UNIVERSITE BLAISE PASCAL

ECOLE DOCTORALE

SCIENCES POUR L'INGENIEUR

DE CLERMONT-FERRAND

THÈSE

présentée par :

Xavier CLADY
Ingénieur E.N.S.P.S.

Formation Doctorale
Electronique et systèmes

Pour obtenir le grade de

DOCTEUR D'UNIVERSITE

(Spécialité : Vision pour la Robotique)

Contributions à la navigation autonome d'un véhicule automobile par vision

---

Table des matières

Communications

Publications Internationales

X. Clady, F. Collange, F. Jurie et P. Martinet. Tracking with a Pan-Tilt-Zoom Camera for an ACC System. In *Proceeding of the 12th Scandinavian Conference on Image Analysis*, Bergeen, Norvège, pages 561-566, Juin 2001 . SCIA'2001.

X. Clady, F. Collange, F. Jurie et P. Martinet. Object Tracking with Pan-Tilt-Zoom Camera : application to car driving assistance. In *Proceedings of International Conference on Robotic and Automation*, pages 1653-1658, Séoul, Corée, Mai 2001 . ICRA'2001.

Publications Nationales

T. Chateau, F. Jurie, M. Dhome et X. Clady. Real-time Tracking using Wavelet Representation. In *Proceeding of the Deutsche Arbeitsgemeinschaft für Mustererkennung -Symposium*, pages 523-530, Zürich, Suisse, Septembre 2002 . DAGM'2002.

X. Clady, F. Collange, F. Jurie, B. Thuilot et P. Martinet. Détection et Suivi de Véhicule par Vision. colloque GRETSI'01, Toulouse, France, Septembre 2001.

X. Clady. Détection et Suivi de Véhicule dans une Séquence d'Images Autoroutières. *Journées Jeunes Chercheurs en Robotique*, pages 50-55, Bourges , France, 3-4 février 2000.

Rapport Technique

X. Clady, C. Blanc, L. Trassoudaine, D. Aubert, F. Le Coat, G. Yahiaoui, F. Nashashibi, A. Bensrhair et S. Mousset Etat de l'art en détection d'obstacles par perception. *ARCOS*, PSI-INSA de Rouen, PSI-INSA/R1.3-v1.0, mai 2002.

Communications Orales

X. Clady Estimation de la pose d'un objet lors de son suivi avec une caméra PTZ. *Journée Scientifique de l'Ecole Doctorale Sciences Pour l'Ingénieur*, Clermont-Ferrand, France, 13 décembre 2001.

X. Clady Suivi d'objet avec une caméra Pan-tilt-Zoom. *GDR-ISIS : GT 5 : Asservissement visuel et vision active*, Rennes, France, 14 et 15 décembre 2000.

X. Clady, F. Collange, P. Martinet Suivi de véhicule par vision active. *Journée de travail GT5-Vision — Géométrie (OT5.2) et détection et suivi dans les séquences (OT5.1)*, Grenoble, France, 20 janvier 2000.

Glossaire

ABS	Anti-lock Braking System
ACC	Adaptive Cruise Control
ACP	Analyse en Composantes Principales
AH	Approximation Hyperplane
AHSRA	Advanced Cruise-Assist Highway System Research Association
AJ	Approximation Jacobienne
ARCOS	Action de Recherche Conduite Sécurisée
ASP	Advanced Snowplow Program
Caltrans	California Departement of Transportation
CAN	Control Area Network
CARSENSE	Car awarness for driving via strategy that valuates numerous systems
CCD	Charge Coupled Device
CETE	Centre d'Etude Technique de l'Equipement
CMOS	Complementary Metal-Oxyd Semiconductor
CMU	Carnegie Mellon University
COR	Caractéristique Opérationnelle du Récepteur
CP-DGPS	Carrier-Phase Differential Global Positioning System
DFFS	Distance From Feature Space
DGPS	Differential Global Positioning System
DIFS	Distance In Feature Space
DNN	Discrete Neural Network
EDF	Electricité De France
ESP	Electronic Stability Program
FADE	Fonction d'Analyse et de Détection de l'Environnement
FOC	Focus Of Contraction
FMCW	Frequency Modulation, Continuous Wave
GRAVIR	GRoupe Automatique : Vision et d'Automatique
GPL	Gaz de Pétrole Liquéfié
GPS	Global Positioning System
HMM	Hidden Markov Model

INRETS	Institut National de Recherche sur les Transports et leur Sécurité
IPM	Inverse Perspective Mapping
IRDAR	Infra Red Detection And Ranging
ITS	Intelligent Transportation Systems
LASMEA	LAboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique
Lavia	Limitateur s'Adaptant à la VItesse Autorisée
LIDAR	LIght Detection And Ranging
LCPC	Laboratoire Central des Ponts et Chaussée
MarVEye	Multi-focal active/reactive Vehicle Eye
MIT	Massachusset Institut of Technology
MLP	Multi-layered Perceptron
PAROTO	Projet Anticollision RADAR et Optotronique pour l'Automobile
PATH	Partners for Advanced Transit and Highways
PC	Personnal Computer
PCA	Principal Component Analysis
PREDIT	Programme National de Recherche et d'Innovation dans les Transports Terrestres
PROMETHEUS	PROgraM for European Traffic with Highest Efficiency and Unprecedent Safety
PSA	Peugeot S.A.
PTZ	Pan Tilt Zoom
RADAR	Radio Detection And Ranging
RBF	Radial Basis Functions
ROI	Region Of Interest
ROADSENSE	Road awarness for driving via strategy that valuates numerous systems
SVM	Support Vector Machine
UBM	Universität der Bunderswehr Müchen
UMTS	Universal Mobile Telecommunications System
VaMP	Veruchsfahrzeug für autonome Mobilität PKW
VELAC	Véhicule Expérimental de Lasmea pour l'Aide à la Conduite
WAP	Wireless Application Protocol

Introduction

FIG. 1 – Tête d'un aigle royal [222].

Dans la nature, la perception visuelle est adaptée au mode de vie de chaque espèce animale, lui conférant des aptitudes particulières, différentes de celles des hommes.

Les yeux de l'aigle royal (cf. figure 1) sont dotés d'un puissant système optique d'agrandissement des images. Ils lui permettent d'avoir deux visions de la scène observée. Il a une vision grand angle, bien utile lorsqu'il doit localiser ses proies. Une seconde vision lui permet de «faire le point» sur sa proie (en l'occurence, un lièvre) en grossissant l'image, tel un zoom. Le facteur d'agrandissement est alors de quatre à huit fois celui du reste de l'œil. Cette vision est centrée sur les pattes en position de capture.

Ce système de vision particulier, illustré par la figure 2, fournit à l'aigle des capacités de navigation adaptées à sa technique de chasse.

FIG. 2 – Illustration de la vision d'un aigle [222] : autour des pattes, il s'agit d'une vision grand angle, entre les pattes, une vision «zoomée» sur une proie.

Dans un contexte moins animalier, la perception visuelle est aussi beaucoup étudiée et utilisée pour les systèmes artificiels de navigation, en particulier pour les véhicules dits *intelligents*, c'est-à-dire présentant un certain degré d'autonomie. La capacité d'un tel véhicule à se localiser et à se mouvoir dépend en grande partie de la connaissance de son environnement. La vision artificielle ou vision par ordinateur s'est avérée un moyen privilégié pour acquérir cette connaissance. L'information fournie par une caméra vidéo est en effet très riche. Encore faut-il savoir l'extraire et l'analyser ?

En effet, une des principales difficultés de l'utilisation de capteurs visuels réside dans le traitement de l'information. Réaliser une analyse exhaustive d'une scène réelle par vision est un objectif utopique mais aussi inefficace. Toute l'information contenue dans une image n'est pas utile à l'interprétation de l'environnement d'un véhicule : seuls certains éléments le sont. Une analyse dédiée est donc nécessaire. Par ailleurs, en analogie avec les systèmes de vision naturelle, il est primordial de mettre en adéquation l'architecture matérielle des systèmes de vision et les aptitudes requises à la navigation. Ainsi un système de vision doit se définir suivant la tâche de navigation à effectuer. Il doit particulièrement s'adapter à l'application visée.

Les travaux présentés dans ce mémoire s'inscrivent dans un projet d'« aide à la conduite » mené depuis plusieurs années au sein du groupe GRAVIR (GRoupe Automatique : VIsion et Robotique) du LASMEA (LAboratoire des Sciences et Matériaux pour l'Electronique, et d'Automatique). Ce type de projet vise à accroître la sécurité des transports routiers, et notamment à diminuer le nombre de tués sur la route. En France, ce nombre s'élevait à 7720 en 2001.

Les collisions frontales sont la cause de près d'un tué sur cinq. Une étude de l'INRETS[1] [5] a montré l'importance des distances intervéhiculaires extrêmement faibles : en trafic moyen sur autoroute, près de 50% des véhicules sont au-dessus de la vitesse limite et à moins d'une seconde du véhicule qui précède. D'après cette étude, le nombre de collisions semble corrélé au ratio de temps intervéhiculaires courts (< 2 secondes) dans un trafic autoroutier.

Un système d'aide à la conduite embarqué informant le conducteur de cette distance et le prévenant lorsque cette distance deviendrait trop faible, pourrait donc permettre de réduire le nombre d'accidents ou leur gravité. Un tel système nécessite de localiser les véhicules (souvent qualifiés de «véhicules obstacles») au devant du véhicule équipé. Cette tâche de navigation peut être effectuée via un capteur de vision.

