et direction) sur l'image de fond et sur l'image courante pour décider d'un label au niveau pixel parmi les labels suivants : $\{$*changement*, *fond*, *inconnu*$\}$. Un critère de NFA est finalement appliqué pour détecter les régions de changement à partir de l'image 3-labels. Les régions de changement sont des pavés 2D rectangulaires (fenêtres) sur lesquels la significativité du nombre de points observés est évaluée par rapport à la loi binomiale (modèle naïf considéré). Un intérêt de cette approche est la prise en compte d'un label *inconnu* dans l'image de points. Cependant la détection proposée n'est pas automatique en raison des paramètres de seuil pour la création de l'image 3-labels.

Dans leurs travaux, Robin et al. [112] proposent une mesure de significativité à partir de la valeur des niveaux de gris des pixels. Les images traitées étant des images (satellitaires) basse résolution (pixels allant jusqu'à 1×1 km2) et les changements recherchés non nécessairement de taille supérieure à celle des pixels, les pixels étaient détectés indépendamment les uns des autres sans notion de groupement. La mesure proposée identifie le sous-domaine de l'image (sous-ensemble de pixels non nécessairement connexes) dont les valeurs de niveau de gris sont les plus étonnantes par rapport au modèle naïf supposé *a priori* (bruit blanc gaussien). Plus précisément, soit une image contenant c classes, de proportions connues (en l'absence de changement) à l'intérieur des pixels basse résolution, mais de moyennes $(\mu_i, i \in [1, c])$ inconnues (dans le contexte applicatif de la télédétection, un changement est généralement un changement dans la cartographie de la surface, i.e. la classification, et non un simple changement de valeurs radiométriques car ces dernières évoluent temporellement en fonction des conditions de la surface et de l'atmosphère). Pour estimer ces moyennes inconnues, les auteurs proposent de maximiser l'étonnement lié à une erreur quadratique très faible entre les valeurs des pixels observées et prédites (selon le modèle linéaire, égale à la somme des μ_i pondérées par les pourcentages d'occupation de chaque classe dans le pixel basse résolution considéré). L'erreur sera d'autant plus faible que les moyennes $\mu_i, i \in [1, c]$ seront correctes et que les proportions des classes intrapixels seront également correctes, i.e. que seuls les pixels sans changement seront pris en compte. Un critère de NFA est utilisé pour mesurer l'étonnement, et le sous-domaine détecté est celui des pixels

2. action d'attribuer un label ou une étiquette de classe ; nous préférons ce terme à celui d'étiquetage usuel en segmentation (les étiquettes étant alors des étiquettes de régions et non de classes)

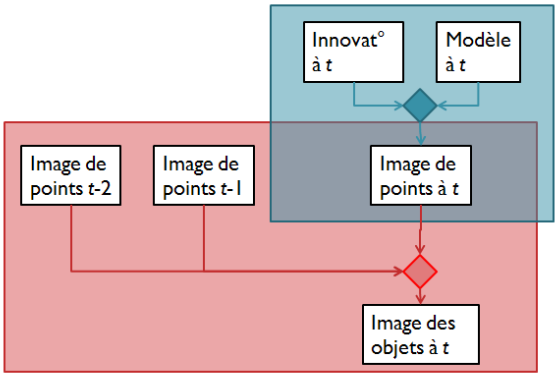


FIGURE 2.5: *Processus de décision et approche proposée à deux niveaux.*
La détection des points (pixels de label *objet*) est effectuée au niveau 1 (bleu) et celle des structures de points au niveau 2 (rose).

de fond, le sous-domaine des pixels de changement étant le complémentaire par rapport au domaine image.

La section 2.3.2 présente la façon dont ces travaux ont été en partie réexploités dans la méthodologie développée.

2.3.2 Méthode développée

Reprenons tout d'abord notre problème afin de comprendre en quoi les approches décrites au paragraphe précédent sont insuffisantes. Trois caractéristiques rendent notre problème complexe :

1. les données en entrée sont des images à niveaux de gris avec éventuellement des valeurs absentes,
2. les conditions d'acquisition sont très variables (scènes en extérieur),
3. la caméra peut être en mouvement (embarquée sur le véhicule).

Des points communs cependant demeurent par rapport au paragraphe précédent :

1. les changements ou les objets à détecter sont difficiles à modéliser (en raison de leur variabilité), les zones de non changement pourraient être plus facilement déduites à partir d'une image de fond et d'un modèle de bruit,
2. les objets sont grands devant les pixels de l'image et le bruit est supposé spatialement décorrélé, ainsi les approches qui considèrent les pixels non indépendamment (e.g. dans un voisinage) devraient être plus robustes.

Nous supposons que les objets sont grands devant les pixels de l'image, c'est à dire qu'ils comprennent plusieurs pixels. Ainsi, en l'absence de toute autre hypothèse sur la forme des objets recherchés, ils seront détectés comme des groupements de pixels répondant à une fonction spécifique au niveau du pixel, à savoir une fonction de détection niveau pixel. La détection niveau pixel repose sur la comparaison de l'image observée à l'instant t, appelée *innovation à l'instant* t dans la figure 2.5, et d'un *modèle à* t. Par exemple, dans la détection de changement par rapport à une image de fond, la fonction de détection est (souvent) la norme L2 de la différence entre la valeur du niveau de gris d'un pixel et la valeur de ce même pixel dans l'image du fond.

Nous avons à définir d'une part une fonction de détection niveau pixel, et d'autre part une fonction de détection niveau groupement.

2.3.2.1 Niveau 1 : Détection des points sur l'image

La première idée est de s'inspirer de la détection niveau pixel de [112]. Les deux différences principales sont : les pixels ici sont des pixels haute résolution, la détection n'est pas fondée sur la comparaison avec une classification mais avec une image modèle (e.g. image fond ou image dérivée d'un modèle de scène 3D).

Pour chaque image I_t de la séquence vidéo, nous considérons l'image de différence

$$\forall s \in \mathcal{D},\ \Delta(s) = I_t(s) - \tilde{B}_t(s), \tag{2.11}$$

où $\tilde{B}_t$ est l'image modèle (nous détaillons ultérieurement les techniques de définition de cette dernière), et s est un pixel du domaine $\mathcal{D}$ des pixels définis (nous verrons que certains pixels peuvent avoir une valeur indéfinie par exemple dans le cas d'utilisation de cartes de disparité).

Classiquement un seuillage est pratiqué sur les valeurs de $|\Delta(s)|$ ou $\Delta^2(s)$ pour déterminer les pixels de valeur étonnamment élevée. Or le choix de ce seuil est généralement très délicat et non robuste. Aussi propose-t-on d'utiliser un critère NFA pour déterminer ce seuil automatiquement.

Critère NFA1

La significativité des pixels de l'image de différence Δ est mesurée en choisissant le modèle naïf, H_1 de la définition 2.

Définition 2. Le modèle naïf, H_1, pour l'image Δ est un champ aléatoire de $|\mathcal{D}|$ variables gaussiennes indépendantes et centrées $\mathcal{N}(0, \sigma^2)$ avec σ l'écart type correspondant une image totalement non structurée, et $|\mathcal{D}|$ le cardinal du domaine $\mathcal{D}$ des pixels définis.

Soit $\Delta^2(s) = [\Delta(s)]^2$ le carré de l'image de différence. L'erreur résiduelle entre l'image observée et $\tilde{B}_t$ sur un sous-domaine $D \subseteq \mathcal{D}$ est mesurée par ϵ_D^2, somme des pixels de l'image de différence Δ^2 appartenant au sous domaine D. La valeur de ϵ_D^2 est difficilement interprétable de manière absolue. Ainsi, suivant la modélisation *a contrario*, nous considérons la probabilité d'observer l'erreur résiduelle ϵ_D^2 par hasard, notée $\mathbb{P}_{H_1}(\epsilon_D^2)$. De là, nous définissons la mesure sur un sous-domaine D par :

$$\begin{aligned} NFA_1(D) &= \eta_1(D)\, \mathbb{P}_{H_1}(\epsilon_D^2, \sigma) \\ &= \eta_1(D)\, \frac{1}{\Gamma(\frac{|D|}{2})} \int_0^{\epsilon_D^2/2\sigma^2} e^{-t} \cdot t^{\frac{|D|}{2}-1} dt, \end{aligned} \tag{2.12}$$

où Γ est la fonction d'Euler, et $\eta_1(D)$ est un terme de normalisation (équivalent du nombre de tests) qui permet de contrôler le nombre moyen des fausses alarmes. Dans [112], Robin et al. ont montré que selon la définition 2, il suffit que :

$$\sum_D \frac{1}{\eta_1(D)} \leq 1. \tag{2.13}$$

Dans notre cas, nous avons choisi $\eta_1 = |\mathcal{D}| \begin{pmatrix} |\mathcal{D}| \\ |D| \end{pmatrix}$, pour distribuer le risque par rapport à la taille du domaine. Ce terme vérifie bien l'inégalité 2.13 :

$$\sum_D \frac{1}{|\mathcal{D}| \begin{pmatrix} |\mathcal{D}| \\ |D| \end{pmatrix}} = \sum_{|D|} \begin{pmatrix} |\mathcal{D}| \\ |D| \end{pmatrix} \frac{1}{|\mathcal{D}| \begin{pmatrix} |\mathcal{D}| \\ |D| \end{pmatrix}} = \sum_{|D|} \frac{1}{|\mathcal{D}|} = \frac{|\mathcal{D}|}{|\mathcal{D}|} = 1.$$

L'équation (2.14) fait apparaitre les dépendances aux paramètres, dans l'équation (2.12) :

$$NFA_1\left(\epsilon_D^2, |D|, \sigma, |\mathcal{D}|\right) = \eta_1\left(|D|, |\mathcal{D}|\right) \frac{1}{\Gamma(\frac{|D|}{2})} \int_0^{\epsilon_D^2/2\sigma^2} e^{-t} t^{\frac{|D|}{2}-1} dt. \tag{2.14}$$

À la fin de ce premier test de significativité, nous avons détecté le sous-domaine $\hat{D}$ des pixels ayant les valeurs les plus significativement faibles dans l'image Δ^2. En combinant le résultat $\hat{D}$ avec $\mathcal{D}$, nous obtenons une image à 3 labels indiquant les pixels d'objet, i.e. ceux qui appartiennent à $\mathcal{D}\backslash\hat{D}$, les pixels de fond, i.e. ceux qui appartiennent à $\hat{D}$, et les pixels *inconnus*, i.e. ceux qui n'appartiennent pas à $\mathcal{D}$.

La détection est totalement automatique. En effet les seuls paramètres qui interviennent dans l'équation (2.12) sont le cardinal des pixels définis $|\mathcal{D}|$, directement accessible, et l'écart-type σ du modèle naïf. Ce dernier représente le cas d'une image Δ totalement non structurée, i.e. par exemple une différence entre deux images quelconques (ne représentant pas la même scène) ou simplement une différence entre une image de la scène et une image noire (tous les pixels à 0). Donc, σ peut être estimé par l'écart-type de I_t. L'algorithme est alors l'algorithme 2.1.

Algorithm 2.1 Algorithme de détection au niveau pixel.

- Calculer le paramètre σ et l'image Δ^2 ;
- Trier les pixels de Δ^2, en un vecteur $\left(\delta_i^2\right)_{1\leq i\leq|\mathcal{D}|}$ par ordre croissant des valeurs ;
- Initialiser NFA_1^{min} à $+\infty$, δ^2 à 0, et $\hat{D}$ à $\varnothing$;
- Pour chaque indice $i \in [1, |\mathcal{D}|]$,
  - Ajouter δ_i^2 à δ^2 et le pixel d'indice i dans le vecteur ordonné à D ;
  - Calculer $NFA_1\left(\epsilon_D^2, |D|, \sigma, |\mathcal{D}|\right)$ (équation (2.14)) ;
  - Si $NFA_1\left(\epsilon_D^2, |D|, \sigma, |\mathcal{D}|\right) < NFA_1^{min}$,
    - $NFA_1^{min} = NFA_1\left(\epsilon_D^2, |D|, \sigma, |\mathcal{D}|\right)$;
    - $\hat{D} = D$;
  - Fin test ;
- Fin boucle.

La figure 2.6 donne un exemple de détection pixelique à partir des images de niveaux de gris *background* et *foreground* dans le cas d'une caméra statique. La figure 2.6(a) est une composition colorée des images *background* et *foreground*, de sorte que les régions sans changement apparaissent en jaune, et les régions présentant des changements ont une couleur dominante dans le rouge ou dans le vert selon la variation radiométrique des pixels induite par le changement au pixel considéré. On remarque principalement les trois personnages ainsi que de très ponctuelles (spatialement) variations dans le fond de la scène ou des branches de cyprès. L'image de différence en valeur absolue $|\Delta|$ (figure 2.6(b)) confirme les observations précédentes et fait également apparaitre des variations de luminosité au niveau de certaines voitures parquées : il s'agit probablement de variations dues à des reflets sur la carrosserie des voitures. La figure 2.6(c) représente la courbe donnant la valeur du NFA_1 en fonction du cardinal du domaine D (pixels étonnamment consistants). On observe un minimum bien prononcé pour la courbe du NFA_1, tandis que la courbe des erreurs quadratiques cumulées parait beaucoup plus difficile à interpréter automatiquement. Enfin, la figure 2.6(d) montre le résultat de la détection des domaines $\hat{D}$ pixels labellisés *fond* codés en blanc, $\mathcal{D}\backslash\hat{D}$ pixels labellisés *objet* codés en noir, et $\overline{\mathcal{D}}$ pixels labellisés *inconnu* codés en gris (absents dans cet exemple). Nous notons que les personnages sont assez bien détectés mais également un nombre élevé de détections pixeliques, d'où l'intérêt de prendre en compte un deuxième critère de densité des points.

Cette première détection niveau pixel appelle plusieurs commentaires.

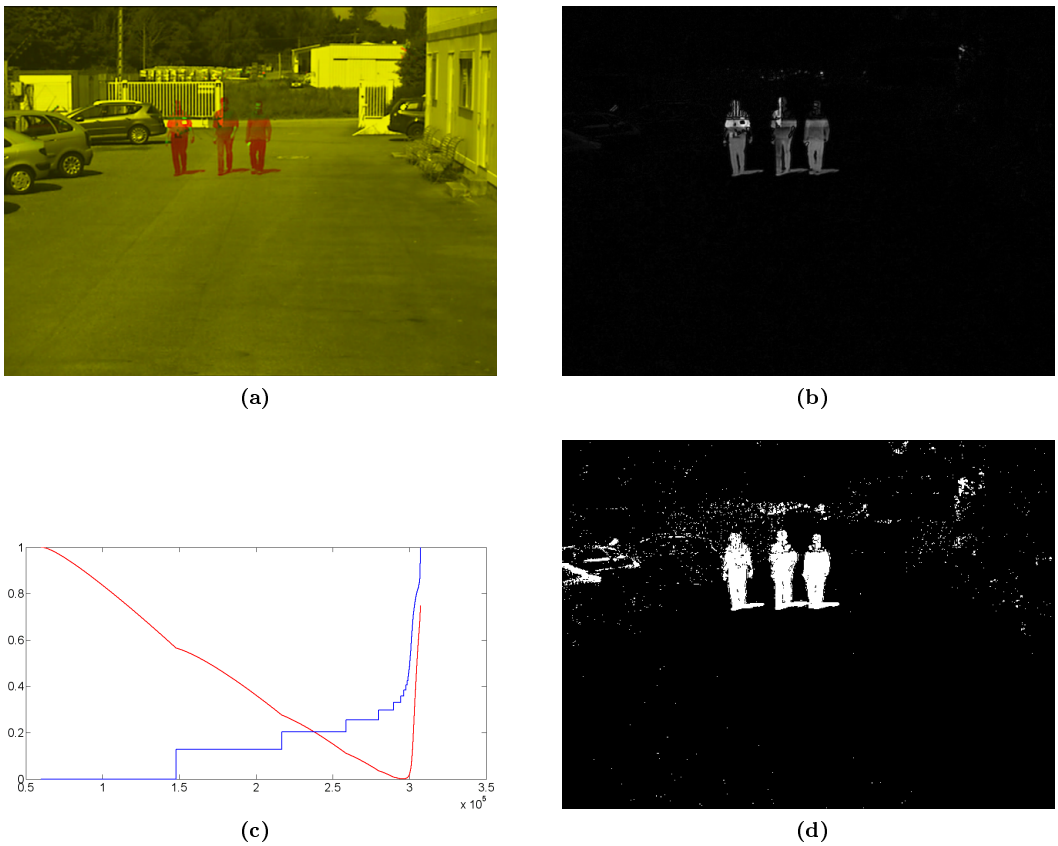


FIGURE 2.6: *Exemple de détection pixelique à base de niveau de gris.*
(a) composition colorée contenant l'image courante (*foreground*) en rouge et l'image de fond (*background*) en vert (la couleur jaune indique donc les régions sans changement, (b) image de différence en valeur absolue $|\Delta|$, (c) courbe donnant la valeur du $log\,(NFA_1)$ en rouge et de la somme des erreurs quadratiques en bleu (i.e. δ^2 dans l'algorithme 2.1) en fonction du cardinal du domaine D (pixels étonnamment consistants), et (d) image des points 3 labels (Noir≡*objet*, Blanc≡*fond* et Gris≡*inconnu*, absents dans cet exemple).

En premier lieu, nous pouvons nous demander s'il n'aurait pas été plus simple de détecter directement les pixels de valeur significativement élevée. Dans ce cas le paramètre σ du modèle naïf aurait été choisi différemment, typiquement l'écart-type effectivement observé sur Δ. Cependant, deux raisons jouent en défaveur de cette modélisation.

1. Premièrement, le fait d'utiliser ϵ_D^2, somme sur les pixels de D, permet de robustifier la détection par rapport à une décision faisant intervenir la valeur du seuil sur les pixels. Or si l'on cherche à s'étonner des valeurs de ϵ_D^2 particulièrement fortes, on ne garantit pas la valeur minimale de $\Delta^2(s)$, $s \in D$ (il suffit d'une valeur très forte pour que ϵ_D^2 reste élevée même en lui ajoutant des valeurs relativement faibles), comme on pouvait garantir la valeur maximale de $\Delta^2(s) \leq \epsilon_D^2, \forall s \in D$ lorsqu'on s'étonnait de valeurs de ϵ_D^2 particulièrement faibles.
2. Deuxièmement, si l'on a des paramètres inconnus, on va naturellement les choisir selon un critère d'erreur minimale. Dans [112], les paramètres à optimiser étaient les moyennes des classes. Dans notre cas, les paramètres à optimiser (éventuellement) seront ceux du mouvement 6D de la caméra : ils seront estimés (cf. un peu plus loin dans cette section) en maximisant l'étonnement d'avoir une valeur d'erreur particulièrement faible.

En second lieu, nous pourrions envisager de faire varier le seuil, d'avoir pour chaque valeur de seuil, une image de points (binaire ou ternaire), de calculer un critère de significativité (à partir d'un NFA, cf. section 2.3.2.2) sur chacune de ces images de points, et de choisir la plus significative, i.e. celle qui correspond au groupement de points les plus significatifs. Nous expliciterons cette variante de création de l'image de points lorsque nous aurons présenté la détection des groupements de points. Cependant, nous pouvons annoncer déjà qu'un obstacle important est le temps de calcul induit. En effet, en l'absence de toute information sur le seuil optimal (du point de vue de la maximisation de la significativité des groupements de points), pour des images codées sur huit bits (niveaux de gris classique), il y a 255 valeurs à tester.

Estimation de l'égo-mouvement précis et apport de la correction du mouvement

Nous donnons ici une seconde illustration du critère NFA_1. Dans la figure 2.6, la caméra était statique. Or, dans notre cas applicatif, la caméra est en mouvement et le mouvement 6D de la caméra est un paramètre de l'estimation de l'image modèle (cf. section 2.2.2.3). Or, les causes d'imprécision sur la mesure de ce mouvement sont fréquentes : outre les limites de la précision intrinsèque des capteurs de mouvement (odomètres), la synchronisation des données (caméra et odomètres) peut ne pas être parfaite. Malgré une même fréquence d'acquisition pour les différents capteurs, il peut exister une différence au niveau des instants d'acquisition de l'ordre de quelques dizaines de millisecondes. Or, la projection de l'image à l'instant $t-1$ vers l'instant t (étape nécessaire dans la plupart des cas applicatifs de la détection de changement, cf. section 2.3.3) est très sensible à ces paramètres de mouvement. Ici, nous proposons d'utiliser le critère NFA_1 pour estimer simultanément les pixels *objet* (contradictoires à $\tilde{B}_t$) et les paramètres du mouvement précis.

Une initialisation des paramètres T et Ω est donnée en entrée car il n'est pas envisageable de cherche un mouvement 6D dans tous l'espace des mouvements. L'algorithme 2.2 décrit l'optimisation de ces paramètres conjointement à l'estimation du sous-domaine des pixels d'objets (i.e. sous-domaine des points) par minimisation du NFA_1.

Pour illustrer l'estimation conjointe des pixels d'objet et du mouvement précis de la caméra, nous devons considérer un cas de caméra embarquée (sur un véhicule mobile). La figure 2.7 présente un exemple avec des données simulées par le simulateur Pro-SiVIC. La figure 2.7(a) montre l'image simulée à l'instant t et la figure 2.7(b) montre l'image simulée à l'instant $t-1$ projetée à l'instant

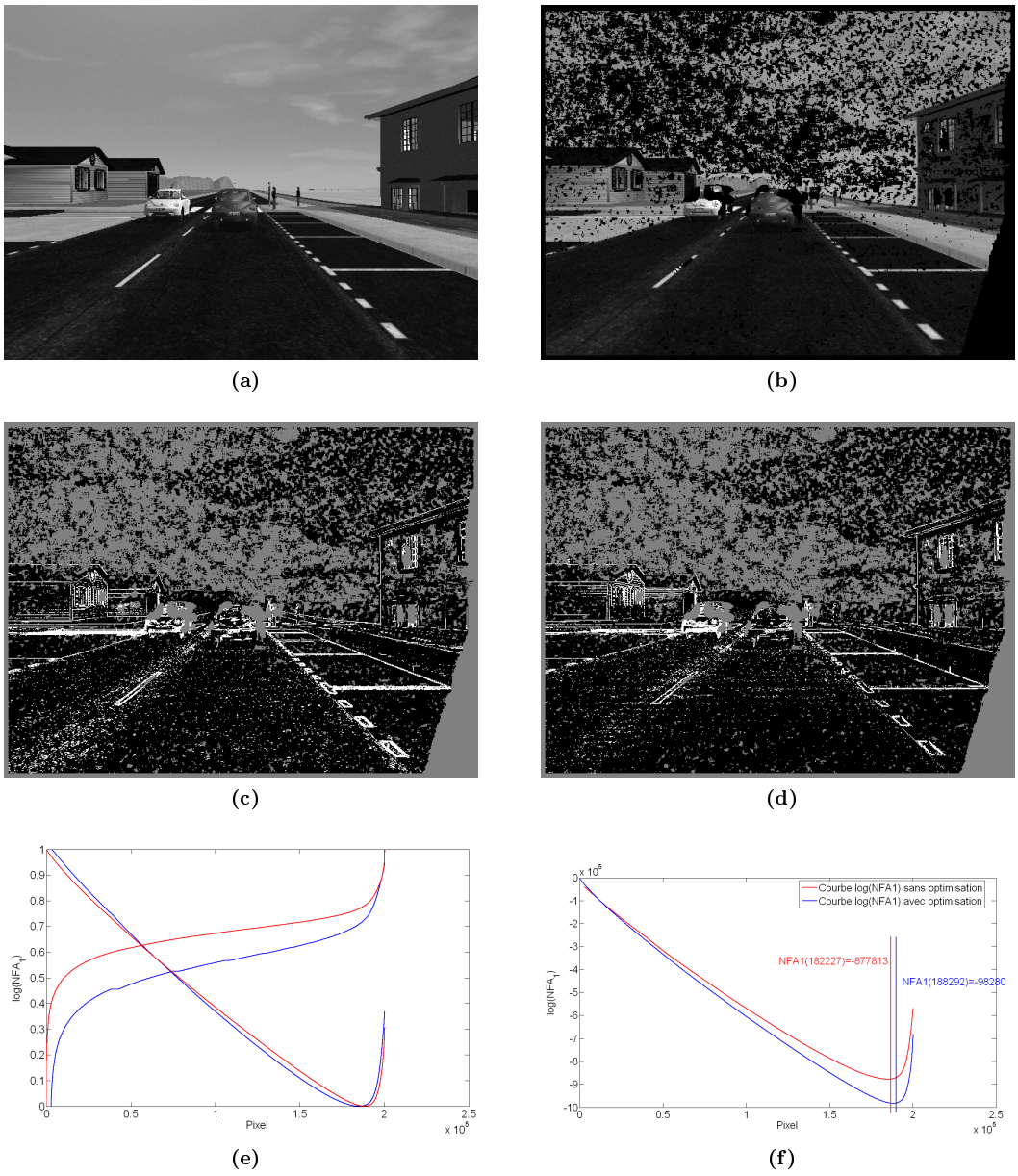


FIGURE 2.7: *Exemple de détection pixelique selon le critère NFA_1 (pour images à niveau de gris).* (a) image à l'instant t créée par le simulateur Pro-SiVIC, (b) image $t-1$ projetée à l'instant t, (c) image des points 3 labels (Noir≡*objet*, Blanc≡*fond* et Gris≡*inconnu*) obtenue sans optimisation des paramètres T et Ω, i.e. avec l'algorithme 2.1, (d) image des points 3 labels obtenue avec optimisation des paramètres T et Ω, i.e. avec l'algorithme 2.2, (e) courbes normalisées donnant la valeur du $\log(NFA_1)$ et de la somme des erreurs quadratiques (δ^2 dans l'algorithme 2.1) en fonction du cardinal du domaine D (pixels étonnamment consistants) respectivement sans et avec optimisation des paramètres T et Ω, (f) courbes non normalisées donnant la valeur du $\log(NFA_1)$ en fonction du cardinal du domaine D respectivement sans et avec optimisation des paramètres T et Ω.

Algorithm 2.2 Algorithme d'estimation conjointe des pixels *objets* et du mouvement précis du véhicule.

- Initialiser $NFA^{min}_{1,(\tilde{T},\Omega)}$ à $+\infty$;
- Initialiser $\left(\hat{T},\hat{\Omega}\right)$ aux valeurs par défaut, et $\hat{D}_{(\tilde{T},\Omega)}$ à $\varnothing$;
- Varier (T,Ω) dans le domaine choisi,
  - Calculer l'image $I_{t-1\to t}(T,\Omega)$;
  - Appliquer l'algorithme 2.1 pour obtenir $\hat{D}(T,\Omega)=\hat{D}$ et $NFA^{min}_1(T,\Omega)=NFA^{min}_1$;
  - Si $NFA^{min}_1(T,\Omega) < NFA^{min}_{1,(\tilde{T},\Omega)}$,
    - $NFA^{min}_{1,(\tilde{T},\Omega)} = NFA^{min}_1(T,\Omega)$;
    - $\hat{D}_{(\tilde{T},\Omega)} = \hat{D}(T,\Omega)$;
    - $\left(\hat{T},\hat{\Omega}\right) = (T,\Omega)$;
  - Fin test ;
- Fin boucle ;
- Assigner le sous-domaine des pixels labelisés *objet* à $\mathcal{D} \setminus \hat{D}_{(\tilde{T},\Omega)}$.

t. On note le déplacement de la caméra visible sur les positions des marquages au sol dans les deux images respectives. La figure 2.7(c) montre le résultat de l'algorithme 2.1, i.e. sans optimisation du mouvement 6D de la caméra (T,Ω), et la figure 2.7(d) montre le résultat de l'algorithme 2.2, i.e. avec optimisation du mouvement (T,Ω). On remarque que sans optimisation il existe de très nombreuses fausses alarmes au niveau des bords des marquages au sol notamment, et qu'avec une optimisation, le nombre de ces fausses alarmes diminue notablement. Enfin les courbes NFA_1 versus le cardinal du domaine D (pixels étonnamment consistants) respectivement sans et avec optimisation du mouvement (T,Ω) sont présentées sur les figures 2.7(e-f), soit en normalisant les valeurs à 1 (comme pour la figure 2.6(c)), soit en ne les normalisant pas, de manière à comparer les valeurs de NFA_1 avec ou sans optimisation. On observe alors que la courbe NFA_1 versus $|D|$ issue de l'algorithme 2.2 est en dessous de celle issue de l'algorithme 2.1 de manière significative.

2.3.2.2 Niveau 2 : Détection des groupements de points

Le paragraphe précédent avait pour objet l'obtention d'une image de points (image à deux ou trois labels, si l'on introduit le label *inconnu* dans le cas de données manquantes). Sous l'hypothèse que les objets recherchés sont grands devant les pixels, la détection va porter sur des amas de points de densité étonnante. Dans [39], notamment (mais pas exclusivement) sont proposés deux critères NFA pour détecter de tels amas de points.

Le premier type de NFA a déjà été cité en illustration du principe de Helmholtz à la figure 2.4, dans le cas de la recherche d'alignements. Il est immédiatement généralisable comme suit : soit une figure F_j géométrique paramétrique de paramètres $\vec{\pi}_j$, le NFA mesure le nombre de Fausses Alarmes associé à la détection des objets contenant des nombres de points supérieurs ou égaux au nombre observé dans la figure F_j . La figure 2.4 illustrait le cas d'un objet géométrique 'bande', avec comme paramètres la largeur, l'orientation et l'abscisse à l'origine. Dans notre cas, les objets géométriques recherchés sont plutôt de type pavé rectangulaire, avec comme paramètres la largeur, la longueur et les coordonnées du coin supérieur gauche (par exemple). La mesure NFA peut être utilisée soit pour détecter toutes les figures F_j telles que le nombre de fausses alarmes associés soit inférieur à un seuil donné, soit pour détecter la(es) figure(s) F_j la(es) plus significative(s), i.e. associée(s) à la (aux) valeur(s) de NFA minimale(s).

Avant de détailler plus avant ce critère NFA, citons un deuxième type de NFA qui permet la détection d'amas de points. Contrairement au précédent, ce dernier ne fait pas d'hypothèse sur la forme

géométrique des objets recherchés, mais sur la distance entre les points à l'intérieur des objets. Les *clusters* C_j (on utilise ici la terminologie anglaise plus légère que le mot regroupement) évalués sont définis à partir d'un paramètre de distance d_j, tel que la distance entre tout point p_i du *cluster* et l'ensemble des autres points du *cluster* $C_j \setminus \{p_i\}$ (minimum des distances entre p_i et p_l) soit inférieure à d_j. Bien qu'il paraisse intéressant dans notre cas de ne pas préjuger de la forme des objets recherchés, deux éléments nous ont pourtant conduits à choisir le NFA associé à des figures géométriques. L'algorithme proposé dans [39] passe par la construction d'un arbre couvrant de l'ensemble des points, tel qu'un nœud corresponde à une valeur de d_j et que le sous-arbre associé contienne tous les points reliés directement ou indirectement par une distance inférieure à d_j. Alors d'une part, pour les images contenant un grand nombre de points, le temps de calcul de cet arbre peut être relativement long, et d'autre part, dans le cas de données manquantes (label *inconnu*), les résultats ne sont pas stables. Par exemple, considérons deux *clusters* significatifs de paramètres d_j et $d_j - 1$, séparés par une distance $2d_j - 1$ et un pixel de label *inconnu* situé à mi-distance des *clusters*. Selon que l'on suppose que le pixel de label *inconnu* est un point (pixel de label *objet*) ou non, les deux *clusters* vont fusionner en un seul ou non. Ainsi, les résultats en terme de *clusters* détectés peuvent varier fortement en fonction d'un seul pixel. Or les labels ont été évalués au niveau d'une détection pixel et sont *a priori* sujets à des erreurs.

Nous choisissons donc l'approche fondée sur une hypothèse de figure géométrique qui est moins sensible à la présence de pixels de label *inconnu* et leur localisation précise. Nous choisissons comme figure géométrique, le pavé rectangulaire qui permet d'approcher en basse résolution la forme des objets. Classiquement un seuillage est pratiqué sur les valeurs des nombres de points dans une fenêtre pour déterminer les fenêtres de densité de points étonnamment élevée. Or le choix de ce seuil est généralement très délicat et non robuste. Aussi propose-t-on d'utiliser un critère NFA pour déterminer ce seuil automatiquement.

Critère NFA2

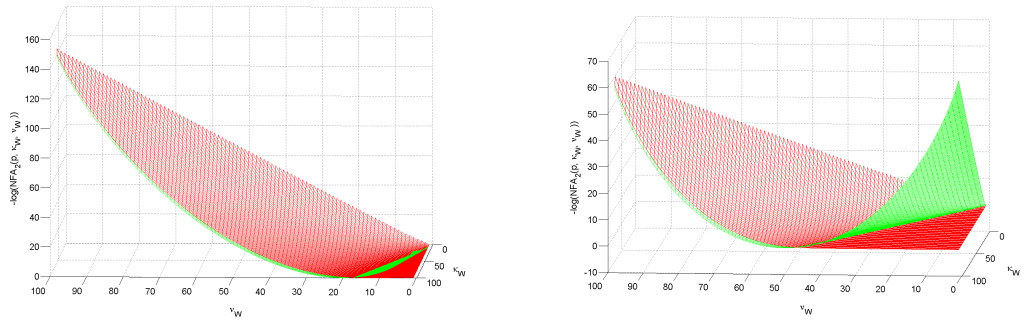
La significativité de l'image de points (à trois labels, *objet*, *fond*, et *inconnu*) est mesurée en choisissant le modèle naïf, H_2 de la définition 3.

Définition 3. Le modèle naïf, H_2, pour l'image à 3 labels (*objet*, *fond*, et *inconnu*), est la loi binomiale de paramètre p.

Examinons d'abord la façon dont Dibos et al. [42] calculent le Nombre de Fausses Alarmes à partir du modèle naïf précédent. Soit p la probabilité des pixels *objet* parmi les pixels de la label connu ($p = \frac{|\mathcal{D}\setminus\hat{D}|}{|\mathcal{D}|}$, où $\mathcal{D}$ est le domaine des pixels définis, i.e. de label connu, et $\hat{D}$ est le domaine des pixels de label *objet*), κ_W est le nombre de pixels *objet* dans le pavé rectangulaire (fenêtre) considéré W ($\kappa_W = \left|W \cap \mathcal{D}\setminus\hat{D}\right|$), ν_W est le nombre total de pixels définis (pixels de label *objet* ou *fond*) dans W ($\nu_W = |W \cap \mathcal{D}|$). La valeur de κ_W est difficilement interprétable de manière absolue. Ainsi, suivant la modélisation *a contrario*, nous considérons la probabilité d'observer l'amas de κ_W points par hasard, notée $\mathbb{P}_{H_2}(\kappa_W)$. De là, nous définissons la mesure sur une fenêtre W par :

$$\begin{aligned} NFA_2(W) &= \eta_2(W)\,\mathbb{P}_{H_2}(p, \kappa_W, \nu_W), \\ &= \eta_2(W) \sum_{i=\kappa_W}^{\nu_W} \binom{\nu_W}{i} p^i (1-p)^{\nu_W - i}, \end{aligned} \tag{2.15}$$

où η_2 est un nombre de tests (terme de normalisation). Dans [42], ce nombre est une constante, et dans [39] ce nombre est le nombre d'objets nécessaires à couvrir l'image entière, c'est-à-dire, transposé



(a) $p = 0.2$ **(b)** $p = 0.5$

FIGURE 2.8: *Approximation de Hoeffding de la queue binomiale.*
La courbe en vert est l'approximation de Hoeffding, la courbe en rouge est celle de $-\log(NFA)$.

à notre cas : $\eta_2(W) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|W|}$. Nous verrons que ce choix a tendance à surestimer les détections, nous proposons donc en alternative le nombre de tests suivant : $\eta_2(W) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|W|} \times 2^{|W|}$. Ces deux nombres de tests vérifient la condition $\sum_j \frac{1}{\eta_2(W_j)} \leq 1$.

Dibos et al. proposent [42] d'utiliser une approximation de l'équation précédente, à savoir l'approximation de Hoeffding, qui calcule $-\log(NFA_2(W))$ en fonction de p et $p_l = \frac{\kappa_W}{\nu_W}$, la probabilité locale à W pour un pixel observable d'être de label *objet*, (κ et ν sont les paramètres de l'équation (2.15)) :

$$\begin{aligned} S(p, p_l, \nu_W, |W|) &= -\log(NFA_2(W)) \\ &\approx \nu_W \left[p_l \log\left(\frac{p_l}{p}\right) + (1 - p_l) \log\left(\frac{1 - p_l}{1 - p}\right) \right] - \log(\eta_2(W)). \end{aligned} \tag{2.16}$$

La figure 2.8 permet de comparer la valeur exacte de $-\log(NFA_2(p, \kappa_W, \nu_W))$ (cf. équation (2.15)) et son approximation ($S(p, p_l, \nu_W, |W|)$, cf. équation (2.16)) dans le cas où $p = 0.2$ et celui où $p = 0.5$. Les deux fonctions ne sont pas définies pour $\kappa_W > \nu_W$. On remarque que des différences apparaissent pour les valeurs de ν_W élevées et de κ_W faibles, et ce d'autant plus que p est grand. Précisément, les valeurs divergentes correspondent à des cas où $p_l < p$, c'est à dire à des cas de fenêtres non significatives. On pose donc $-\log(NFA_2(p, p_l, \nu_W, |W|)) = -\log(\eta_2)$, si $p_l \leq p$. Remarquons ici que le fait qu'on ne considère que les cas où $p_l \geq p$, signifie que la taille maximale de la fenêtre pour détecter un groupement de κ_W points est κ_W/p, avec p probabilité d'un point dans l'image, ou autrement dit qu'avec une taille de fenêtre égale à $|W|$, on ne peut détecter que des groupements de taille supérieure à $p \times |W|$ points.