Ce mémoire présente des méthodes et des algorithmes pour détecter et suivre des véhicules, en situation routière, à l'aide de capteurs de type caméra, embarqués dans un véhicule porteur. Ces méthodes sont conçues dans l'optique d'un système de perception autonome : aucune coopération (via des marqueurs visuels, par exemple) n'est requise de la part de l'infrastructure routière ou des autres véhicules.

Nous proposons notamment d'utiliser un capteur combinant une caméra à focale courte et une caméra Pan-Tilt-Zoom (PTZ), commandée en angles site et azimut, et

[1] INRETS, Institut National de Recherche sur les Transports et leur Sécurité.

pilotée en zoom. Ce capteur permet une vision similaire à celle de l'aigle : la première caméra fournissant une vision globale de la scène frontale et la seconde une capture locale mais avec une plus grande résolution.

Ce manuscrit est organisé en quatre chapitres.

Le premier introduit la définition du concept de véhicule intelligent, limité ici aux véhicules de type routier. L'état de l'art sur la perception extéroceptive dans les véhicules intelligents qui suivra, sera l'occasion d'évoquer les principales caractéritisques des différents capteurs utilisés dans de telles applications. Nous pourrons ainsi situer notre capteur par rapport à l'existant dans ce domaine. L'intérêt et le rôle d'un tel capteur pour la capture de la scène frontale seront ainsi définis et développés par rapport à son analyse.

Dans le second chapitre, nous décrirons un algorithme de détection de véhicules par vision monoculaire. A partir des méthodes de détection d'objets, nous distingons deux classes d'approches selon que nous recherchons des «primitives» ou que nous effectuons une classification selon l'«apparence» entre différents objets. La première nécessite le schéma classique de détection puis de comparaison avec un modèle, la seconde nécessite un apprentissage sur une base de données. Nous développons une méthode hybride qui permet d'associer la rapidité de l'approche par primitives et la robustesse de celle basée sur l'apparence.

Le chapitre suivant sera consacré à la description et à l'expérimentation d'un algorithme de suivi temps réel d'objet original développé au sein du LASMEA par *Jurie et Dhome* [107, 108]. Les résultats montrent que cette méthode présente des avantages tels que la généricité de l'approche et le fonctionnement en temps réel. Nous verrons ainsi que cet algorithme est particulièrement bien adapté au suivi de véhicules, et permet une estimation assez précise de la cinématique de ceux-ci, en considérant l'hypothèse de route plane et via un filtrage de Kalman. Cet algorithme est utilisé aussi bien en vision globale, qu'en vision locale, lors de l'asservissement de la caméra PTZ sur un véhicule.

Le dernier chapitre concerne la commande de la caméra PTZ. Nous y établissons un asservissement des positions en angles site et azimut et du zoom de sorte à conserver le véhicule suivi dans l'image. Nous définissons ainsi un module nommé *«PTZ Tracking»*, essentiel à notre capteur. Nous utilisons une approche de type commande référencée capteur, à partir d'informations visuelles fournies par la caméra. Ce chapitre est l'occasion de vérifier la faisabilité et l'utilité de notre capteur pour le suivi d'un véhicule. Il a été en effet testé avec le véhicule VELAC[2] du laboratoire en situation réelle de conduite.

Nous terminerons en donnant quelques perspectives.

[2]VELAC = Véhicule Expérimental du Lasmea pour l'Aide à la Conduite.

Chapitre 1

La perception de l'environnement dans les véhicules intelligents

Dans ce chapitre, nous verrons rapidement comment les nouvelles technologies peuvent être utiles à la Mobilité. Par Mobilité, nous entendons les moyens techniques permettant le déplacement des biens et des personnes.

Ceci nous permettra d'introduire puis de discuter du concept de véhicules intelligents. Nous verrons quelles sont les fonctionalités mises en jeu dans un véhicule en situation de conduite, et comment les différents dispositifs envisagés par les constructeurs et les chercheurs s'y intégrent. Ce sera l'occasion de vérifier l'importance dans ces concepts des systèmes de perception, notamment extéroceptifs.

Aussi la suite de ce chapitre s'intéressera aux différents capteurs extéroceptifs pouvant être utilisés dans le cadre d'un système d'aide à la conduite. Nous aborderons plus particulièrement le cas des capteurs de vision.

Ce qui nous amènera à la description du sujet de cette thèse, c'est-à-dire le développement de différentes méthodes permettant la mise en œuvre d'un nouveau capteur de vision intégrant une caméra grand-angle et une caméra commandée en angles site et azimut et pilotée en zoom, pour la détection et le suivi d'obstacles routiers.

1.1 Généralités : la Mobilité.

La Mobilité est un des grands défis technologiques de la fin du siècle dernier et de ce siècle. Face à l'augmentation générale du trafic en hommes et en marchandises constatée depuis les années 1950, de nombreux et graves problèmes se posent : pollution, congestion (embouteillages), sécurité,...

Ces problèmes sont particulièrement vérifiés dans le cadre du trafic automobile. Et dès le début des années 1970, les gouvernements et les constructeurs ont lancé des études afin d'y remédier. L'intégration de nouvelles technologies permettent d'envisager plusieurs types de solutions [204] :

- utilisation de la télématique embarquée (GPS[1], WAP[2], UMTS[3],...) pour aider le conducteur dans le choix de son itinéraire (augmentation du confort de conduite, diminution des congestions), informer rapidement les services concernés en cas d'accidents ou d'embouteillages,...

- intégration des nouvelles technologies, notamment en communication [10], dans les infrastructures routières pour une meilleure régulation du trafic,...

- développement de véhicules intelligents.

- recherche de nouvelles sources énergétiques (moteur électrique, hybride) ou une meilleure exploitation de celles existantes comme le GPL[4] afin de diminuer la pollution induite par la circulation.

- diminution du poids des véhicules par l'intégration de nouveaux matériaux (composites, aluminium,...) et par l'amélioration du réseau de communication interne (diminution du poids du cablage).

1.2 Les véhicules intelligents

Parmi tous ces projets, il faut distinguer ceux qui consistent à mettre en œuvre des systèmes d'assistance embarqués. Ces systèmes ont pour objectif d'accroître les performances des véhicules en terme de confort et de sécurité. Classiquement [161], deux classes sont distinguées :

- les systèmes passifs (Airbag, ceinture de sécurité, carrosserie déformable,...) qui tentent de minimiser les conséquences d'un accident.

- les systèmes actifs (ABS[5], ESP[6] , ACC[7],...) qui interviennent en amont, dans la prévention ou l'évitement d'une situation dangereuse.

Ils participent au développement de *véhicules* dits *intelligents*.

[1] GPS : Global Positioning System

[2] WAP : Wireless Application Protocol

[3] UMTS : Universal Mobile Telecommunications System

[4] GPL, Gaz de Pétrole Liquéfié

[5] ABS, *Anti-lock Braking System* : système évitant le blocage des roues au freinage afin de garder le contrôle directionnel du véhicule.

[6] ESP, *Electronic Stability Program* : système aidant le véhicule à maintenir la trajectoire voulue par le conducteur. Il détecte la moindre tendance au dérapage et corrige en agissant sur une ou plusieurs roues par l'intermédiaire des freins ou du moteur.

[7] *ACC, Adaptive Cruise Control* ou régulateur intelligent de vitesse : un véhicule ACC doit réguler sa vitesse de façon à respecter les distances de sécurité par rapport aux véhicules «les plus dangereux» situés au devant.

1.2.1 Qu'est ce qu'un véhicule automobile ?

Avant de présenter le concept de véhicule intelligent, il paraît intéressant de se pencher sur le concept de véhicule automobile en lui même, notamment lorsque celui-ci est en situation de conduite, d'abord d'un point de vue matériel, ensuite d'un point de vue fonctionnel.

Il serait faux de limiter un véhicule à sa seule structure matérielle. Lorsque celui-ci est en situation de conduite, il est avant tout défini par l'interaction de trois composantes :

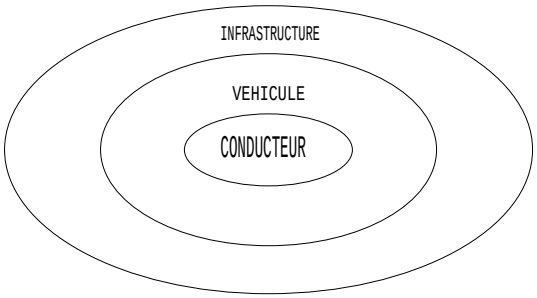


FIG. 1.1 – Représentation matérielle d'un véhicule automobile : l'homme est au centre d'une «trinité» véhicule automobile.

- l'environnement, dans lequel il évolue. C'est-à-dire l'infrastructure routière ou autoroutière (route, pont, signalisation,...), ainsi que les autres utilisateurs de cette infrastructure (voitures, camions, piétons,...).

- la machine, c'est-à-dire le véhicule réduit à sa seule structure matérielle : habitacle, motorisation, actionneurs (freins, direction, accélérateurs),...

- l'homme, c'est-à-dire le conducteur.

La représentation matérielle d'un véhicule automobile de la figure 1.1 illustre bien le fait que l'homme est au centre de cette *«trinité»*, comme il est au centre de l'habitacle de sa voiture. Ceci est vrai d'un point de vue matériel, et se confirme lorsqu'on examine les différentes fonctionnalités mises en œuvre dans un véhicule. Ces différentes fonctionnalités sont une fonction de perception, une de décision et une d'action (ou de commande).