On s'intéresse à présent à ne sélectionner que les fenêtres les plus significatives. Considérons d'abord le cas où l'ensemble des fenêtres réalise une partition de l'image (i.e. il n'y a pas de recouvrement entre les fenêtres). Pour une taille de fenêtre donnée, la valeur de $S(p, p_l, \nu_W, |W|)$ croît avec la valeur de p_l. On peut donc dire que l'ordonnancement des fenêtres selon leur significativité et le même que l'ordonnancement selon les valeurs p_l. Si l'on s'intéresse maintenant à la significativité d'un sous-ensemble de fenêtres noté $\bigcup_{j=1}^{K} W_j$, en utilisant le même modèle naïf, et la même approximation de

Hoeffding, nous pouvons écrire :

$$\begin{aligned} S\left(p, \overline{p}_l, \nu_\cup, |W_j|\right) &= -\log\left(NFA_2\left(\bigcup_{j=1}^{K} W_j\right)\right) \\ &\approx \nu_\cup\left[\overline{p}_l \log\left(\frac{\overline{p_l}}{p}\right) + (1-\overline{p_l})\log\left(\frac{1-\overline{p_l}}{1-p}\right)\right] - \log\left(\eta_2\left(\bigcup_{j=1}^{K} W_j\right)\right). \quad (2.17) \end{aligned}$$

où $\overline{p_l}$ est la probabilité locale au sous-ensemble de fenêtres considéré pour un pixel observable d'être de label *objet* : $\overline{p_l}$ est la moyenne des probabilités locales aux W_j du sous-ensemble de fenêtres (cas de fenêtre de même taille), $\nu_\cup = \left(\sum_{j=1}^{K} \nu_{W_j}\right)$, et $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|} 2^{|\bigcup_{j=1}^{K} W_j|}$ (ou $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$ pour comparaison des résultats en fonction du choix de η_2).

L'algorithme est alors l'algorithme 2.3.

Algorithm 2.3 Algorithme de détection niveau fenêtre.

- Calculer le paramètre p et l'image 3-labels ;
- Calculer les valeurs de p_l associées aux $N_W = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|W|}$ fenêtres réalisant la partition de l'image ;
- Trier les valeurs de p_l , en un vecteur $(p_i)_{1\leq i\leq N_W}$ par ordre décroissant des valeurs ;
- Initialiser S_2^{max} à $-\infty$, $\nu_\cup$ à 0, $\kappa_\cup$ à 0 et $\hat{K}$ à 0 ;
- Pour chaque indice $K \in [1, N_W]$,
  - Ajouter ν_{W_j} à $\nu_\cup$ et κ_{W_j} à $\kappa_\cup$, et calculer $\overline{p_l} = \kappa_\cup/\nu_\cup$ et ;
  - Calculer $S\left(p, \overline{p}_l, \nu_\cup, \left|\bigcup_{j=1}^{K} W_j\right|\right)$ (équation (2.17)) ;
  - Si $S\left(p, \overline{p}_l, \nu_\cup, \left|\bigcup_{j=1}^{K} W_j\right|\right) > S_2^{max}$,
    - $S_2^{max} = S\left(p, \overline{p}_l, \nu_\cup, \left|\bigcup_{j=1}^{K} W_j\right|\right)$;
    - $\hat{K} = K$;
  - Fin test ;
- Fin boucle ;
- $\widehat{D} = \bigcup_{j=1}^{\hat{K}} W_j$

La figure 2.9 donne un exemple de détection fenêtre à partir l'image des points "3 labels" présentée en figure 2.6(d) et rappelée ici en (a). La figure 2.9(b) donne les courbes des valeurs de $\overline{p_l}$ et $S\left(p, \overline{p}_l, n_\cup, |W|\right)$ dans les deux cas $|W| = 20 \times 20$ et $|W| = 40 \times 40$ pour le nombre de tests $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$: on remarque que les courbes de $\overline{p_l}$ sont monotones décroissantes (de part l'ordonnancement des valeurs dans l'algorithme 2.3), tandis que les courbes présentent chacune un maximum bien marqué. L'abscisse de ce maximum correspond au nombre de fenêtres à sélectionner pour la détection, à savoir respectivement 34 pour $|W| = 20 \times 20$ et 13 pour $|W| = 40 \times 40$. Les fenêtres sélectionnées sont tracées sur les images présentées en 2.9(c-d). On remarque quelques fausses détections au niveau des carrosseries des voitures et dans le fond de la scène, fausse détection compréhensible au vu de l'image de points rappelée en (a). Les images 2.9(e-f) présentent les résultats obtenus avec le nombre de tests proposé, $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|} 2^{|\bigcup_{j=1}^{K} W_j|}$: on remarque que les fausses détections ont disparu.

Cette première détection niveau fenêtre appelle plusieurs commentaires.

Tout d'abord nous notons que plusieurs travaux exploitant la méthodologie *a contrario* calculent un Nombre de Fausses Alarmes, sur lequel ils effectuent un seuillage. C'est le cas par exemple dans [42], où ne sont retenues que les fenêtres de valeur de NFA_2 supérieure à une valeur minimale, NFA_2^{min} (ou de significativité mesurée supérieure à $T = -\log\left(NFA_2^{min}\right)$). Le choix de ce seuil NFA_2^{min} est à peine moins difficile qu'un seuil sur la probabilité ou sur la grandeur mesurée (par exemple densité

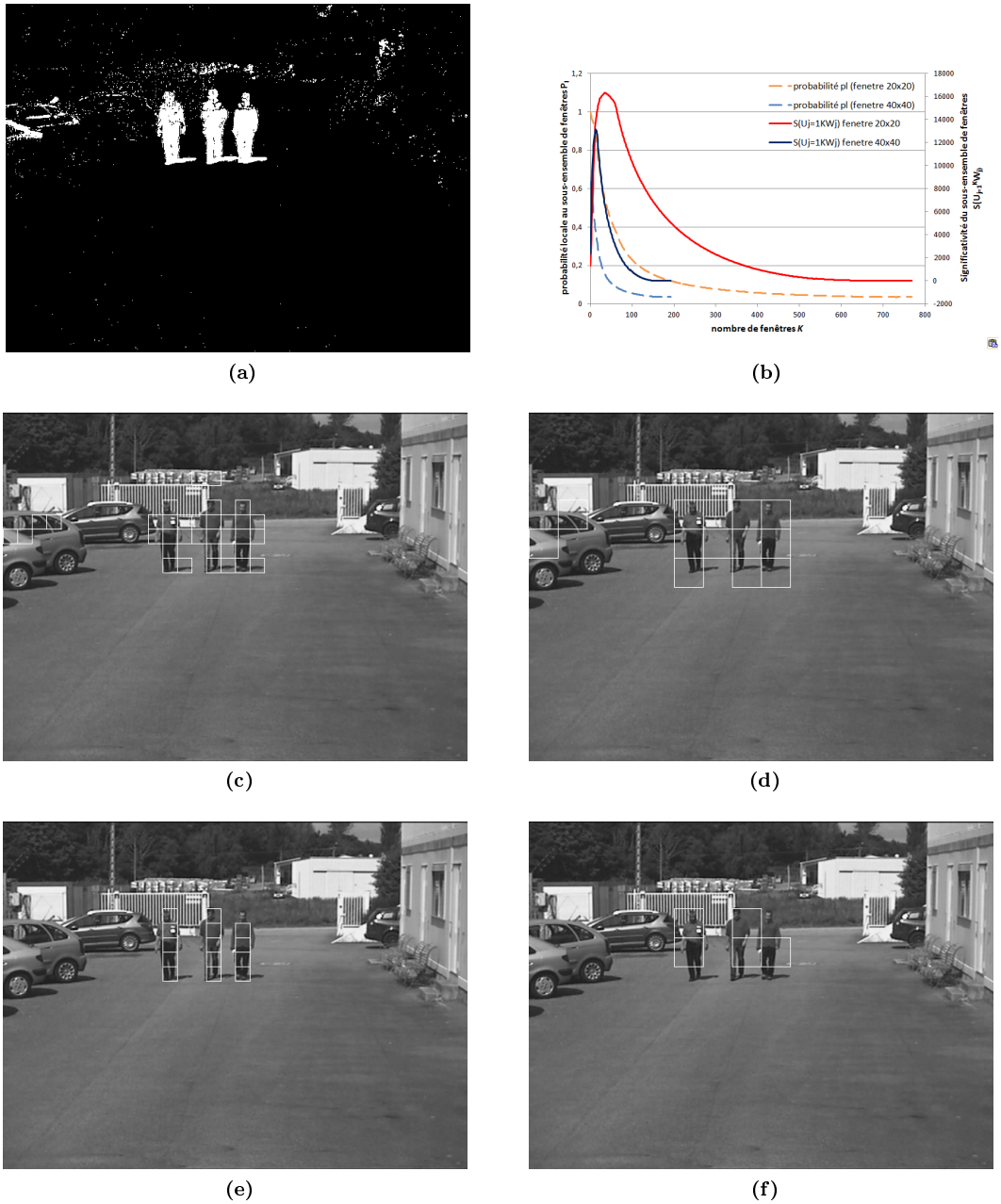


FIGURE 2.9: *Exemple de détection niveau fenêtre.*
(a) est l'image des points 3 labels (Noir≡*objet*, Blanc≡*fond* et Gris≡*inconnu*, cf. figure 2.6(d)), (b) donne les courbes des valeurs de $\overline{p_l}$ et $S\left(p, \overline{p}_l, n_\cup, |W|\right)$ dans les deux cas $|W| = 20 \times 20$ et $|W| = 40 \times 40$ pour le nombre de tests $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$, (c-d) sont les images *foreground* avec en superposition les fenêtres détectées par l'algorithme 2.3 et $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$ respectivement dans les cas $|W| = 20 \times 20$ et $|W| = 40 \times 40$, et (e-f) sont les images *foreground* avec en superposition les fenêtres détectées par l'algorithme 2.3 et $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}\, 2^{|\bigcup_{j=1}^{K} W_j|}$ respectivement dans les cas $|W| = 20 \times 20$ et $|W| = 40 \times 40$.

p_l) : en fait dans le cas d'un nombre de tests constant, c'est même équivalent. Or, un des intérêts de la mesure en terme de Nombre de Fausses Alarmes est d'être un nombre, c'est-à-dire directement comparables avec d'autres valeurs de Nombre de Fausses Alarmes. A l'instar de la méthodologie mise en œuvre pour le niveau pixel (NFA_1), nous avons alors proposé de minimiser le Nombre de Fausses Alarmes, ou de manière équivalente de maximiser la significativité. L'algorithme défini est alors sans aucun paramètre de seuil : le seul paramètre est celui du modèle naïf, à savoir p de la loi binomiale. Insistons ici sur le fait que les algorithmes de détection au niveau pixel (algorithme 2.1) comme au niveau fenêtre (algorithme 2.3) n'utilisent pas de paramètres de seuil, et estiment automatiquement les paramètres du modèle naïf. Pour l'algorithme 2.3, le seul paramètre *a priori* est la taille des fenêtres utilisées, nous verrons juste après comment il est possible de s'en affranchir. Pour terminer sur ces remarques sur l'absence de seuil, précisons tout de même qu'il y a en fait un paramètre NFA_{min} implicite auquel est comparé la valeur du minimum obtenu (le maximum de significativité est supposé reconverti en valeur de NFA) de façon à ne rien détecter en l'absence d'objet. Cependant ce paramètre est moins un seuil qu'une borne supérieure admissible (pour les minima obtenus par l'algorithme), et des travaux futurs pourraient essayer de s'en affranchir en regardant par exemple la profondeur du minima obtenu sur la courbe des NFA en fonction du nombre d'éléments considérés (soit $|D|$ dans le cas niveau pixel, soit K dans le cas niveau fenêtre).

Dans ce qui précède, nous avions considéré un ensemble de fenêtres réalisant une partition de l'image. L'algorithme 2.3 est fondé sur la comparaison, en terme de significativité, de sous-ensembles de fenêtres reliées par une relation d'inclusion. Les calculs dans l'algorithme 2.3 étaient simples de part le non-recouvrement des fenêtres. Généralisons à présent cet algorithme au cas de fenêtres présentant des recouvrements éventuels. Définissons, d'après [42], alors la notion de fenêtres de significativité maximale pour des fenêtres se recouvrant partiellement.

Définition 4. Une fenêtre W_j est dite de significativité maximale si, pour toute fenêtre W_k telle que $W_j \cap W_k \neq \emptyset$, on a :

- $NFA_2(W_j) < NFA_2(W_k)$, ce qui s'interprète comme : W_j approxime mieux l'objet (éventuel) que W_k (figure 2.10a) ;

ou

- $\exists W_i$, avec $W_i \cap W_k \neq \emptyset$, et $NFA_2(W_i) < NFA_2(W_k) < NFA_2(W_j)$, ce qui s'interprète comme : W_k englobe un autre objet localisé sur W_i (figure 2.10b).

L'algorithme qui permet alors de considérer des fenêtres de taille variable avec recouvrement est alors l'algorithme 2.4.

La première partie de l'algorithme vise à éliminer les fenêtres de significativité non maximale d'après la définition 4, ne restent alors que des fenêtres sans recouvrement, sur l'ensemble desquelles on réapplique l'algorithme 2.3 adapté au cas de fenêtres de taille quelconque. La figure 2.11 présente les résultats obtenus sur les images tests considérées précédemment dans le cas de fenêtres variables de tailles 20×20, 20×40, 40×20, et 40×40. Rappelons que le choix de la taille maximale où minimale d'une fenêtre peut être relié à la taille en nombre de points minimale des objets à détecter. Ainsi, tant pour des raisons de temps de calcul que pour des hypothèses sur la taille minimale des objets recherchés, on n'explore pas toutes les tailles de fenêtres. On remarque que les résultats obtenus avec l'un ou l'autre des deux nombres de tests sont tous deux satisfaisants, et *a priori* plus précis dans la détection des objets (ici personnages) que l'utilisation d'un seul type de fenêtre (cf. figure 2.9).

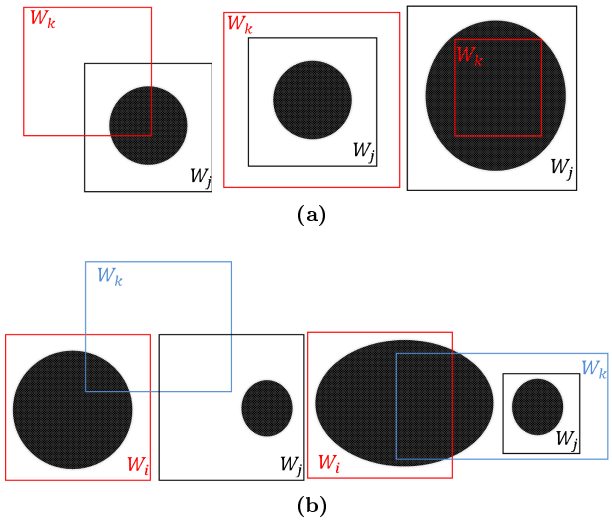


FIGURE 2.10: *Illustration de fenêtres W_j de significativité maximale en cas de recouvrement avec la fenêtre W_k (seuls des cas avec présence d'un objet sont illustrés).*
(a) cas où W_j approxime mieux l'objet que W_k, (b) W_k englobe un autre objet localisé sur W_i.

Algorithm 2.4 Algorithme de détection niveau fenêtre, cas de fenêtres variables.

- Calculer le paramètre p et l'image 3-labels ;
- Soit N_W le nombre total initial de fenêtres considérées, initialiser $\{W\}$ à l'ensemble des fenêtres : $\{W\} = \{W_j, j \in [1, N_W]\}$;
- Calculer les valeurs de (κ_W, ν_W, p_l) associées aux N_W fenêtres ;
- Calculer $S(p, p_l, \nu_W, |W|)$ (équation (2.17)) pour les N_W fenêtres ;
- Trier les valeurs de $S(p, p_l, \nu_W, |W|)$, en un vecteur $(S_i)_{1 \leq i \leq N_W}$ par ordre décroissant des valeurs ;
- Pour chaque indice $i \in [1, N_W]$,
  - Soit W_i la fenêtre d'indice i dans le vecteur trié $(S_i)_{1 \leq i \leq N_W}$;
  - Pour chaque indice $k \in [i+1, N_W]$,
    - Soit W_k la fenêtre d'indice k dans le vecteur trié $(S_i)_{1 \leq i \leq N_{W_j}}$, si $W_k \cap W_i \neq \emptyset$,
      - retirer W_k de l'ensemble $\{W\}$, du vecteur trié $(S_i)_{1 \leq i \leq N_W}$ et décrémenter N_W ;
    - Fin test ;
  - Fin boucle ;
- Fin boucle ;
- Trier les valeurs de p_l des fenêtres de $\{W\}$, en un vecteur $(p_i)_{1 \leq i \leq N_W}$ par ordre décroissant des valeurs ;
- Initialiser $S_2^{max} = -\infty$, $\nu_\cup = 0$, $\kappa_\cup = 0$ et $\hat{K}$=0 ;
- Pour chaque indice $K \in [1, N_W]$, soit W_K la fenêtre d'indice K dans le vecteur trié $(p_i)_{1 \leq i \leq N_W}$;
  - Ajouter ν_{W_K} à $\nu_\cup$, ajouter κ_{W_K} à $\kappa_\cup$, et calculer $\overline{p_l} = \kappa_\cup / \nu_\cup$;
  - Calculer $S\left(p, \overline{p}_l, \nu_\cup, \left|\cup_{i=1}^{K} W_i\right|\right)$ (équation (2.17)) ;
  - Si $S(p, \overline{p}_l, \nu_\cup) > S_2^{max}$,
    - $S_2^{max} = S\left(p, \overline{p}_l, \nu_\cup, \left|\cup_{i=1}^{K} W_i\right|\right)$;
    - $\hat{K} = K$;
  - Fin test ;
- Fin boucle ;
- $\widehat{D} = \bigcup_{j=1}^{\hat{K}} W_j$

(a) (b)

FIGURE 2.11: *Exemple de détection niveau fenêtre.*
Détection à partir de l'image des points 3 labels (Noir≡*objet*, Blanc≡*fond* et Gris≡*inconnu*, cf. figure 2.6(d)) et selon l'algorithme 2.4 fonctionnant avec des fenêtres de taille variable (ici $|W_j| \in \{20 \times 20, 20 \times 40, 40 \times 20, 40 \times 40\}$) : (a) image *foreground* avec en superposition les fenêtres détectées en utilisant le nombre de tests $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$, (b) image *foreground* avec en superposition les fenêtres détectées en utilisant le nombre de tests $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D} \cup \overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|} 2^{|\bigcup_{j=1}^{K} W_j|}$.

Extension de la mesure au cas multitemporel

Nous proposons une extension de la mesure précédente au cas d'images de points multidimensionnelles. Le cas applicatif typique est celui d'images de points multitemporelles, i.e. issues non d'un seul couple d'images (cas dit monotemporel), par exemple $(t-1, t)$, mais d'un ensemble de couples d'images (cas dit multitemporel) , par exemple $((t-3, t-2), (t-2, t-1), (t-1, t))$.

L'extension proposée du critère NFA_2 précédent est alors comme suit.

Soit non pas une image de points mais N images de points, les paramètres (p, κ, ν) sont calculés non plus sur des domaines 2D (image, pavés rectangulaires i.e. fenêtres) mais sur des domaines 3D, tels que les deux premières dimensions soient celles correspondantes des domaines 2D précédents et la troisième dimension est celle associée à N, i.e. la dimension temporelle dans notre cadre applicatif.

L'algorithme 2.4 reste valide sauf que les fenêtres sont maintenant 3D (en particulier leur cardinal en 2D est multiplié par N) et les paramètres (p, κ_W, ν_W) sont maintenant définis par :

$$p = \frac{\sum_{t=1}^{N} \left|\mathcal{D}_t \backslash \hat{D}_t\right|}{\sum_{t=1}^{N} |\mathcal{D}_t|}, \tag{2.18}$$

$$\kappa_{W_j} = \sum_{t=1}^{N} \left|W_j \cap \mathcal{D}_t \backslash \hat{D}_t\right|, \quad \nu_{W_j} = \sum_{t=1}^{N} |W_j \cap \mathcal{D}_t|, \quad p_l = \frac{\kappa}{\nu} \tag{2.19}$$

où l'indice t réfère l'image de points considérée parmi les N images de points. Comme déjà dit, une telle extension peut se faire selon d'autres dimensions que la dimension temporelle. Cependant, comme elle a pour effet de renforcer les détections des points corrélés selon la troisième dimension, la dimension temporelle parait un choix valide pour notre application de détection d'objets (contrairement à la couleur par exemple). Notons aussi que dans notre cas le véhicule étant en mouvement, ce mouvement doit être compensé (voir section 2.2.2.3). Cette opération est faite par projection de toutes les images de points selon une géométrie donnée : celle de l'image à l'instant t dans notre cas. Cette projection induisant elle-même des distorsions et erreurs nous nous limitons à $N = 3$ dans nos expériences.

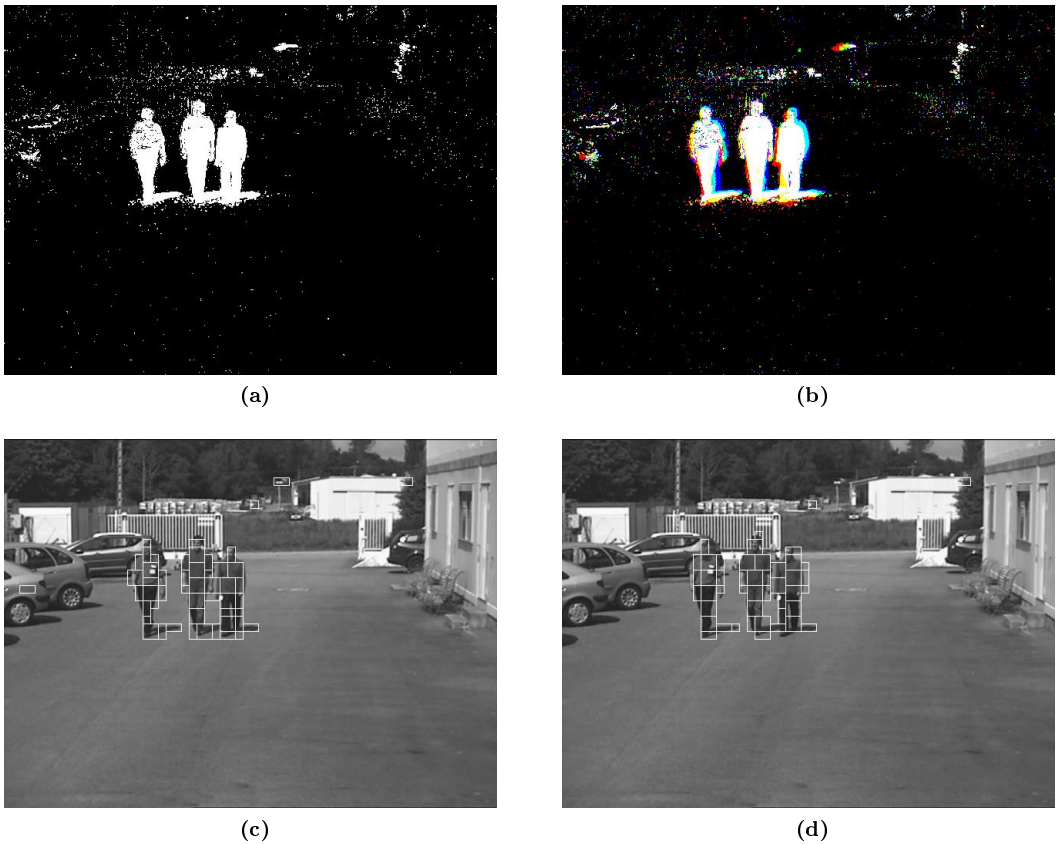


FIGURE 2.12: *Exemple de résultat de détection de groupement de points dans le cas multitemporel.* (a) image de points résultat de la détection selon le critère NFA_1, (b) image de points RGB résultat de la détection selon le critère NFA_1 aux instants $((t-3,t-2),(t-2,t-1),(t-1,t))$, (c) résultat de détection des objets selon le critère NFA_2 appliqué sur l'image (a), pour des tailles de fenêtres 2D $\in \{10\times 10, 10\times 20, 20\times 10, 20\times 20\}$, (d) résultat de détection des objets selon le critère NFA_2 appliqué sur l'image (b), pour des tailles de fenêtres 3D $\in \{10\times 10\times 3, 10\times 20\times 3, 20\times 10\times 3, 20\times 20\times 3\}$.

La figure 2.12 présente un exemple de résultat illustrant l'apport de la dimension multitemporelle. Notons tout d'abord que la taille verticale et horizontale des fenêtres 2D utilisées est ici de 10 ou 20 (on utilise les quatre combinaisons possibles). En effet, pour des fenêtres de taille quadruple (double dans chaque dimension), les résultats obtenus en monotemporel sont déjà très satisfaisants (cf. figure 2.11). La première colonne de la figure 2.12 montre les résultats en monotemporel pour les 'petites' fenêtres. On note sur la figure 2.12(c) quatre fausses alarmes : sur la carrosserie de la voiture et dans le fond de la scène. La seconde colonne de la figure 2.12 donne les résultats en multitemporel. On note sur la figure 2.12(d) que deux des fausses alarmes ont été éliminées. La figure 2.12(b), qui est une composition colorée des résultats de la détection pixelique (critère NFA_1), montre en effet que ces erreurs sont temporellement décorrélées : elles apparaissent en couleur et non en blanc signalant ainsi leur présence dans seulement une ou deux des trois images considérées conjointement dans le traitement multitemporel.

Retour sur la définition de la probabilité locale (sur une fenêtre) versus la probabilité globale (sur l'image)

Basiquement, la mesure de significativité compare une mesure de probabilité locale à une fenêtre (ou à un sous-ensemble de fenêtres) avec une mesure de probabilité globale à l'image : ce sont les

paramètres p et p_l de l'équation (2.16) (ou p et $\overline{p}_l$ de l'équation (2.17)). Nous discutons (ici) de la définition de p, ν_W et κ_W. En effet, si pour Dibos et al. [42], $\frac{\kappa_W}{\nu_W}$ représente la probabilité d'un point localement à la fenêtre W, de sorte à détecter les fenêtres sur lesquelles cette probabilité s'éloigne étonnamment de la probabilité à l'échelle de l'image, p, pour Desolneux et al. [39], le sens donné aux paramètres p, ν_W et κ_W est différent : p est la surface relative de la fenêtre par rapport au domaine image donc dépendant de W, soit p devient p_W ; ν_W est le nombre de points dans l'image donc maintenant indépendant de W soit ν_W devient ν ; et κ_W est le nombre de points dans la fenêtre. Ainsi, dans [39], $\frac{\kappa_W}{\nu}$ représente la probabilité de la fenêtre W localement à l'ensemble des points, de sorte à détecter les fenêtres sur lesquelles cette probabilité s'éloigne étonnamment de la probabilité (de la fenêtre) estimée par rapport aux pixels de l'image, p_W.

Dans [39], Desolneux et al. ne considéraient que des images définies en tout pixel. Nous proposons alors d'étendre ses définitions de p_W, ν et κ_W dans le cas d'une image avec des pixels indéfinis, de la façon suivante : p_W est le ratio entre le nombre de pixels définis dans la fenêtre W et le nombre total de pixels définis sur l'image, soit avec nos notations $p_W = \frac{|W \cap \mathcal{D}|}{|\mathcal{D}|}$, κ_W est le nombre de points dans la fenêtre W, soit $\kappa_W = \left|W \cap \mathcal{D} \backslash \hat{D}\right|$, et ν est le nombre total de points sur l'image, soit $\nu = \left|\mathcal{D} \backslash \hat{D}\right|$. On rappelle par ailleurs les définitions des paramètres p, ν_W et κ_W, selon Dibos et al. [42], en présence de pixels de label *inconnu* : p est estimé par le ratio entre le nombre total de points sur l'image et le nombre total de pixels définis, soit avec nos notations $p = \frac{\left|\mathcal{D} \backslash \hat{D}\right|}{|\mathcal{D}|}$, κ_W est le nombre de pixels *objet* dans la fenêtre W_W, soit $\kappa_W = \left|W \cap \mathcal{D} \backslash \hat{D}\right|$, et ν_W est le nombre total de pixels définis dans W, soit $\nu_W = |W \cap \mathcal{D}|$.

L'expression de la significativité d'une fenêtre W devient alors :

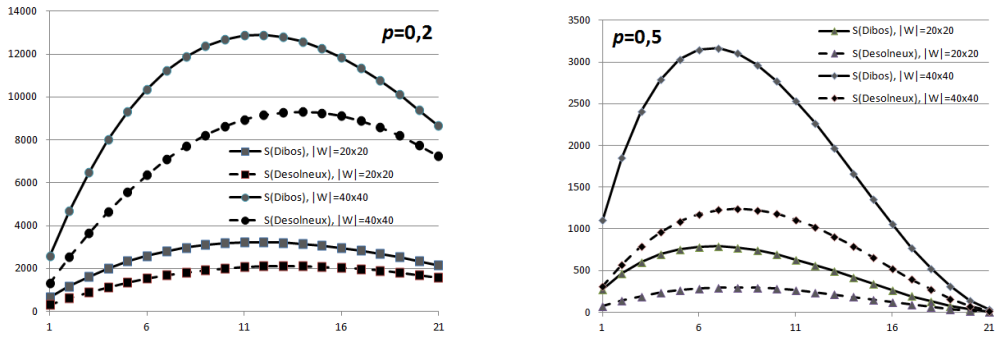
$$S\left(p_W, p_l, \nu, |W|\right) = \nu \left[p_l \log\left(\frac{p_l}{p}\right)\right] + \nu \left[(1-p_l) \log\left(\frac{1-p_l}{1-p}\right)\right] - \log\left(\eta_2\left(W\right)\right),$$

avec $p_l = \frac{\kappa_W}{\nu}$.

Notons d'abord que la borne sur la taille maximale de la fenêtre pour détecter un groupement de κ_W points est κ_W / p, avec p probabilité d'un point dans l'image puisque dans [39], κ_W divisé par le nombre total de points doit être inférieur à la surface relative de la fenêtre rapportée au domaine image. Cette borne est la même que trouvée précédemment. Remarquons par ailleurs que

$$\nu \left[\frac{\kappa_W}{\nu} \log\left(\frac{\kappa_W}{\nu} \frac{1}{p_W}\right)\right] = \left[\kappa_W \log\left(\frac{\kappa_W \left|\mathcal{D}\right|}{\left|\mathcal{D} \backslash \hat{D}\right| \left|W \cap \mathcal{D}\right|}\right)\right] = \nu_W \left[\frac{\kappa_W}{\nu_W} \log\left(\frac{\kappa_W}{\nu_W} \frac{1}{p}\right)\right],$$

donc le premier terme de l'expression de la significativité est identique pour les deux définitions de paramètres (de même que le terme correspondant au nombre de tests). Le second terme de l'expression de la significativité est négatif puisque $p_l \geq p$. Il vaut $\left[(|W \cap \mathcal{D}| - \kappa_W) \log\left(\frac{1 - \frac{\kappa_W}{|W \cap \mathcal{D}|}}{1 - \frac{\left|\mathcal{D} \backslash \hat{D}\right|}{|\mathcal{D}|}}\right)\right]$ avec les définitions des paramètres de [42], donc tend vers zéro quand $\kappa_W \to |W \cap \mathcal{D}|$, c'est à dire que les pixels définis dans la fenêtre considérée sont quasiment tous des points. Maintenant, avec les définitions des paramètres de [39], ce terme vaut $\left[\left(\left|\mathcal{D} \backslash \hat{D}\right| - \kappa_W\right) \log\left(\frac{1 - \frac{\kappa_W}{\left|\mathcal{D} \backslash \hat{D}\right|}}{1 - \frac{|W \cap \mathcal{D}|}{|\mathcal{D}|}}\right)\right]$, donc tend vers zéro quand $\kappa_W \to \left|\mathcal{D} \backslash \hat{D}\right|$, c'est à dire que la fenêtre considérée englobe quasiment tous les points. Ainsi, on a l'interprétation suivante des différences de définition des paramètres : selon Desolneux et al. [39], une fenêtre (ou un sous-ensemble de fenêtres) est d'autant plus significative qu'elle englobe un maximum de points (au plus près) ; et selon Dibos et al. [42], une fenêtre (ou un sous-ensemble de fenêtres) est d'autant plus significative que la densité des points y est élevée (et les points nombreux). Dans le cas



(a) (b)

FIGURE 2.13: *Courbes de significativité du sous-ensemble de fenêtres en fonction de son cardinal.* le nombre de tests $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$, (a) $p = 0.2$ et (b) $p = 0.5$.

(a) (b)

FIGURE 2.14: *Exemples de détection niveau fenêtre selon l'algorithme 2.4 et avec la définition des paramètres du modèle de [39].*
(a) cas $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$, (b) cas $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|} 2^{|\bigcup_{j=1}^{K} W_j|}$.

d'une image de points sans bruit (fausses détections au niveau pixel), les deux définitions donneront des résultats très proches. Par contre en présence de bruit, la définition de [39] sera plus favorable que [42] à la détection (via l'algorithme 2.3 ou l'algorithme 2.4) de fenêtres même de densité moyenne à faible de façon à détecter un nombre significatif de points.

Pour illustrer cette différence de comportement, considérons un ensemble de 21 fenêtres (sans recouvrement et de taille fixe $|W|$) de probabilités locales $p_l = \frac{\kappa_W}{\nu_W}$ variant linéairement de 1 à 0.05, et observons la variation de la courbe de significativité $S\left(p, \overline{p}_l, \nu_\cup, \left|\bigcup_{j=1}^{K} W_j\right|\right)$ en fonction du nombre K de fenêtres considérées dans le sous-ensemble. La figure 2.13 présente les courbes obtenues pour deux probabilités globales de points : $p = 0.2$ et $p = 0.5$, et deux tailles de fenêtre : $|W| = 20 \times 20$, et $|W| = 40 \times 40$. On a pris $\eta_2\left(\bigcup_{j=1}^{K} W_j\right) = \frac{|\mathcal{D}\cup\overline{\mathcal{D}}|}{|\bigcup_{j=1}^{K} W_j|}$ qui ne dépend pas des définitions des paramètres. On remarque d'abord qu'avec les définitions de [42], les valeurs de significativités sont globalement plus élevées. On remarque également que, comme annoncé, les maxima de significativité correspondent à des nombres de fenêtres ($\hat{K}$) légèrement plus élevés (d'une ou deux unités) pour les significativités calculées selon les définitions de [39]. Cette remarque est confirmée par les résultats obtenus sur notre image test et présentés en figure 2.14.

Critère NFA_1 versus critère NFA_2

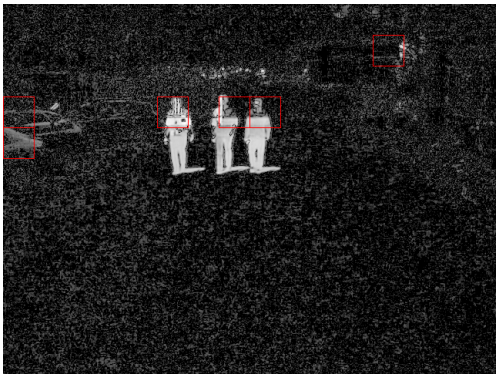
L'algorithme 2.1 (présenté au paragraphe 2.3.2.1) qui exploite le critère NFA_1, est semblable à une étape de détection au niveau du pixel qui produit une image binaire (i.e. de points). Une propriété fondamentale des mesures NFA est d'être des mesures absolues donc potentiellement comparables. Ceci nous permet une nouvelle approche : nous proposons d'utiliser le seul critère NFA_2 pour faire la détection directement à partir des images à niveaux de gris, selon le seul critère NFA_2. Pour cela, nous faisons simplement varier les seuils sur les images $|\Delta|$ (valeur absolue en chaque pixel de la valeur du pixel sur l'image différence), et calculons le Nombre de Fausses Alarmes supposant la loi binomiale pour modèle naïf en fonction du seuil. Le seuil optimal pour une fenêtre W est alors le seuil maximisant la valeur de $S\left(p\left(\tau\right), p_l\left(\tau\right), \nu_W\left(\tau\right), |W|\right)$ où les dépendances des paramètres envers τ, le seuil utilisé sur les niveaux de gris, sont indiquées : $\tau\hat{}_W = \underset{\tau}{\text{argmax}} S\left(p\left(\tau\right), p_l\left(\tau\right), \nu_W\left(\tau\right), |W|\right)$. Il est clair qu'une telle approche parait irréaliste dans l'état actuel des puissances de calculs des ordinateurs embarqués, surtout si elle est couplée à l'algorithme 2.4. Cependant elle est intéressante au niveau théorique, aussi présentons nous quelques résultats obtenus dans le cas de notre image test.

La figure 2.15 présente pour six cas de fenêtres 40×40 les courbes de significativité en fonction du seuil τ, et les images seuillées à la valeur $\hat{\tau}_W$. Les fenêtres choisies à titre d'illustration sont indiquées sur la figure 2.15(a) : il s'agit de fenêtres incluant une partie du haut de chacun des personnages et d'une fenêtre incluant les branches du cyprès sur lesquelles il y a un très léger mouvement, et de deux fenêtres au niveau des carrosseries des voitures. La figure 2.15(b) présente les courbes des significativité en fonction du seuil sur les niveaux de gris τ. On remarque que les courbes des fenêtres incluant une partie des personnages (courbes intitulées "fenêtre (120,200)", "fenêtre (120,280)" et "fenêtre (120,320)") présentent un maximum de significativité élevé (de ≥ 1000 à ≥ 1700 environ pour des personnages de gauche à droite de plus à moins apparents) et correspondant à des valeurs de $\tau\hat{}_W$ respectivement égaux à 7, 8 et 10. Les courbes des fenêtres incluant soit le feuillage (courbe intitulée "fenêtre (40,480)") soit une partie des voitures non réfléchissante (courbe intitulée "fenêtre (120,0)") présentent un maximum de significativité faible (de ≈ 35 pour le feuillage ≈ 160 pour les voitures garées) et correspondant à des valeurs de $\tau\hat{}_W$ respectivement égaux à 3 et 4, tandis que la courbe incluant la carrosserie réfléchissante (courbe intitulée "fenêtre (160,0)") présente un maximum de significativité élevé (≈ 1000) correspondant à une valeur de $\tau\hat{}_W$ égale à 4. Il paraît donc difficile d'éliminer les fausses détections associées à cette fenêtre.

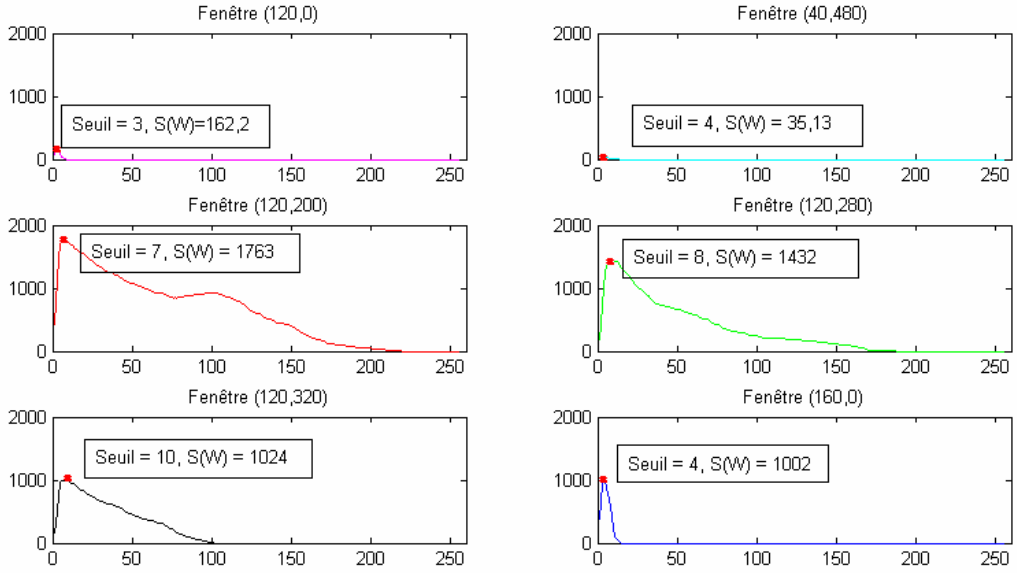
La figure 2.16 présente les résultats de la détection fenêtre par fenêtre (dans cet exemple $|W| = 40 \times 40$) avec comme seuil sur les niveaux de gris et pour la fenêtre considérée

$$\tau\hat{}_W = \underset{\tau}{\text{argmax}} S\left(p\left(\tau\right), p_l\left(\tau\right), \nu_W\left(\tau\right), |W|\right),$$

à condition que $\underset{\tau}{\max} S\left(p\left(\tau\right), p_l\left(\tau\right), \nu_W\left(\tau\right), |W|\right) \geq T$. On observe que cette approche est donc plus élégante que l'enchainement des deux critères NFA_1 et NFA_2 puisqu'elle ne fait intervenir qu'un seul critère (NFA_2) mais elle a l'inconvénient de réintroduire un seuil non automatique, T. Pour s'en affranchir il faudrait (probablement, car nous ne l'avons pas testé) coupler la variation du seuil sur les niveaux de gris et la détection du sous-ensemble de fenêtres de significativité maximale selon l'algorithme 2.4. Cependant un tel couplage paraît excessivement lourd, et nous laissons cette piste (éventuelle) pour des travaux futurs. Notons tout de même que la détection est relativement robuste par rapport à la valeur de T comme l'illustre la figure 2.16.

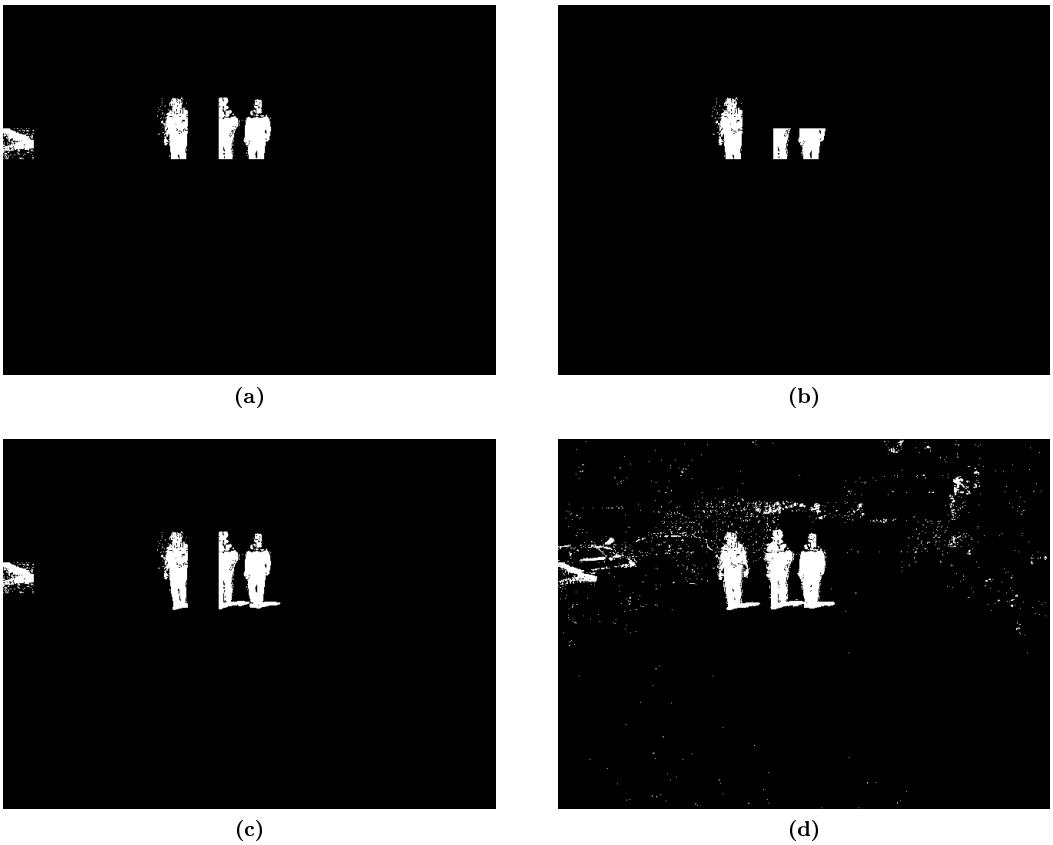


(a)



(b)

FIGURE 2.15: *Exemple de résultat de détection de points à partir du critère NFA_2.*
(a) image des différences absolues $|\Delta|$ (échelle logarithmique) avec tracé des six fenêtres de taille 40×40 étudiées, (b) courbes $S\left(p\left(\tau\right), p_l\left(\tau\right), \nu_W\left(\tau\right), |W|\right)$ en fonction du seuil sur les niveaux de gris τ pour chacune des cinq fenêtres.



(a) **(b)** **(c)** **(d)**

FIGURE 2.16: *Exemple de résultat de détection de points à partir du seul critère NFA_2.*
(a-c) mosaïque de fenêtres seuillées au seuil maximisant la significativité si cette dernière est supérieure à T; (a) $T = 1000$, (b) $T = 1500$, (c) $T = 500$; (d) image résultat de la détection de points selon le critère du NFA_1.

Conclusion

Nous avons proposé une nouvelle approche qui suit le principe de la détection *a contrario*. Un atout fort de cette approche est qu'elle ne modélise pas directement les objets mais le bruit selon un modèle dit naïf. Aussi les paramètres de ce modèle sont-ils beaucoup plus simples à estimer que s'il s'agissait des objets eux-mêmes. Dans notre cas, ces paramètres sont déduits directement des images à traiter : σ, écart-type sur les valeurs de niveaux de gris (observés à t) de l'ensemble du domaine image, et ratio entre le nombre de points sur l'image (deux ou trois labels), et p, taille du domaine image. La section 2.3.3 présente l'application de cette approche aux problèmes de la détection de certains objets.

La méthode développée ne fait d'hypothèse que sur le bruit donc elle n'a pas d'information *a priori* sur les objets à détecter, elle n'utilise que les données d'entrée.

2.3.3 Application

Dans la section précédente nous avons présenté une approche en détection basée sur le test successif de deux critères : le fait qu'un pixel n'appartient pas à l'ensemble des pixels ayant une valeur significativement faible dans $\Delta^2(s)$), et le fait qu'il appartient à un groupement (*cluster*) de pixels. Pour les deux critères, la significativité est mesurée en terme d'incohérence par rapport un modèle naïf. Ici, nous illustrons cette approche en sa généricité vis-à-vis des données en entrée, ce qui est un atout important de cette approche (en plus d'être sans paramètre).

2.3.3.1 Trois différentes déclinaisons de l'approche proposée

Précisons ici les trois façons dont la méthode a été instanciée dans cette étude. Notre objectif est la détection d'objets dans une séquence vidéo pour le cas d'une caméra fixé sur un véhicule, généralement en mouvement. Or dans la scène extérieure, les d'objets d'intérêt sont généralement les piétons, les autres véhicules et les obstacles sur la chaussée ou à proximité (piétons sur le trottoir susceptibles de traverser). L'approche *a contrario* ne fait pas d'hypothèse sur les caractéristiques des objets recherchés (seulement sur le modèle de bruit), aussi l'*a priori* sur les objets lié à l'application sera entièrement contenu dans le type de données présenté en entrée de la méthode.

La figure 2.17 illustre les différentes données en entrée testées (images radiométriques versus cartes de disparité, modèle de déplacement versus modèle de route), conduisant à différents types d'objets détectés.

Pour les objets en mouvement, les images en entrée amenant la construction de l'image Δ (algorithme 2.1 ou algorithme 2.2) sont l'image à l'instant t, et la projection de l'image précédente dans la géométrie de l'image courante, permettant de compenser le mouvement propre engendré par le véhicule et de mettre en évidence les objets en mouvement. Les types d'images considérées peuvent soit être des images usuelles, appelées ici images radiométriques (on réserve le terme "images à niveaux de gris" à la désignation générique d'images ayant plus de deux, images binaires, ou trois, images ternaires, valeurs, sans préjuger de la signification physique de ces valeurs), soit des cartes de disparité. Pour les objets statiques, l'image Δ est construite à partir de l'image à l'instant t, et d'une image représentant la scène vierge, i.e. sans objets. Pour des raisons liées à l'estimation de cette scène vierge, seul le cas d'images de disparité est considéré dans notre étude.

Le tableau 2.1 résume les caractéristiques des objets détectés selon les combinaisons considérées de type de données en entrée de notre approche, avec en colonne le *modèle à t* et en ligne l'*innovation à t* (cf. figure 2.5). Les paragraphes suivants précisent ces combinaisons : nous fixons d'abord le type de *modèle à t*, puis les sous cas possibles d'*innovation à t*.

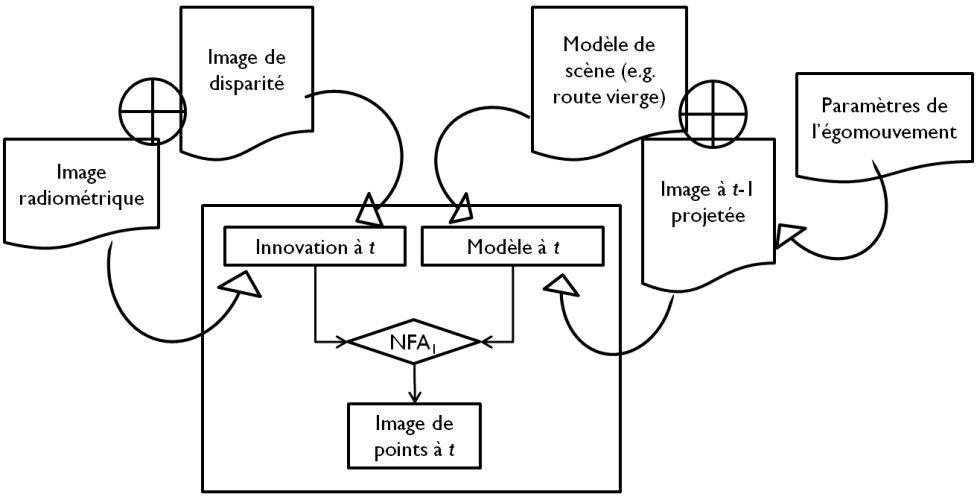


FIGURE 2.17: *Approche proposée : Polyvalence et Généricité de la méthode.*

	Image radiométrique	Image de disparité
Image radiométrique projetée	Objets en mouvement	/
Image de disparité projetée	/	Objets en mouvement
Carte de disparité de la scène 3D vierge	/	Objets saillants

TABLE 2.1: *Type d'objets détectés selon les combinaisons entre les images d'entrées : en colonne, les possibles* modèle à t *et en ligne les possibles* innovation à t.

Premier type de *modèle à* t : scène 3D *a priori*. Selon ce premier modèle, l'image *background* est une scène vierge à l'exception de la route. Cette dernière est estimée directement à partir de l'analyse de la v-disparité (cf. section 2.2.2.2). Ainsi, dans la scène *a priori*, chaque point 3D soit appartient à la route, i.e. il satisfait l'équation de la route, soit est situé à l'infini. Ainsi, les caractéristiques *a priori* des éléments de la scène (la route) sont les coordonnées 3D. Aussi, l'image de *background* que l'on est capable d'en déduire est-elle nécessairement une image de disparité (et non de valeurs radiométriques).

Dans l'équation (2.11), nous posons donc : $I_t \equiv \delta_t$ et $\tilde{B}_t \equiv \delta_{road_t}$; ainsi, l'image de différence, s'écrit :

$$\Delta(s) = \delta_t(s) - \delta_{road_t}(s) \tag{2.20}$$

avec s un pixel, $\delta_t(s)$ la valeur de la disparité pour s de l'image observée à l'instant t, et $\delta_{road_t}(s)$ la valeur de la disparité pour s de l'image du *background* représentant la scène vierge à l'exception de la route.

En se référant à la figure 2.17, les images en entrée sont l'image de disparité à t qui joue le rôle de l'innovation, et la carte de disparité associée à la scène vierge à l'exception de la route, qui joue le rôle du *modèle à* t, et les objets sont détectés comme des objets saillants par rapport à ce modèle, i.e. non situés dans le (ou proches du) plan de la route. Des exemples de résultats sont présentés dans la section 2.3.3.2.

Deuxième type de *modèle à* t : projection de l'image à $t-1$. Selon ce second modèle, l'image de *background* est déduite de la scène observée à l'instant précédent. Comme la caméra est *a priori* mobile, il ne s'agit pas directement de l'image observée à $t-1$, mais de cette dernière après compensation du mouvement de la caméra (mouvement du véhicule dans notre cas où la caméra est fixée de façon rigide au véhicule). Notons que pour construire ce *background*, nous avons besoin (cf. section 2.2.2.3) de : l'image des valeurs radiométriques à $t-1$, de la carte de disparité à $t-1$, et du mouvement 6D de la caméra, entre les instants $t-1$ et t, à savoir les composantes de translation $T = (T_X, T_Y, T_Z)$ et rotation $\Omega = (\Omega_X, \Omega_Y, \Omega_Z)$. Plus le mouvement (en particulier la composante rotationnelle) est faible, plus précise est la projection de l'image. En sortie de la projection de $(t-1) \rightarrow t$, nous disposons d'estimations de : l'image des valeurs radiométriques à t, de l'image de disparité à t. Ces images contiennent des valeurs manquantes en certains pixels, d'une part en raison de valeurs manquantes dans la carte de disparité à $t-1$, et d'autre part du mouvement de la caméra qui a modifié le champ de vision.

Généralement, les méthodes de détection se basent sur les valeurs radiométriques comme information pertinente pour la détection. Dans notre travail, nous proposons d'utiliser aussi l'image de disparité pour la détection.

Premier type d'*innovation à* t : les images radiométriques. Le type d'images considérées est ici classiquement des images de valeurs radiométriques. Dans l'équation (2.11), nous posons : $I_t \equiv I_t$ et $\tilde{B}_t \equiv I_{t-1 \rightarrow t}$; ainsi, l'image de différence, s'écrit :

$$\Delta(s) = I_t(s) - I_{t-1 \rightarrow t}(s) \tag{2.21}$$

avec s un pixel, $I_t(s)$ la valeur radiométrique pour s dans l'image observée à l'instant t, et $I_{t-1 \rightarrow t}(s)$ la valeur radiométrique pour s de l'image du *background* résultat de la projection de I_{t-1} à t (connaissant la disparité δ_{t-1} et le mouvement 6D (T, Ω)).

En se référant à la figure 2.17, les images en entrée sont l'image radiométrique à t qui joue le rôle de l'innovation, l'image radiométrique à $t-1$ projetée à t, qui joue le rôle du *modèle à t*, et les objets sont détectés comme des objets en mouvement, i.e. non statiques. Des exemples de résultats sont présentés dans la section 2.3.3.2 qui commentent les types de mouvement perceptibles.

Deuxième type d'*innovation à t* : les images de disparité. Dans l'équation (2.11), nous posons : $I_t \equiv \delta_t$ et $\tilde{B}_t \equiv \delta_{t-1\to t}$; ainsi, l'image de différence, s'écrit :

$$\Delta(s) = \delta_t(s) - \delta_{t-1\to t}(s) \tag{2.22}$$

avec s un pixel, $\delta_t(s)$ la valeur de la disparité pour s de l'image observée à l'instant t, et $\delta_{t-1\to t}(s)$ la valeur de la disparité pour s de l'image du *background* résultat de la projection de δ_{t-1} à t (connaissant le mouvement 6D (T,Ω)).

En se référant à la figure 2.17, les images en entrée sont l'image de disparité à t qui joue le rôle de l'innovation, et l'image de disparité à $t-1$ projetée à t, qui joue le rôle du *modèle à t*, et les objets sont détectés comme des objets en mouvement, i.e. non statiques. Des exemples de résultats sont présentés dans la section 2.3.3.2 qui commentent les types de mouvement perceptibles.

2.3.3.2 Premiers exemples de résultats

La scène considérée ici est simulée par Pro-SiVIC. Elle comprend des bâtiments le long de la route, une voiture qui arrive de face dans la voie d'à côté, ainsi que deux piétons marchant lentement au niveau du passage piéton. Il y a aussi une voiture très loin au fond de la scène et un autre piéton également à une distance élevée. La caméra est embarquée sur un véhicule ayant une vitesse d'environ $10\,m/s$. La figure 2.18 présente les résultats obtenus en monotemporel (un seul couple d'images considéré). On s'intéresse ici à la comparaison des trois combinaisons possibles de type de données et de type de *background* ou *modèle à t*, et on observe :

- les données radiométriques conduisent à des images de points, en sortie de l'algorithme 2.2, avec plus de fausses détections (niveau pixel) que les données de disparité, qui ont tendance elles plutôt à sous-détecter. Après détection des fenêtres significatives selon l'algorithme 2.4, les résultats sont très proches avec une légère sous détection pour les données de disparité. On remarque aussi que dans les deux cas, seule la voiture en sens inverse est détectée, mais ni les piétons, probablement car ils sont trop lents, et donc vus comme des objets statiques, ni la voiture au fond, probablement car elle est trop petite en pixels et a un mouvement apparent sur l'image trop faible (à cause de la distance).
- L'utilisation d'un *modèle à t* de type scène vierge sauf le plan de la route permet de détecter les objets verticaux, à savoir les bâtiments, la voiture en face et les deux piétons. La voiture au fond de la scène et le piéton plus loin ne sont pas perçus, probablement en raison de leur proximité (distance mesurée en pixels) par rapport au plan de la route.

Une des premières observations est donc que l'utilisation des images radiométriques conduit à des résultats à la fois moins robustes et plus précis que l'utilisation des images de disparité (dans le cas où le *modèle à t* est l'image à $t-1$ projetée à t). Commentons plus avant cette observation.

Le manque de précision sur les images de disparité est simplement dû à leur calcul qui effectue la mise en correspondance des pixels dans chacune des images de la paire stéréo par corrélation de fenêtres spatiales, induisant donc une perte de résolution spatiale de l'ordre de la taille de la fenêtre utilisée (même en l'absence de sous-échantillonnage de l'image, i.e. avec des fenêtres dites glissantes).

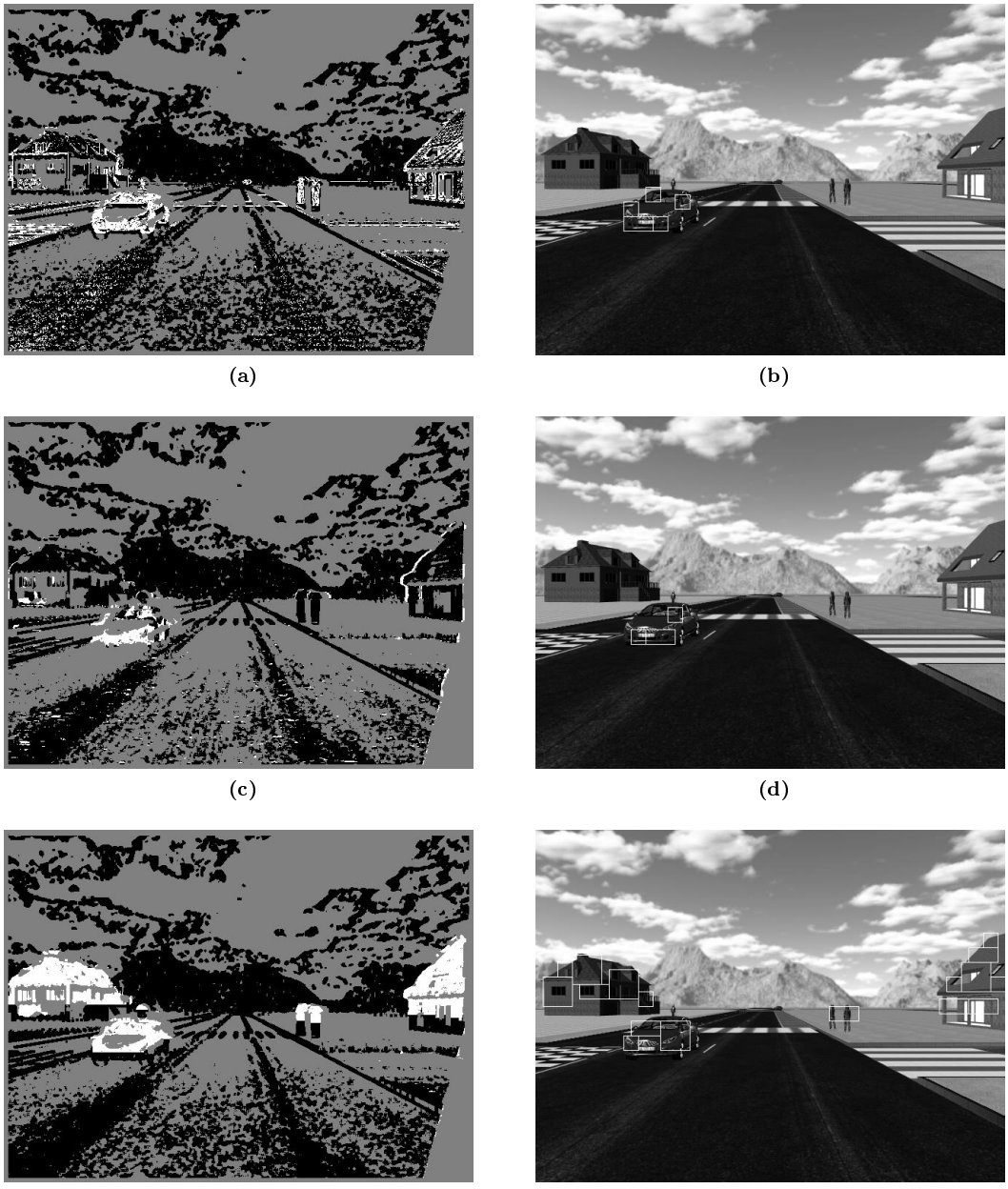


FIGURE 2.18: *Exemples de résultats obtenus en monotemporel dans le cas d'images simulées sous Pro-SiVIC.* 1ère colonne (a,c,e) images de points (sorties de l'algorithme 2.2) : les 3 labels sont : Noir≡*objet*, Blanc≡*fond* et Gris≡*inconnu* (pixels pour lesquels la valeur de disparité n'a pu être estimée) ; 2ème colonne (b,d,f) images de détections niveau fenêtres (sorties de l'algorithme 2.4) ; 1ère ligne (a-b) données d'entrée ≡ images radiométriques, *modèle à* $t \equiv$ projection de l'image à $t-1$; 2ème ligne (c-d) données d'entrée ≡ images de disparité, *modèle à* $t \equiv$ projection de l'image à $t-1$; 3ème ligne (e-f) données d'entrée ≡ images de disparité, *modèle à* $t\equiv$ scène vierge à l'exception du plan de la route.

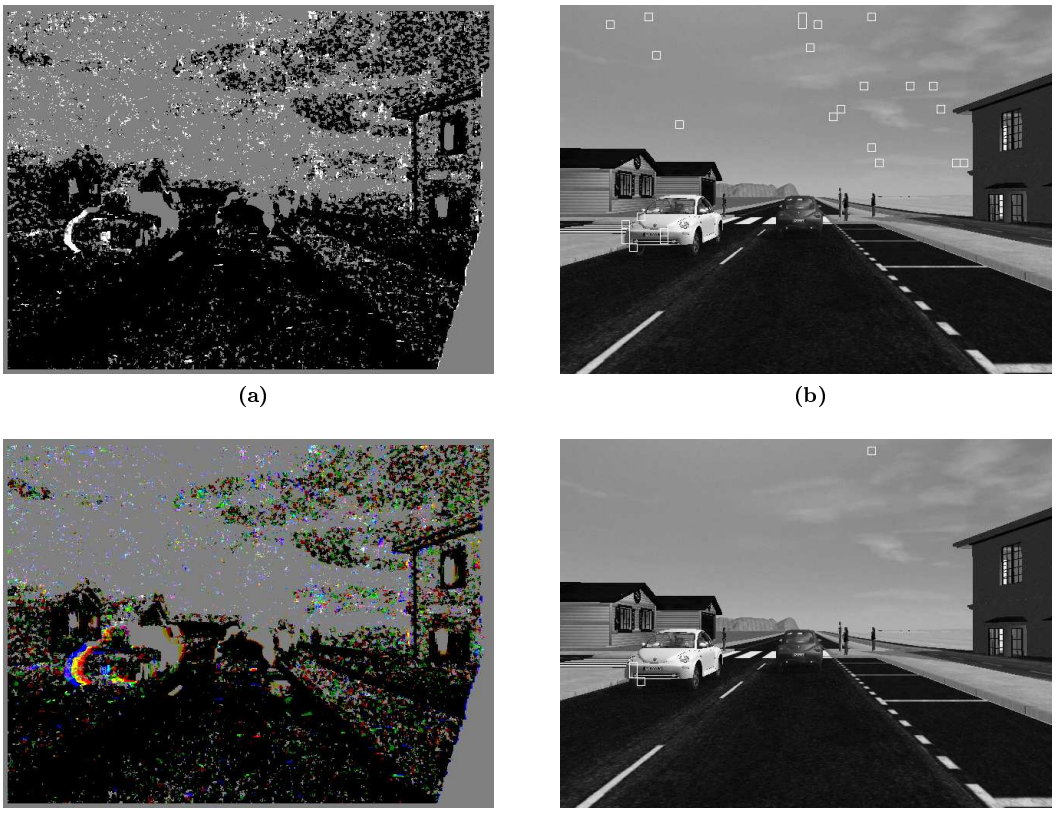
Un deuxième aspect concernant les images de disparité est l'ampleur des variations attendues. Les valeurs de disparité sont en $\frac{1}{Z_M}$ (précisément $\delta_m = \frac{fb_s}{Z_M}$), où Z_M est la coordonnée selon l'axe perpendiculaire au plan image (cas d'images rectifiées, i.e. de plan image parallèle). Ainsi, plus l'objet est proche et se déplace selon l'axe Z, plus la variation de valeur de disparité sera importante : soit une voiture située à une distance de $10m$ de la caméra, et qui se déplace dans la direction opposée à la caméra (enregistrement vidéo à $30\,images/s$) à une vitesse inférieure à $50\,km/h$; supposons par ailleurs que la *baseline*, distance entre les deux caméras stéréoscopiques, est $b_s = 0.5\,m$ (comme pour nos données simulées), la variation des valeurs de la disparité des pixels à l'intérieur de la voiture est inférieure à 2-3 pixels. Or, les valeurs de disparité de l'image à l'extérieur des objets varient entre 0 et 100 pixels selon leur distance Z_M par rapport à la caméra. Ainsi, les changements détectés, dans les valeurs de disparité, seront principalement des changements dus à des occultations/désoccultations du fond de la scène (ou d'objets plus loin selon Z), i.e. principalement au niveau des frontières des objets.

Un troisième aspect concernant l'utilisation des images de disparité est la dépendance à leur qualité (nous y reviendrons dans le cas des données réelles testées). Dans les images de la séquence Pro-SiVIC dont la figure 2.18 est un extrait, les cartes de disparité sont très 'propres', i.e. elles ne comportent que très peu d'erreurs : notamment au niveau du ciel les valeurs de disparité sont absentes plutôt que d'être erronées (valeurs en gris sur la figure 2.18, 1ère colonne). La figure 2.19 présente un cas moins favorable obtenu à partir de l'ancienne version de Pro-SiVIC. Des erreurs notamment au niveau du ciel entraînent des fausses détections sur l'image des points. On observe alors l'apport de l'approche multitemporelle concernant ces fausses détections. Dans la scène, les objets à détecter sont les deux voitures (ayant un mouvement propre), contrairement aux piétons (immobiles) et aux bâtiments (fixes) qui sont statiques (et n'ont pas à être détectés avec le *modèle à* $t \equiv$ projection de l'image à $t-1$ utilisé pour cet exemple). La voiture dans la voie d'en face est partiellement détectée sur les bords : comme annoncé précédemment, les variations de disparité situées au bord de l'objet et dues à des phénomènes d'occultation/désoccultation priment sur les variations de l'objet selon la coordonnée en Z. Notons ainsi que la voiture roulant devant la caméra n'est pas elle détectée, car, se déplaçant à une vitesse proche du mouvement de la caméra, elle est projetée dans l'image suivante sur elle-même, et donc il n'y pas de phénomène d'occultation/désoccultation.

En conclusion, l'utilisation de la disparité couplée avec un *modèle à* t qui est l'image précédente (projetée) est robuste et plus sujette aux non détections (de part les faibles variations de disparité hors des phénomènes d'occultation/désoccultation) à condition que les cartes de disparité soient de qualité suffisante (quitte à être plus incomplètes). Dans le cas d'un bruit sur les valeurs de disparité, l'ajout d'une dimension multitemporelle agit comme un filtre multitemporel et permet de réduire le nombre des fausses détections de manière significative pourvu que ce bruit soit temporellement décorrélé. Soulignons tout de même que dans le cas où l'objet se déplace suffisamment vite pour qu'il n'y ait pas recouvrement entre sa position aux instants successifs, le filtre multitemporel peut conduire à non détections. Un cas typique que nous avons rencontré est celui d'un piéton qui se traverse juste devant la voiture à l'arrêt.

2.3.3.3 Classification des objets selon les labels *mobile* ou *statique*

Nous avons vu précédemment, que l'algorithme de détection par le *modèle à* t correspondant à la scène vierge sauf la route, et celui correspondant à la projection de l'image à $t-1$ vers t étaient complémentaires. Nous proposons de combiner ces deux algorithmes pour affiner le résultat par chacun d'eux en distinguant les objets statiques, i.e. $\in \{Obj_{Statique}\}$, ou mobiles, i.e. $\in \{Obj_{Mobile}\}$. En effet



(a) (b)

(c) (d)

FIGURE 2.19: *Exemple de résultat dans un cas moins favorable en terme de qualité des cartes de disparité.* les données en entrée sont des images de disparité et le *background* est l'image (de disparité) à $t-1$ (projetée à t) : $1^{\text{ère}}$ ligne : cas monotemporel ; $2^{\text{ème}}$ ligne : cas multitemporel ; $1^{\text{ère}}$ colonne : images de points (critère NFA_1) ; $2^{\text{ème}}$ colonne : fenêtres significatives (critère NFA_2, taille 10×10 à 20×20) sur fond d'image radiométrique.

très simplement, le *modèle à t* correspondant à la scène vierge sauf la route, fournit tous les objets statiques ou mobiles (hors du plan de la route), tandis que le *modèle à t* correspondant à la projection de l'image à $t-1$ ne fournit que les objets ayant un mouvement propre (plus quelques fausses détections), soit $\{Obj_{Change}\}$ l'ensemble de ces objets. Ainsi, nous utilisons ce deuxième modèle pour identifier les objets en mouvement, $\{Obj_{Mobile}\}$, parmi les objets extraits à partir du premier modèle qui identifie $\{Obj_{\neg Route}\}$, soit :

$\{Obj_{Mobile}\} = \{Obj_{Change}\} \cap \{Obj_{\neg Route}\}$, $\{Obj_{Statique}\} = \overline{\{Obj_{Change}\}} \cap \{Obj_{\neg Route}\}$ et $\{FP\} = \{Obj_{Change}\} \cap \overline{\{Obj_{\neg Route}\}}$.

Alors, l'algorithme de classification est simplement :

Algorithm 2.5 Algorithme de classification.

- Pour chaque objet $O_i \in \{Obj_{Change}\}$,
  - Chercher l'objet $O_{j \leftrightarrow i} \in \{Obj_{\neg Route}\}$ lui correspondant : $O_{j \leftrightarrow i} = \underset{O_j \in \{Obj_{\neg Route}\}}{argmax} |O_j \cap O_i|$;
  - Si $O_{j \leftrightarrow i} \cap O_i \neq \emptyset$,
    - $(O_{j \leftrightarrow i} \cup O_i) \in \{Obj_{Mobile}\}$;
  - Sinon,
    - $O_i \in \{FP\}$;
  - Fin test ;
- Fin boucle ;
- Pour chaque objet $O_j \in \{Obj_{\neg Route}\} \cap \overline{\{Obj_{Mobile}\}}$,
  - $O_i \in \{Obj_{Statique}\}$;
- Fin boucle.

Notons que dans l'algorithme 2.5, nous manipulons des objets et non plus des fenêtres. Ces derniers correspondent simplement à des régions définies comme des ensembles de fenêtres connexes.

La figure 2.20 présente un exemple de résultat de classification. Pour l'estimation de $\{Obj_{Change}\}$, nous pouvons nous servir soit des images radiométriques, soit des images de disparité. Les résultats correspondant à cette alternative sont présentés sur la figure 2.20 et nous constatons qu'ils sont très proches. Dans les deux cas, les maisons et les piétons sont bien classés comme des objets statiques. La voiture blanche, roulant en sens opposé de la caméra, est bien classée comme un objet en mouvement. La voiture grise, roulant dans le même sens et à vitesse similaire par rapport à la caméra, est mal classée, comme un objet statique, car elle n'est pas détectée parmi les objets $\{Obj_{Change}\}$ que ce soit en utilisant la disparité ou en utilisant les radiométrie.

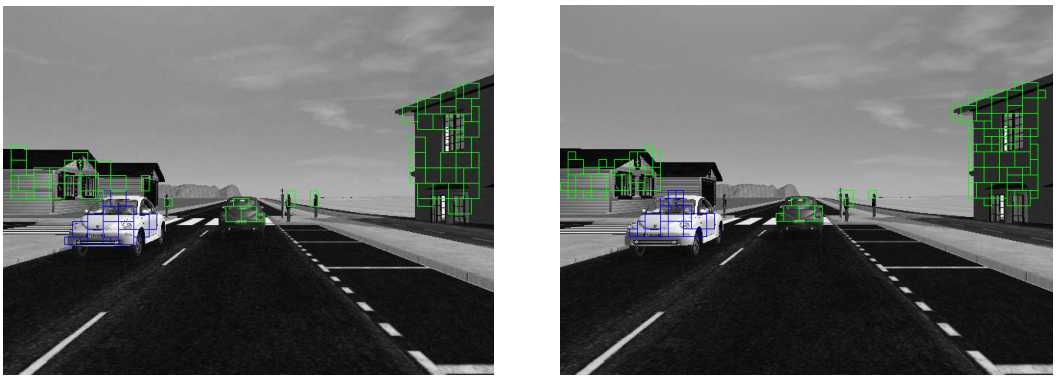
2.3.3.4 Résultats

L'objectif dans cette section est de valider l'approche proposée non sur quelques images mais sur l'ensemble d'une séquence vidéo. Pour effectuer une validation quantitative, il est nécessaire de disposer d'une vérité-terrain. Pour cela nous utilisons le simulateur Pro-SiVIC (de la société Civitech).