Le schéma 1.2 permet de remarquer que, dans un véhicule classique, le conducteur a la charge de l'essentiel de ces fonctions. Il cumule donc une charge de travail assez importante. Il doit être attentif à l'ensemble de l'environnement et à son véhicule, détecter les moindres changements, prendre les décisions appropriées aux situations rencontrées, et traduire ces décisions en jouant sur les actionneurs du véhicule. Par ailleurs, on peut aussi remarquer que les interactions entre les trois composantes d'un véhicule sont unilatérales : la défaillance d'une composante, en particulier du conducteur, entraîne la

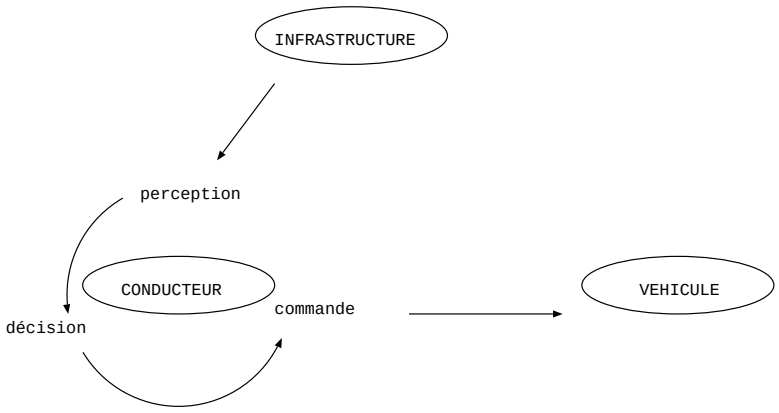


FIG. 1.2 – Représentation fonctionelle d'un véhicule automobile.

défaillance de l'ensemble.

Ainsi, l'amélioration du concept «véhicule automobile» passe par :

- l'allègement de la charge de travail du conducteur, notamment par la délocalisation partielle ou totale de certaines fonctions (perception, décision, commande) au niveau des autres composantes (l'infrastructure ou le véhicule).

- l'amélioration des interactions et des échanges d'informations entre les différents intervenants (fonctionnalités ou composantes) au sein du véhicule, notamment en terme de pertinence.

Les dispositifs *véhicules intelligents* sont la mise en pratique de ces objectifs.

1.2.2 Les différents dispositifs de véhicules intelligents

La plupart des concepts de véhicules intelligents ont été imaginés dans les années 80. Plusieurs programmes nationaux et internationaux, souvent sous une impulsion gouvernementale, ont permis l'exploration et la mise en pratique dans des véhicules expérimentaux de nombreuses solutions possibles et de différentes approches.

Nous allons dans les pages qui suivent, présenter d'abord les principales organisations développant des projets ITS, *Intelligent Transportation Systems*, dont les véhicules intelligents sont un des axes de recherche. Nous ferons un découpage selon la localisation continentale de ces projets (Europe, Etats-Unis, Asie).

Pour chacun, nous donnerons un exemple de démonstrations de conduite en convoi ou *platooning*. Une conduite en convoi consiste à l'asservissement d'un ou plusieurs véhicules esclaves sur un véhicule de tête, nommé maître ou *leader*. Bien qu'il s'agisse d'une application parmi d'autres, les démonstrations présentées ici sont assez représentatives des progrès réalisés grâce aux programmes nationaux et internationaux cités, au cours des 20 dernières années. Elles montrent notamment l'évolution dans l'intégration de

situations de plus en plus complexes et de capteurs de plus en plus nombreux.

Puis, nous énumérerons rapidement les différents dispositifs de véhicules intelligents que ces programmes ont permis de concevoir.

Les grands projets de véhicules intelligents

En Europe, l'exploration des différents dispositifs a été essentiellement amorcée par le projet PROMETHEUS (*PROgraM for European Traffic with Highest Efficiency and Unprecedent Safety*). Il a démarré en 1986. Il a réuni plus de 13 concepteurs automobiles et de nombreux laboratoires de recherche, de 19 pays européens. Une démonstration finale a eu lieu en 1994 à Paris.

Un exemple de ses résultats est, par exemple, la démonstration réalisée par Daimler-Benz et l'UBM (*Universität der Bunderswehr Müchen*). Deux véhicules jumeaux (équipés du même dispositif), de type Mercedes SEL 500, se sont mis en convoi (avec les phases de transition vers ce mode) sur quelques milliers de kilomêtres d'une autoroute à trois voies près de Paris. Elle fut réalisée avec les véhicules test VITA 2 de Daimler-Benz et VaMP (*Veruchsfahrzeug für autonome Mobilität PKW*, cf. figure 1.3) de l'UBM. Ceux-ci étaient équipés de quatre tourelles de capteurs stéréoscopiques (cf. figure 1.4), munies de caméras de focales différentes (8 et 24 *mm*) : deux pour la capture de la scène frontale et deux pour la scène arrière.

FIG. 1.3 – Le véhicule VaMP de l'UBM

FIG. 1.4 – Une tourelle stéréoscopique utilisée dans les véhicules VaMP et VITA 2.

Depuis les recherches dans ce domaine ne se sont pas interrompues en Europe. De nouveaux programmes prennent la relève de PROMETHEUS. Nous pouvons, par exemple, citer les projets ARCOS (Action de Recherche Conduite Sécurisée [39], une des actions du programme PREDIT), PAROTO (Projet Anticollision RADAR[8] et Optotronique pour l'Automobile), ROADSENSE et CARSENSE (*Road/Car awarness for driving via strategy that valuates numerous systems*) qui réunissent plusieurs partenaires parmi les industriels et les centres de recherche autour de sujets de recherche communs. Citons aussi les projets européens CyberCars (*Cybernetic technologies for the car in the city*) et CyberMove (*Cybernetic transportation technologies for the cities*) débutées en 2001 pour une durée de trois ans. Ils visent à évaluer différentes technologies intéressant la navigation, la localisation, le pilotage, la programmation temps réel et la perception de

[8] RADAR, Radio Detection And Ranging

véhicules individuels automatiques pour leur intégration dans un contexte urbain.

Aux Etats-Unis, un grand nombre d'initiatives ont été menées simultanément par des universités, des centres de recherche et par des compagnies automobiles [43]. En 1995, une institution nationale, le *National Automated System Consortia (NAHSC)*, est mise en place par le gouvernement fédéral.

Parmi les intervenants de ce consortium, nous pouvons citer le programme PATH (*Partners for Advanced Transit and Highways*), né d'une collaboration entre le Département Californien des Transports, *California Departement of Transportation (Caltrans)* et l'Université de Californie [94].

(a) un plot magnétique

(b) véhicule et marqueurs magnétiques

FIG. 1.5 – Le système de guidage magnétique PATH

Un des projets phare de ce programme fut la mise en place d'un système de guidage automatique de véhicules sur voies autoroutières dédiées, afin d'améliorer la densité de véhicules y circulant [223]. La régulation de la position latérale des véhicules au sein du convoi est réalisée avec l'aide d'aimants permanents installés dans la chaussée tous les 1.2 mètres (cf. figure 1.5). Des capteurs de flux magnétiques sont installés sous les pare-chocs avant et arrière de chaque véhicule et mesurent le champ magnétique suivant 3 axes. Ce système permet une estimation de la position latérale avec une précision de $5mm$, et de la position longitudinale à $5cm$. L'association de ces données avec celles fournies par un capteur de type RADAR pour la régulation de la distance inter-véhicule (fixée à 6.5 mètres) et par un système de communications radio permettant la transmission des mouvements du véhicule *leader* aux autres, permet la conduite en convoi. En 1997, ces technologies ont permis la mise en convoi de 8 Buicks, tel que l'illustre la figure 1.6.

Ce système est aussi testé depuis 1998 pour la navigation de chasse-neige. C'est le programme ASP, *Advanced Snowplow Program* (cf. figure 1.7). Un moniteur vidéo permet d'informer le conducteur sur sa position par rapport à la chaussée et sur la présence d'obstacles potentiels.

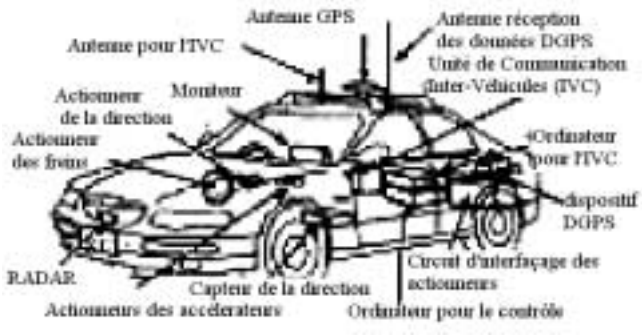
Au Japon aussi, où les problèmes de mobilité sont encore plus accrus, de tels programmes de recherche se sont mis en place. Depuis 1996, un grand nombre d'industries automobiles et de centre de recherches se sont regroupés au sein de l'*Advanced Cruise-Assist Highway System Research Association (AHSRA)*, qui développe différentes ap-

FIG. 1.6 – NAHSC Demo '97, San Diego : démonstration de conduite en convoi.

FIG. 1.7 – ASP : démonstration de navigation de chasse-neige.

proches autour du problème de l'*Automatic Vehicle Guidance*.

L'*ITS Group Research* du *National Institut of Advanced Industrial Science and Technology* a, par exemple, réalisé une démonstration, nommée Demo2000 [197], de *platooning* flexible à faibles distances, incluant cinq véhicules équipés. Les communications inter-véhicules étaient assurées par une ligne radio, la localisation assurée par DGPS[9] et la mesure de la distance inter-véhicule par télémètre laser (cf. figure 1.8). Cette démonstration a eu lieu en novembre 2000 sur une piste à Tsukuba.