Calcul de la vérité-terrain de la séquence Pro-SiVIC

La vérité-terrain est calculée de la façon suivante. Le simulateur fournit des informations sur certains objets de la scène 3D, via différents observateurs en distinguant :

1. objets mobiles (e.g. piétons),
2. objets statiques (e.g. bâtiments, voitures immobiles),
3. voitures mobiles (voiture principale qui contient les caméras embarquées, ou autres voitures qui roulent).



(a) (b)

FIGURE 2.20: *Exemple de classification (algorithme 2.5) exploitant les deux* modèles à t *pour séparer les objets mobiles (en bleu) des objets statiques (en vert).*
(a) cas où les objets mobiles sont estimés à partir des images radiométriques, (b) cas où les objets mobiles sont estimés à partir des images de disparité.

Les informations accessibles concernant les objets sont les orientations angulaires et la position de chaque objet (position du centre de gravité), par rapport au référentiel absolu R_a. La vérité terrain s'obtient alors en trois étapes.

$1^{\text{ère}}$ étape : Exprimer les coordonnées des objets dans le repère R_{cam} d'une des deux caméras (droite dans notre cas) en prenant en compte la position (à l'instant t) du repère lié à la caméra (mobile) par rapport au repère absolu R_a. Les angles (α, β, γ) fournis par Pro-SiVIC sont ceux des quaternions qui représente l'orientation 3D des objets par rapport au repère absolu.

$2^{\text{ème}}$ étape : Pour chaque objet de la scène, après la détermination des coordonnées des huit sommets du parallélépipède englobant chaque objet dans le repère lié à caméra, les huit sommets sont projetés sur l'image sous l'hypothèse du modèle *pinhole*.

$3^{\text{ème}}$ étape : Pour chaque objet de la scène, le rectangle englobant des huit points projetés est déterminé.

Performance estimée sur la séquence simulée sous Pro-SiVIC

Nous présentons à présent des résultats sur l'ensemble de la séquence simulée. Tout d'abord précisons la façon dont sont définis les indicateurs de sensibilité (S_e) et de spécificité (S_p) :

$$S_e = \frac{VP}{(VP+FN)}, \quad S_p = \frac{VP}{(VP+FP)}, \tag{2.23}$$

avec VP le nombre de vrais positifs, FP le nombre de faux positifs, et FN le nombre de faux négatifs. Notre vérité terrain n'est connue qu'au niveau des boites englobantes des objets. Aussi pour ne pas pénaliser nos résultats, où les objets sont représentés (partiellement) par des groupes de fenêtres connexes, les paramètres VP, FP, et FN sont calculés comme suit.

Pour chaque image I_t de la séquence, FN est le nombre d'objets de la vérité-terrain projetés sur I_t ayant une intersection vide avec l'union des fenêtres détectées sur I_t, FP est le nombre de groupes de fenêtres connexes détectés sur I_t ayant une intersection vide avec l'union des objets de la vérité-terrain projetés sur I_t, et VP est le nombre de groupes de fenêtres connexes détectés sur I_t ayant une intersection non vide avec l'union des objets de la vérité-terrain projetés sur I_t.

La figure 2.21 montre les résultats obtenus sur la séquence lorsqu'on combine le modèle *scène vierge sauf route* avec le modèle *image radiométrique à* $t-1$ (projetée à t). La partie supérieure

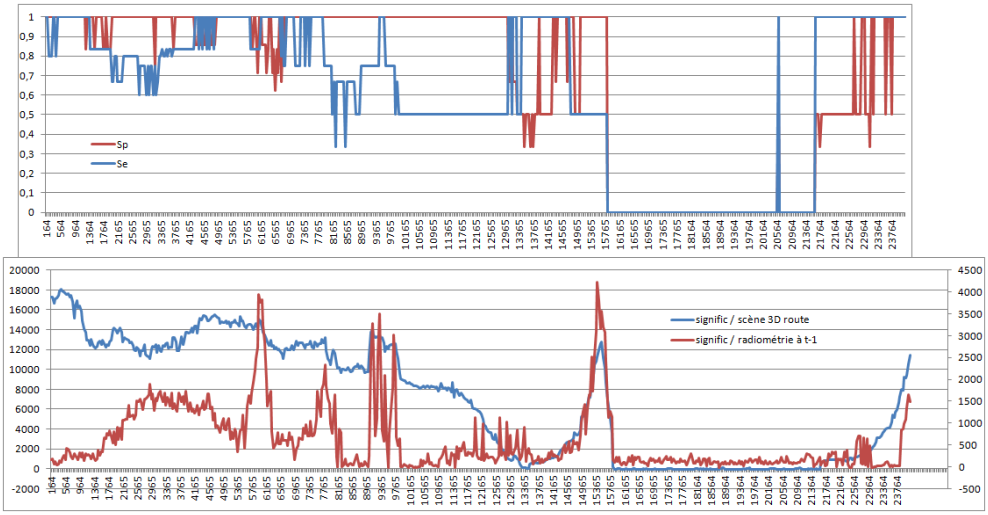


FIGURE 2.21: *Résultats obtenus sur la séquence par combinaison de la détection monotemporelle par rapport au modèle* scène vierge sauf route *et la détection monotemporelle par rapport au modèle* image radiométrique à $t-1$ *(projetée à t).*
en haut, S_e et S_p en fonction de l'échantillon temporel sur la séquence (*timestamp*) ; en bas, $-log\,(NFA_2)$ (significativité) en fonction du *timestamp*).

de la figure donne les valeurs de S_e et S_p en fonction de l'échantillon temporel sur la séquence (t). On note que la spécificité est globalement meilleure que la sensibilité. Rappelons que la spécificité mesure la probabilité que l'objet détecté soit effectivement un objet, tandis que la sensibilité mesure la probabilité qu'un objet soit effectivement détecté. Une valeur de S_p élevée correspond donc à très peu de fausses détections, et une valeur de S_p élevée correspond donc à très peu de non détections. Maintenant, le nombre d'objets réels dans les images de la séquence varie au cours du temps. En particulier, à partir de l'échantillon temporel 15765, la voiture embarquant la caméra a dépassé les quelques maisons au bord de la route, et il n'y a plus qu'un objet qui est une voiture au loin qui roule en sens inverse. Cette voiture qui s'approche de la caméra est détectée ponctuellement à l'échantillon temporel 20565, puis à partir de l'échantillon temporel 21565, avec cependant une fausse alarme (ce qui induit $S_p = 0.5$). La partie inférieure de la figure donne les valeurs de significativité (i.e. $-log\,(NFA_2)$) obtenues (pour chacun des deux modèles). On note que la significativité dans cet intervalle de temps est très faible, en particulier par rapport au modèle *scène vierge sauf route*. Dans cette étude nous n'avons pas voulu ajuster de seuil sur la significativité globale (hormis 0) car nous voulions rester avec une méthode sans paramètre. Rappelons également que les deux détections par rapport aux modèles respectifs ne sont pas prises en compte de façon symétrique : les détections par rapport au modèle *scène vierge sauf route* correspondent à des détections sur le résultat final (aucune fausse alarme ne sera éliminée) tandis que les détections par rapport au modèle *image radiométrique à* $t-1$ permettent la labelisation des détections sur le résultat final (une détection non reconnue, au moins partiellement, parmi les détections finales, i.e. les détections par rapport au modèle *scène vierge sauf route*, sera éliminée comme une fausse alarme).

La figure 2.22 présente des résultats similaires à ceux de la figure 2.21 excepté qu'on considère le cas multitemporel. Les figures 2.23 et 2.24 sont le pendant des deux figures précédentes dans le cas où les données par rapport auxquelles la détection est faite sont d'une part la carte de disparité de la scène 3D vierge sauf la route et d'autre part l'image de disparité projetée de $t-1$ vers t. Les commentaires précédents restent valables.

Le tableau 2.2 présente les résultats globaux en terme de S_e et S_p. A l'image des figures 2.21, 2.23,

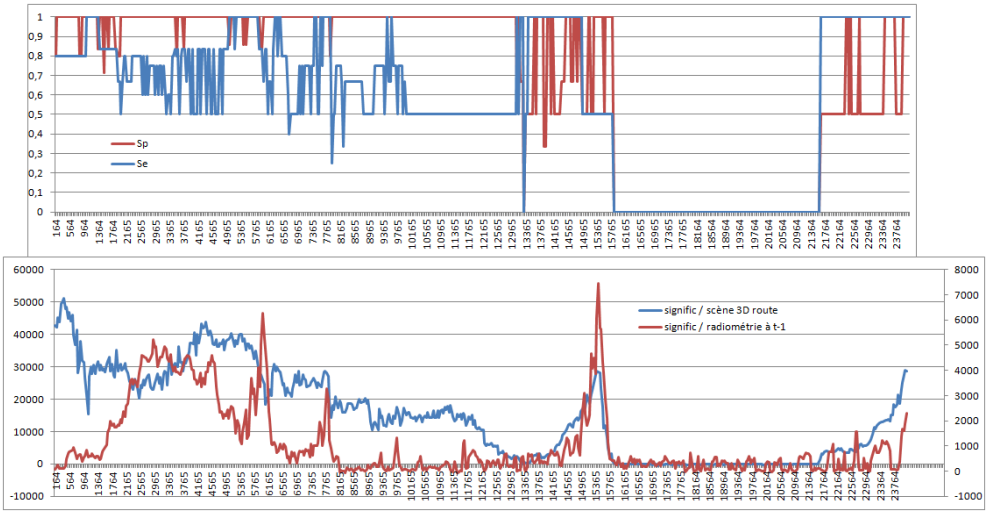


FIGURE 2.22: *Résultats obtenus sur la séquence par combinaison de la détection multitemporelle par rapport au modèle* scène vierge sauf route *et la détection multitemporelle par rapport au modèle* image radiométrique à $t-1$ *(projetée à t).*
en haut, S_e et S_p en fonction de l'échantillon temporel sur la séquence (*timestamp*) ; en bas, $-log\,(NFA_2)$ (significativité) en fonction du *timestamp*).

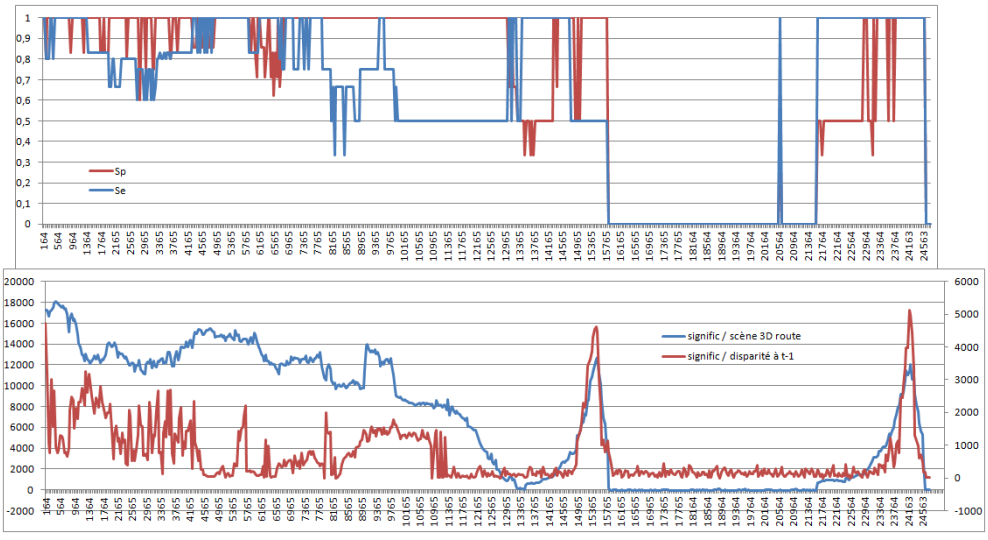


FIGURE 2.23: *Résultats obtenus sur la séquence par combinaison de la détection multitemporelle par rapport au modèle* scène vierge sauf route *et la détection multitemporelle par rapport au modèle* image de disparité à $t-1$ *(projetée à t).*
en haut, S_e et S_p en fonction de l'échantillon temporel sur la séquence (*timestamp*) ; en bas, $-log\,(NFA_2)$ (significativité) en fonction du *timestamp*).

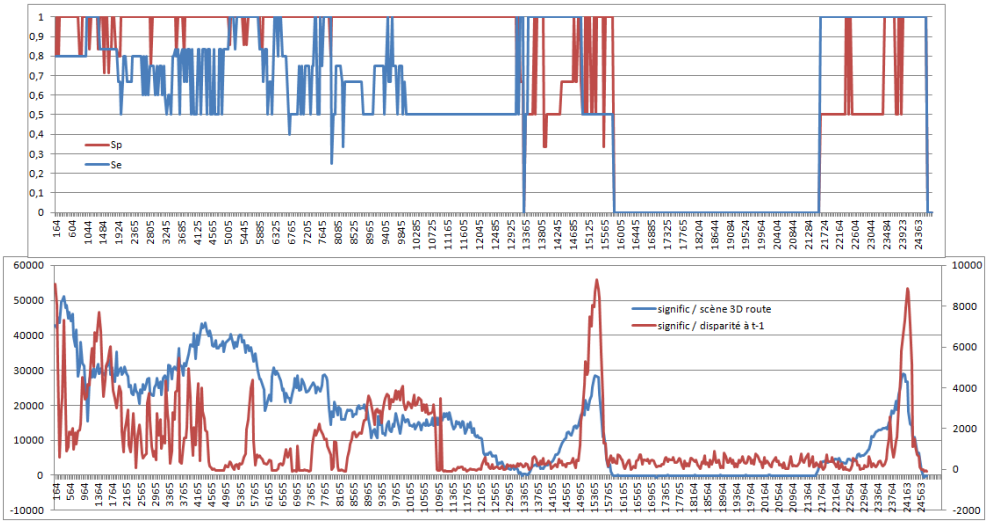


FIGURE 2.24: *Résultats obtenus sur la séquence par combinaison de la détection multitemporelle par rapport au modèle* scène vierge sauf route *et la détection multitemporelle par rapport au modèle* image de disparité à $t-1$ *(projetée à t).*
en haut, S_e et S_p en fonction de l'échantillon temporel sur la séquence (*timestamp*) ; en bas, $-log\,(NFA_2)$ (significativité) en fonction du *timestamp*).

	S_e	S_p
Carte de disparité de la scène 3D vierge + Image radiométrique projetée	0.754 (monotemporel) 0.683 (multitemporel)	0.897 (monotemporel) 0.914 (multitemporel)
Carte de disparité de la scène 3D vierge + Image de disparité projetée	0.755 (monotemporel) 0.684 (multitemporel)	0.892 (monotemporel) 0.904 (multitemporel)

TABLE 2.2: *Statistiques globales sur la séquence en fonction de des données en entrée de la méthode de détection.*

2.22, et 2.24, on constate que les résultats sont très proches.

Application à des données réelles

Dans ce paragraphe, on présente quelques exemples de résultats obtenus dans le cas de données réelles. Ces données ont été acquises dans le cadre d'un projet de recherche, LOVe pour Logiciel d'Observation des Vulnérables qui réunissait plusieurs partenaires académiques (du CNRS à savoir LASMEA/Université de Clermont-Ferrand, Heudiasyc/Université de Compiègne, IEF/Université Paris Sud, du CEA, de l'INRIA, du LIVIC, et Armines) ainsi que des partenaires industriels (Renault, Valeo). Nous citons la description du projet : "LOVe [3] propose de contribuer à la sécurité routière en mettant principalement l'accent sur la sécurité des piétons. L'objectif est d'aboutir à des logiciels embarqués d'observation des vulnérables fiables et sûrs rapidement implantables sur la gamme. A dessein, une démarche de conception industrielle de ces logiciels a été adoptée afin de spécifier clairement les contraintes techniques (capteurs, informatique), contextuels (scenarii envisagés) et méthodes de validation des algorithmes. LOVe est riche d'une multitude d'approches exploitant des données laser, mono et stéréovision tant au niveau de la détection, de la localisation, de la reconnaissance que du suivi des piétons. Ces solutions peuvent ensuite être combinées à différents niveaux par des algorithmes de fusion de données afin d'accroître précision de localisation et certitude dans les évènements. L'en-

3. http ://www.pole-moveo.org/pdf-projets-das/Love-F.pdf

semble des modules est évalué afin de retenir les chaînes de traitement les plus pertinentes parmi le foisonnement de solutions possibles."

Comparaison méthode de disparité

Pour conclure notre brève analyse qualitative des résultats sur données réelles, nous présentons un exemple de comparaison avec les résultats obtenus par uv-disparité (cf. section 2.18).

Il est certainement difficile de tirer des conclusions générales de cette comparaison. On constate simplement que les fausses alarmes détectées sur la route ne le sont pas par uv-disparité car justement appartenant au plan de la route et que pour les autres objets, la détection/classification proposée a tendance à moins détecter (ce qui rejoins le commentaire sur $S_p > S_e$), mais peut-être plus précisément. Le bas des véhicules (ou même le panneau avec la flèche vers le milieu gauche de l'image) n'est pas détecté car trop proche du plan de la route pour l'écart des valeurs de disparité soit significatif. Il est clair que l'algorithme de uv-disparité, en reconstituant la composante connexe jusqu'au plan de la route (hypothèse que l'obstacle touche le plan de la route) n'a pas ce défaut. Il est également clair qu'un post traitement dans ce sens pourrait largement améliorer nos résultats.

2.4 Conclusion

Nous avons proposé une méthode pour la détection d'objets 'inattendus'. L'intérêt majeur de cette approche est d'être fondée mathématiquement et sans paramètre. L'algorithme proposé a été validé sur des données simulées et des premiers tests sur des données réelles ont pu être effectués.

Les travaux doivent être poursuivis dans au moins deux directions afin de rendre la méthode opérationnelle :

- Les résultats doivent être améliorés en enrichissant le *modèle à t*. Notamment, la méthode devra bénéficier des travaux de la communauté pour robustifier la détection par rapport à une image de fond (actualisation du background par filtrage sigma-delta ou CodeBook). Une étape de suivi d'objet pourrait également être mise en place.
- Ces enrichissements vont nécessiter des ressources en calcul supplémentaires. Aussi une optimisation de code sera-t-elle certainement nécessaire.

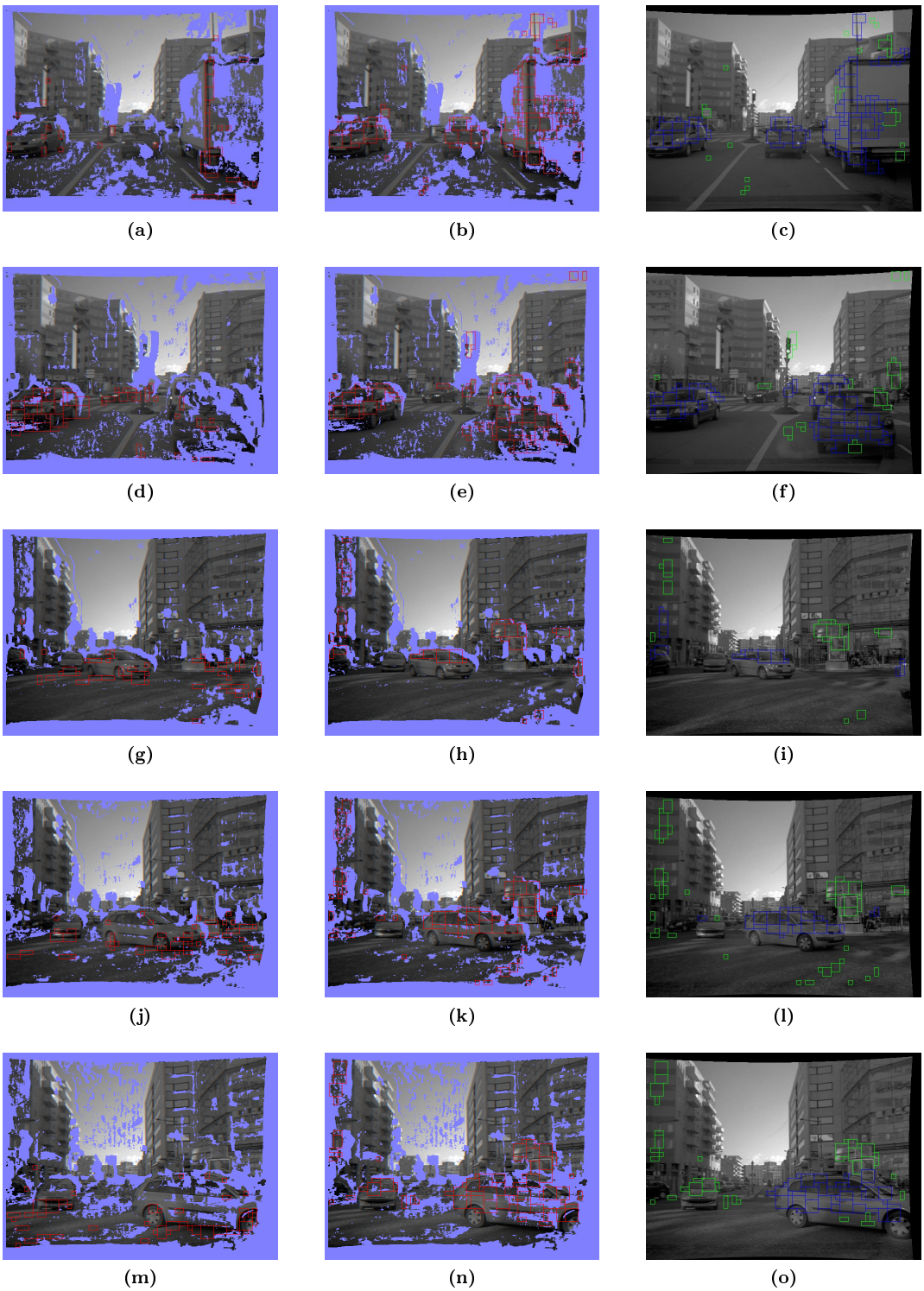
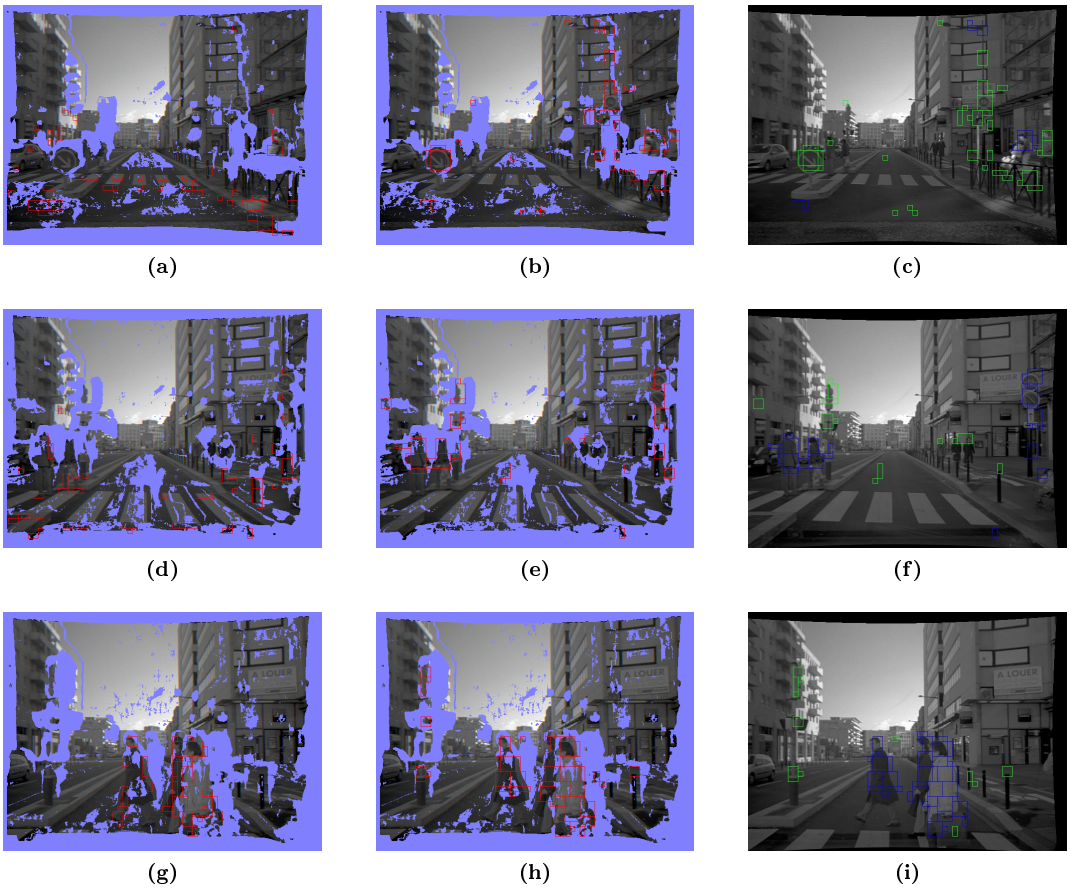


FIGURE 2.25: *Premiers exemples de résultats sur les données réelles.*
(colonne de gauche) détections par rapport au modèle *image radiométrique à* $t-1$ (en mauve le sous-domaine image des pixels pour lesquels la disparité n'a pu être calculée) ; (colonne centrale) détections par rapport au modèle *scène vierge sauf route* ; (colonne de droite) résultat de la classification : en bleu les objets mobiles et en vert les objets statiques.

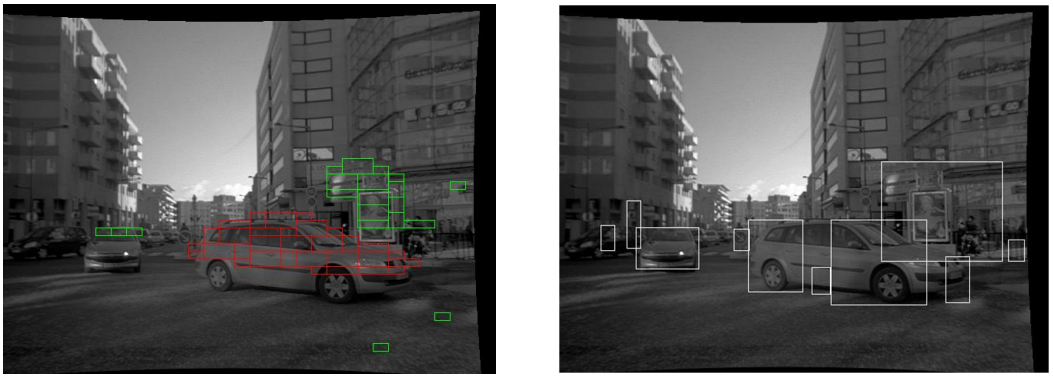


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 2.26: *Seconds exemples de résultats sur les données réelles.*
(colonne de gauche) détections par rapport au modèle *image radiométrique à* $t-1$ (en mauve le sous-domaine image des pixels pour lesquels la disparité n'a pu être calculée) ; (colonne centrale) détections par rapport au modèle *scène vierge sauf route* ; (colonne de droite) résultat de la classification : en bleu les objets mobiles et en vert les objets statiques.



(a) (b)

FIGURE 2.27: *Exemples de comparaison avec la* uv*-disparité.*
(a) résultat de la classification : en rouge les objets mobiles, et en vert les objets statiques ; (b) : résultat de la *uv*-disparité.

Chapitre 3

Signalisation routière

Sommaire

3.1 Introduction 64
3.2 Détection des panneaux routiers 64
3.3 Détection des marquages au sol 65
3.3.1 Approche proposée 68
3.3.2 Prétraitement des images 69
3.3.3 Classification niveau pixel 69
3.3.4 Classification niveau composante connexe 71
3.3.5 Classification niveau objet 72
3.3.6 Résultats et discussion 73
3.3.7 Conclusion 78
3.4 Conclusion 79

3.1 Introduction

Dans l'introduction nous avions évoqué deux grandes classes d'objets d'intérêt pour un véhicule doté d'autonomie : les objets 'inattendus' (sur la détection desquels la section précédente s'est focalisée) et les objets normalisés (ou parfaitement connus et attendus) liés à l'application de circulation automobile, notamment la signalisation routière. La détection et la reconnaissance de ces derniers a fait l'objet de nombreuses études. Notre étude dans le cadre de cette thèse a pour intérêt principal de mettre en avant la dualité avec l'approche précédente, qui elle n'utilisait pas d'*a priori* sur les objets puisque par définition ils étaient 'inattendus'.

La forme et la taille des symboles de la signalisation routière sont normalisées pour chaque pays (tout en pouvant varier d'un pays à un autre). Nous distinguons deux types de signalisation routière : les marquages verticaux (comme les feux de circulation et les panneaux routiers) et les marquages horizontaux dessinés sur la route (tels que les marques de bords de la route, les lignes de séparation de voie, continues ou discontinues, zébra, passages piétons, places de parking). Dans ce qui suit nous présentons un état de l'art des méthodes de détection de la signalisation verticale. Ensuite nous présentons une étude de détection et d'identification de marquages au sol afin de définir l'environnement contextuel de la route, c'est-à-dire que nous déduirons de la signalisation routière le type d'environnement extérieur (urbain, autoroute, zone piétonne, etc.).

Précisons également que nous nous plaçons dans ce chapitre dans le cas monovision, de façon à nous affranchir de la nécessité d'avoir un système stéréoscopique.

3.2 Détection des panneaux routiers

Quand les différentes formes d'un objet recherché sont énumérables et connues, une approche classique est la détection basée sur la reconnaissance (à l'opposé des approches *a contrario*). Généralement, il y a une étape de détection préalable permettant de se concentrer sur une région d'intérêt de l'image où l'objet est susceptible d'être présent, suivie d'une étape de reconnaissance pour valider ou non la présence de l'objet recherché dans la région d'intérêt. Cette étape est basée classiquement sur l'identification des caractéristiques spécifiques des objets recherchés, comme la forme et la couleur. Si la variabilité de ces caractéristiques est relativement faible alors il est possible de faire appel à un dictionnaire (vase de données) cataloguant ces variations. Les panneaux de signalisation routière sont normalisés et énumérables. Ainsi, le type de détection par reconnaissance mentionné précédemment s'applique-t-il.

La couleur et la forme géométrique sont les premières caractéristiques des panneaux (dans la suite on utilise ce terme pour désigner la signalisation routière verticale). Ainsi, la plupart des méthodes existantes utilisent une segmentation basée sur la couleur (e.g. [106, 34, 55]). Bien qu'il existe d'autres types de panneaux avec fond jaune pour les travaux, ou bleu pour une ensemble de panneaux d'obligation et d'information, la plupart des panneaux sont soit bleus, soit rouges, soit noirs, sur fond blanc. Maintenant si un avantage des méthodes exploitant l'*a priori* sur la couleur est leur rapidité, leur inconvénient principal est un manque de robustesse par rapport aux variations des conditions d'éclairage, malgré l'utilisation de différents espaces colorimétrique, tels que l'espace RGB, les ratios de RGB, ou, le plus souvent [106], l'espace HST (Intensité, Saturation, Teinte), ou HSV.

La seconde caractéristique des panneaux est leur forme géométrique régulière (cercle, triangle ou rectangle) et d'une taille semblable. On distingue les travaux qui utilisent cette caractéristique en complément de la couleur pour robustifier la détection/reconnaissance. Dans le premier cas, généralement la segmentation couleur est suivie soit par une méthode d'une détection de contours [106, 55],

soit par une méthode de détection de coins [34]. Dans [46], les deux caractéristiques sont considérées de manière équilibrée puisque les auteurs proposent de construire deux réseaux de neurones séparés, pour extraire respectivement les couleurs et les formes géométriques apprises sur les et associées aux panneaux. Notons que les réseaux de neurones paraissent pertinents dans ce cas où il est possible de constituer une base d'apprentissage consistante, cependant les auteurs signalent que le temps de calcul reste important.

Afin de s'affranchir des problèmes d'éclairage, et en contre-pied des méthodes qui mettent la caractéristique couleur en avant, plusieurs méthodes (e.g. [5], [57], [43]) de détection ont été proposées sans tenir compte de la caractéristique couleur. La plupart des méthodes utilisent soit des méthodes avec apprentissage (on retrouve des réseaux de neurones, e.g. [43], mais également des classifieurs de type *Adaboost* [115] ou des algorithmes génétiques) ou des transformées, notamment la transformée de Hough sur les cercles (e.g. [96]), ou sur les lignes (e.g. [56]), la transformée en symétrie radiale (e.g. [33]), la transformée en symétrie généralisée (e.g. [111]), ou encore la transformée chinoise (e.g. [10] qui par ailleurs présente un bibliographie intéressante sur l'utilisation de ces transformées pour la détection des panneaux) Par exemple, dans [10], Belaroussi et Tarel proposent d'utiliser l'orientation du gradient pour la détection des panneaux, indépendamment de leur position et de leur orientation : la détection des panneaux circulaires et polygonaux (non triangulaires) est réalisée par transformation chinoise latérale (TSB), et les panneaux triangulaires par la transformation en sommets et bissectrices (TCB). Notons cependant que la non utilisation de la couleur, ne permet pas de distinguer certains panneaux entre eux, aussi une étape de reconnaissance, subséquente à la détection est-elle nécessaire.

Les méthodes de reconnaissance à proprement parler (supposant la détection préalable des panneaux soit sur des critères de couleur, de forme géométrique, ou des critères mixtes), sont souvent des approches de classification supervisée (i.e. avec base d'apprentissage) : réseaux de neurones ([34]), corrélation [32, 106], etc. Par exemple, dans [55], la classification exploite un vecteur de caractéristiques contenant des informations sur les orientations des contours et les densités locales du voisinage qu'elle compare aux vecteurs des images de référence. Dans [119], après avoir détecté les panneaux (par une technique de type RANSAC et selon les caractéristiques couleur, forme générale et texture), la méthode proposée cherche à apparier par corrélation d'images, une région supposée contenir un panneau avec l'ensemble des panneaux de référence de même forme. Cette approche nécessite la constitution d'une base recensant pour chaque panneau son image selon les transformations géométriques susceptibles d'être appliquées à ces panneaux (point de vue).

3.3 Détection des marquages au sol

La plupart des travaux traitant la détection des marquages au sol se sont concentrés sur la détection de la route en se basant sur différents modèles de route [7, 128, 30, 87]. D'après leurs auteurs, ces algorithmes peuvent être perturbés par certains marquages de route non pris en charge par le modèle, tels que les sorties et entrées d'autoroutes, les passages piétons, les marquages zébra. Dans la plupart des cas, ces applications ont pour objet de surveiller la position du véhicule par rapport à la route, mais ne s'intéressent pas au contexte routier.

Dans cette étude, nous visons à la fois la détection et la reconnaissance du type de marquage au sol. La détection ou la reconnaissance des panneaux routiers (marquages routiers verticaux) est généralement effectuée sur la base des caractéristiques géométriques et radiométriques (forme et couleur). Or, dans notre cas, pour la détection des marquages au sol, la couleur n'est pas discriminante et la forme géométrique n'est pas conservée à cause de la projection du plan de la route vers le plan de l'image (à l'inverse du cas des panneaux routiers qui sont dans un plan vertical perpendiculaire ou

quasi-perpendiculaire à l'axe de la route, donc celui de la caméra si le véhicule est orienté parallèlement à la route...).

Citons cependant le cas de l'imagerie satellitaire à haute résolution (pixels de taille inférieure à 0.25 m^2) qui n'est pas affectée par ces problème de projection. Ainsi, dans [124], Tournaire et al. proposent de détecter les marquages à partir de ce type d'images selon une approche stochastique utilisant un processus ponctuel marqué (PPM) qui modélise les objets dans l'image ainsi que les interactions qu'ils présentent les uns avec les autres. L'utilisation de ces processus permet de s'affranchir du problème de l'occlusion des bandes par des véhicules ou des ombres projetées. En effet, pour les problèmes de détection et de reconnaissance, outre les approches méta-heuristiques (telles que les réseaux de neurones, la classification *Adaboost*, etc.) qui nécessitent de grandes bases de données d'apprentissage, les modèles probabilistes tels que les champs de Markov (*Markov Random Fields*, MRF, selon la terminologie anglaise), dont les PPM sont une forme d'extension au cas de graphes avec un nombre variable de nœuds, permettent d'introduire directement l'information *a priori* sur la classe ou les objets d'intérêt et leurs interactions spatiales. Dans cette étude, nous proposons d'utiliser cette modélisation à différents niveaux d'abstraction : du pixel à l'objet.

Rappelons brièvement le formalisme des champs de Markov en classification d'images. Dans [59], le problème de classification est formalisé dans un cadre probabiliste, comme suit :

Problème 5. Soit deux champs aléatoires, à savoir : le champ des labels (classes) X, et le champ des observations Y. Nous cherchons à estimer la réalisation x de X à partir de l'observation de la réalisation de y de Y.

Par exemple, dans le cas d'une classification d'une image de niveau de gris, X est le champ des étiquettes de classes (labels) des pixels, et Y le champ des valeurs de niveaux de gris des pixels. Pour le cas d'une classification d'objets, X est le champ des labels ou des classes des objets dans l'image, et Y est le champ des caractéristiques de ces objets, qui peuvent être radiométriques, géométriques, topologiques, etc... .

Suivant le critère du maximum *a posteriori* (MAP), l'estimateur optimal est celui qui maximise la probabilité *a posteriori* du champ x connaissant la réalisation y, notée $P(x/y)$:

$$\hat{x} = \underset{x}{argmax} P\left(x/y\right). \tag{3.1}$$

Or pour la maximisation, $P(y)$ est une constante, et par ailleurs la loi conditionnelle qui est connue (modélisée) est généralement celle d'une observation conditionnellement à la classe (i.e. les statistiques des classes sont supposées connues). Aussi préfère-t-on souvent écrire :

$$\hat{x} = \underset{x}{argmax} P\left(y/x\right) P\left(x\right). \tag{3.2}$$

Ainsi, nous devons modéliser deux probabilités, à savoir : la probabilité conditionnelle de l'observation et celle *a priori* de la classe.

L'hypothèse markovienne intervient dans la modélisation de la probabilité *a priori*. Elle permet d'introduire des informations concernant les interactions entre les éléments (pixels ou objets) voisins et d'en déduire la probabilité *a priori* qui ne dépend plus alors que d'interactions locales (et non sur l'ensemble du champ). Ces interactions sont représentées sous forme d'un graphe dont les nœuds représentent les éléments du champ aléatoire (e.g. pixels ou objets) considéré, et les arêtes du graphe représentent les relations de voisinage entre les éléments (l'hypothèse markovienne revient alors à restreindre les arêtes du graphe à celles entre éléments voisins).

Pour un système de voisinage donné, un système de clique est défini, tel qu'une clique est un sous-graphe complet du voisinage. Les interactions entre pixels voisins s'expriment via une valeur de potentiel de clique, valeur qui dépend de la probabilité de la configuration des cliques. Pratiquement, les configurations à favoriser (selon nos connaissances *a priori*) sont associées à de faibles valeurs de potentiel de clique, et les configurations à défavoriser sont associées à de fortes valeurs de potentiel.

En utilisant le théorème d'Hammersley-Clifford qui établit l'équivalence entre un champ de Markov et un champ de Gibbs, le problème de la classification devient un problème de minimisation d'une fonction d'énergie globale composée de deux termes d'énergie. Le premier terme, noté U_0, provient de la probabilité conditionnelle des observations conditionnellement aux classes et le deuxième terme, noté U, provient de la probabilité *a priori*. Formellement : soit $\mathcal{G}$ l'ensemble des nœuds d'un graphe, et $\mathcal{C}$ l'ensemble des cliques. $\mathcal{C}$ est l'ensemble des sous-ensembles de $\mathcal{G}$ tels que tous les nœuds d'un élément de $\mathcal{C}$ soient connectés sur le graphe (i.e. voisins entre eux).

En supposant l'indépendance conditionnelle, la probabilité du champ des observations conditionnellement au champ des labels, $P(Y = y/X = x)$, est le produit, sur les pixels, des probabilités de l'observation en un pixel conditionnellement à la classe de ce pixel, et donc le terme d'attache aux données U_0 est la somme des énergies associées aux nœuds du graphe :

$$U_0 = \sum_{v \in \mathcal{G}} U_0(v) \tag{3.3}$$

et l'énergie U est la somme des potentiels des cliques (entre nœuds voisins) :

$$U = \sum_{c \in \mathcal{C}} U(c). \tag{3.4}$$

Par suite, l'énergie globale est la somme des deux termes d'énergie précédents :

$$E = U_0 + U = \sum_{v \in \mathcal{G}} U_0(v) + \sum_{c \in \mathcal{C}} U(c). \tag{3.5}$$

Ainsi, les champs de Markov ont été largement utilisés en traitement d'images, car ils fournissent une solution au problème de causalité : sous l'hypothèse Markovienne, $P(x_s \mid x_t, \forall t \in \mathcal{G}) = P(x_s \mid x_t, \forall t/(s,t) \in \mathcal{C})$, la différence d'énergie entre deux configurations x, différant seulement par leur valeur en un pixel s devient traitable informatiquement car elle ne dépend plus que des valeurs des configurations de X dans le voisinage de s. Le recuit simulé est une des techniques d'optimisation qui peut être utilisée pour obtenir la solution globale [60].

Durant les deux dernières décennies, la recherche dans le domaine s'est focalisée (notamment) d'une part sur l'optimisation des algorithmes de minimisation de E et d'autre part sur l'extension de la modélisation markovienne initiale. En effet, l'algorithme de recuit simulé étant très long pour converger (théoriquement temps de calcul infini), certains travaux [76, 18] ont cherché à développer des techniques d'optimisation efficaces, notamment en un nombre fini d'itérations. En particulier, ils ont montré que les méthodes de *max-flot/min-cut* issues de la théorie des graphes pouvaient être utilisées dans le cas de certaines formes d'énergie d'interaction, dont le célèbre modèle de Potts. Cependant, le gain obtenu en terme de temps de calcul a un cout élevé en terme de mémoire, ce qui peut rendre ce type d'approche inefficace quand le nombre de classes est élevé.

D'autres travaux se sont intéressés à des champs de Markov hiérarchiques. Par exemple, Le Hégarat-Mascle et André [83] proposent une détection automatique des nuages basée sur la formalisation des interactions spatiales entre un nuage et son ombre à l'aide d'un champ de Markov à deux niveaux : le

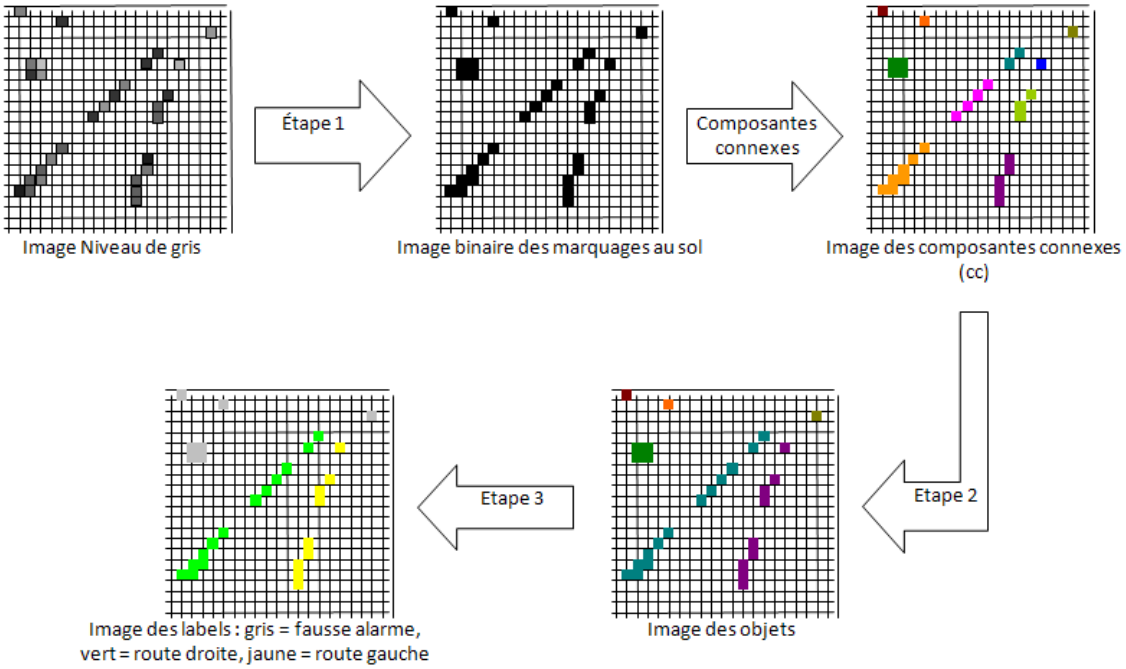


FIGURE 3.1: *Détection des marquages au sol, approche par niveau : du pixel à l'objet*

niveau des pixels pour la détection de régions nuageuses ou de l'ombre, et le niveau objet pour supprimer des faux positifs. Dans [25, 26, 133], Chen et al. proposent des champs de Markov hiérarchiques pour détecter et analyser des objets déformables. Afin de faire face aux différentes configurations d'un objet, soit dues à la position, la taille et l'orientation de l'objet qui varient, soit dues à des occultations de sous-parties de l'objet, ils procèdent par construction hiérarchique à partir de structures élémentaires vers des structures plus complexes. Selon les approches hiérarchiques classiques, les points d'intérêt sont détectés au niveau 0 ou niveau pixel. Ensuite, au niveau 1, ces points d'intérêt créent une structure élémentaire, selon des critères généralement invariants par rotation, translation et homothétie de façon à être identifiable quel que soit le point de vue sur l'objet. Au niveau suivant (niveau 2), ces structures élémentaires sont les nœuds d'un graphe, qui modélise leurs relations, de façon à créer des structures d'un niveau supérieur en complexité. Ainsi de suite, les structures de plus en plus complexes sont assemblées jusqu'à la construction de l'objet entier et sa reconnaissance.

3.3.1 Approche proposée

En s'inspirant des méthodes précédentes, l'approche proposée est de type hiérarchique, avec une différence fondamentale par rapport à l'approche markovienne hiérarchique classique : dans notre cas, les différents niveaux sont sémantiquement différents. Pratiquement, nous effectuons la classification en trois étapes successives correspondant à trois différents niveaux d'abstraction entre le pixel et l'objet. A chaque niveau, nous considérons des domaines différents de Y et X, et différents modèles de probabilité conditionnelle et *a priori*. La figure 3.1 présente les trois étapes successives de traitement associées à chaque niveau d'abstraction, du pixel vers l'objet.

A chaque niveau, nous utilisons un MRF pour modéliser notre connaissance *a priori* sur les caractéristiques des objets recherchés :

1. Au niveau pixel (section 3.3.3), le MRF modélise les interactions entre les pixels voisins. Le graphe est celui des pixels de l'image, et les arcs représentent le système de voisinage 8-connexité.

Le modèle *a priori* fait l'hypothèse que les marquages au sol ont une largeur constante.

2. Au niveau composante connexe (section 3.3.4), le MRF modélise les interactions entre les composants connexes. Le graphe considéré est celui des composantes connexes. Le système de voisinage est défini à partir des directions : soit parallèle soit perpendiculaire à la direction de l'axe principal des composantes connexes. Le potentiel des cliques modélise la connaissance *a priori*, à savoir que les objets recherchés (par exemple les séparations de voies ou les passages pour les piétons) sont formés de composantes connexes alignées.
3. Au niveau objet (section 3.3.5), le MRF modélise les interactions spatiales entre les objets. Le graphe considéré est celui des objets, et le voisinage est défini horizontalement. On suppose alors que la classe des objets peut être estimée à partir de la position (relation spatiale) des objets entre eux, par exemple la ligne droite (resp. gauche) est à droite (resp. gauche) des autres lignes.

Avant de préciser chacun de ces niveaux, nous présentons les données des images et le prétraitement que nous avons appliqué.

3.3.2 Prétraitement des images

Les images traitées sont, soit en niveaux de gris ou soit en couleur RVB. Les marquages au sol sont des marques blanches sur une route généralement de couleur foncée (cas des routes asphaltées). Ainsi, ces marquages se caractérisent par leur haut niveau de contraste. Généralement ils ont une faible largeur. Ainsi, nous proposons la transformation suivante pour aider le traitement de la classification :

Soit I l'image initiale de niveaux de gris (dans le cas des images en couleur, I est l'image d'intensité), et soit ES un élément structurant, suffisamment grand pour qu'une érosion par ES efface toutes les marques de la route sur I. Pratiquement, nous en choisissons ES connaissant la largeur normalisée des marquages (les plus larges, e.g. bandes des passages piétons) et les caractéristiques de la caméra. Soit $\varepsilon_{ES}(I)$ le résultat de l'érosion fonctionnelle de I par ES. Soit $E^I_{ES}(\varepsilon_{ES}(I))$ la reconstruction géodésique de I à partir de $\varepsilon_{ES}(I)$:

$$\left\{ \begin{array}{c} E^I_{ES}(\varepsilon_{ES}(I)) = \sup_{n\geq 0} \left\{ \left[\delta^I_{ES}\right]^n (\varepsilon_{ES}(I)) \right\}, \\ \text{avec } [\delta^I_{ES}]^n(I) = \left\{ \begin{array}{rl} I & \text{si } n = 0, \\ \delta_{ES}\left(\left[\delta^J_{ES}\right]^{n-1} (I) \right) \cap J & \text{si } n > 0, \end{array} \right. \end{array} \right. \tag{3.6}$$

où $\delta_{ES}(I)$ désigne la dilatation fonctionnelle de l'image en niveaux de gris I par l'élément structurant ES.

L'image considérée en entrée de la classification est alors l'image différence entre I et $E^I_{ES}(\varepsilon_{ES}(I))$: $Y = I - E^I_{ES}(\varepsilon_{ES}(I))$.

Un autre prétraitement couramment effectué est de procéder à une correction de l'image de type *birdview*, en connaissant la position approximative du plan de la route, afin de s'assurer que les dimensions des bandes sont constantes dans l'image. Dans le cas d'images stéréoscopiques, cette correction peut se faire en même temps que la rectification. Ici nous présentons le modèle markovien hiérarchique qui est indépendant de ce prétraitement.

3.3.3 Classification niveau pixel

Nous adoptons les notations suivantes : n_{lig} et n_{col} sont respectivement les dimensions en ligne et en colonne de l'image. S est l'ensemble des positions des pixels dans l'image et s l'emplacement d'un pixel donné.

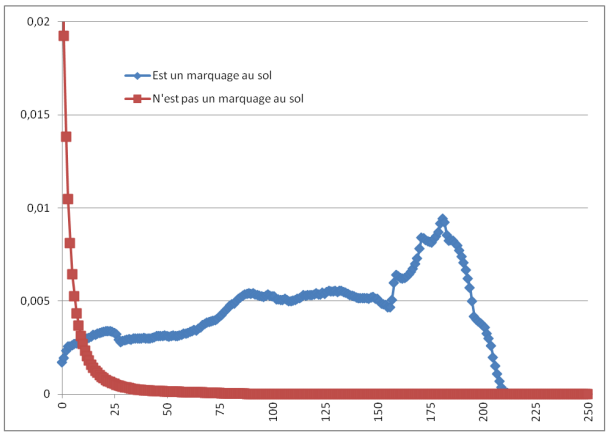


FIGURE 3.2: *Distributions empiriques des classes λ_{mk} et $\lambda_{\overline{mk}}$*

Au niveau pixel, nous considérons deux classes pour les nœuds du graphe associé à S : la classe des marquages routiers (marquages au sol sur la chaussée), notée λ_{mk}, et la classe des pixels n'appartenant pas à un marquage au sol, notée $\lambda_{\overline{mk}}$.

La figure 3.2 montre les deux distributions empiriques des deux classes, marquages routiers et non marquages. Ces deux courbes ont été estimées à partir d'une 'vérité terrain' établie sur huit images.

De façon *ad hoc*, nous modélisons les deux distributions conditionnelles aux classes par deux sigmoïdes. Le terme d'attache aux données U_0, opposé du logarithme de la sigmoïde de la classe donnée, est défini comme suit :

$$\begin{cases} U_0\left(y_s | x_s = \lambda_{mk}\right) = \ln\left(1 + e^{-\alpha(y_s - t_{mk})}\right), \\ U_0\left(y_s | x_s = \lambda_{\overline{mk}}\right) = \ln\left(1 + e^{\alpha(y_s - t_{mk})}\right), \end{cases} \tag{3.7}$$

où t_{mk} est le seuil pour les marquages routiers, $\Delta_{t_{mk}}$ est l'imprécision sur t_{mk} et $\alpha = \frac{10}{\Delta_{t_{mk}}}$. Dans notre cas, le paramètre t_{mk} est déterminé de façon empirique à partir de la moyenne μ_y et l'écart-type σ_y de la réalisation observée y de Y : $t_{mk} = \mu_y + 2\sigma_y$.

Terme de régularisation

Nous supposons *a priori* que les marquages au sol ont une largeur constante, bien que dû à un effet de perspective (en l'absence de projection *birdview*), la largeur du marquage au sol n'est pas exactement constante, mais varie lentement.

Soient i_s et j_s la ligne et la colonne d'un pixel s de S, et $\Delta_{col}(i_s, j_s)$ la longueur horizontale du marquage au sol sur le voisinage de s, défini comme un voisinage vertical d'une ligne. Le terme de régularisation est alors défini comme suit :

$$\begin{cases} l_{mk}\left(V_s\right) = \max\left\{\Delta_{col}(i_s - 1, j_s), \Delta_{col}(i_s + 1, j_s)\right\}, \\ U\left(x_s | x_t, x_t \in V_s\right) = \begin{cases} \left|\Delta_{col}(i_s, j_s) - l_{mk}(V_s)\right| & \text{si } l_{mk}(V_s) > 0, \\ \beta_0 \delta(x_s, \lambda_{mk}) & \text{si } l_{mk}(V_s) = 0. \end{cases} \end{cases} \tag{3.8}$$

Le coefficient β_0 est de valeur plutôt élevée, de façon à pénaliser très fortement des marquages d'épaisseur une seule ligne.

3.3.4 Classification niveau composante connexe

Au niveau composante connexe, les nœuds du graphe sont les composantes connexes (CC). Ces dernières sont issues de l'étiquetage en composantes connexes de l'image binaire (deux classes) obtenue en sortie du niveau précédent (niveau pixel). Les classes recherchées sont les différents objets présents dans l'image, $\{\lambda_{o_1}, \ldots, \lambda_{o_i}, \ldots\}$. L'information pour décider quelles composantes connexes doivent être réunies pour former un objet est essentiellement une information spatiale. Par conséquent, pour ce niveau, le terme d'attache aux données est supposé constant quel que soit λ_{o_i}, et la fonction d'énergie est composée uniquement de termes déduits de la probabilité *a priori*.

Terme de régularisation

D'après la forme géométrique des lignes de la route (séparations de voies, bords de route, etc.) on définit les *a priori* suivants : les composantes connexes appartenant à un même marquage sont alignées et proches spatialement. Numériquement, l'évaluation des critères d'alignement et de proximité est réalisée de la façon suivante :

- pour deux composantes connexes (CCs) :
  - la direction de l'axe principal d'au moins une de ces CCs est bien définie (avec une imprécision estimée $\Delta\alpha$ inférieure à $\Delta\alpha_{max}$) et elle est proche de la direction du vecteur reliant les barycentres des CCs ;
  - la distance des CCs est inférieure à un seuil d_{align}^{max}.
- pour trois CCs considérées simultanément :
  - leurs distances deux à deux sont inférieures à au seuil d_{align}^{max},
  - les directions des 3 vecteurs reliant respectivement chacun des trois couples de barycentres des CCs sont proches.

Le fait de considérer trois composantes connexes simultanément est pertinent dans le cas des lignes discontinues de la route et pour la résolution des images considérées, qui est telle que les composantes connexes peuvent être trop petites pour avoir un axe principal bien défini. La figure 3.3 présente une illustration du calcul de l'alignement des composantes connexes. Le terme de régularisation utilisé pour pénaliser les configurations ne respectant pas le double critère d'alignement et de distance est comme suit :

$$U\left(x_s | x_t, x_t \in V_s\right) = -H\left(d_{align}^{max} - dist(x_s, x_t)\right) H\left(\Delta_{align}^{max} - \Delta_{align}(x_s, x_t)\right) \tag{3.9}$$

$H(u)$ est la fonction de Heaviside : $H(u) = 1$ si $u \geq 0$ et $H(u) = 0$ sinon,

d_{align}^{max} est le seuil sur la distance entre CCs supposées appartenir à un même objet,

$dist(x_i, x_j)$ est la distance entre les CCs x_i et x_j dans le cas de deux CCs considérées à la fois, et le maximum des distances entre les couples CCs dans le cas de trois CCs considérées à la fois,

Δ_{align}^{max} est le seuil sur l'alignement des CCs permettant de tenir compte de l'imprécision lors de l'estimation d'angle, et

$\Delta_{align}(x_s, x_t)$ est la différence d'angles entre CCs (angles estimés à partir de la direction de l'axe principal cas de 2 CCs ou des vecteurs des couples barycentriques dans le cas de 3 CCs).

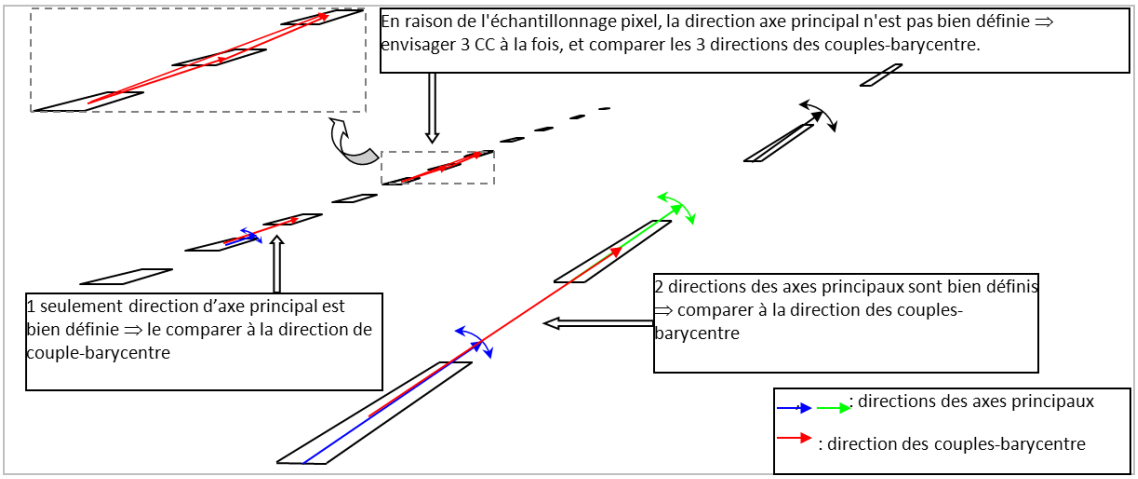


FIGURE 3.3: *Illustration du calcul d'alignement des composantes connexes (CC). La direction de l'axe principal est bien définie pour au moins une des CCs, ou n'est pas bien définie pour aucune des CCs.*

3.3.5 Classification niveau objet

Au niveau objet, les nœuds du graphe sont les objets dont on cherche à évaluer la classe parmi les types des marquages au sol de la route. Par exemple, des classes considérées sont : ligne gauche de la route (rL), ligne droite de la route (rR), ligne centrale de la route (rM), ligne discontinue (rD) ou ligne continue (rC) (lignes gauche, droite ou centrale), passage des piétons (pd), arrêt de bus, place de parking, etc. Dans notre travail, nous nous limitons à Ω, espace de discernement, tel que $\Omega = \left\{\lambda_{rL}, \lambda_{rR}, \lambda_{rM}, \lambda_{rD}, \lambda_{rC}, \lambda_{pd}, \lambda_{other}, \lambda_{\overline{mk}}\right\}$, tout en considérant les également hypothèses composées (disjonctions) suivantes $\lambda_{re} = \bigcup\{\lambda_{rL}, \lambda_{rR}, \lambda_{rM}\} = \bigcup\{\lambda_{rD}, \lambda_{rC}\}$, $\lambda_{unknown} = \bigcup\{\lambda_{re}, \lambda_{pc}, \lambda_{other}\}$, et toutes les hypothèses complémentaires des hypothèses composées ou des hypothèses des singletons, par exemple : $\lambda_{\overline{re}} = \lambda_{unknown} \setminus \{\lambda_{re}\}$, $\lambda_{\overline{pd}} = \lambda_{unknown} \setminus \{\lambda_{pd}\}$, etc.

Les marquages au sol peuvent être distingués entre eux par le rapport entre les longueurs du grand axe et du petit axe des composantes connexes les constituants (ici, nous supposons que les CCs réunies au sein d'un même objet ont toute la même forme géométrique). Soit y_r ce ratio : $y_r = \frac{majoraxis}{minoraxis}$. Ainsi on définit le terme d'attache aux données U_0 comme suit :

$$\begin{cases} U_0\left(y_r | x_r = \lambda_{re}\right) = \ln\left(1 + e^{-\alpha(y_r - t_{Raxis,r})}\right), \\ U_0\left(y_r | x_r = \lambda_{\overline{re}}\right) = \ln\left(1 + e^{\alpha(y_r - t_{Raxis,r})}\right), \end{cases} \tag{3.10}$$

avec $t_{Raxis,r}$ le seuil sur le rapport des axes du marquage au sol, et

$$\begin{cases} U_0\left(y_r | x_r = \lambda_{pd}\right) = -\ln\left(\frac{1}{1+e^{-\alpha(y_r - t_{inf_Rax})}} + \frac{1}{1+e^{\alpha(y_r - t_{sup_Rax})}} - 1\right), \\ U_0\left(y_r | x_r = \lambda_{\overline{pd}}\right) = -\ln\left(2 - \frac{1}{1+e^{-\alpha(y_r - t_{inf_Rax})}} - \frac{1}{1+e^{\alpha(y_r - t_{sup_Rax})}}\right), \end{cases} \tag{3.11}$$

avec t_{inf_Rax} et t_{sup_Rax} les limites de l'intervalle représentant l'imprécision sur le rapport des axes dans le cas d'un passage piétons ($t_{inf_Rax} < t_{sup_Rax}$).

Les classes correspondant à des marquages au sol se situent au-dessous de la ligne d'horizon. Ainsi, nous définissons un terme d'énergie pénalisant l'attribution d'une classe de marquage routier aux objets au-dessus de la ligne d'horizon :

$$\begin{cases} U_{up}\left(i_r^{up} | x_r \in \lambda_{mk}\right) = 0, \\ U_{up}\left(i_r^{up} | x_r \in \lambda_{\overline{mk}}\right) = -\beta_{up} \times \left[\frac{1}{1+e^{-\alpha_{up} \cdot \max(f_{flat} n_{lin} - i_r^{up}, 0)}} - 0.5\right], \end{cases} \tag{3.12}$$

$f_{flat} n_{lin}$ est la position la ligne d'horizon,

i_r^{up} est le numéro de la ligne inférieure de la région, et

β_{up} est un facteur de pondération accordant une importance relative aux différents termes d'attache aux données.

Finalement, nous introduisons des termes d'énergie ad-hoc pour caractériser les classes spécifiquement. Par exemple, un marquage *ligne discontinue* doit contenir plusieurs CCs :

$$\begin{cases} U_{disk}\left(y_r | x_r = \lambda_{rD}\right) &= -\beta_{disc}\left[1 - n_{cc}(y_r)\right], \\ U_{disk}\left(y_r | x_r = \lambda_{\overline{rD}}\right) &= 0. \end{cases} \tag{3.13}$$

Terme de régularisation

Tout d'abord, nous définissons un terme par défaut : s'il n'y a aucun objet de label x_t dans le voisinage de l'objet s, alors le terme d'énergie est :

$$U_N\left(x_r = \lambda_i | x_t, t \in V_r\right) = \beta_N \min\left\{1 - \delta(x_r, x_t)\right\}, \tag{3.14}$$

où β_N est un facteur de pondération accordant une importance relative aux différents termes d'énergie (U_0, U_N).

Ici, le voisinage comprend toutes les objets situés à une distance inférieure à $dist_{V_r}$ de l'objet r : $t \in V_r \Leftrightarrow dist(r,t) \leq dist_{V_r}$. De manière à prendre en compte la taille apparente de l'objet dans l'image en fonction de la coordonnée en ligne, $dist_{V_r}$ dépend de la ligne de r (cas du plan de la route où la coordonnée en ligne et la coordonnée profondeur en 3D sont liées).

Le dernier terme d'énergie proposé sert à spécifier plus précisément certaines classes, par exemple, en distinguant entre les lignes gauche et droites, ou l'identification des entrées et/ou des sorties des voies d'autoroute et ainsi de suite. Pour ce faire, nous considérons des relations de voisinage horizontales (en ligne), de sorte que si r et t sont deux objets, ils sont voisins si et seulement si $|N(r,t)| > 0$ avec $N(r,t)$ l'ensemble des pixels appartenant à r , tels que un pixel de la même ligne appartienne à l'objet t : $N(r,t) = \{s | s \in r, \exists s' \in t : i_{s'} = i_s\}$. Soit alors $D_{col}(i_s)$ la fonction de régression linéaire entre les pixels correspondants entres les objets r et t, qui donne, par rapport à la coordonnée de la colonne j_s des pixels $N(r,t)$ de s, la coordonnée de la colonne des pixels correspondant dans t , et $rmse[D_{col}(r,t)]$ l'erreur quadratique moyenne associée à la régression linéaire $D_{col}(i_s)$. Alors :

$$U\left(x_r = \lambda_{rL} | x_t, t \in V_s\right) = \begin{cases} \gamma rmse[D_{col}(r,t)] & \text{si } |N(r,t)| > 0 \quad \text{et } r \uparrow t, \\ +A & \text{si } |N(r,t)| = 0 \quad \text{ou } r \downarrow t, \end{cases} \tag{3.15}$$

avec '$r \uparrow t$' signifie 'r est à gauche de t' et '$r \downarrow t$' signifie 'r n'est pas à gauche de t', γ est une constante de pondération des énergies, et A est constante à valeur très élevée.

3.3.6 Résultats et discussion

Nous considérons des exemples d'images simulées par le simulateur Pro-SiVIC. La caméra simulée est frontale et acquiert des images dans le domaine visible (c'est-à-dire la gamme de 0.4 à 0.75 μm), soit en couleur (les canaux Rouge, Vert, et Bleu) soit panchromatique. Les différentes scènes simulées (figure 3.4) sont de complexité croissante. Sur la première, figure 3.4a, on trouve un exemple classique de route à deux voies. Dans le second exemple, figure 3.4b, il y a une voie de sortie de route. Dans les troisième et quatrième exemples, figure 3.4c et figure 3.4d, il y a une voie d'entrée sur la route, avec comme différence que, dans le troisième cas, la voiture se trouve sur la route principale, alors que dans

code couleur	interprétation de la classe
Noir	n'est pas un marquage au sol
Rouge	passage de piétons
Vert	ligne continue ou marquage non spécifié
Bleu foncé	ligne discontinue
Jaune	ligne gauche de la route
Magenta	ligne droite de la route
cyan	ligne centrale

Table 3.1: *Codes des couleurs des classes représentées dans les figures 3.6, 3.7, 3.8, 3.9, 3.10, et 3.11*

le quatrième cas la voiture est sur la voie d'entrée. Dans les cinquième et sixième exemples, figure 3.4e et figure 3.4f, on observe des passages pour piétons, ayant un niveau de complexité différent entre eux : dans le cinquième cas la ligne centrale intersecte le passage pour piétons, ce qui n'est pas le cas dans le sixième exemple. La figure 3.5 présente les résultats du prétraitement à base de morphologie mathématique (section 3.3.2). Les marquages ressortent clairement.

Pour chacun des trois niveaux considérés (pixel, composante connexe et objet), la minimisation de la fonction d'énergie globale correspondante est réalisée en utilisant un algorithme ICM (*Iterative Conditionnal Modes*) [11]. Il s'agit d'une procédure d'optimisation locale permettant d'assurer une convergence rapide en quelques itérations, et telle que, dans la plupart des cas, on obtienne des résultats proches des résultats optimaux.

Le tableau 3.1 donne une représentation des codes des couleurs de la classification finale représentée sur les figures 3.6, 3.7, 3.8, 3.9, 3.10, et 3.11.

Dans la première scène, figure 3.6, les voies de circulation sont bien classifiées, à savoir la ligne de droite, la ligne de gauche et la ligne centrale, le marquage de bus étant détecté comme une ligne continue. Le résultat n'est pas faux, d'une part puisque nous n'avons pas défini un marquage spécifique pour les bus, et d'autre part nous avons associé les marquages non spécifiés à la couleur verte.

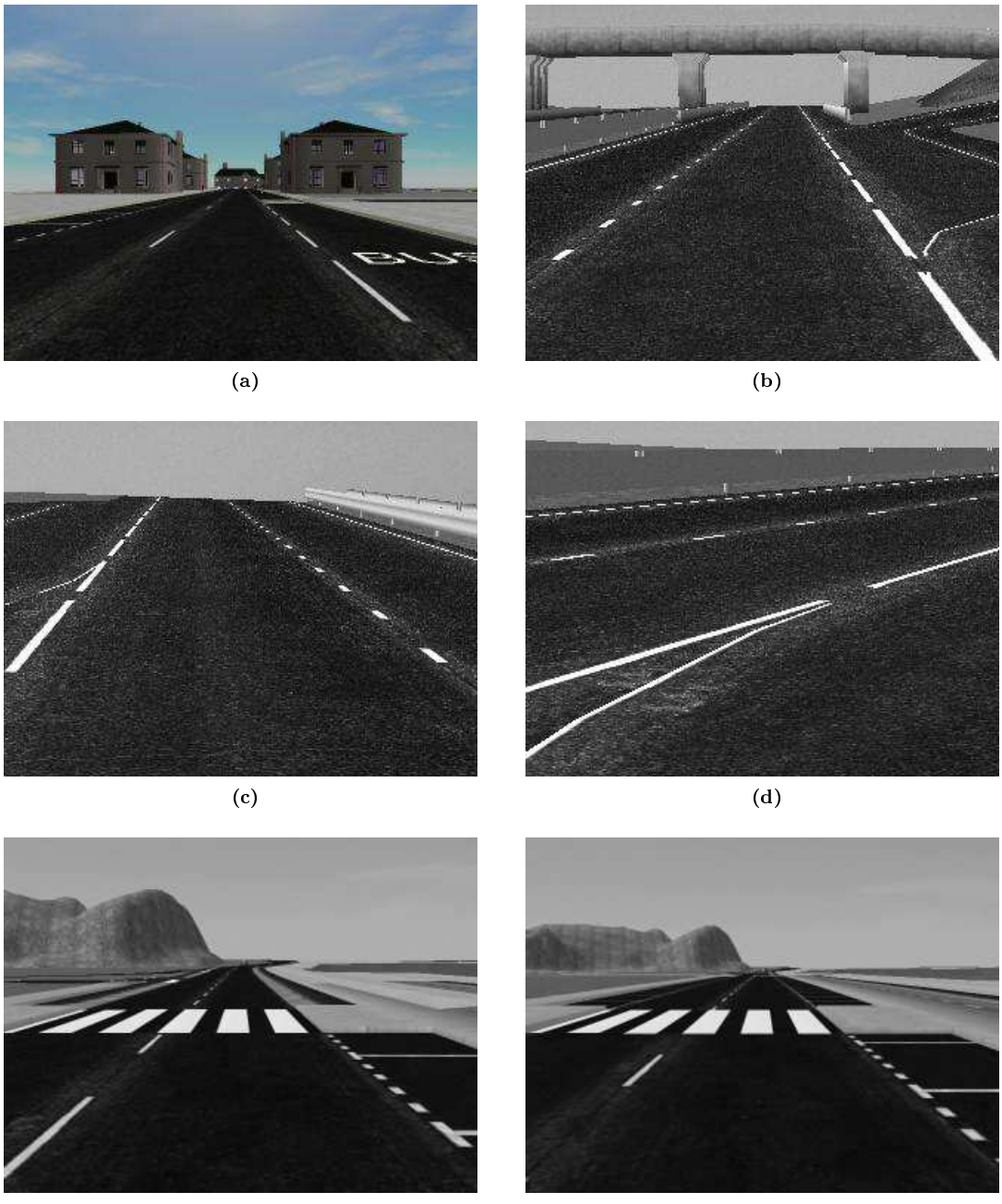
Dans la deuxième scène, figure 3.7, les deux lignes, gauche et droite, de la route sont correctement classifiées, sauf pour de petites parties des lignes de sortie de la route. Les deux autres lignes de la route sont classées comme des lignes centrales (ou intermédiaires), ce qui est assez cohérent.

Dans la troisième scène, figure 3.8, les lignes gauche et centrale sont bien classifiées, mais il y a une erreur sur la classification de la ligne droite en raison de la détection de la barrière routière comme une ligne de circulation.

Dans la quatrième scène, figure 3.9, l'erreur de classification pour les lignes gauche et centrale est due au fait que les lignes sont presque horizontale et que le voisinage horizontal utilisé pour le terme de régularisation au niveau objet n'est pas adapté à cette direction de la route. Cet artéfact était prévisible, mais nous avons tenu à considérer cet exemple pour illustrer les limites de notre approche.