(a) équipement d'un véhicule

(b) photo des 5 véhicules

FIG. 1.8 – Les véhicules expérimentaux ayant servi à la Demo 2000.

Les cinq véhicules étaient pilotés automatiquement à des vitesses comprises entre 40 et 60 km.h^{-1}. Les scénari de conduite incluaient des démonstrations de *Stop and Go*, de conduite en convoi, de division en deux convois, d'évitement d'obstacles et de fusion de deux convois en un. La figure 1.9 illustre la fusion de deux convois en un.

[9] DGPS : Differential GPS ; cf. section 1.3.2.

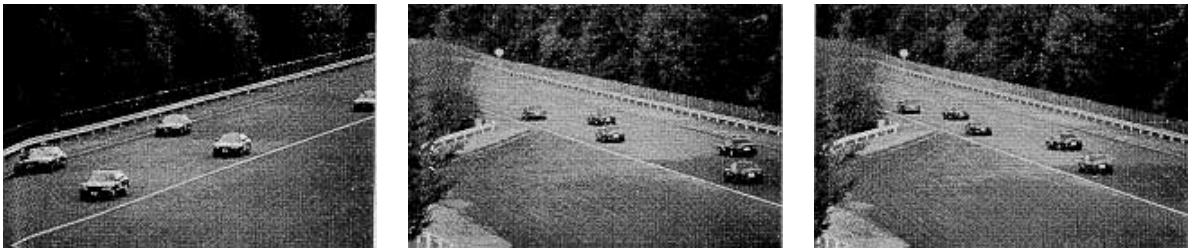


FIG. 1.9 – Fusion de deux convois en un lors de la Demo 2000.

Les dispositifs de véhicules intelligents

Ainsi, durant les 20 dernières années, de nombreux projets dans le monde s'intéressent aux systèmes de véhicules intelligents. Leurs applications concernent les véhicules dits de tourisme, les poids lourds, les transports publics (bus) ou d'autres véhicules (moissoneuse [115], tracteur [49], chasse-neige,...). Richard Bishop propose dans [22] une segmentation en trois classes :

- Les systèmes d'avertissement ou *collision warning systems* : il s'agit de systèmes qui informent ou avertissent le conducteur. Ils peuvent intervenir en cas de risque de collision frontale, pour détecter les ralentissements de véhicules dans la scène frontale, d'empiètement sur les voies latérales, lors d'un changement de voies, pour la détection de piétons, lors d'une marche arrière, pour prévenir une collision arrière ou le basculement pour des véhicules lourds. Il existe une catégorie spéciale de ce type de systèmes qui supervisent l'état du conducteur, pour détecter et l'avertir des cas d'endormissement ou d'autres défaillances.

- Les systèmes d'évitement ou *collision avoidance*, ou semi-automatiques : il s'agit de systèmes qui prennent partiellement le contrôle du véhicule soit dans une manœuvre spécifique soit pour répondre à une situation d'urgence (typiquement, une collision). En effet, si le conducteur ne semble pas répondre efficacement aux avertissements, ces systèmes peuvent agir sur la direction, les freins ou/et l'accélérateur pour réaliser les manœuvres nécessaires à la sécurité du véhicule et de ses occupants. Ces systèmes d'assistance à la conduite peuvent inclure des fonctions, tels que le régulateur de vitesse intelligent, ou *Adaptive Cruise Control* (ACC), suivi automatique de la voie, l'accostement ou autre manœuvre demandant une grande précision d'exécution.

- Les systèmes automatiques ou *vehicle automation* : ce sont des systèmes qui prennent totalement le contrôle du véhicule. Ils peuvent inclure des systèmes d'automatisation à faibles vitesses (par exemple, les systèmes *Stop and Go* qui arrêtent et démarrent automatiquement les véhicules dans les embouteillages), conduite autonome, conduite en convoi ; certains véhicules peuvent aussi être guidés automatiquement dans des zones dédiées, tels que les quais de frêt ou les voies de bus.

Ces dispositifs peuvent être implémentés comme des *systèmes autonomes*, avec toute l'instrumentation et les systèmes de décision à bord du véhicule, ou comme des *systèmes coopératifs*, où une certaine assistance peut être fournie par l'infrastructure routière ou/et par les autres véhicules. La première assistance peut être typiquement l'installation de marqueurs ou autres signalisations facilement détectables ou reconnaissables. La coopération inter-véhicules est particulièrement intéressante lorsque les véhicules doivent manœuvrer à faibles distances les uns des autres. Ceci demande en effet une grande précision. Généralement, celle-ci est obtenue en transmettant les paramètres internes de véhicule(s) clé(s) (par exemple, le *leader* d'un convoi), et ses intentions, aux autres véhicules (dans le cas d'un convoi, il s'agit des véhicules suiveurs).

D'une manière générale, les systèmes autonomes sont conçus pour fonctionner sur toute voie et dans toute situation à un certain niveau de performances, et utilisent la coopération, si elle existe, afin d'accroître ces performances.

Nous avons donc vu que ces différents dispositifs ont besoin d'une perception de l'environnement extérieur au véhicule. Plusieurs types de capteurs extéroceptifs sont étudiés et développés. La section suivante en propose un aperçu.

1.3 Les capteurs extéroceptifs

La perception de l'environnement pour un véhicule intelligent est un problème ardu, du fait de la complexité de la scène à analyser. S'y ajoutent les variations rapides de cet environnement, dues aux conditions d'éclairement non contrôlées et bien sûr au mouvement propre du véhicule équipé ou de ceux partageant son espace d'évolution.

Pour une scène routière (ou autoroutière), deux tâches peuvent être globalement distinguées :

- la perception de l'infrastructure. Il s'agit essentiellement de détecter les voies de circulation et de localiser le véhicule embarqué par rapport à celles-ci. Ces données sont essentielles pour guider le conducteur ou le véhicule lors d'un suivi de trajectoire (suivi de la route, changement de voies,...).

- la détection et la localisation des obstacles. Ils peuvent avoir différentes localisations (en avant, en arrière ou sur les côtés du véhicule équipé) ou être de différentes natures (autres véhicules, piétons, vélos,...). Il est souvent nécessaire de localiser ces obstacles non seulement par rapport au véhicule embarqué, mais aussi par rapport à l'infrastructure. La position des obstacles par rapport aux voies de circulation est une information essentielle pour décider de la «dangerosité» de ceux-ci, comme le démontre la figure 1.10, extraite de [142]. De plus, l'estimation du vecteur cinématique (position, orientation et vitesses) de ces obstacles est souvent requise. Par exemple, elle est primordiale pour l'asservissement longitudinal d'un système de type ACC.

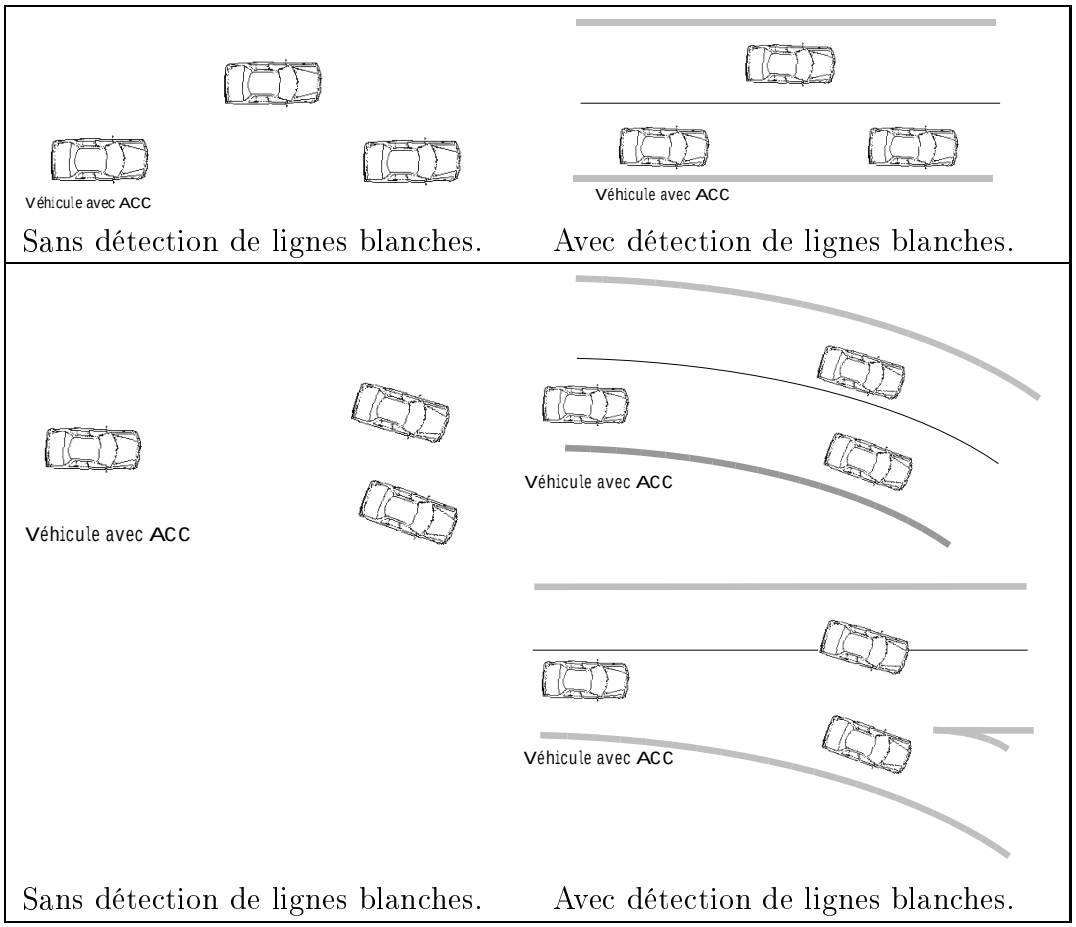


FIG. 1.10 – Quel est l'obstacle le plus dangereux ? [142]

Les différents capteurs étudiés apportent leurs solutions à ces problèmes de perception. Certains sont particulièrement bien adaptés à une tâche spécifique : par exemple, le RADAR et les télémètres, pour la détection d'obstacles. D'autres, comme les caméras, permettent d'effectuer toutes les tâches de perception. C'est ce que nous verrons dans les paragraphes qui suivent.

1.3.1 RADAR, Télémètres, Ultrasons

Les capteurs de type RADAR, télémètres laser ou à ultrasons sont des capteurs dits actifs. Ils sont composés d'un émetteur, qui envoie un rayonnement, et d'un récepteur, qui le reçoit (après réflexion par la scène observée). La comparaison entre l'élément émis et l'élément reçu permet d'établir une image de profondeur. C'est une mesure par temps de vol. L'observation fournie par le capteur est de type (ρ, θ) où ρ représente la distance entre le capteur et le point visé, et θ, l'angle entre le rayon et l'axe du capteur (cf fig.1.11).

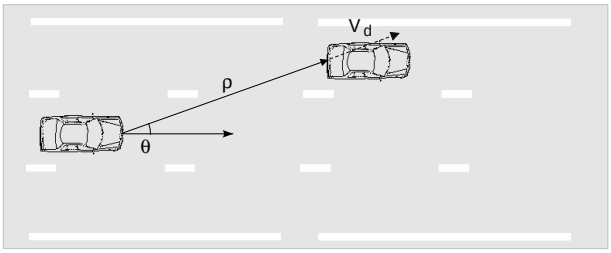


FIG. 1.11 – Mesures RADAR.

Les télémètres laser sont conçus à l'aide de lasers émettant dans le visible (*LIght Detection And Ranging*, LIDAR) ou dans l'infrarouge (*Infra Red Detection And Ranging*, IRDAR). De nombreuses méthodes de mesures de temps de vol pour les télémètres laser sont explicitées dans [87]. Aujourd'hui, la plupart des télémètres lasers renvoient l'intensité du signal pour chaque pixel en mesurant l'énergie du signal laser retourné. Donc, un balayage complet en deux dimensions peut donner une image de profondeur et une image d'intensité. Les obstacles peuvent donc être détectés en observant les discontinuités apparaissant dans l'image de profondeur et dans l'image d'intensité.

La faible divergence du faisceau permet d'obtenir de bonnes résolutions ($4cm$ à $200m$ [193]). Toutefois, ces capteurs ont besoin d'un système de balayage mécanique afin de parcourir différents points de la scène. Ce système est constitué de miroirs plan permettant de diriger le faisceau émis par le laser. Le tableau 1.1 donne un exemple des caractéristiques d'un IRDAR développé au laboratoire [166].

longueur d'onde λ	0,9 μm
divergence du faisceau	approx. $3mrad$ ($1mrad$ correspond à $10cm$ d'écart latéral à $100m$ de distance).
portée	2 à $150m$, pour des objets naturels.
précision	entre $\pm 25mm$
angle de vue	$\pm 40^{\circ}$(pour une ligne, par déplacement des miroirs) entre 0°et 333°par déplacement de la tête.

TAB. 1.1 – Un exemple de capteur IRDAR : LMS-Z210-60 de RIEGL

En 1994, *M. Xie et al.* [213] proposent un système embarqué couplant un télémètre laser à une caméra CCD pour avoir une image d'intensité. Ils détectent premièrement les obstacles dans l'image d'intensité délivrée par la caméra, ensuite le télémètre est utilisé pour valider ou non la présence d'obstacles mais aussi pour obtenir une information de distance des obstacles détectés. Une seconde approche [196], dans laquelle la scène est scrutée par balayage du faisceau laser pour obtenir une image 3D de faible résolution, repose sur la segmentation et l'interprétation des données de profondeur.

Plus récemment, J. Hancock [87] démontre comment l'intensité du laser peut être utilisée pour détecter les obstacles sur autoroute. En effet, l'intensité du laser fournit des

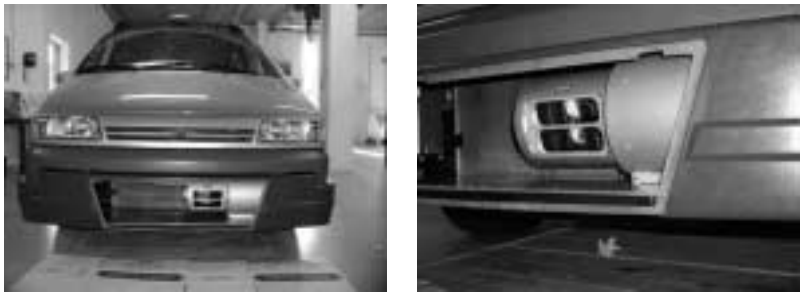


FIG. 1.12 – Vues du capteur IRDAR embarqué dans le véhicule VELAC.

informations différentes des données de vidéo ordinaire puisque les directions d'éclairage et de vue sont coïncidentes. Leur système de détection d'obstacle (pour des obstacles statiques) utilise un scanner laser haute performance qui procure rapidement une ligne de différents scans. L'analyse sous forme d'histogramme de l'intensité laser retournée est utilisée pour sélectionner les obstacles potentiels. Ensuite, après avoir mis en correspondance les candidats des lignes précédentes, la distance de chaque obstacle est estimée. Finalement, la position de chaque obstacle est mise à jour, avant que la prochaine ligne soit acquise, en se basant sur le mouvement du véhicule. Ils détectent facilement toutes sortes d'obstacles (cageot de bois, parpaing, réverbère, voiture) jusqu'à $35m$. La plupart de ces obstacles apparaissent détectables à des distances de $50m$ ou plus en utilisant une bonne configuration des différents paramètres. En particulier, ce système est capable de détecter un parpaing à une distance de $60m$.

D'autres méthodes récentes [58, 75, 117] utilisant la télémétrie laser sont présentes dans la littérature. Dans [117], les auteurs combinent les données d'estimation des bords de la route (barrière de sécurité, borne réfléchissante) et les données de détection des obstacles pour avoir une estimation plus robuste et plus précise de la situation dans le trafic (position, vitesse des obstacles) jusqu'à une distance maximale de $100m$. Un capteur, utilisant la télémétrie laser haute portée (i.e. $150m$), est proposé dans [75]. Ce capteur retourne une image de profondeur haute-résolution, et utilise une méthode de détection d'obstacle basée sur la segmentation et un algorithme de tracking pour envoyer via une interface CAN les informations (vitesse, taille) concernant les différents objets détectés. Dans [58], les obstacles détectés (jusqu'à $100m$) sont différenciés par segmentation des images de profondeur et par utilisation de modèles (voiture, camions/bus, moto/vélo, petits objets dynamiques).

D'une manière générale, les télémètres laser sont relativement bien adaptés à un usage en milieu extérieur. En effet, ils ne sont pas sensibles aux conditions de luminosité et sont également peu perturbés par la pluie. Par contre, dans le brouillard, des fausses détections sont générées, la portée de fonctionnement est limitée.

Le RADAR (cf figure 1.13) est un excellent moyen pour détecter les autres véhicules puisqu'il travaille à de très grandes portées et puisqu'il n'est pas affecté par la pluie ou la neige [105, 112, 121, 123].

FIG. 1.13 – Un exemple de RADAR.

Les capteurs RADAR utilisent le plus souvent une technique FMCW (*Frequency Modulation, Continuous Wave*). L'utilisation de RADAR pour l'automobile étant prometteuse, une bande de fréquence leur a été spécialement réservée entre 76 et $77Ghz$ en Europe et aux Etats Unis, et entre 59 et $77Ghz$ au Japon. Le signal émis est modulé et la mesure de distance est réalisée par le calcul de la différence de phase entre le signal émis et le signal reçu. Par effet Doppler, ce capteur renvoie également la vitesse des cibles.

Un système ACC utilisant un RADAR 60.5 Ghz MMW est présenté dans [121]. Le tableau 1.2 en donne les principales caractéristiques. Ce RADAR est capable de mesurer la distance, l'angle azimut et la vitesse (ρ, θ, v_d) des véhicules en amont.

type	FMCW (fsk)
fréquence de fonctionnement	60,5 Hz
portée	1 à 120 m $\pm 3\%$
vitesse	0.4 à 180 km/h
azimuth	11°
élévation	4°

TAB. 1.2 – Un exemple de capteur RADAR à 60.5 Ghz MMW [121]

Un autre RADAR à $77Ghz$ MMW pour la détection d'obstacle est présenté dans [123]. La position des obstacles est estimée grâce à la reconstruction du front d'onde, et en combinant cette estimation avec l'information géométrique de la route, la position et l'orientation des obstacles potentiels relativement à leur voie sont calculées. Dans [200], un RADAR courte portée permet de détecter les obstacles à quelques dizaines de centimètres.

D'une manière générale, les RADARs sont bien adaptés à la détection de véhicules et à leur localisation en distance. La carrosserie est en effet un bon réflecteur RADAR. Cependant, sa résolution latérale faible ne permet pas une localisation latérale précise. Par ailleurs, la facilité avec laquelle un RADAR détecte un objet à une distance donnée est directement liée à sa surface équivalente RADAR SER. Par exemple, un véhicule à

type	FMCW (fsk)
fréquence de fonctionnement	$77Hz$
portée	jusqu'à $200m$ $\pm 0.1m$
azimuth	12°
élévation	3°

TAB. 1.3 – Un exemple de capteur RADAR à $77Ghz$ MMW [123]

une SER ($10m^2$) supérieure à celle d'un piéton ($2m^2$) [123]. La plupart des petits obstacles ont une SER encore plus faible ce qui les rend indétectables. Ainsi il est surtout dédié à la seule détection de véhicules.