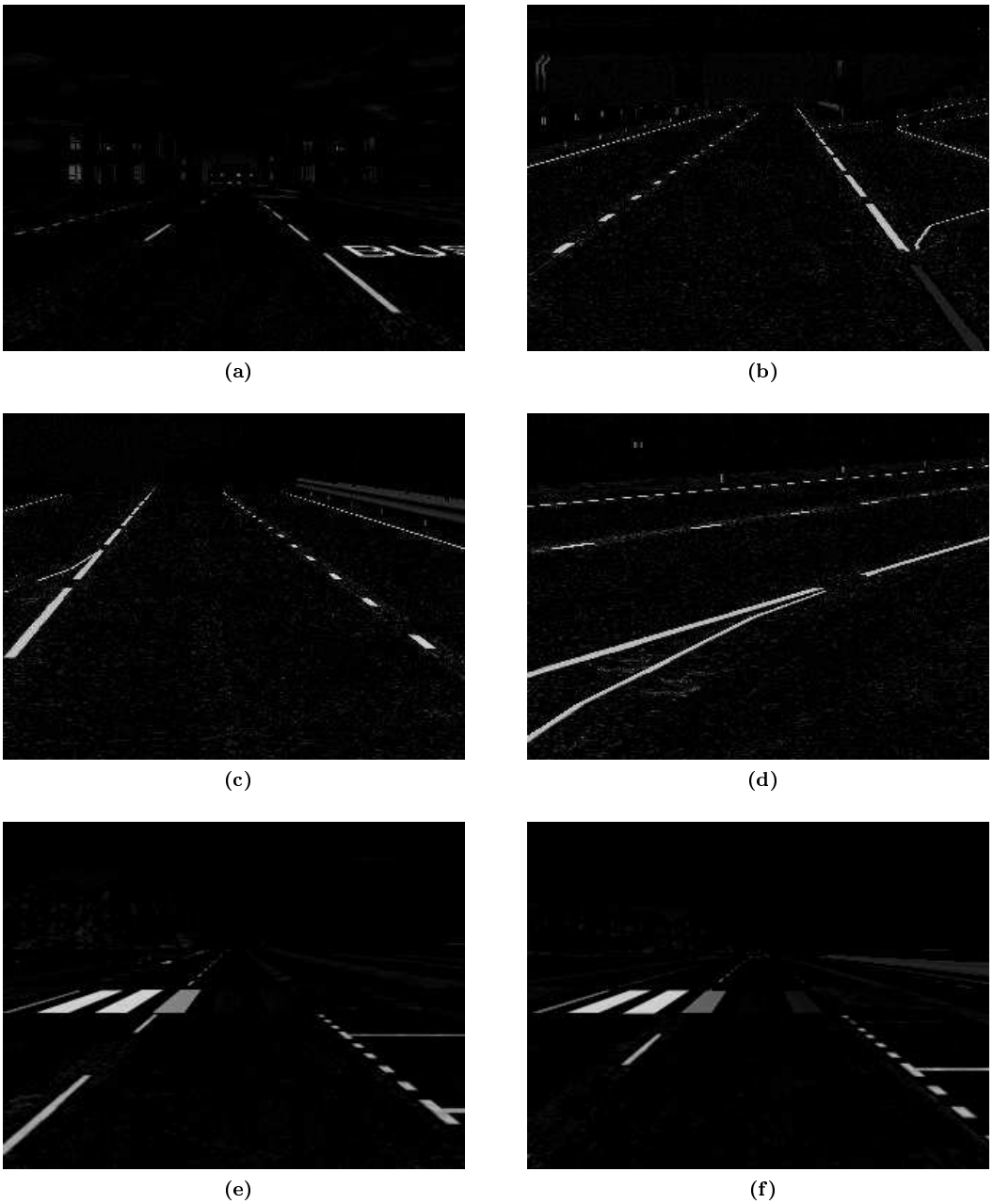
La cinquième scène, figure 3.10, montre le premier exemple avec des passages de piéton. Premièrement, nous notons que la taille de l'élément structurant utilisé, à savoir une boule de rayon 5, est trop petite pour détecter correctement toutes les bandes du passage des piétons, mais nous avons voulu garder une même taille d'élément structurant sur tous les exemples. En outre, pour le calcul de l'indice vigilance, la détection d'une partie du passage des piétons est suffisante. En ce qui concerne les lignes gauche, droite et centrale de la route, elles sont bien classifiées, même si le passage des piétons qui est relié au bord de la ligne centrale est classé en tant ligne centrale.

Finalement, la sixième scène, figure 3.11, montre un exemple plus complexe avec quelques erreurs : la ligne centrale est classée comme une ligne gauche, dû au fait qu'il n y a pas une ligne sur sa gauche. Et une partie de la ligne droite est considérée comme une ligne discontinue (qui est vrai aussi).



(a) (b) (c) (d) (e) (f)

Figure 3.4: *Six exemples d'images simulées.*

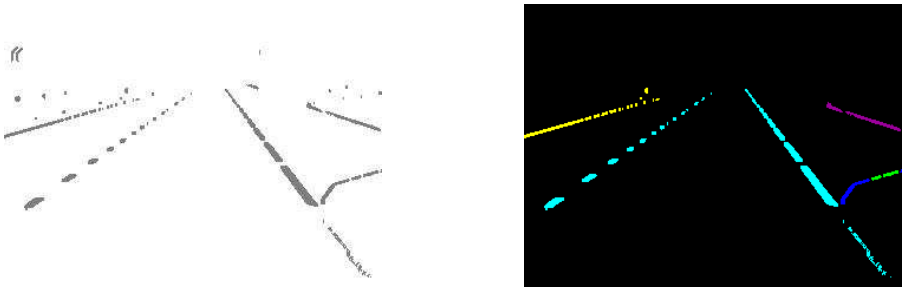


(a) (b) (c) (d) (e) (f)

FIGURE 3.5: *Six exemples d'images en sortie du prétraitement de morphologie mathématique (différence entre entre I et sa reconstruction géodésique $E^I_{ES}(\varepsilon_{ES}(I))$) et en entrée de la classification.*

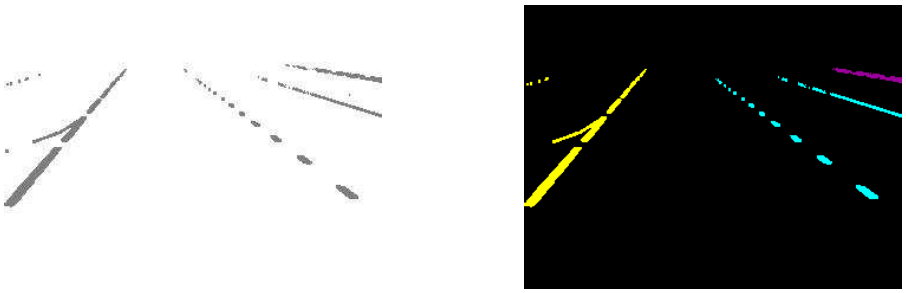
(a) (b)

FIGURE 3.6: *Exemple 1 : (a) classification pixelique, et (b) classification au niveau objet.*



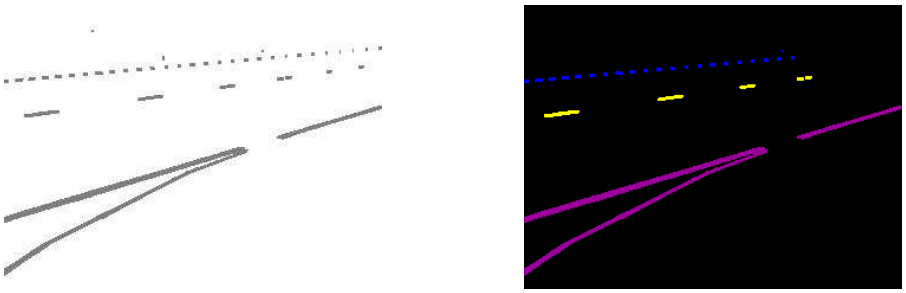
(a) (b)

Figure 3.7: *Exemple 2 : (a) classification pixelique, et (b) classification au niveau objet.*



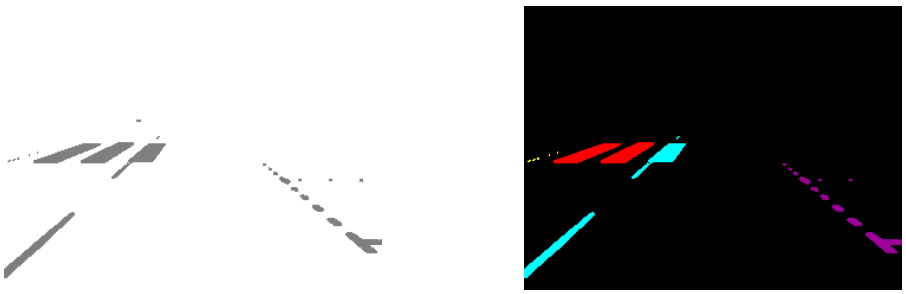
(a) (b)

Figure 3.8: *Exemple 3 : (a) classification pixelique, et (b) classification au niveau objet.*



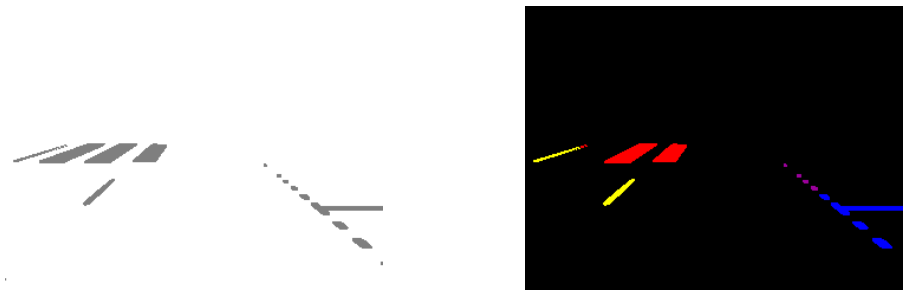
(a) (b)

Figure 3.9: *Exemple 4 : (a) classification pixelique, et (b) classification au niveau objet.*



(a) (b)

Figure 3.10: *Exemple 5 : (a) classification pixelique, et (b) classification au niveau objet.*



(a) (b)

FIGURE 3.11: *Exemple 6 : (a) classification pixelique, et (b) classification au niveau objet.*

	vrai positifs	faux positifs	faux négatifs	vrai négatifs
CL_T	10144	749	4085	587138
CL_P	10191	752	4038	587135
CL_O	9337	271	4892	587616

TABLE 3.2: *Performance quantitative de la classification : seuil CL_T, pixel CL_P et objet CL_O estimé sur les six images comparées à une vérité de terrain.*

Quantitativement, le tableau 3.2 indique le nombre de vrais positifs, faux positifs, faux négatifs et vrais négatifs calculés à partir d'une vérité de terrain (VT) et une classification binaire en deux classes λ_{mk} et $\lambda_{\overline{mk}}$.

Nous avons comparé la vérité de terrain, obtenue par une interprétation "manuelle" des images de données, avec différentes méthodes de classification. La première méthode est un simple seuillage à la valeur t_{mk} effectué sur l'image $Y = I - ES(\varepsilon_{ES}(I))$ et qui fournit une image notée CL_T. Pour notre méthode, la classification au niveau pixel est notée CL_P, et la classification objet est notée CL_O.

Entre la classification par seuillage (CL_T) et la classification au niveau pixel (CL_P), nous notons une légère baisse des faux négatifs probablement due à une meilleure définition des bords des marquages au sol. Entre la classification au niveau pixel (CL_P) et la classification au niveau objet (CL_O), nous constatons à la fois une augmentation notable des faux négatifs et une diminution des faux positifs, ce qui est dû à une diminution globale des pixels ou des composantes connexes de label λ_{mk} (parmi eux certains sont erronées et d'autres étaient justes). Ainsi, nous concluons que l'intérêt principal de la classification au niveau objet est l'amélioration de l'espace de discernement : il gère beaucoup plus de classes que dans le cas de deux classes, à savoir λ_{mk} et $\lambda_{\overline{mk}}$.

3.3.7 Conclusion

Dans ce chapitre, nous avons proposé un modèle markovien à trois niveaux pour extraire et classifier les marquages routiers au sol. L'approche proposée a été validée en utilisant des données simulées. Ce travail reste un travail préliminaire qui devra être étendu dans trois directions.

- Tout d'abord, il s'agit d'augmenter le nombre de classes au niveau objet : entrée et sortie de route, arrêt de bus, parking, etc.
- Ensuite, on espère que la prise en compte du résultat de classification précédente (obtenu à l'instant $t-1$) permettra de réduire le temps de calcul : connaissant l'égomouvement du véhicule, des régions d'intérêt seront définies à partir des détections précédentes, et seulement une sous-partie de l'image (les zones d'intérêt ainsi que les zones autour de la ligne d'horizon) seront traitées.

– Enfin, l'algorithme doit être validé sur des données réelles, notamment sa robustesse devra être évaluée par rapport aux conditions réelles de circulation incluant le vieillissement des marquages au sol (ce qui induit une faible contraste) et des conditions de visibilité réduite (en cas de pluie, brouillard).

3.4 Conclusion

Contrairement au chapitre précédent, nous nous sommes intéressés à la détection d'objets parfaitement connus et normalisés. Notre contribution a été au niveau de la classification des marquages au sol (signalisation verticale). Nous avons vu au paragraphe précédent les travaux restants à mener.

Afin d'enrichir les algorithmes embarqués sur le véhicule d'ACCIS, nous souhaiterions également proposer une contribution sur la détection de la signalisation verticale. Nous croyons que les approches à base de transformée (e.g. [10]) sont bien adaptées pour modéliser l'*a priori* sur les formes (en petit nombre) des panneaux recherchés.

Deuxième partie

Automatique

Chapitre 4

Commande du véhicule

Sommaire

4.1 État de l'art 84
4.1.1 Planification de trajectoire 84
4.1.2 Faisceaux de trajectoires 85
4.1.3 Platitude et suivi de trajectoire 86
4.2 Modélisation cinématique augmentée du véhicule 91
4.2.1 Modèle cinématique augmenté du véhicule 91
4.2.2 Platitude du modèle 92
4.2.3 Suivi de trajectoire avec stabilité 93
4.2.4 Simulation 94
4.3 Conclusion 100

4.1 État de l'art

4.1.1 Planification de trajectoire

La planification de trajectoires pour les robots, et pour les véhicules à contraintes non holonomes, a été largement développé ces dernières années. En [44], l'auteur introduit les premiers travaux en planification de trajectoires allant d'un point à un autre en minimisant le temps de parcours et en se basant sur des lignes droites et des arcs de cercles. Dans ce contexte de planification en environnement libre, plusieurs extensions de ces travaux ont été réalisées [110, 69, 117, 95].

Il existe un grand nombre d'algorithmes différents permettant de réaliser la tâche de planification de trajectoire. Dans ce qui suit, nous présentons certaines de ces méthodes en expliquant brièvement leurs principes ainsi que leurs avantages et leurs inconvénients.

Champs de potentiel

Les méthodes de navigation à base de champs de potentiel ont été proposées par [75] dans le cadre de la navigation en robotique. Son travail consiste à assimiler le robot à une particule, qui, en se déplaçant, subit un champ de potentiel fictif composé d'un champ attractif, modélisant le point d'arrivée, et d'un ensemble de champs répulsifs, modélisant les obstacles présents. Ainsi, pour chaque position donnée, une force résultante des actions des obstacles et du point d'arrivée est calculée, qui correspond à la direction à suivre. Plusieurs adaptations de cette technique ont été proposées (comme par exemple [15]).

Cet algorithme est totalement réactif et peut donc être très facilement implémenté en temps réel. Cependant, comme l'environnement est rarement convexe, cette méthode entraine facilement le robot dans des minimas locaux. Ce problème peut être résolu de différentes manières. En [9, 72], les auteurs proposent de déclencher un mouvement particulier (suivi de mur, déplacement aléatoire, ...) lorsqu'on rencontre un tel minimum.

Méthode par décomposition de l'environnement en cellules

Cette méthode de planification de trajectoire est basée sur la décomposition cellulaire de l'environnement [81]. Elle consiste à discrétiser l'environnement en un ensemble de régions connexes adjacentes. Plusieurs techniques existent ; citons comme exemples la partition de Voronoï [28] où les graphes de visibilité [24]. L'environnement discrétisé est représenté par un graphe de connectivité dont les nœuds correspondent aux différentes régions et les arcs aux relations d'adjacence entre elles. Trouver une trajectoire revient à chercher un chemin dans le graphe. Ainsi, Le problème peut être facilement traité informatiquement [116]. Ce type d'algorithme est facile à implémenter. Cependant la discrétisation de l'espace de travail limites l'espace des trajectoires admissibles.

La méthode de la fenêtre dynamique

Cette méthode a été proposée par [51]. Elle calcule les vitesses accessibles du robot pour que celui-ci ne rentre en collision avec aucun obstacle. Une fois l'espace de recherche des vitesses calculé, l'algorithme trouvera la commande optimale à envoyer au robot en minimisant une fonction de coût sur ce domaine de recherche (minimisation du temps de parcourt, de l'énergie dépensée, maximisation de la vitesse, etc.). Cette méthode permet de prendre en compte les contraintes cinétiques du robot. Son utilisation est très intéressante pour un robot se déplaçant rapidement. Par contre, elle manque de flexibilité, ce qui rend difficile son implémentation dans un cadre de véhicule autonome.

La planification par juxtaposition de splines polynomiales

Cette méthode a été proposée par [35]. Elle génère des morceaux de splines polynomiales en respectant les contraintes cinématiques du robot. On peut simplifier l'algorithme en le forçant à générer non pas des splines, mais simplement des polynômes. La trajectoire du robot est alors assimilée à la juxtaposition de plusieurs polynômes. Cette méthode permet donc de respecter les contraintes cinématiques du robot.

La méthode de la bande élastique

Le concept de bande élastique a été proposé par [108] pour les plates-formes mobiles. Cette méthode tend une "bande élastique" entre le robot et l'objectif. Cette bande étant capable de se déformer en présence d'un obstacle, celle-ci génère dont une trajectoire envisageable par le robot. Cette technique a été étendue à un robot de type véhicule par [74].

Rapidly exploring Random Trees

Une méthode, relativement rapide consiste à construire un arbre, dit "Rapidly exploring Random Tree ou RRT" [27]. Partant d'une configuration initiale, l'algorithme explore incrémentalement l'espace des configurations jusqu'à trouver un chemin connectant P_i à P_f. Le processus de construction de l'arbre est relativement simple. On tire, pour chaque configuration donnée, des configurations aléatoires et on vérifie si cette configuration est réalisable ou non, en tenant compte de la cinématique du véhicule [105].

Planification Kinodynamique déterministe

Cette technique (enveloppé dans [53]) permet de respecter les contraintes du véhicule en évitant de problèmes de minima locaux. La première étape construit un arbre de solutions locales contenant des morceaux de trajectoires polynomiales correspondant à différents horizons temporels. La deuxième étape est un planificateur global de type A* qui sélectionne les morceaux précédents pour obtenir une trajectoire solution. L'algorithme obtenu est notablement plus rapide que le RRT et fournit des trajectoires avec bien moins de variations de courbure.

4.1.2 Faisceaux de trajectoires

La plupart des systèmes de conduite autonomes, véhicule ou robot, se réfèrent au suivi d'une trajectoire de référence. D'autres auteurs ont considéré la planification d'un ensemble de trajectoires que l'on nomme faisceau. Dans [68], les auteurs proposent de construire un faisceau sous la forme d'un ensemble d'antennes virtuelles, (*tentacles* en anglais) pour chaque étape lors de l'avancement du véhicule.

Dans ce même contexte, les auteurs de [73] proposent d'évaluer un ensemble de trajectoires candidates. Dans [80], les auteurs proposent un ensemble d'arcs comme première étape pour la planification de trajectoire.

En [123], les auteurs ont proposé, dans le cadre du Grand Challenge Defense Advanced Research Projects Agency (DARPA) en 2005, d'utiliser un ensemble de chemins candidats pour la planification de chemin. Leur robot, Stanley, a remporté, cette année, là la première place[1]. La planification de trajectoire est basée sur une recherche qui minimise une combinaison linéaire de fonctions de coût en continu, suivant le modèle du véhicule. Ce modèle comprend plusieurs contraintes cinématiques

1. http ://cs.stanford.edu/group/roadrunner/stanley.html

et dynamiques, notamment l'accélération latérale maximale, l'angle de braquage maximal, etc. Pour chaque pas temporel, l'algorithme considère un ensemble de trajectoires choisies dans un espace 2D de manœuvres. La première dimension décrit le décalage latéral à ajouter à la trajectoire en cours ; ce premier paramètre permet à Stanley d'aller à gauche et à droite, tout en restant sensiblement parallèle à la trajectoire de base. La deuxième dimension définit la vitesse avec laquelle Stanley modifie le décalage latéral, tous les chemins candidats devant impérativement obéir aux contraintes cinématiques et dynamiques du véhicule. Ce deuxième paramètre permet au processeur de contrôler le niveau de danger pour l'évitement d'obstacles. Deux types d'évitements d'obstacle existent, à savoir l'évitement ultra-rapide pour le cas des obstacles dans la rue et l'évitement en douceur pour éviter la sortie de route du véhicule et suivre en douceur le centre de la voie. Une fois lA trajectoire prévue définie, il détermine les commandes d'accélération, de freinage et de direction nécessaires à la réalisation de ce chemin.

4.1.3 Platitude et suivi de trajectoire

4.1.3.1 Suivi de trajectoire

Le suivi de trajectoire consiste à calculer les commandes à envoyer aux actionneurs du véhicule permettant de suivre une trajectoire préalablement calculée et ce, avec stabilité. Le véhicule est considéré comme un système dynamique, et l'on utilise des méthodes de commande par retour d'état pour asservir le système sur la trajectoire de référence.

Afin d'assurer ce suivi de trajectoire, des commandes linéaires non stationnaires continues ont été développées [98]. Une famille de lois de commandes non linéaires permettant de réaliser le suivi de trajectoire est donnée par l'application directe de commandes issues de la platitude [49]. Kanayama [71] suppose que la trajectoire de référence est réalisable par le robot ou le véhicule mobile.

Le suivi de la trajectoire planifiée ne permet pas de garantir l'absence de collision. En effet, des collisions peuvent se produire lors de l'exécution de la trajectoire, dues à un mouvement des obstacles dans l'environnement, mouvement qui n'était pas préalablement prévu dans le modèle de planification.

Ainsi, le mouvement initialement planifié doit être adapté lors de son exécution, en mettant en place des stratégies d'évitement réactif d'obstacles appropriées.

4.1.3.2 Platitude

Les divers objectifs que l'on souhaite atteindre avec un système dynamique se ramènent très souvent à une modification du comportement de ce système. On dispose pour ce faire de certaines variables, nommées commandes, qui servent à piloter le système. Ces modifications de comportement consistent généralement en une poursuite de trajectoires prescrites à l'avance de certaines variables du système, poursuite complétée par un bouclage stabilisant, assurant la tenue du suivi dans le temps. Notons que la première phase est de nature anticipative (ou *feedforward* en anglais) alors que la seconde est rétroactive (ou *feedback* en anglais). Notons en outre que la majorité des études de la littérature automaticienne ainsi que celles des manuels pédagogiques se concentre sur la seconde phase prenant l'état du système égal à zéro pour la première. Si l'importance des bouclages est bien évidemment incontestable, l'utilité pratique de méthodologies mêlant anticipation et rétroaction l'est tout autant.

La méthodologie que nous adoptons pour la commande des systèmes se décompose principalement en deux étapes :

1. Un suivi de trajectoires dit en boucle ouverte, parce que supposant le modèle parfait et les conditions initiales connues.

2. La stabilisation par bouclage autour des trajectoires précédentes, afin de compenser les inexactitudes de modèle et les méconnaissances sur les conditions initiales.

Des solutions simples et naturelles au problème (1) sont obtenues en utilisant une propriété structurelle fondamentale ; elle se nomme platitude différentielle en non linéaire de dimension finie (notion due à Michel Fliess, Jean Lévine, Philippe Martin et Pierre Rouchon).

Historiquement, le concept de la platitude fut introduit dans un cadre d'algèbre différentielle [48, 49]. Un grand nombre de systèmes pratiques sont plats (voir par exemple [63, 84, 85, 86, 91, 92, 100, 113, 114, 118])

Un système est différentiellement plat s'il existe une suite de quantités nommée sortie plate, dont les composantes sont indépendantes entre elles ainsi que leurs dérivées, et sont telles que toute variable du système s'exprime en fonction de ces composantes et de leurs dérivées. Cette paramétrisation fournit des expressions sans intégration d'équation différentielle, ce qui assure un calcul des plus rapides ainsi qu'une certaine robustesse naturelle. La commande est en particulier donnée comme expression de la base ; on réalise ainsi une inversion de système sans aucune intégration.

Considérons un système non linéaire décrit par un ensemble d'équations différentielles

$$F_l\left(z, ..., z^{(i)}, ..., z^{(v_l)}\right) = 0, \qquad l = 1, ..., N.$$

La notion de la platitude peut s'exprimer ainsi : un système non linéaire est dit plat s'il existe un m-uple $\mathbf{y} = (y_1, ..., y_m)$ de fonctions, nommé sortie plate, doté des trois propriétés suivantes :

1. Elle fait partie du système (*caractère endogène*) :
   Les composantes de $\mathbf{y}$ peuvent s'exprimer en fonction des variables du système z par des relations différentielles de la forme

   $$y_i = P_i\left(z, ..., z^{(k)}, ..., z^{(p_i)}\right)$$

   pour $i = 1, ..., m$.
2. Ses composantes sont indépendantes (*indépendance*) :
   Les composantes de $\mathbf{y}$ sont différentiellement indépendantes, c.à.d. elles ne sont reliées par aucune équation différentielle (non triviale)

   $$Q\left(y, ..., y^{(k)}, ..., y^{(\alpha)}\right) = 0.$$

3. Elle fournit une paramétrisation complète du système (*forme canonique de suivi*) :
   Toutes variables z_i du système, par exemple l'état ou les entrées, s'expriment directement à partir de $\mathbf{y}$ en utilisant seulement des dérivations. En d'autres termes, tout $z_i$satisfait une relation de la forme

   $$z_i = R\left(y, ..., y^{(k)}, ..., y^{(\gamma_i)}\right).$$

Remarque 6. La troisième propriété fournit une solution simple au problème du suivi de trajectoires de références $\mathbf{y}_r(t) = (y_{1r}(t), ..., y_{mr}(t))$ en boucle ouverte. La deuxième propriété assure que les différentes composantes de $\mathbf{y}_r(t)$ peuvent être choisies indépendamment.

Remarque 7. Le nombre m de sorties plates est le degré de sous-détermination du système (correspondant, pour un système d'équations de rang plein, au nombre de variables du système moins le nombre d'équations). Si le système admet des entrées, m est le nombre des entrées indépendantes.

Cette notion peut donc se définir, dans le cas de système d'état $\mathbf{x}$ et de commande $\mathbf{u}$ par ce qui suit.

Définition 8. Le système

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \tag{4.1}$$

où $\mathbf{x}(t) \in \mathbb{R}^n$ et $\mathbf{u}(t) \in \mathbb{R}^m$ est *différentiellement plat* s'il existe un m-uplet de variables nommé *sortie plate*, noté $y(t)$ où

$$\mathbf{y}(t) = h\left(\mathbf{x}(t), \mathbf{u}(t), \dot{\mathbf{u}}(t), ..., \mathbf{u}^{(\eta)}(t)\right), \qquad \mathbf{y}(t) \in \mathrm{R}^m, \eta \in \mathrm{N}^m \tag{4.2}$$

telles que d'une part

$$\mathbf{x}(t) = A\left(\mathbf{y}(t), \dot{\mathbf{y}}(t), ..., \mathbf{y}^{(q)}(t)\right) \tag{4.3a}$$

$$\mathbf{u}(t) = B\left(\mathbf{y}(t), \dot{\mathbf{y}}(t), ..., \mathbf{y}^{(q)}(t)\right) \tag{4.3b}$$

où h, A et B sont des fonctions continues en leurs arguments et q est un entier naturel, et telle que d'autre part les équations du système soient satisfaites :

$$\frac{dA}{dt}\left(\mathbf{y}(t), ..., \mathbf{y}^{(q)}(t)\right) = f\left(A\left(\mathbf{y}(t), ..., \mathbf{y}^{(q)}(t)\right), B\left(\mathbf{y}(t), ..., \mathbf{y}^{(q)}(t)\right)\right).$$

La notion de système plat généralise ainsi les systèmes linéaires mono-entrée dont les fonctions de transfert entre une variable quelconque z du système et la base ω sont à numérateur constant $\hat{z}(s)/\hat{\omega}(s) = C/p(s)$ (où C est une constante et p un polynôme).

Exemple de robotique mobile Un exemple simple de système plat est fourni par un robot mobile à une roue avant folle et avec un moteur monté sur chacune des deux roues arrières. Un modèle cinématique est le suivant :

$$\dot{x} = v\cos\psi \tag{4.4a}$$

$$\dot{y} = v\sin\psi \tag{4.4b}$$

$$\dot{\psi} = \omega \tag{4.4c}$$

où x et y sont les coordonnées du milieu de l'essieu arrière, ψ est l'angle dit de lacet (angle d'orientation du robot), v et ω les vitesses de propulsion et de braquage (voir Figure (4.1)). La vitesse v est reliée à la somme des vitesses de rotation des roues arrière et la vitesse ω est reliée à leur différence[2].

Ce système est plat, de sortie plate (x, y). En effet, nous avons les expressions suivantes pour les différentes variables du système :

$$v = \sqrt{\dot{x}^2 + \dot{y}^2}, \quad \text{(d'après la combinaison de } (4.4a)^2 + (4.4b)^2\text{)} \tag{4.5a}$$

$$\psi = \operatorname{atan}\left(\frac{\dot{y}}{\dot{x}}\right), \quad \text{(d'après la combinaison de } (4.4b)\,/\,(4.4a)\text{)} \tag{4.5b}$$

$$\omega = v\kappa, \qquad \text{(Utilisant } (4.4c)\text{)} \tag{4.5c}$$

où $\kappa = (\ddot{x}\dot{y} - \dot{x}\ddot{y})/v^3$ est la courbure de la trajectoire en (x, y).

2. Plus précisément, notant ω_1 et ω_2 les vitesses de rotation des roues, on a $v = (\omega_1 + \omega_2)/2$ et $\omega = (\omega_1 - \omega_2)/2$.

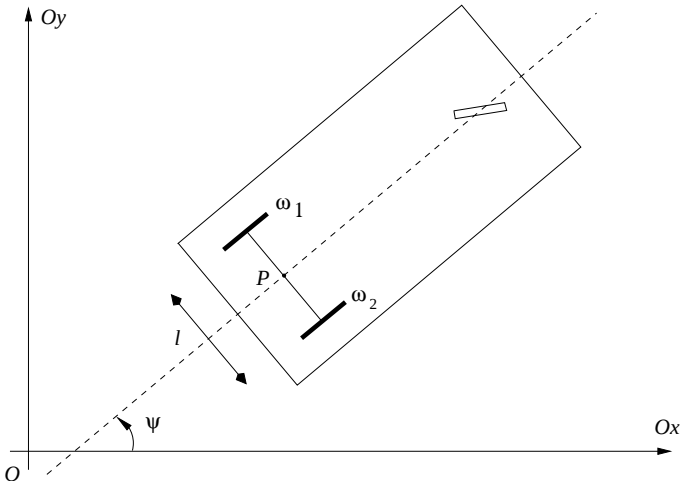


FIGURE 4.1: *Robot à roue avant folle.*

L'intérêt de cette propriété pour le suivi de trajectoires est alors évident : se fixant une trajectoire de référence $t \mapsto (x_r(t), y_r(t))$ à suivre, les expressions (4.5a) et (4.5c) où x et y sont remplacés par $x_r(t)$ et $y_r(t)$ fournissent des lois de suivi en boucle ouverte obtenues de manière simple et naturelle. Sachant qu'une confiance aveugle a été accordée au modèle, il convient de la tempérer par une phase de stabilisation autour de la trajectoire de référence $(x_r(t), y_r(t))$ permettant de compenser les inexactitudes de modèle ainsi que les méconnaissances sur les conditions initiales.

4.1.3.3 Paramétrisation

Pour chaque sortie plane donnée par une fonction de la forme $t \in \mathbf{R} \to y(t)$, la trajectoire du système $\mathbf{x}(t), \mathbf{u}(t)$ est donné par :

$$\mathbf{x}(t) = A\left(\mathbf{y}(t), \dot{\mathbf{y}}(t), ..., \mathbf{y}^{(q)}(t)\right) \tag{4.6a}$$

$$\mathbf{u}(t) = B\left(\mathbf{y}(t), \dot{\mathbf{y}}(t), ..., \mathbf{y}^{(q)}(t)\right) \tag{4.6b}$$

Il y a une correspondance une à une entre les trajectoires du système et ceux fournies par la sortie plate.

4.1.3.4 Méthodologie

Les notions précédentes seront utilisées pour obtenir ce qu'on appelle des lois en "boucle ouverte", qui assureront le suivi de la sortie plate lorsque le modèle est supposé être parfait et les conditions initiales sont supposées connues exactement. Étant donné que ce n'est jamais le cas en pratique, on a besoin de lois de rétroaction qui assureront la convergence asymptotique des erreurs de suivi à zéro. Ainsi, le schéma peut être décomposé en deux étapes :

- Élaboration de la trajectoire de référence des sorties plates ; calcul hors ligne de la commande en boucle ouverte.
- Calcul en ligne de la commande complémentaire en boucle fermée afin de stabiliser le système autour de la trajectoire de référence.

Cette conception en deux étapes convient mieux qu'un régime de stabilisation classique. En effet, La première étape donne une solution grossière pour le problème de suivi, tout en suivant le modèle au lieu de le forcer, comme dans une méthode de stabilisation pure. La deuxième étape est un raffinement,

et l'erreur entre les valeurs réelles et la trajectoire de référence sera beaucoup plus faible que dans le cas de stabilisation pure.

Plus précisément, considérant le système dynamique non linéaire $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$, nous cherchons à définir un contrôleur capable de suivre une trajectoire de référence $t \mapsto y_r(t)$. Afin de compenser les incertitudes de modèle et les méconnaissances de conditions initiales, il est nécessaire de compléter la boucle ouverte par un terme de correction en boucle fermée en fonction de l'erreur $y(t) - y_r(t)$.

4.1.3.5 Platitude et la linéarisation du bouclage

Une caractérisation des systèmes plats qui apparaît très utile pour le suivi de trajectoire est la suivante :

Proposition 9. *Un système est plat si, et seulement si, il est linéarisable par le bouclage endogène et changement de coordonnées.*

Un bouclage est dit endogène s'il ne comprend pas de dynamique externe. Plus précisément

Définition 10. Considérons le système dynamique non linéaire $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$. Le bouclage

$$\mathbf{u} = \varphi(\mathbf{x}, \mathbf{z}, v) \tag{4.7a}$$

$$\dot{\mathbf{z}} = \zeta(\mathbf{x}, \mathbf{z}, v) \tag{4.7b}$$

(où v est la nouvelle entrée) est appelé endogène si la dynamique originelle $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t))$ est équivalente à la dynamique transformée :

$$\mathbf{u} = f(\mathbf{x}, \varphi(\mathbf{x}, \mathbf{z}, v)) \tag{4.8a}$$

$$\dot{\mathbf{z}} = \zeta(\mathbf{x}, \mathbf{z}, v) \tag{4.8b}$$

Deux systèmes sont dits équivalents s'il existe une transformation inversible qui échange leurs trajectoires.

4.1.3.6 Algorithme d'extension dynamique

Cette procédure nous permet de savoir si un m-uplet $(y_1, ..., y_m)$ est une sortie plate ou non. Dans le même temps, nous obtiendrons un bouclage linéarisant.

Phase 1 : Obtenir les indices de Brunovsky faibles.

- Dériver y_1 jusqu'à ce qu'une combinaison de commandes apparaisse
- Noter k_1 le nombre des dérivations successives $y_1^{(k_1)} = f_1$
- Dériver y_2 jusqu'à ce qu'une combinaison de contrôles, indépendante de la précédente, apparaisse
- Noter k_2 le nombre des dérivations successives $y_2^{(k_2)} = f_2$

  $\vdots$
- Dériver y_m jusqu'à ce qu'une combinaison de contrôles, indépendante de la précédente, apparaisse
- Noter k_m le nombre des dérivations successives $y_m^{(k_m)} = f_m$

Phase 2 : Décider du caractère plat

Ainsi, si $k_1 + ... + k_m = n$, avec n est la dimension de l'état, alors le système admet $(y_1, ..., y_m)$ comme sortie plate. Sinon $(y_1, ..., y_m)$ n'est pas une sortie plate

Phase 3 : Obtenir le bouclage linéarisant

Le bouclage linéarisant est donné par $f_1 = v_1, ..., f_m = v_m$.

4.1.3.7 Suivi de trajectoire en boucle fermée

Sachant le système dynamique est plat, de sortie plate y, il peut être transformé par un bouclage endogène et un changement de coordonnées en une dynamique linéaire de la forme :

$$\begin{aligned} y_1^{(k_1)} &= v_1 \\ &\vdots \\ y_m^{(k_m)} &= v_m \end{aligned}$$

avec la nouvelle entrée $(v_1, ..., v_m)$. Le bouclage de suivi s'écrit :

$$\begin{aligned} v_i &= y_{ir}^{(k_i)} - \sum_{j=0}^{k_i-1} k_{ij} \left(y_i^{(j)} - y_{ir}^{(j)} \right), \qquad i = 1, ..., m \\ &= y_{ir}^{(k_i)} - \sum_{j=0}^{k_i-1} k_{ij} e_i^{(j)}. \end{aligned} \tag{4.9}$$

Un choix approprié des gains k_{ij} permet de rendre la dynamique d'erreur asymptotiquement stable :

$$e_i^{k_i} = \sum_{j=0}^{k_i-1} - k_j e_i^j, \qquad i = 1, ..., m. \tag{4.10}$$

4.2 Modélisation cinématique augmentée du véhicule

Nous utiliserons ici un modèle très simplifié de véhicule. Il s'agit d'un modèle dit cinématique augmenté en ce sens qu'il tient compte d'une inertie en dynamique longitudinale. Notons que le glissement n'est pas pris en compte, ni les dynamiques verticales, de tangage, de roulis de roues ou de suspension.

4.2.1 Modèle cinématique augmenté du véhicule

Considérons le modèle cinématique augmenté du véhicule :

$$\dot{x} = v \cos\psi \tag{4.11a}$$

$$\dot{y} = v \sin\psi \tag{4.11b}$$

$$\dot{\psi} = \frac{v}{L} \tan\delta \tag{4.11c}$$

$$M\dot{v} = \frac{C}{r} \tag{4.11d}$$

où (x, y) sont les coordonnées du milieu P de l'essieu arrière, v est la vitesse longitudinale du véhicule, du point P, ψ est l'angle de lacet, δ l'angle de braquage des roues avant et C le couple moteur. Les constantes L, M et r sont respectivement l'empattement (distance inter essieu), la masse du véhicule et le rayon de roue. Les variables C et δ sont prises comme commandes. La Figure (4.2) présente une illustration de ce modèle.

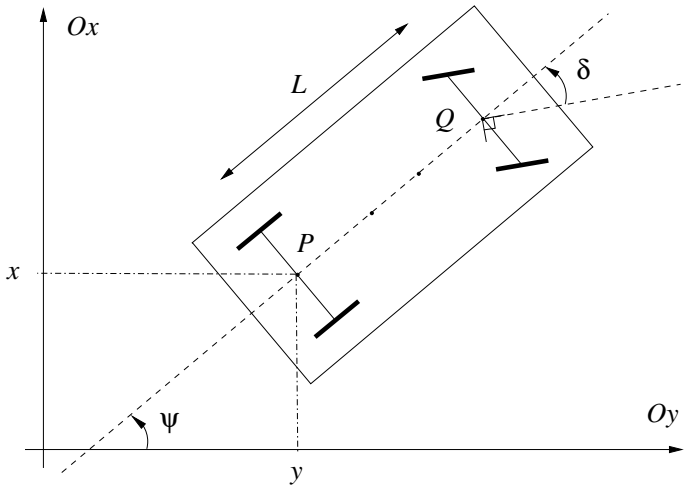


FIGURE 4.2: *Modèle cinématique d'un véhicule automobile.*

4.2.2 Platitude du modèle

Une sortie plate de ce système est (x, y). En effet, la combinaison d'équations $(4.11a)^2 + (4.11b)^2$ fournit v :

$$v = \sqrt{\dot{x}^2 + \dot{y}^2}. \tag{4.12}$$

La division $(4.11b) / (4.11a)$ donne ψ :

$$\psi = \arctan\left(\frac{\dot{y}}{\dot{x}}\right). \tag{4.13}$$

Par dérivation de cette dernière équation, nous obtenons :

$$\dot{\psi} = \frac{1}{1 + \left(\frac{\dot{y}}{\dot{x}}\right)} \frac{\ddot{y}\dot{x} - \dot{y}\ddot{x}}{\dot{x}^2} = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2} \tag{4.14}$$

d'où

$$\delta = \arctan\left(\frac{L\dot{\psi}}{v}\right) = \arctan\left(L\frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}\right). \tag{4.15}$$

Par ailleurs

$$\dot{v} = \frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{v} \tag{4.16}$$

d'où l'expression de C :

$$C = Mr\dot{v} = Mr\frac{\dot{x}\ddot{x} + \dot{y}\ddot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}}. \tag{4.17}$$

4.2.3 Suivi de trajectoire avec stabilité

Définition

Une **trajectoire** d'un point M par rapport un repère $\mathcal{R}$ entre deux instants t_0 et t_1, est l'ensemble des positions successives de ce point entre ces dates. C'est une courbe continue liée au repère $\mathcal{R}$. Une trajectoire doit satisfaire un ensemble de critères :

- Continuité du chemin
- Continuité de la courbure

Une **trajectoire de référence** est une configuration de points continue construite soit par une fonction mathématique soit par récupération d'une vérité terrain, c.à.d., par enregistrement d'un déplacement véhicule. Dans cette partie nous nous limitons à la simulation, du suivi d'une seule trajectoire de référence.

Suivi de trajectoire

Pour établir le bouclage, commençons par obtenir la dynamique de la sortie plate. La dérivation des deux premières équations (4.11a) et (4.11b) du modèle s'écrit :

$$\ddot{x} = \dot{v}\cos\psi - v\dot{\psi}\sin\psi \tag{4.18a}$$

$$\ddot{y} = \dot{v}\sin\psi + v\dot{\psi}\cos\psi \tag{4.18b}$$

d'où la dynamique de sortie plate :

$$\ddot{x} = \frac{1}{M}C\cos\psi - \frac{v^2}{L}\tan\delta\sin\psi \tag{4.19a}$$

$$\ddot{y} = \frac{1}{M}C\sin\psi + \frac{v^2}{L}\tan\delta\cos\psi \tag{4.19b}$$

qui s'écrit encore :

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \frac{C}{M} \\ \frac{v^2\tan\delta}{L} \end{pmatrix} = \mathcal{R}(\psi) \begin{pmatrix} \frac{C}{M} \\ \frac{v^2\tan\delta}{L} \end{pmatrix}. \tag{4.20}$$

Afin d'assurer le suivi d'une trajectoire de référence $(x_r(t), y_r(t))$, on doit poser :

$$\mathcal{R}(\psi) \begin{pmatrix} \frac{C}{M} \\ \frac{v^2\tan\delta}{L} \end{pmatrix} = \begin{pmatrix} \ddot{x}_r - \lambda_x\dot{e}_x - \mu_x e_x \\ \ddot{y}_r - \lambda_y\dot{e}_y - \mu_y e_y \end{pmatrix} \tag{4.21}$$

avec

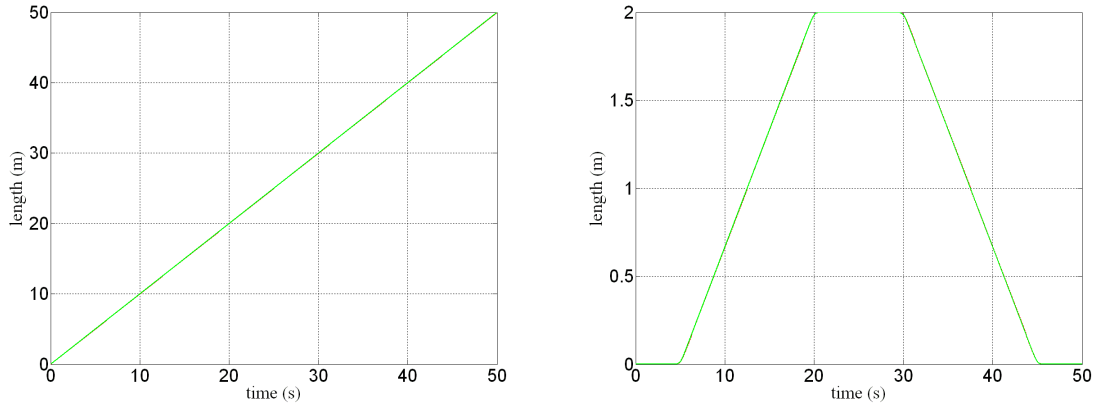
$$e_x = x - x_r,\ e_y = y - y_r.$$

D'où les expressions des commandes :

$$\begin{pmatrix} C \\ \tan\delta \end{pmatrix} = \begin{pmatrix} M & 0 \\ 0 & \frac{L}{v^2} \end{pmatrix} \mathcal{R}^{-1}(\psi) \begin{pmatrix} \ddot{x}_r - \lambda_x\dot{e}_x - \mu_x e_x \\ \ddot{y}_r - \lambda_y\dot{e}_y - \mu_y e_y \end{pmatrix} \tag{4.22}$$

avec

$$v = \sqrt{\dot{x}^2 + \dot{y}^2},\quad \cos\psi = \cos\left(\arctan\left(\frac{\dot{y}}{\dot{x}}\right)\right),\quad \sin\psi = \frac{\dot{x}}{\dot{y}\sqrt{1+\left(\frac{\dot{y}}{\dot{x}}\right)^2}}$$
$$= \frac{1}{\sqrt{1+\left(\frac{\dot{y}}{\dot{x}}\right)^2}} = \frac{\dot{x}}{\sqrt{\dot{x}^2+\dot{y}^2}} \qquad = \frac{\dot{x}^2}{\dot{y}\sqrt{\dot{x}^2+\dot{y}^2}}.$$



(a) (b)

FIGURE 4.3: *Courbes des positions, de type* log *de* cos *hyperbolique : (a) trajectoire en x, (b) trajectoire en y, (rouge) référence et (vert) simulation .*

On obtient :

$$\begin{pmatrix} C \\ \tan\delta \end{pmatrix} = \begin{pmatrix} M & 0 \\ 0 & \frac{L}{v^2} \end{pmatrix} \begin{pmatrix} \cos\psi\left(\ddot{x}_r - \lambda_x\dot{e}_x - \mu_x e_x\right) + \sin\psi\left(\ddot{y}_r - \lambda_y\dot{e}_y - \mu_y e_y\right) \\ -\sin\psi\left(\ddot{x}_r - \lambda_x\dot{e}_x - \mu_x e_x\right) + \cos\psi\left(\ddot{y}_r - \lambda_y\dot{e}_y - \mu_y e_y\right) \end{pmatrix}. \quad (4.23)$$

On obtient finalement les lois en boucle fermée :

$$C = M\left[\frac{\dot{x}}{\sqrt{\dot{x}^2+\dot{y}^2}}\left(\ddot{x}_r - \lambda_x\dot{e}_x - \mu_x e_x\right) + \frac{\dot{x}^2}{\dot{y}\sqrt{\dot{x}^2+\dot{y}^2}}\left(\ddot{y}_r - \lambda_y\dot{e}_y - \mu_y e_y\right)\right], \quad (4.24a)$$

$$\delta = L\left[\frac{-\dot{x}^2}{\dot{y}\left(\dot{x}^2+\dot{y}^2\right)^{3/2}}\left(\ddot{x}_r - \lambda_x\dot{e}_x - \mu_x e_x\right) + \frac{\dot{x}}{\left(\dot{x}^2+\dot{y}^2\right)^{3/2}}\left(\ddot{y}_r - \lambda_y\dot{e}_y - \mu_y e_y\right)\right], \quad (4.24b)$$

qui assurent le suivi de la trajectoire de référence $\left(x_r(t), y_r(t)\right)$.

4.2.4 Simulation

Cas d'une trajectoire de référence en log de cosinus hyperbolique

Considérons une trajectoire de référence en log cosinus hyperbolique selon l'axe des Y et proportionnel à t selon l'axe des X, entre l'instant $t_b = 0\,\text{s}$ et $t_e = 50\,\text{s}$. Plus précisément, nous prenons

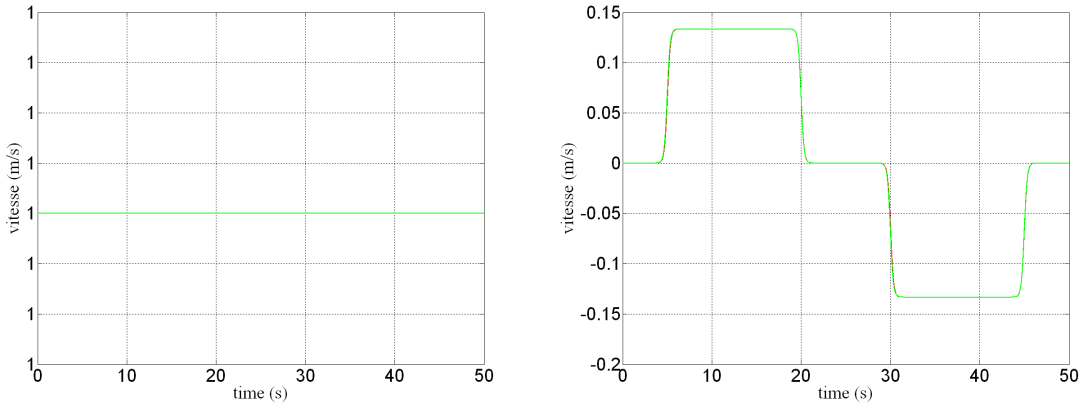
$$y(x) = \frac{y_h - y_l}{2\left(t_e - t_b\right)}\left(\text{logc}_\sigma\left(t - t_b\right) + \text{logc}_\sigma\left(t - t_e\right)\right) + \frac{y_h + y_l}{2}, \quad (4.25a)$$

$$\text{logc}_\sigma = \frac{1}{\sigma}\log\left(\cosh\left(\sigma t\right)\right), \quad (4.25b)$$

et x est proportionnel à t.

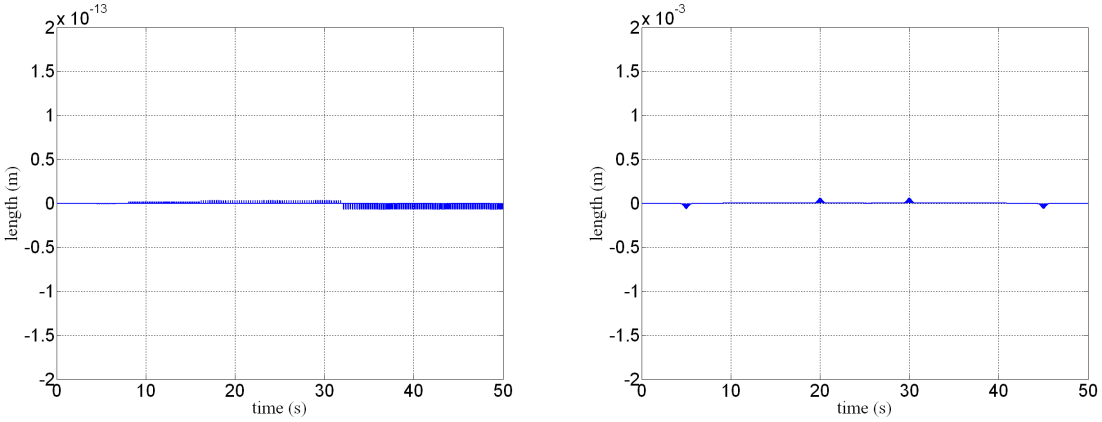
Les figures 4.3 et 4.4 représentent respectivement la trajectoire et la vitesse de référence en rouge. L'état initial est $x_0 = 0$, $y_0 = 0$, $v_0 = 0$ et $\psi_0 = 0$. La réponse du modèle de véhicule simplifié est donnée en 4.3 et 4.4 en vert. Les figures 4.6 et 4.5 mettent en valeur les erreurs observées entre la trajectoire de référence (resp. vitesse de référence) et la trajectoire simulée (resp. vitesse simulée).

Les commandes de couple moteur C et d'angle de braquage δ sont données en figure 4.7.



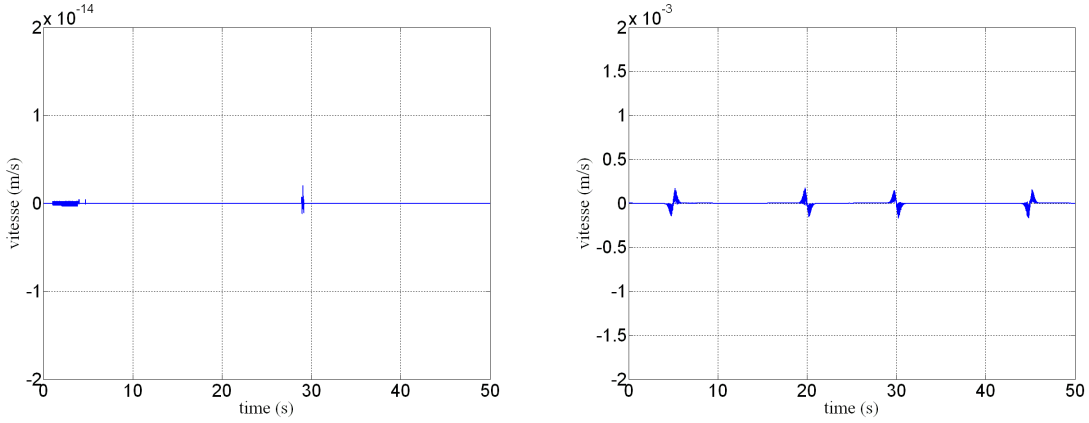
(a) (b)

FIGURE 4.4: *Courbes des vitesses, de type* log *de* cos *hyperbolique : (a) vitesse en x, (b) vitesse en y, (rouge) référence et (vert) simulation.*



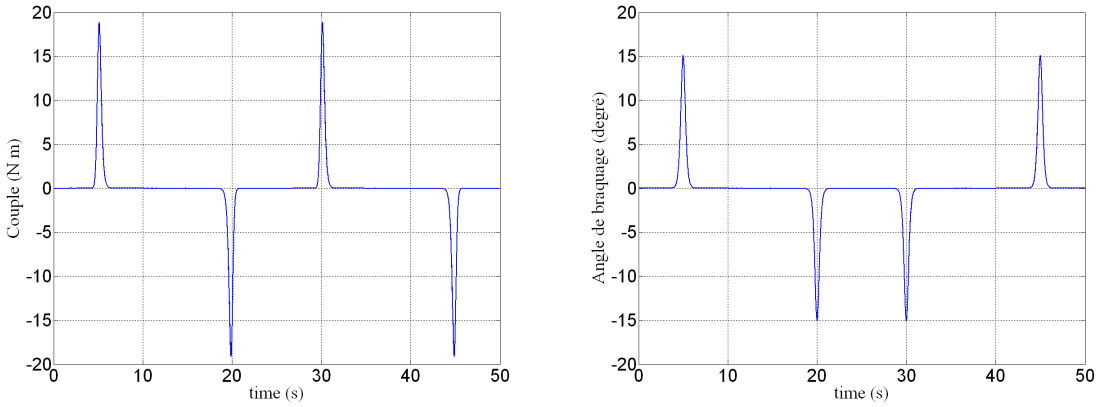
(a) (b)

FIGURE 4.5: *Courbes d'erreurs sur les positions dans le cas du* log *de* cos *hyperbolique, (a) erreur sur la position en x, (b) erreur sur la position en y.*



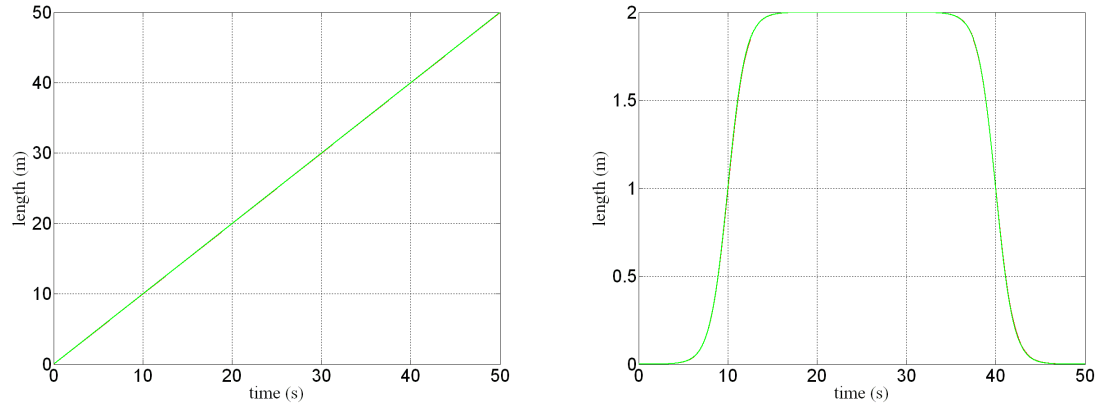
(a) (b)

FIGURE 4.6: *Courbes d'erreurs sur les vitesses dans le cas du* log *de* cos *hyperbolique, (a) erreur sur la vitesse en x, (b) erreur sur la vitesse en y.*



(a) (b)

FIGURE 4.7: *(a) Couple moteur et (b) angle de braquage pour une trajectoire de type log de cos hyperbolique.*



(a) (b)

FIGURE 4.8: *Courbes des positions, de type tangente hyperbolique : (a) position en x, (b) position en y, (rouge) référence et (vert) simulation.*

Cas d'une trajectoire de référence de type tangente hyperbolique

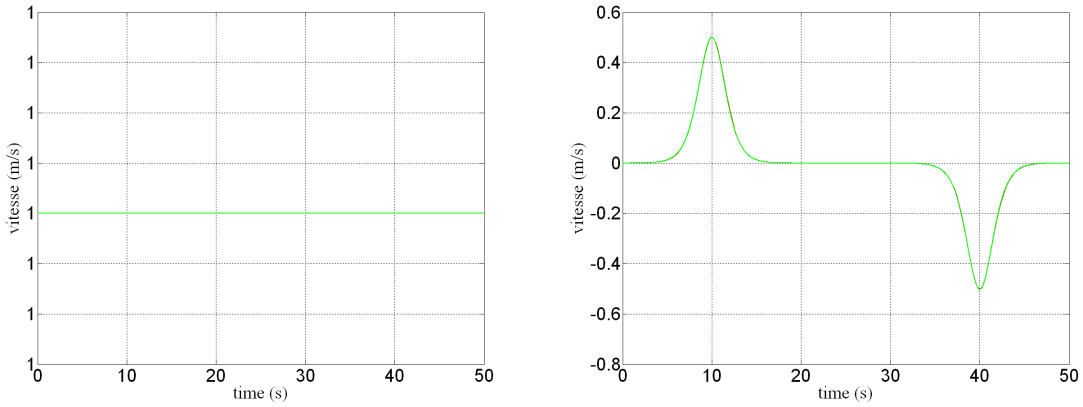
Considérons une trajectoire de référence en tangente hyperbolique selon l'axe des Y et proportionnel à t selon l'axe des X, entre l'instant $t_0 = 0\,\mathrm{s}$ et $t_e = 50\,\mathrm{s}$. Plus précisément, nous prenons

$$y(x) = \frac{y_h - y_l}{2(t_e - t_b)} \left(\tanh(\sigma(t - t_b)) + \tanh(-\sigma(t - t_b))\right) \tag{4.26}$$

et x est proportionnel à t.

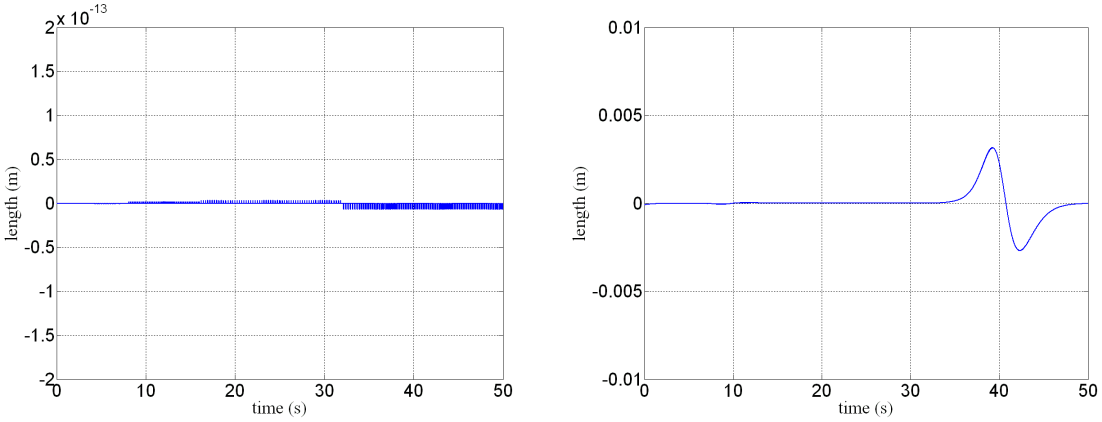
Les figures 4.8 et 4.9 représentent respectivement la trajectoire et la vitesse de référence en rouge. L'état initial est $x_0 = 0$, $y_0 = 0$, $v_0 = 0$ et $\psi_0 = 0$. La réponse du modèle de véhicule simplifié est donnée en 4.8 et 4.9 en vert. Les figures 4.11 et 4.10 mettent en valeur les erreurs observées entre la trajectoire de référence (resp. vitesse de référence) et la trajectoire simulée (resp. vitesse simulée).

Les commandes de couple moteur C et d'angle de braquage δ sont données en figure 4.12.



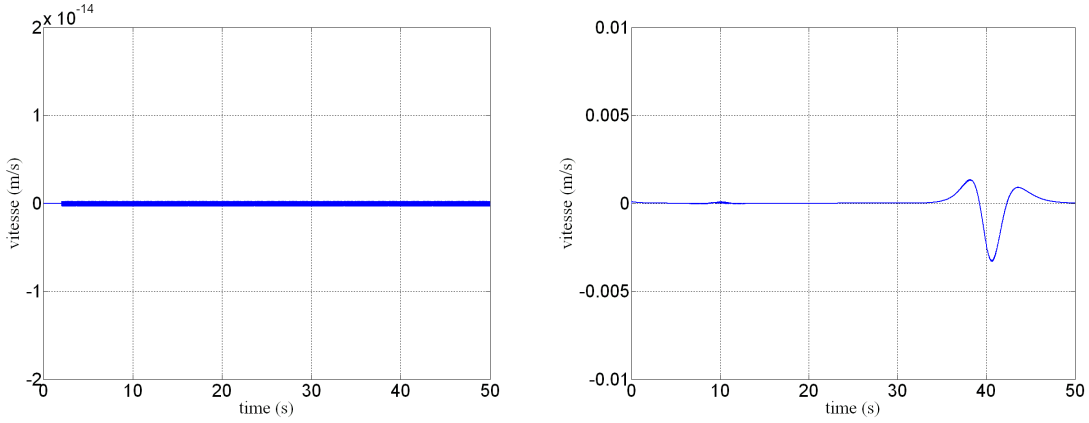
(a) (b)

FIGURE 4.9: *Courbes des vitesses, de type tangente hyperbolique : (a) vitesse en x, (b) vitesse en y, (rouge) référence et (vert) simulation.*



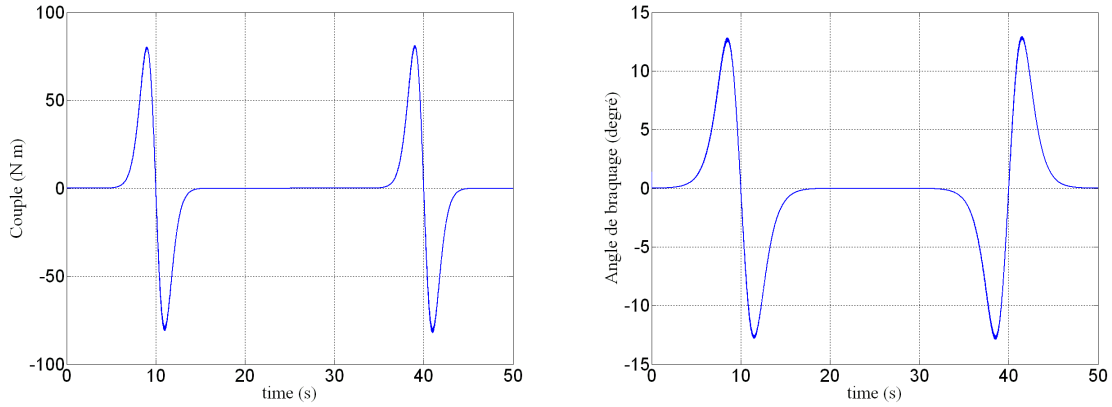
(a) (b)

FIGURE 4.10: *Courbes d'erreurs sur les positions dans le cas du tangente hyperbolique, (a) erreur sur la position en x, (b) erreur sur la position en y.*



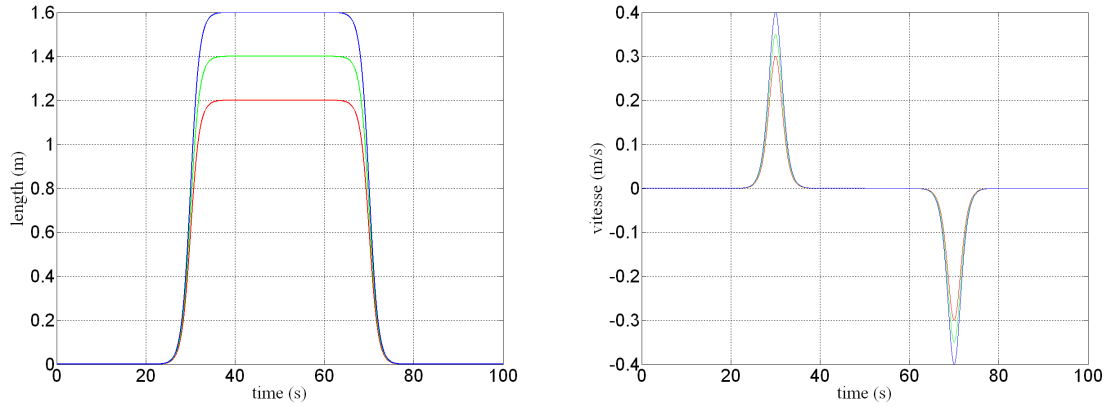
(a) (b)

FIGURE 4.11: *Courbes d'erreurs sur les vitesses dans le cas du tangente hyperbolique, (a) erreur sur la vitesse en x, (b) erreur sur la vitesse en y.*



(a) (b)

FIGURE 4.12: *(a) Couple moteur et (b) angle de braquage pour une trajectoire de type tangente hyperbolique.*



(a) (b)

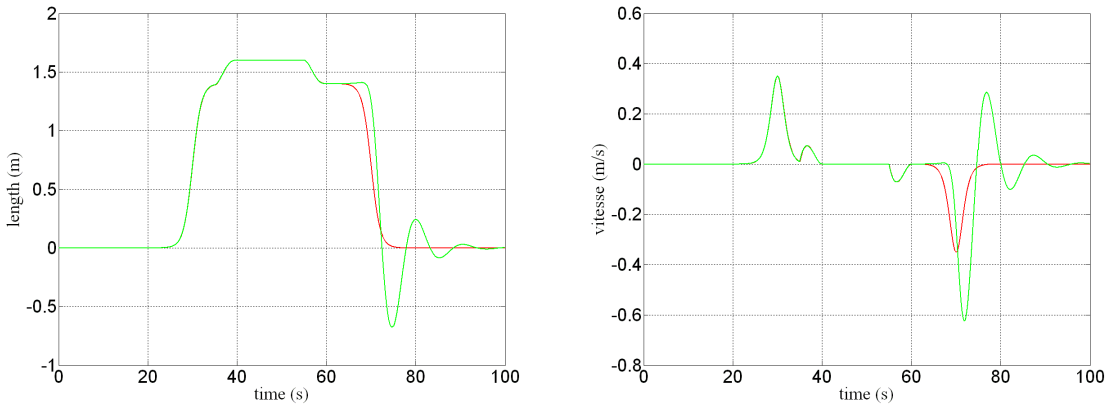
FIGURE 4.13: *Faisceau de trajectoires de référence, de type tangente hyperbolique : (a) faisceau des positions en y, (b) faisceau des vitesses en y.*

Cas d'un faisceau de trajectoires de référence de type tangente hyperbolique

Considérons un faisceau de trajectoires de référence en tangente hyperbolique selon l'axe des Y et proportionnel à t selon l'axe des X, entre l'instant $t_0 = 0\,\text{s}$ et $t_e = 100\,\text{s}$. Nous créons ce faisceau en se référant à l'équation (4.26), et en choisissant différentes valeurs de y_h pour chaque trajectoire. Les figures 4.13(a-b) représentent le faisceau de trajectoires de références en positions et en vitesses. Nous créons un scénario de changement de trajectoire parmi ce faisceau, et l'interconnexion entre deux trajectoires est réalisée suivant une trajectoire de type polynomial.

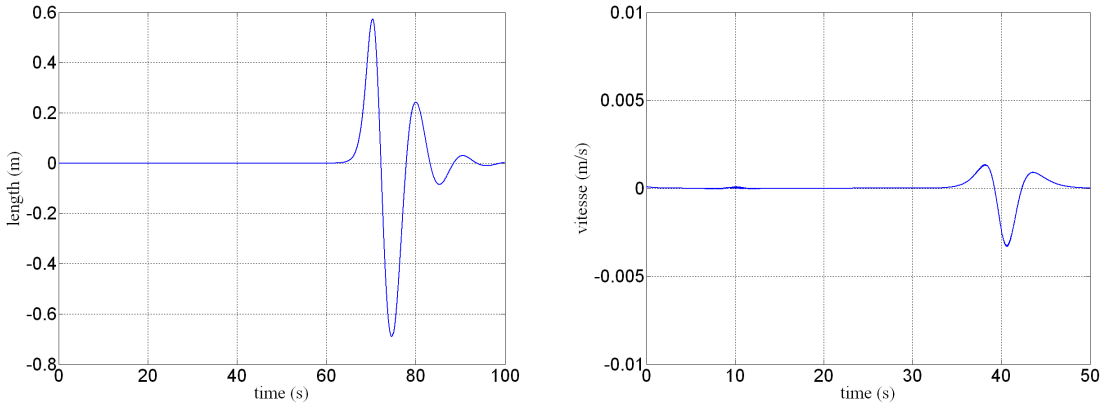
Les figures 4.14(a-b) représentent respectivement la trajectoire et la vitesse de référence en rouge. La réponse du modèle de véhicule est donnée en 4.14(a-b) en vert. Les figures 4.15(a-b) mettent en valeur les erreurs observées entre la trajectoire de référence choisie parmi le faisceau et la trajectoire de transition (resp. vitesse de référence) et la trajectoire simulée (resp. vitesse simulée).

Les commandes de couple moteur C et d'angle de braquage δ sont données en 4.16.



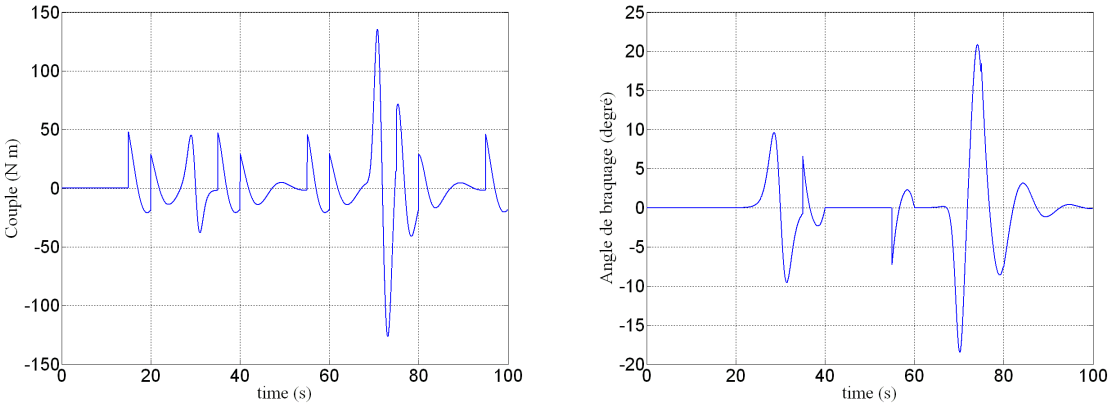
(a) (b)

FIGURE 4.14: *Courbes des positions et des vitesses, dans le cas d'un faisceau de trajectoires de type tangente hyperbolique : (a) positions en y, (b) vitesses en y, (rouge) référence et (vert) simulation.*



(a) (b)

FIGURE 4.15: *Courbes d'erreurs sur les positions et les vitesses, dans le cas d'un faisceau de trajectoires de type tangente hyperbolique, (a) erreur sur la position en y, (b) erreur sur la vitesse en y. .*



(a) (b)

FIGURE 4.16: *(a) Couple moteur et (b) angle de braquage pour une trajectoire de type tangente hyperbolique.*

4.3 Conclusion

Nous avons proposé une technique de planification volontairement simple parce que conduisant à des calculs rapides. Un ensemble de situations possibles, matérialisées par un faisceau de trajectoires, est élaboré pour tenir compte des divers cas susceptibles de se produire. Le passage d'un membre du faisceau à un autre se fera en fonction de la valeur d'indices de vigilance, élaborés en sortie de l'étage de vision.

Le suivi d'un faisceau avec passage d'un membre à l'autre a été construit en utilisant la propriété de la platitude (très bien adaptée au problème de suivi) d'un modèle cinématique augmenté.

Chapitre 5

Conclusion générale et perspectives

Les systèmes ADAS sont aujourd'hui encore l'objet de nombreuses recherches. Dans cette étude, nous avons voulu montrer comment le contexte devait être pris en compte par de tels systèmes, et nous nous sommes intéressés à la notion de faisceaux de trajectoires contextuels. Ainsi notre travail a été divisé en deux partie, la première ayant pour objectif d'estimer le contexte et la seconde de le prendre en compte dans la commande du véhicule.

Le contexte auquel nous nous sommes principalement intéressés est l'environnement extérieur. Nous avons distingué deux grands types d'objets ou évènements à prendre en compte dans la mesure du contexte : les objets connus *a priori* et ceux 'inattendus'. La signalisation routière fournit un exemple des premiers, tandis que les vulnérables (piétons, cyclistes) fournissent un exemple des seconds. Supposant que la détection de ces objets d'intérêt se ferait à partir de caméras, nous avons proposé deux contributions en traitement des images.

La principale concerne une approche *a contrario* permettant de détecter des évènements (à l'échelle du groupe de fenêtres) contradictoires par rapport à un modèle. Deux modèles ont été considérés : l'image précédente (projetée à l'instant courant), et une scène avec la route pour seul objet. Notre méthode est sans seuil puisque les évènements détectés sont ceux présentant un maximum de signification.

La seconde contribution concerne un modèle markovien hiérarchique pour la détection et reconnaissance des marquages au sol (signalisation horizontale). Contrairement au cas précédent, cette méthode comprend de nombreux paramètres.

Plusieurs perspectives peuvent être énoncées à ces travaux.

Tout d'abord, les approches proposées ont été principalement validées sur données simulées. Il s'agira donc de les évaluer sur des données réelles et selon différentes configurations, notamment milieu urbain versus campagne, conditions météorologiques (et donc de visibilité) variables. L'implantation des algorithmes doit être optimisée pour diminuer le temps de calcul.