Les capteurs ultra-sonores fonctionnent sur des principes semblables. Ils sont souvent utilisés pour des systèmes de type *collisions warning*. Ils peuvent fonctionner dans des conditions de visibilité réduite (en cas de brouillard).

Les plus courants sont conçus avec une paire transmetteur-récepteur de piézo-éléctriques. Cependant, la portée de ces systèmes est très faible (de l'ordre du mètre). De plus, du fait de leur bande passante étroite, la reconstruction de la vitesse par effet Doppler est restreinte. Pour palier à cette limitation, depuis quelques années, des auteurs [62] proposent des capteurs ultrasoniques avec modulation de fréquence (FM) à la manière du RADAR. Ce type de système permet une estimation de la vitesse. Cependant, ces vitesses doivent être de l'ordre de 10 $km.h^{-1}$.

En effet, la vitesse de propagation du son dans l'air limite fortement l'utilisation des ultrasons dans les applications de véhicules intelligents. Ils sont ainsi réservés pour des manœuvres à faibles vitesses et à de relatives faibles portées (stationnement, accostage,...). Ils sont aussi très sensibles au déplacement de l'air, c'est-à-dire au vent. De plus, leur résolution latérale est assez faible, et nécessite donc souvent l'emploi de plusieurs microphones pour y remédier [62, 218].

L'avantage de ces capteurs actifs est qu'ils renvoient directement une information 3D. De plus, ils ne sont pas perturbés par les variations de luminosité. Toutefois, ils ne renvoient aucune information quant à la position des véhicules par rapport aux voies de circulation ce qui peut conduire à des situations difficiles à analyser en virage. La détermination du véhicule le plus dangereux est alors ardue. Enfin, les risques d'interférence entre différents capteurs actifs fonctionnant simultanément ne sont pas nuls.

1.3.2 GPS, DGPS

L'utilisation de GPS pour des applications non militaires est une activité relativement récente. En dehors de ces applications militaires, les GPS ont été utilisés pour la localisation de bateaux. La précision de positionnement alors obtenue (inférieure à $100m$) était suffisante pour ce genre d'applications. Aujourd'hui, l'utilisation de GPS se généralise et apparaît dans les véhicules routiers. Associés à une cartographie de la

route, ces appareils permettent d'aider les conducteurs à trouver leur chemin (système de navigation par satellites).

Le positionnement par satellite est rendu possible grâce à une constellation de 24 satellites qui décrivent chaque jour la même orbite. Entre 5 et 8 satellites sont toujours visibles de n'importe quel point de la terre, ce qui permet d'obtenir une bonne précision du positionnement. La précision de l'ordre du centimètre obtenue par les CP-DGPS (*Carrier-Phase Differential Global Positioning System*) permet de les utiliser comme capteur en vu de la commande de véhicule.

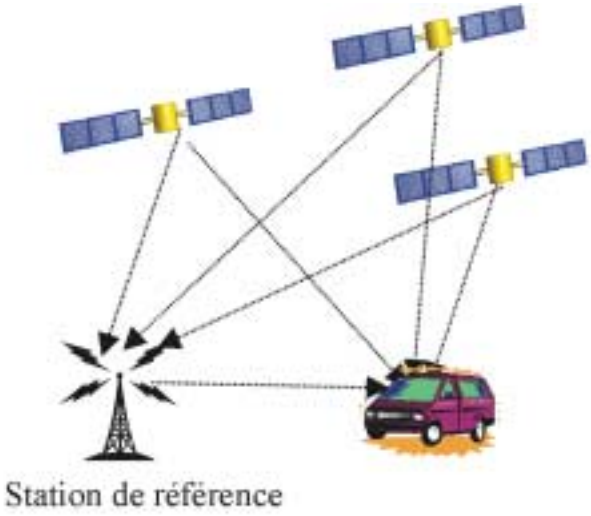


FIG. 1.14 – Système DGPS

Les travaux actuels concernent essentiellement le guidage de véhicule par rapport à une trajectoire de référence :

- dans le domaine agricole, le guidage automatique de tracteurs ou de moissonneuses batteuses par GPS fait l'objet de nombreux travaux [49]. La trajectoire de référence peut alors être définie par un précédent passage.

- dans le domaine routier avec la création de convois de véhicules, notamment lors de la Demo 2000 où un système de DGPS permettait aux véhicules d'un ou deux convois de s'informer de leurs positions respectives (cf. le dernier paragraphe de la section 1.2.2).

- toujours dans le domaine routier, James W. Sinko [185] propose d'utiliser les GPS en coordination avec d'autres informations afin de guider un véhicule routier. Il propose notamment le guidage latéral d'un véhicule à partir de données GPS et d'une base de données précise du positionnement de la chaussée.

Le GPS est insensible aux conditions atmosphériques. Cependant, lors de traversées

de tunnels, ou dans des zones de réception difficile, le GPS ne reçoit plus le signal satellite, donc ne peut pas renvoyer l'information position. De même, la constellation de satellites visibles doit être suffisante afin d'obtenir une bonne précision dans la mesure.

1.3.3 Vision : stéréo-vision, vision monoculaire

Du fait de l'analogie avec le système de perception humain, les capteurs de vision, i.e. les caméras, sont des capteurs «naturels» pour les problèmes d'aide à la conduite. En effet, l'essentiel des informations perçues par le conducteur sur son environnement provient de sa vision. Elle lui permet d'assurer les principales tâches de perception requises à la conduite, bien que sensible aux variations d'éclairements et aux conditions atmosphériques.

Deux types de capteurs de vision peuvent être globalement distingués dans la littérature. Le premier est la vision stéréoscopique, c'est-à-dire la combinaison de deux ou plusieurs caméras observant la même scène. Le second est l'utilisation d'une caméra unique.

La plupart des caméras utilisées dans les véhicules intelligents sont des caméras C.C.D., qui renvoient les luminances de la scène. Récemment, des caméras Infra-rouge [148, 214] ou des rétines CMOS [61] sont étudiées pour ces applications. Ces capteurs, bien que prometteurs pour l'intégration dans les véhicules, sont encore à un stade expérimental et importent souvent des techniques déjà développées pour les caméras C.C.D.

La **vision stéréoscopique** s'appuie sur le formalisme développé et bien défini de la géométrie épipolaire. Cette géométrie est fondée sur l'alignement des capteurs optiques (il s'agit de la contrainte épipolaire) qui est la configuration habituelle en vision naturelle (par exemple, les yeux d'un homme). La plupart des systèmes stéréoscopiques développés pour les applications de véhicules intelligents exploitent deux caméras. Certains [210] en exploitent trois afin de rendre le système plus fiable, soit du fait de la redondance d'informations soit par l'élimination de certaines ambiguïtés.

Ce capteur est généralement utilisé pour la détection d'obstacles. Cette détection se fait soit par appariement, soit par rectification homographique.

La première technique [14, 118, 150, 172] consiste à mettre en correspondance des informations entre les images du système stéréo. Cet appariement fournit une carte de disparité, qui peut être traduite en carte de profondeur moyennant un calibrage préalable. Une sélection des informations selon des critères de cohérence (disparité similaire [119, 122], distance et voisinage spatiale semblable [31, 73]) met, alors, en évidence les obstacles potentiels.

La seconde technique [89, 17] exploite la transformation homographique qui permet de rectifier une image du système stéréo pour que les pixels, issus de la projection de points sur le plan de la route, se retrouvent à l'identique dans l'autre image. Cette transformation est déterminée *a priori*, suite à un calibrage préalable du système sté-

réoscopique. Les objets situés au dessus ou en-dessous de la route apparaissent comme étant très dissemblables. La comparaison entre une image et une image tranformée par homographie permet ainsi de les mettre en évidence.

Un capteur stéréoscopique est aussi utilisé par *Bertozzi et al.* [18] pour la détection de la chaussée. Elle se base sur une technique de reconstruction (*Inverse Perspective Mapping* ou IPM) d'une vue de dessus de la route.

La portée d'un capteur stéréoscopique est fortement liée à la distance inter-caméra. Cette distance en embarqué est donc limitée par la taille du véhicule. Par exemple, dans [19], un tel système est proposé pour la détection d'obstacles de différentes tailles (25x60cm, 50x90cm, 40x180cm). Il est composé de deux caméras séparées de 120cm et de focale 6mm. La portée de ce système pour les objets testés est approximativement évaluée entre 10 et 27m.

Williamson [210] présente aussi un système de stéréovision embarqué et l'évalue pour la détection et la localisation de petits objets facilement détectables (boites noire ou blanche, de 30cm de longueur et de hauteur variant entre 10 et 30cm, et canette de soda). Lorsque le véhicule est statique, il peut détecter certains de ces objets (placés à une hauteur de 14cm) jusqu'à 150m de distance. Lorsque le véhicule est en mouvement (entre 10 et 25 mph, soit entre 16 et 40 $km.h^{-1}$) et en traitant les données *«off-line»* , la détection est encore fiable à partir de 110m pour des objets, de type boites. Bien sûr, ces résultats sont grandement facilités par la nature simple (achromaticité, forme) des objets.