Une perspective plus fondamentale consiste à combiner plusieurs informations. Concernant la détection de la signalisation (horizontale ou verticale), la détection de la route n'a pas été exploitée. Hormis le fait que la signalisation horizontale s'y trouve, la détection des carrefours ou de la géométrie de la route peut guider la détection de certains types de signalisation (de même que signalisation verticale et horizontale sont corrélées dans certains cas, e.g. *stop*). Concernant la détection d'évènements ou objets 'inattendus', le pistage de ces évènements (informations multitemporelles) est un point clé de l'analyse de leur comportement. Un point critique dans le cas de la comparaison avec une image précédent est le cumul d'erreur de la projection de cette image précédent à l'instant courant lorsque l'intervalle de temps s'accroit (et en cas de mouvement de la caméra). Toujours est-il que nous souhaitons analyser si la détection d'un véhicule lointain, vu à l'heure actuelle comme non significatif (d'après nos tests sur les données simulées, cf. section 2.3.3), deviendrait significatif en raison du caractère régulier (non aléatoire) de sa trajectoire.

En planification de trajectoires, nous avons élaboré une technique privilégiant la simplicité où un faisceau de trajectoires s'offre à chaque instant comme possibilités d'évolution pour le véhicule. La bifurcation vers un autre membre du faisceau se faisant en fonction d'indices de vigilance liés au contexte externe au véhicule (proximité de vulnérables par exemple). Le suivi d'une trajectoire se réalise en exploitant la propriété de platitude différentielle du modèle cinématique augmenté de véhicule adopté. Plus précisément, un bouclage dynamique endogène permet de linéariser de manière exacte le modèle, ce qui rend la tâche de suivi sur le linéarisé triviale.

Les perspectives de ce travail sont nombreuses et sont directement liées à la genèse du projet Digiteo associé à ce travail de thèse sur l'élaboration de schémas de traitement d'images et de commande

adaptés à des contextes embarqué et temps réel. De tels schémas doivent être d'une part souples, adaptables et à faible coût temporel, de façon à satisfaire les échéances et d'autre part doivent être capables réagir à divers changements d'environnement. Citons quelques-unes de ces perspectives :

1. Élaboration d'un critère d'ordonnancement ou d'un modèle de contrôle de la fusion entre capteurs virtuels (entendus comme sorties des algorithmes de traitements d'images).
2. Réalisation d'un échantillonnage variable, s'adaptant aux conditions externes. On peut tirer parti de la propriété de platitude différentielle des modèles envisagés, afin de régler cet échantillonnage selon les caractéristiques des trajectoires à suivre. L'on sent en effet que plus une trajectoire à suivre est raide (en pente et en courbure, donc géométriquement) et rapide (temporellement), plus les lois de commande à appliquer, ainsi que leurs dérivées, seront elles-mêmes sujettes à de grandes variations. En effet la commande de braquage est directement reliée à la courbure et la commande de propulsion directement reliée à l'accélération longitudinale. Et plus un signal a une pente et/ou une courbure importante, plus l'échantillonnage doit être important, pour une approximation donnée par un bloqueur, par exemple d'ordre zéro. Voir à ce sujet [47] pour une réflexion fort intéressante sur la notion de fréquence instantanée d'un signal, définie comme le produit de la vitesse par sa courbure. Plus précisément, pour un signal $\zeta(t)$, la fréquence y est définie par

   $$\phi_\zeta(t) = \frac{\ddot{\zeta}(t)}{\sqrt{1+\dot{\zeta}^2(t)}}$$

   sachant que $\sqrt{1+\dot{\zeta}^2(t)}$ s'interprète comme une vitesse de parcours d'un point matériel le long de la courbe composée par le signal et que la courbure est égale à

   $$\frac{\ddot{\zeta}(t)}{(1+\dot{\zeta}^2(t))^{3/2}}$$

3. De manière très corrélée au point précédent, moduler la vitesse de traitement. En fonction de la complexité et de l'indice de danger de l'environnement, une commande d'horloge peut ralentir ou accélérer le temps avec lequel les trajectoires sont parcourues. Plus spécifiquement, ayant les commandes en boucle ouverte de la forme

   $$C = \frac{\dot{x}_r\ddot{x}_r + \dot{y}_r\ddot{y}_r}{\sqrt{\dot{x}_r^2+\dot{y}_r^2}}, \qquad \delta = \mathrm{Atan}(L\kappa) = \mathrm{Atan}\left(L\,\frac{\ddot{y}_r\dot{x}_r - \dot{y}_r\ddot{x}_r}{(\dot{x}_r^2+\dot{y}_r^2)^{3/2}}\right)$$

   si l'on considère des trajectoires de référence $x_r(t) = \phi_r(\alpha t)$ et $y_r(t) = \psi_r(\beta t)$ la modulation des paramètres α et β (commandes d'horloge) permet de ralentir ou accélérer le temps de parcours de x_r et y_r.
4. Réaliser une adaptation au modèle, éventuellement variable. Selon la situation de vie du système, tel ou tel type de modèle sera envisagé. Pour un véhicule, on distinguera typiquement des cas de parking, de conduite urbaine et de conduite autoroutière. D'autres types de modèles, comme des modèles de type simple voie, incluant des inerties et des modèles de force, pourraient être envisagés. Plus spécifiquement, un tel modèle serait de la forme :

$$
\begin{aligned}
M(\dot{V}_x - V_y\dot{\psi}) &= F_{xf}\cos\delta + F_{xr} - F_{yf}\sin\delta \\
M(\dot{V}_y + V_x\dot{\psi}) &= F_{xf}\sin\delta + F_{yf}\cos\delta + F_{yr} \\
I_z\ddot{\psi} &= L_f F_{xf}\sin\delta + L_f F_{yf}\cos\delta - L_r F_{yr} \\
I_f\dot{\omega}_f = T_f - rF_{xf} &\qquad I_r\dot{\omega}_r = T_r - rF_{xr}
\end{aligned}
$$

Les variables d'un tel modèle sont : V_x et V_y les vitesses longitudinale et latérale, ψ l'angle de lacet, F_{xf} et F_{yf} (resp. F_{xr} et F_{yr}) les forces d'adhérences pneumatique/sol longitudinale et latérale avant (resp. arrière), δ l'angle de braquage des roues avant, ω_f et ω_r les vitesses de roue avant et arrière, T_f et T_r les couples de propulsion/freinage avant et arrière. Et les constantes sont : M la masse du véhicule, I_z l'inertie de lacet, I_f et I_r l'inertie de roue avant et arrière, L_f et L_r les distances du centre de gravité à l'essieu avant et arrière, r le rayon moyen de roue.

5. Pour pallier aux incertitudes de modèles (un modèle cinématique est trop simpliste, les modèles habituels d'adhérence pneumatique/sol sont bien trop approchés), des techniques de type commande sans modèle [50] pourraient être utilisées avec profit. Plus précisément, on considère un modèle par exemple SISO

$$E(t, y, \dot{y}, \ldots, y^{(\iota)}, u, \dot{u}, \ldots, u^{(\kappa)}) = 0$$

où E est une fonction suffisamment régulière en ses arguments. Supposons que pour un certain entier n, $0 < n \leqslant \iota$, $\partial E/\partial y^{(n)} \not\equiv 0$. Le théorème des fonctions implicites donne l'expression, valable localement :

$$y^{(n)} = \mathfrak{E}(t, y, \dot{y}, \ldots, y^{(n-1)}, y^{(n+1)}, \ldots, y^{(\iota)}, u, \dot{u}, \ldots, u^{(\kappa)})$$

Équation que l'on réécrit, en posant $\mathfrak{E} = F + \alpha u$:

$$y^{(n)} = F + \alpha u \tag{5.1}$$

où $\alpha \in \mathbb{R}$ est un paramètre non nécessairement physique, choisi de telle sorte que F et αu soient du même ordre de grandeur ; F est déterminé grâce à la connaissance de u, α, est d'estimées de $y^{(n)}$.

Une estimée de F (qui représente l'ensemble des méconnaissances de modèle) s'obtient comme suit :

$$\widehat{F} = \widehat{y^{(n)}} - \alpha u^{-}$$

où $\widehat{y^{(n)}}$ est une estimée de la dérivée $n^{\text{ième}}$ de la mesure y, et u^{-} est une approximation de u, prise pour éviter les boucles algébriques dans la loi de bouclage résultant. Un choix possible pour u^{-} est la valeur de u à l'échantillon précédent. Le contrôleur obtenu s'écrit alors :

$$u = \frac{1}{\alpha}\left(y_r^{(n)} - \widehat{F} - \Lambda(\mathbf{e}^{\langle -\xi,\zeta\rangle})\right)$$

où y_r est une trajectoire de référence, choisie d'une manière analogue à ce qui est réalisé en commande par platitude ; $e = y_r - y$ est l'erreur de suivi ; $\mathbf{e}^{\langle -\xi,\zeta\rangle} = (\int^{\xi} e, \int^{\xi-1} e, \ldots, e, \dot{e}, e^{(\zeta)})$, $\xi, \zeta \in [0, n]$, $\int^k$ est la $k^{\text{ième}}$ intégrale itérée, et Λ est une fonction telle que la dynamique d'erreur

en boucle fermée

$$e^{(n)} = -\Lambda(\mathbf{e}^{\langle -\xi, \zeta \rangle})$$

soit stable.

6. Hiérarchiser les objectifs d'un point de vue temporel. Divers horizons temporels pourraient être envisagés ; typiquement une échéance à court terme de l'ordre de la seconde, une à moyen terme de la dizaine de secondes, et une à plus long terme de l'ordre de la minute et plus. Ces horizons seraient associés à des objectifs de niveaux sémantiques croissants, utilisant des techniques allant de la régulation simple et robuste à de la prise de décision haut niveau en passant par des techniques de suivi utilisant la platitude différentielle. Les objectifs pourront être une trajectoire à court terme, un faisceau de trajectoires à moyen terme et un faisceau de faisceaux à long terme.

Bibliography

[1] M.S. Allili and D. Ziou. Active contours for video object tracking using region, boundary and shape information. *Signal, Image and Video Processing*, 1(2):101–117, 2007.

[2] M.S. Allili and D. Ziou. Object tracking in videos using adaptive mixture models and active contours. *Neurocomputing*, 71:2001–2011, June 2008.

[3] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):502–507, 2003.

[4] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[5] Y. Aoyagi and T. Asakura. A study on traffic sign recognition in scene image using genetic algorithms and neural networks. In *Proceedings of the 22nd International Conference on Industrial Electronics, Control, and Instrumentation (IEEE IECON)*, volume 3, pages 1838–1843, Aug. 1996.

[6] R. Aufère, F. Marmoition, R. Chapuis, F. Collange, and J.P. Dérutin. Détection de route et suivi de véhicules par vision pour l'acc. *Traitement du Signal*, 17(3):233–248, 2000.

[7] R. Aufrere, R. Chapuis, and F. Chausse. A model-driven approach for real-time road recognition. *Machine Vision and Applications*, 13:95–107, 2001.

[8] A. Bak. *Cooperation stereo mouvement pour la detection des objets dynamiques.* PhD thesis, Université Paris-Sud (France), 2011.

[9] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man and Cybernetics*, 22(2):224–241, 1992.

[10] R. Belaroussi and J.-P. Tarel. Détection des panneaux de signalisation routière par accumulation bivariée. *Traitement du Signal*, 27(3):265–296, May 2010.

[11] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48:259–302, 1986.

[12] S.J. Besson, M. Barlaud, and G. Aubert. Detection and tracking of moving objects using a new level set based method. In *Proceedings of the 15th International Conference on Pattern Recognition.*, volume 3, pages 1100–1105, 2000.

[13] R. Bishop. *Intelligent Vehicle Technologies and Trends.* Artech House, Inc., 2005.

[14] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[15] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, 1991.

[16] N. Bouaynaya, W. Qu, and D. Schonfeld. An online motion-based particle filter for head tracking applications. In *proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'05)*, volume 2, pages 225–228, 2005.

[17] Y. Boykov and D.P. Huttenlocher. Adaptive bayesian recognition in tracking rigid objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 697–704, 2000.

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[19] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi. Shape-based pedestrian detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium, (IV'00)*, pages 215–220, 2000.

[20] A. Broggi, C. Caraffi, R.I. Fedriga, and P. Grisleri. Obstacle detection with stereo vision for off-road vehicle navigation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR'05)*, volume 3, page 65. IEEE Computer Society, 2005.

[21] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, pages 211–231, 2005.

[22] J. Byrne, M. Cosgrove, and R. Mehra. Stereo based obstacle detection for an unmanned air vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA'06)*, pages 2830–2835, May 2006.

[23] F. Cao, J. Delon, A. Desolneux, P. Musé, and F. Sur. A unified framework for detecting groups and application to shape recognition. *Journal of Mathematical Imaging and Vision*, 27(2):91–200, 2007.

[24] B. Chazelle and L. Guibas. Visibility and intersection problems in plane geometry. *Discrete & Computational Geometry*, 4:551–581, 1989.

[25] Y. Chen, L.L. Zhu, A. Lin, C. Yuille, and H. Zhang. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In *Proceedings of the Neural Information Processing Systems, (NIPS'07)*, Vancouver, Canada, 2007.

[26] Y. Chen, L.L. Zhu, A. Yuille, and H. Zhang. Unsupervised learning of probabilistic object models (poms) for object classification, segmentation and recognition using knowledge propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1747–1761, 2009.

[27] P. Cheng, Z. Shen, and S.M. LaValle. RRT-based trajectory design for autonomous automobiles and spacecraft. *Archives of Control Sciences*, 11(3–4):167–194, 2001.

[28] H. Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, California Institute of Technology, Pasadena, California, 1996.

[29] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000*, volume 2, pages 142–149, 2000.

[30] P. Coulombeau and C. Laurgeau. Vehicle yaw, pitch, roll and 3d lane shape recovery by vision. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, volume 2, pages 619–625, June 2002.

[31] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

[32] A. de la Escalera, J.M. Armingol, J.M. Pastor, and F.J. Rodriguez. Visual sign information extraction and identification by deformable models for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 5(2):57–68, 2004.

[33] A. De la Escalera, J.M.A. Armingol, and M. Mata. Traffic sign recognition and analysis for intelligent vehicles. *Image and Vision Computing*, 21(3):247–258, 2003.

[34] A. de la Escalera, L.E. Moreno, M.A. Salichs, and J.M. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6):848–859, 1997.

[35] M. Defoort, T. Floquet, A. Kökösy, and W. Perruquetti. Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach. *Robotics and Autonomous Systems*, 57(11):1094–1106, 2009.

[36] A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40:7–23, 2000.

[37] A. Desolneux, L. Moisan, and J.-M. Morel. Edge detection by helmholtz principle. *Journal Mathematical Imaging and Vision*, 14(3):271–284, May 2001.

[38] A. Desolneux, L. Moisan, and J.-M. Morel. Computational destalts and perception thresholds. *Journal of Physiology*, 97:311–324, 2003.

[39] A. Desolneux, L. Moisan, and J.-M. Morel. A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):508–513, 2003.

[40] A. Desolneux, L. Moisan, and J.-M. Morel. Gestalt theory and computer vision. In Arturo Carsetti, editor, *Seeing, Thinking and Knowing*, volume 38 of *Theory and Decision Library A:*, pages 71–101. Springer Netherlands, 2004.

[41] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis*. Springer-Verlag, 2008.

[42] F. Dibos, S. Pelletier, and G. Koepfler. Real-time segmentation of moving objects in a video sequence by a contrario detection. In *Proceedings of the IEEE International Conference on Image, (ICIP'05)*, volume 1, pages 1065–8, Sept. 2005.

[43] P. Douville. Real-time classification of traffic signs. *Real-Time Imaging*, 6(3):185–193, 2000.

[44] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[45] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background substraction. In *Proceedings of the Computer Vision, (ECCV'00)*, volume 1843 of *Lecture Notes in Computer Science*, pages 751–767, 2000.

[46] C.Y. Fang, S.W. Chen, and C.S. Fuh. Road-sign detection and tracking. *IEEE Transactions on Vehicular Technology*, 52(5):1329–1341, 2003.

[47] M. Fliess. Réflexions sur la question fréquentielle en traitement du signal. Technical report, Laboratoire d'informatique de l'école polytechnique-LIX. eprint arXiv:cs/0510084.

[48] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Sur les systèmes non linéaires différentiellement plats. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 315(5):619–624, 1992.

[49] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[50] M.l Fliess and C. Join. Commande sans modèle et commande à modèle restreint. *e-STA*, 5(4):1–23, 2008.

[51] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(1):23–33, 1997.

[52] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 2006.

[53] F. Gaillard, M. Soulignac, C. Dinont, and P. Mathieu. Deterministic kinodynamic planning with hardware demonstrations. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems, (IROS'11)*, pages 3519–3525, sept. 2011.

[54] T. Gandhi and M.M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):413–430, 2007.

[55] X.W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova. Recognition of traffic signs based on their colour and shape features extracted using human vision models. *Journal of Visual Communication and Image Representation*, 17(4):675–685, 2006.

[56] M.A. Garcia-Garrido, M.A. Sotelo, and E. Martm-Gorostiza. Fast traffic sign detection and recognition under changing lighting conditions. In *Proceedings of the IEEE Intelligent Transportation Systems Conference , (ITSC'06)*, pages 811–816, sept. 2006.

[57] D.M. Gavrila. Multi-feature hierarchical template matching using distance transforms. In *Proceedings of the Fourteenth International Conference on Pattern Recognition*, volume 1, pages 439–444, Aug. 1998.

[58] D.M. Gavrila. Sensor-based pedestrian protection. *IEEE Intelligent Systems*, 16(6):77–81, 2001.

[59] D. Geman. Random fields and inverse problems in imaging. *Lectures Notes in Mathematics*, 1427:113–193, 1990.

[60] S. Geman and D. Geman. Stochastic relaxation gibbs distribution and bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[61] D. Geronimo, A.M. Lopez, A.D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1239–1258, 2010.

[62] B. Grosjean and L. Moisan. A-contrario detectability of spots in textured backgrounds. *Journal of Mathematical Imaging Vision*, 33(3):313–337, March 2009.

[63] V. Hagenmeyer and E. Delaleau. Continuous-time non-linear flatness-based predictive control: an exact feedforward linearisation setting with an induction drive example. *International Journal of Control*, 81:1645–1663, 2008.

[64] I. Haritaoglu, D. Harwood, and L. Davis. W4: A real time system for detecting and tracking people. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 962, June 1998.

[65] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[66] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[67] Z. Hu, F. Lamosa, and K. Uchimura. A complete u-v-disparity study for stereovision based 3d driving environment analysis. In *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, (3DIM)*, pages 204–211, June 2005.

[68] F. Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche. Driving with tentacles: Integral structures for sensing and motion. In Martin Buehler, Karl Iagnemma, and Sanjiv Singh, editors, *The DARPA Urban Challenge*, volume 56 of *Springer Tracts in Advanced Robotics*, pages 393–440. Springer Berlin/Heidelberg, 2009.

[69] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Proceeding of the IEEE International Conference on Robotics and Automation*, volume 1, pages 2–7, may 1989.

[70] W.D. Jones. Building safer cars. *Spectrum, IEEE*, 39(1):82–85, Jan. 2002.

[71] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for a non-holonomic mobile robot. In *Proceedings of the International Workshop on Intelligent Robots and Systems (IROS'91)*, volume 3, pages 1236–1241, Nov. 1991.

[72] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions onRobotics and Automation*, 12(4):566–580, 1996.

[73] A Kelly and A. Stentz. Rough terrain autonomous mobility - part 2: An active vision, predictive control approach. *Autonomous Robots*, 5:163–198, 1998.

[74] M. Khatib, H. Jaouni, R. Chatila, and J.P. Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation.*, volume 4, pages 2920–2925, apr. 1997.

[75] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, apr. 1986.

[76] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[77] K. Kyungnam, T.H. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *Proceedings of the International Conference on Image Processing, (ICIP'04)*, volume 5, pages 3061–3064, Oct. 2004.

[78] K. Kyungnam, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, June 2005.

[79] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection on non flat road geometry through 'v-disparity' representation. In *Proceedings of IEEE Intelligent Vehicle Symposium*, volume 2, pages 646–651, 2002.

[80] P. Lamon, S. Kolski, R. Triebel, and W. Burgard. The smartTer for ELROB 2006 - a vehicle for fully autonomous navigation and mapping in outdoor environments. Technical report, Autonomous Systems Lab, Ecole polytechnique Fédérale de Lausanne, Switzerland, Autonomous Intelligent Systems, Albert-Ludwigs-University of Freiburg, Germany, 2006.

[81] J-C. Latombe. *Robot motion planning*. Kluwer Academic Publishers, 1991.

[82] G. Le Besnerais and F. Champagnat. Dense optical flow by iterative local window registration. In *Proceedings of the IEEE International Conference on Image Processing, (ICIP'05)*, volume 1, pages 137–40, Sept. 2005.

[83] S. Le Hégarat-Mascle and C. André. Automatic detection of clouds and shadows on high resolution optical images. *Journal of Photogrammetry and Remote Sensing*, 64(4):351–366, 2009.

[84] Y. Lenoir, Ph. Martin, and P. Rouchon. 2k-pi, the juggling robot. In *Proceedings of Control and Decision Conference*, pages 1995–2000, 1998.

[85] J. Lévine, J. Lottin, and J.-C. Ponsart. A nonlinear approach to the control of magnetic bearings. *IEEE Transactions on Control Systems Technology*, 4(5):524–544, 1996.

[86] J. Lévine, P. Rouchon, G. Yuan, C. Grebogi, B.R. Hunt, E. Kostelich, E. Ott, and J.A. Yorke. On the control of us navy cranes. In *Proceedings of the 4-th European Control conference.*, 1997.

[87] P. Lombardi, M. Zanin, and S. Messelodi. Switching models for vision-based on-board road detection. In *Proceedings of the IEEE Intelligent Transportation Systems*, pages 67–72, Sept. 2005.

[88] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, San Francisco, CA, USA, 1981.

[89] A.-R. Mansouri. Region tracking via level set pdes without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):947–961, 2002.

[90] A. Manzanera and Richefeu J.C. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Proceedings of the Fourth Indian Conference on Computer Vision, Graphics and Image Processing, (ICVGIP'04)*, pages 46–51, Dec. 2004.

[91] P. Martin, S. Devasia, and B. Paden. A different look at output tracking: control of a VTOL aircraft. In *Proceedings of the 33nd IEEE Conference on Decision and Control*, volume 3, pages 2376–2381, dec. 1994.

[92] Ph. Martin and P. Rouchon. Flatness and sampling control of induction motors. In *Proceeding of the International Federation of Automatic Control World Congress*, pages 189–394, San Francisco, 1996.

[93] J.J. Martinez and C. Canudas de Wit. A safe longitudinal control for adaptive cruise control and stop-and-go scenarios. *IEEE Transactions on Control Systems Technology*, 15(2):246–258, 2007.

[94] E. Mémin and P. Pérez. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision*, 46(2):129–155, Feb. 2002.

[95] B. Mirtich and J. Canny. Using skeletons for nonholonomic path planning among obstacles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2533–2540, 1992.

[96] J. Miura, T. Kanda, and Y. Shirai. An active vision system for real-time traffic sign recognition. In *Proceedings of the Intelligent Transportation Systems*, pages 52–57, 2000.

[97] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3):201–218, May 2004.

[98] P. Morin and C. Samson. Trajectory tracking for nonholonomic vehicles. In Krzysztof Kozlowski, editor, *Robot Motion and Control*, volume 335 of *Lecture Notes in Control and Information Sciences*, pages 3–23. Springer Berlin/Heidelberg, 2006.

[99] D.W. Murray and B.F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):220–228, 1987.

[100] R. M. Murray, M. Rathinam, and W. Sluis. Differential flatness of mechanical control systems: A catalog of prototype systems. In *Proceedings of the ASME International Congress and Exposition*, San Francisco, Nov. 1995.

[101] H.H. Nagel. Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In *Proceedings of the first european conference on Computer vision, (ECCV'90)*, pages 139–148, 1990.

[102] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[103] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97:259–282, March 2005.

[104] M. Parent. The car of the future. In Ljubo Vlacic, Michel Parent, and Fumio Harashima, editors, *Intelligent Vehicle Technologies*, pages 3–17. Butterworth-Heinemann, 2001.

[105] R. Pepy, A. Lambert, and H. Mounier. Path planning using a dynamic vehicle model. In *Proceedings of International Conference on Information Communication Technologies from Theory to Applications ICTTA*, 2006.

[106] G. Piccioli, E. De Micheli, and M. Campani. Robust method for road sign detection and recognition. *Image and Vision Computing*, 14(3):209–223, 1996.

[107] F. Porikli and O. Tuzel. Bayesian background modeling for foreground detection. In *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*, VSSN '05, pages 55–58, New York, NY, USA, 2005. ACM.

[108] S. Quinlan. *Real-time modification of collision-free paths.* PhD thesis, CA, USA, 1995.

[109] J. Rabin, J. Delon, and Y. Gousseau. A statistical approach to the matching of local features. *SIAM Journal on Imaging Sciences*, 2(3):931–958, 2009.

[110] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145:367–393, 1990.

[111] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators: the generalized symmetry transform. *International Journal of Computer Vision*, 14:119–130, 1995.

[112] A. Robin, L. Moisan, and S. Le Hégarat-Mascle. An a-contrario approach for subpixel change detection in satellite imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1977–1993, 2010.

[113] R. Rothfuss, J. Rudolph, and M. Zeitz. Flatness based control of a nonlinear chemical reactor model. *Automatica.*, 32:1433–1439, 1996.

[114] P. Rouchon, M. Fliess, J. Lévine, and P. Martin. Flatness, motion planning and trailer systems. In *Proceedings of the 32nd IEEE Conference on Decision and Control.*, volume 3, pages 2700–2705, San Antonio, 1993.

[115] A. Ruta, F. Porikli, S. Watanabe, and Y. Li. In-vehicle camera traffic sign detection and recognition. *Machine Vision and Applications*, 22(2):359–375, Mars 2011.

[116] K. Shin and N. McKay. A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Transactions on Automatic Control*, 31(6):491–500, 1986.

[117] T. Simeon, S. Leroy, and J.P. Laumond. Computing good holonomic collision-free paths to steer nonholonomic mobile robots. In *Proceedings of the 1997 IEEE-RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1004–1010, sep 1997.

[118] H. Sira-Ramirez and S. K. Agrawal. *Differentially Flat Systems.* CRC, 2004.

[119] B. Soheilian, A. Arlicot, and N. Paparoditis. Extraction de panneaux de signalisation routière dans des images couleurs. In *Proceedings of the Reconnaissance des Formes et Intelligence Artificielle*, pages 1–8, France, Jan. 2010.

[120] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2246–2252, 1999.

[121] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006.

[122] N. Suvonvorn, F. Le Coat, and B. Zavidovique. Marrying level-line junctions for obstacle detection. In *Proceedings of the International Conference on Image Processing, (ICIP'07)*, pages 305–308, 2007.

[123] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the darpa grand challenge. *Journal of Field Robotics*, 2006.

[124] O. Tournaire, N. Paparoditis, and F. Lafarge. Rectangular road marking detection with marked point processes. In *Proceedings of the Photogrammetric Image Analysis*, volume 36, pages 149–154, Sept. 2007.

[125] M. Vasiliu and M. Gouiffes. Complex automotive applications. In *GPU Technical Conference, NVidia Research Summit*, San José, USA, Sept. 2010.

[126] P. Venhivens, K. Naab, and B. Adiprasito. Stop and go cruise control. *International journal of automotive technology*, 1(2):61–69, 2000.

[127] R.G. von Gioi, J. Jakubowicz, and J.-M. Morel. LSD: A fast line segment detector with false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, 2010.

[128] R. Wang, Y. Xu, and Y. Zhao. A vision-based road edge detection algorithm. In *IEEE Intelligent Vehicle Symposium*, volume 1, pages 141–147, June 2002.

[129] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, May 2001.

[130] M. Wertheimer. Untersuchungen zur lehre von der gestalt. *Psychological Research*, 1:47–58, 1922.

[131] M. Wertheimer. Untersuchungen zur lehre von der gestalt II. *Psychological Research*, 4:301–350, 1923.

[132] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[133] L.L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1029–1043, 2010.

Table des figures

1.1 Illustration des différents modules du mesure du contexte et du contrôle du véhicule. 7

2.1 Modélisation d'un capteur de stéréovision ([8]). 16
2.2 UV-disparité. 19
2.3 Exemples d'illusions visuelles : (a) illusion de Hering, (b) triangle impossible. 22
2.4 Illustration du principe de Helmholtz pour la détection d'alignement ([38]). 25
2.5 Processus de décision et approche proposée à deux niveaux. 27
2.6 Exemple de détection pixelique à base de niveau de gris. 30
2.7 Exemple de détection pixelique selon le critère NFA_1 (pour images à niveau de gris). 32
2.8 Approximation de Hoeffding de la queue binomiale. 35
2.9 Exemple de détection niveau fenêtre. 37
2.10 Illustration de fenêtres W_j de significativité maximale en cas de recouvrement avec la fenêtre W_k (seuls des cas avec présence d'un objet sont illustrés). 39
2.11 Exemple de détection niveau fenêtre. 40
2.12 Exemple de résultat de détection de groupement de points dans le cas multitemporel. 41
2.13 Courbes de significativité du sous-ensemble de fenêtres en fonction de son cardinal. . 43
2.14 Exemples de détection niveau fenêtre selon l'algorithme 2.4 et avec la définition des paramètres du modèle de [39]. 43
2.15 Exemple de résultat de détection de points à partir du critère NFA_2. 45
2.16 Exemple de résultat de détection de points à partir du seul critère NFA_2. 46
2.17 Approche proposée : Polyvalence et Généricité de la méthode. 48
2.18 Exemples de résultats obtenus en monotemporel dans le cas d'images simulées sous Pro-SiVIC. 51
2.19 Exemple de résultat dans un cas moins favorable en terme de qualité des cartes de disparité. 53
2.20 Exemple de classification (algorithme 2.5) exploitant les deux *modèles à t* pour séparer les objets mobiles (en bleu) des objets statiques (en vert). 55
2.21 Résultats obtenus sur la séquence par combinaison de la détection monotemporelle par rapport au modèle *scène vierge sauf route* et la détection monotemporelle par rapport au modèle *image radiométrique à* $t-1$ (projetée à t). 56
2.22 Résultats obtenus sur la séquence par combinaison de la détection multitemporelle par rapport au modèle *scène vierge sauf route* et la détection multitemporelle par rapport au modèle *image radiométrique à* $t-1$ (projetée à t). 57
2.23 Résultats obtenus sur la séquence par combinaison de la détection multitemporelle par rapport au modèle *scène vierge sauf route* et la détection multitemporelle par rapport au modèle *image de disparité à* $t-1$ (projetée à t). 57

2.24 Résultats obtenus sur la séquence par combinaison de la détection multitemporelle par rapport au modèle *scène vierge sauf route* et la détection multitemporelle par rapport au modèle *image de disparité à* $t-1$ (projetée à t). 58
2.25 Premiers exemples de résultats sur les données réelles. 60
2.26 Seconds exemples de résultats sur les données réelles. 61
2.27 Exemples de comparaison avec la uv-disparité. 61

3.1 Détection des marquages au sol, approche par niveau : du pixel à l'objet 68
3.2 Distributions empiriques des classes λ_{mk} et $\lambda_{\overline{mk}}$ 70
3.3 Illustration du calcul d'alignement des composantes connexes (CC). La direction de l'axe principal est bien définie pour au moins une des CCs, ou n'est pas bien définie pour aucune des CCs. 72
3.4 Six exemples d'images simulées. 75
3.5 Six exemples d'images en sortie du prétraitement de morphologie mathématique (différence entre entre I et sa reconstruction géodésique $E^I_{ES}(\varepsilon_{ES}(I))$) et en entrée de la classification. 76
3.6 Exemple 1 : (a) classification pixelique, et (b) classification au niveau objet. 76
3.7 Exemple 2 : (a) classification pixelique, et (b) classification au niveau objet. 77
3.8 Exemple 3 : (a) classification pixelique, et (b) classification au niveau objet. 77
3.9 Exemple 4 : (a) classification pixelique, et (b) classification au niveau objet. 77
3.10 Exemple 5 : (a) classification pixelique, et (b) classification au niveau objet. 77
3.11 Exemple 6 : (a) classification pixelique, et (b) classification au niveau objet. 78

4.1 Robot à roue avant folle. 89
4.2 Modèle cinématique d'un véhicule automobile. 92
4.3 Courbes des positions, de type log de cos hyperbolique : (a) trajectoire en x, (b) trajectoire en y, (rouge) référence et (vert) simulation 94
4.4 Courbes des vitesses, de type log de cos hyperbolique : (a) vitesse en x, (b) vitesse en y, (rouge) référence et (vert) simulation. 95
4.5 Courbes d'erreurs sur les positions dans le cas du log de cos hyperbolique, (a) erreur sur la position en x, (b) erreur sur la position en y. 95
4.6 Courbes d'erreurs sur les vitesses dans le cas du log de cos hyperbolique, (a) erreur sur la vitesse en x, (b) erreur sur la vitesse en y. 95
4.7 (a) Couple moteur et (b) angle de braquage pour une trajectoire de type log de cos hyperbolique. 96
4.8 Courbes des positions, de type tangente hyperbolique : (a) position en x, (b) position en y, (rouge) référence et (vert) simulation. 96
4.9 Courbes des vitesses, de type tangente hyperbolique : (a) vitesse en x, (b) vitesse en y, (rouge) référence et (vert) simulation. 97
4.10 Courbes d'erreurs sur les positions dans le cas du tangente hyperbolique, (a) erreur sur la position en x, (b) erreur sur la position en y. 97
4.11 Courbes d'erreurs sur les vitesses dans le cas du tangente hyperbolique, (a) erreur sur la vitesse en x, (b) erreur sur la vitesse en y. 97
4.12 (a) Couple moteur et (b) angle de braquage pour une trajectoire de type tangente hyperbolique. 98

4.13 Faisceau de trajectoires de référence, de type tangente hyperbolique : (a) faisceau des positions en y, (b) faisceau des vitesses en y. 98
4.14 Courbes des positions et des vitesses, dans le cas d'un faisceau de trajectoires de type tangente hyperbolique : (a) positions en y, (b) vitesses en y, (rouge) référence et (vert) simulation. 99
4.15 Courbes d'erreurs sur les positions et les vitesses, dans le cas d'un faisceau de trajectoires de type tangente hyperbolique, (a) erreur sur la position en y, (b) erreur sur la vitesse en y. 99
4.16 (a) Couple moteur et (b) angle de braquage pour une trajectoire de type tangente hyperbolique. 99

Liste des tableaux

2.1 Type d'objets détectés selon les combinaisons entre les images d'entrées : en colonne, les possibles *modèle à t* et en ligne les possibles *innovation à t*. 48
2.2 Statistiques globales sur la séquence en fonction de des données en entrée de la méthode de détection. 58

3.1 Codes des couleurs des classes représentées dans les figures 3.6, 3.7, 3.8, 3.9, 3.10, et 3.11 74
3.2 Performance quantitative de la classification : seuil CL_T, pixel CL_P et objet CL_O estimé sur les six images comparées à une vérité de terrain. 78

Liste des algorithmes

2.1 Algorithme de détection au niveau pixel. 29
2.2 Algorithme d'estimation conjointe des pixels *objets* et du mouvement précis du véhicule. 33
2.3 Algorithme de détection niveau fenêtre. 36
2.4 Algorithme de détection niveau fenêtre, cas de fenêtres variables. 39
2.5 Algorithme de classification. 54

Liste des publications

Communications internationales avec actes

1. Moez Ammar, Sylvie Le Hégarat-Mascle, Roger Reynaud and Amandine Robin. Video Scene Object Detection Using An A Contrario Approach. *Workshop on Image Processing Theory, Tools and Applications*), p.92-99, Nov. 23-28, 2008, Sousse, Tunisia.
2. Moez Ammar, Sylvie Le Hégarat-Mascle and Hugues Mounier. Road surface marking classification based on a hierarchical Markov model. *In Proceedings of International Conference on Image Analysis and Recognition ICIAR*, June 22-24, 2011, Burnaby, BC, Canada.
3. Moez Ammar, Sylvie Le Hégarat-Mascle, Marius Vasiliu and Hugues Mounier. An A-contrario Approach for Obstacle Detection in Assistance Driving Systems. *In Proceedings of International Conference on Image Analysis and Recognition ICIAR*, June 25-27, 2012, Aveiro, Portugal.
4. Moez Ammar, Sylvie Le Hégarat-Mascle, Marius Vasiliu and Hugues Mounier. An A-contrario Obstacle Detection from Disparity Analysis using an A-contrario Approach. *In Proceedings International Conference on Image Processing, Computer Vision, and Pattern Recognition*, July 16-19, 2012, Las Veags, Nevada, USA.

Communications sans actes

1. Moez Ammar, Sylvie le Hégarat-Mascle, Hugues Mounier, Roger Reynaud, Brigitte d'Andréa-Novel. ComFuCet : Embedded and real-time control and fusion. Forum annuel Digiteo, 21 oct. 2009, à l'École Polytechnique, France.
2. Moez Ammar, Sylvie le Hégarat-Mascle, Hugues Mounier, Roger Reynaud. Markov model based road surface marking classification. Forum annuel Digiteo, 12 oct. 2010, à l'École Polytechnique, France.