En **vision monoculaire** (avec une caméra), deux approches sont possibles :

- soit les techniques de reconnaissance de formes sont utilisées sur des scènes statiques (les images sont traitées indépendament les unes des autres).

- soit des séquences d'images sont analysées pour détecter le mouvement de la caméra ou des obstacles.

Pour les scènes statiques, deux techniques sont distingables pour détecter un objet : soit par extraction de primitives et leur mise en correspondance avec un modèle, soit via l'apparence.

La première est donc une *approche par primitives*. Elle considère une connaissance explicite de l'objet, exprimée dans un modèle de l'objet. Elle suit un schéma classique en deux passes : extraction des primitives dans l'image, puis analyse de celles-ci par rapport au modèle. Les primitives utilisées peuvent être de plusieurs types : niveaux de gris, segments, couleur, symétrie ou texture. Il s'agit souvent de systèmes pouvant être exploitables en temps réel.

La seconde est une *approche selon l'apparence* qui propose une classification directe des pixels (ou de groupes de pixels) entre ceux qui sont un objet et ceux qui n'en sont pas. Elle utilise une connaissance implicite de l'objet, obtenue via un apprentissage sur une base de données. Ces techniques récentes, souvent issues de travaux en reconnaissance,

s'avèrent très robustes, mais sont très couteuses en temps de calcul car elles requièrent une recherche multi-échelles et multi-résolutions. Aussi, elles n'ont été que rarement exploitées dans des applications de type véhicules intelligents.

Nous reviendrons en détails sur ces techniques dans le chapitre 2.

Pour analyser le mouvement dans des séquences d'images, les techniques de flot optique sont généralement utilisées. La théorie du flot optique considère que les changements de luminance dans un intervalle de temps réduit sont dûs aux seuls mouvements dans l'image. Plusieurs méthodes peuvent être utilisées pour calculer le flot optique. *Enkelmann* dans [64] en teste trois : une analytique basée sur les gradients, une autre basée sur les contours, et une discrète (basée sur les pixels). Il montre une préférence pour la méthode basée sur les gradients pour sa fiabilité. Cependant, nécessitant un lissage, elle montre des points faibles pour la détection de petits obstacles. Les méthodes basées sur les contours, utilisées pour détecter et suivre les véhicules dans [25, 64], ont elles leurs performances liées à celles des détecteurs de bords, puisqu'elles consistent à apparier ceux-ci. Les algorithmes de calcul de flot optique sont très coûteux en temps. De plus, ils ne peuvent s'appliquer qu'à des véhicules non-stationnaires dans l'image ou par rapport à la scène (par exemple, en phase de dépassement [11]).

D'autres techniques utilisent aussi les séquences d'images pour la détection et le suivi d'obstacles. Les travaux de *Cohen et al.* [40] et ceux d'*Araki et al.* [4] utilisent des techniques basées sur le suivi d'éléments pour éliminer l'arrière-plan. Elles s'apparentent beaucoup à celles utilisées en flot optique. En premier lieu, le mouvement reprojeté de plusieurs points appartenant à l'arrière plan est estimé. Ceux-ci permettent de constituer un modèle affine de mouvement, et de le projeter pour chaque point de l'image. La présence d'un véhicule potentiel est alors indiquée par une différence forte du mouvement locale par rapport à celle estimée via le modèle. Comme pour le flot optique, le défaut majeur de cette méthode est l'importante complexité des algorithmes et les nombreuses fausses détections dues aux erreurs d'estimation du mouvement reprojeté (ou *projected motion*).

Dans les travaux de Betke et al [20] et aussi dans ceux de *Leeuwen et al.* [202], la comparaison d'image à image (ou *temporal differencing*) est utilisée pour détecter des véhicules en situation de dépassement (qui doublent ou en train de se faire doubler par le véhicule équipé). Ces véhicules occupent de larges zones dans l'image et donc introduisent localement d'importantes variations de luminance. La comparaison d'image à image est donc dans ce cas une méthode très efficace. Cependant, les objets de l'arrière plan sont aussi susceptibles de provoquer de pareilles variations (par exemple, le passage d'un pont, l'ombre d'un arbre,...).

Ainsi, certaines techniques ou indices sont utilisables ou non suivant la localisation dans l'image de l'objet de la scène routière. En effet, l'apparence d'un objet dans une scène routière en dépend ; la transformation géométrique mise en jeu n'est pas la même lorsque le véhicule est sur les côtés du champ de vision, ou lorsqu'il se trouve à l'horizon. D'une manière générale, trois zones de l'image sont distinguées :

– Les coins inférieurs droit et gauche de l'image, où les variations de l'image sont

importantes. Il s'agit de zones où peuvent se présenter des véhicules en situation de dépassement. Dans cette zone, l'information mouvement est souvent primordiale.
- L'horizon, ou plus exactement le *focus of contraction* (FOC) de la route. Dans cette zone, le mouvement d'un obstacle ou de son environnement n'est pas distingable.
- Enfin une zone mitoyenne, comprise entre les deux précédentes. C'est une zone où les déformations géométriques dues à la projection ne sont souvent pas considérées.

Il existe donc une relation entre le type d'informations à utiliser dans un système de vision monoculaire et les régions de l'image. Le tableau 1.4 [202] fait la synthèse des diverses informations utilisables pour la détection de véhicules par vision monoculaire. Ces informations sont classées suivant le type d'obstacle (position par rapport à la caméra) et la complexité des algorithmes mis en œuvre.

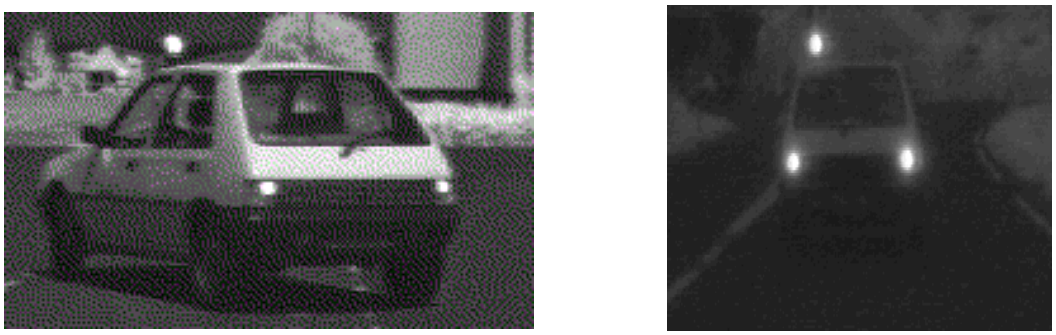
COMPLEXITÉ ALGORITHMÉTIQUE	proche	mitoyenne	lointaine
importante	flot optique par corrélation flot optique contraint	flot optique par corrélation flot optique contraint flot optique gradient apparence	apparence
faible	*projected motion temporal differencing* couleur ombre texture	symétrie segments couleur ombre texture	symétrie segments couleur ombre texture

DISTANCE

TAB. 1.4 – Relation entre le type d'informations et les régions de l'image où ils peuvent être utilisés

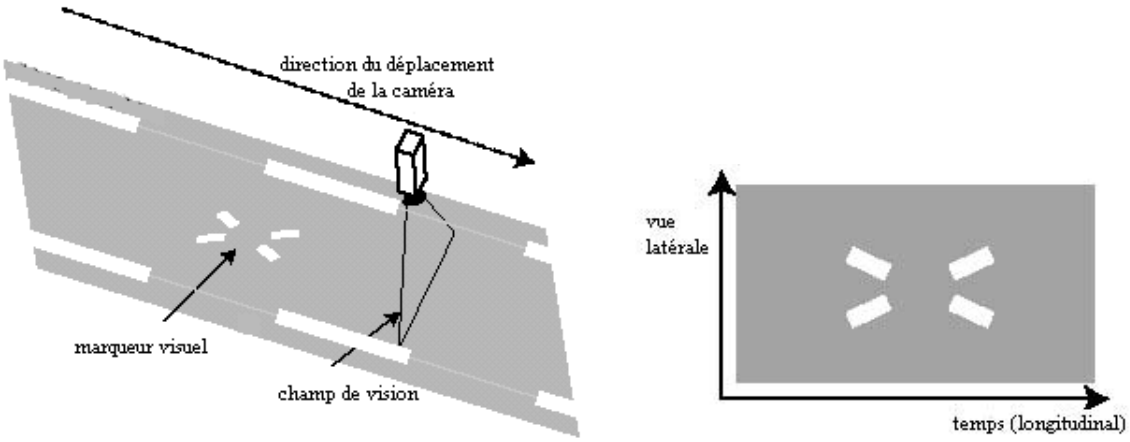
La localisation de la route ou d'un obstacle en monovision se fait à partir d'hypothèses, la plus fréquente étant celle d'un monde plan, ou à partir d'une connaissance *a priori* de la géométrie 3D de l'objet à localiser. Cette connaissance peut être introduite par le placement d'amers visuels sur les obstacles routiers [142, 54] ou sur la route [113]. Les figures 1.15 et 1.16 donnent des exemples d'amers utilisés.

La portée maximale de tels systèmes de vision est souvent estimée entre $60m$ et $100m$ dans de bonnes conditions de visibilité. Cette limitation est induite par la focale de la caméra. Une caméra grand angle est souvent utilisée pour pouvoir détecter la route à proximité du véhicule ainsi que les obstacles en situation de dépassement. Cette caractéristique fait qu'un objet situé à une distance relativement lointaine apparaîtra de petite taille dans l'image, et sera donc difficilement identifiable.



(a) vue 3/4 arrière (b) vue d'arrière (avec un filtre infrarouge)

FIG. 1.15 – Un exemple d'amers visuels sur un véhicule coopératif [142].



(a) vue de la route (b) marqueur vu de la caméra

FIG. 1.16 – Un exemple de marqueurs visuels sur la route [113].

Enfin, l'utilisation de capteurs de vision nécessite une grande puissance de calcul, pour détecter les différents éléments routiers (route, obstacles,...) et les localiser. Leur développement a donc été grandement facilité cette dernière décennie par celui des ordinateurs personnels de type PC.

1.3.4 Bilan sur les capteurs pour les véhicules intelligents

Nous avons vu que différents capteurs peuvent être utilisés dans les véhicules intelligents. Ils ont chacun des caractéristiques différentes, aussi bien en termes de technologie qu'en aptitudes et performances. Dans les paragraphes qui suivent, nous allons rappeler celles-ci. Nous évoquerons aussi différentes solutions pour faire collaborer plusieurs de ces capteurs afin d'améliorer la perception de l'environnement. Ce bilan nous permettra d'introduire notre approche.

Avantages et défauts des différents capteurs.

Les capteurs actifs sont particulièrement intéressants car ils fournissent directement la position 3D des obstacles routiers. Des systèmes d'aide à la conduite intégrant certaines de ces technologies sont dès à présent disponibles dans le commerce.

Des systèmes ACC avec RADAR sont disponibles en option sur des véhicules haut de gamme [57]. Par exemple, la Mercedes Classe S propose en option (à 2500 euros) depuis l'année 2000, un tel système nommé *Distronic* (cf. figure 1.17).

FIG. 1.17 – Illustration du système *Distronic* (à gauche), disponible sur la Classe S de Mercedes-Benz (à droite)

De manière plus modeste, des capteurs ultrasoniques sont aussi intégrés sur certaines voitures. Dans des situations de mise en stationnement, ils fournissent approximativement (sous la forme d'un signal sonore modulé) au conducteur la distance par rapport à un obstacle arrière (typiquement, un autre véhicule en stationnement) et lui permettent ainsi de se garer sans avoir le visuel total de la scène arrière.

Cependant, ces capteurs ne permettent pas de localiser le véhicule par rapport aux voies de circulation, ce qui limite leurs performances dans beaucoup de situations. Par exemple, un système ACC muni seulement d'un RADAR est incapable de fonctionner

correctement dans des virages. Ce défaut est compensé dans les systèmes commerciaux par l'apport de capteurs proprioceptifs, qui permettent d'estimer la trajectoire suivie par le véhicule équipé, et ainsi de «diriger» la détection d'obstacles par RADAR. Mais d'une manière générale, sans la localisation par rapport à l'infrastructure, ces capteurs sont limités à des obstacles de proximité, et donc à des véhicules circulant à des vitesses restreintes.

Le capteur GPS semble aussi intéressant. Il est fort probable que les véhicules soient équipés en série de GPS pour les systèmes d'aide à la conduite ; à la manière de flotte de camions pour certaines compagnies de transport routier.

Actuellement, un système nommé Lavia[10] (PSA[11], Renault, LCPC[12], INRETS, CETE[13]) est testé en France [3] sur un panel d'une centaine d'automobilistes sur un site expérimental. Ce dispositif limite automatiquement la vitesse du véhicule à la vitesse réglementaire du secteur (route nationale, zone urbaine ou autoroute) où il se trouve. Le système s'appuie sur une carte digitalisée embarquée. Grâce au GPS, le véhicule connaîtra sa position sa localisation ainsi que les vitesses limitées correspondantes.

Par ailleurs, un système civil de localisation par satellites, Galileo, concurrent au système américain GPS, devrait être mis en place par l'Europe. Ceci devrait rendre encore plus attractifs et plus courants de tels systèmes.

Toutefois, ces systèmes de navigation souffrent de problèmes de pertes de signal satellite et, pour la coopération inter-véhicules, sont très exigeants au niveau communication.

Les systèmes basés sur la vision ont démontré leur efficacité pour des applications de type véhicules intelligents.

En détection de la route, plusieurs démonstrations ont montré qu'un système muni de caméras pouvait être robuste. Dès 1995, le NavLab 5 du *Robotics Institute* de la *Carnégie Mellon University* a parcouru les Etats Unis d'est en ouest avec un système automatique de contrôle latéral [162]. Celui-ci était guidé par un simple système de vision de reconnaissance de la courbure horizontale et de la position latérale des voies. Près de 98% des $5000km$ parcourus ont pu l'être sans intervention humaine, à une vitesse moyenne de $102km.h^{-1}$. Quelques mois plus tard, le véhicule VaMP de l'UBM a parcouru plus de $1600km$ avec un controle automatique de la vitesse et de la direction sur près de 95% de la distance. Plus récemment (en 1998), *Broggi et al.* ont fait une démonstration similaire en Italie [27]. 94% des $2000km$ du parcours se sont effectués en contrôle automatique avec le véhicule ARGO.

En détection et suivi d'obstacles, de semblables expérimentations ont été menées avec des véhicules coopératifs. Ceux-ci étaient munis d'amers visuels connus. Nous pouvons par exemple citer le projet Praxitèle [54], lancé en 1993 par l'INRETS, Renault, EDF[14] et Dassault, pour la création de convois de véhicules électriques en ville, et le projet Promote

[10] Lavia, Limitateur s'Adaptant à la VItesse Autorisée.
[11] Peugeot S.A.
[12] Laboratoire Central des Ponts et Chaussées
[13] Centre d'Etude Technique de l'Equipement
[14] EDF, Electricité de France.

(a) Véhicule NavLab 5 du CMU

(b) Véhicule ARGO de l'Université de Parme

FIG. 1.18 – Exemples de véhicules expérimentaux.

Chauffeur (1996-1999) de Daimler-Chrysler [74] qui concerne la mise en convoi de deux poids-lourds. A noter que dans cette derniere, certaines informations (mouvement du camion maître) étaient transmises par radio.

(a) Projet PRAXITEL

(b) Projet CHAUFFEUR

FIG. 1.19 – Illustrations d'asservissement longitudinal par vision.

Ainsi le capteur de vision est le seul qui informe à la fois sur les voies de circulation et sur le contenu de la scène. De plus, ce sont généralement des équipements moins lourds et moins coûteux à embarquer au sein d'un véhicule, bien qu'ils nécessitent un calibrage rigoureux, et posent donc des problèmes de maintenance à long terme. Cependant, cette difficulté pourrait voir fin avec les nombreux travaux récents sur l'auto-calibrage de caméras. Nous pouvons citer, par exemple, les travaux de *A. Broggi et al.* [28] sur le calibrage d'une caméra embarquée à partir d'amers visuels posés sur le capot du véhicule équipé.

Par ailleurs, la localisation par capteurs de vision reposant sur des hypothèses (telle que la route localement plane) est souvent approximative, surtout lorsqu'il s'agit de la position longitudinale d'un obstacle.

Collaboration entre ces capteurs.

Ainsi chacun de ces capteurs possède ses points forts et ses points faibles pour une application d'aide à la conduite. Aussi un des axes de recherche actuels dans ce domaine est de faire collaborer ces différents capteurs dans un même système, afin d'accroître sa

performance globale.

Ceci peut être notamment obtenu par fusion multi-capteurs. Les qualités de certains capteurs sont alors utilisées pour compenser les défauts des autres. Plusieurs approches sont proposées dans la littérature :

- *S. Jouannin* [105] propose de fusionner des données sur la position de la route et des obstacles par rapport au véhicule embarqué, provenant de capteurs proprioceptifs, de vision et RADAR. Il obtient alors un positionnement des obtacles plus sûr et plus robuste (en limitant les erreurs).

- La fusion de données RADAR et vision est aussi utilisée dans [79, 67, 189, 80]. Une telle collaboration est motivée par le fait que le RADAR mesure la distance avec un obstacle avec précision mais n'a pas une bonne résolution latérale alors qu'un système de vision présente les caractéristiques inverses. Par exemple, dans le système FADE[15] mis au point par *Bruno Steux* [189], la mauvaise localisation latérale fournie par le RADAR est alors corrigée par une détection de symétrie visuelle. Ce même système permet d'élargir le champ de détection du RADAR avec celui de la caméra.

- D'autres systèmes font collaborer les données issues de la télémétrie laser et de la vision. Dans [193], une caméra détecte les obstacles potentiels. Puis le télémètre laser est utilisé pour leur détermination finale, afin d'éliminer notamment les fausses détections, et leur localisation. Dans [184], le télémètre laser est utilisé pour la détection des obstacles. La vision sert à la reconnaissance des lignes blanches de la route et ainsi à déterminer le type d'obstacles présents sur la route.

- Enfin, des systèmes plus complets [12, 79, 124] se proposent de fusionner les données issues des trois technologies de capteurs extéroceptifs (vision, RADAR, télémètre).

Notre choix d'un capteur de vision

Dans nos travaux, nous nous sommes intéressés à la mise en œuvre de méthodes et d'algorithmes pour la détection et le suivi d'obstacles via un capteur de vision. Ce choix d'un capteur de vision s'explique pour plusieurs raisons. Il s'agit d'un capteur «naturellement» bien adapté à l'aide à la conduite. Il permet une perception globale de la scène ; il est le seul capteur extéroceptif capable de fournir des informations à la fois sur la route (et les voies de circulation) et les obstacles. Il est peu coûteux et peu encombrant. Il consomme relativement peu d'énergie. Par ailleurs, il s'agit d'une technologie plus souple. Le champ de perception d'une caméra est essentiellement liée à sa focale.

C'est d'ailleurs en grande partie pour cette dernière raison que des capteurs de vision

[15] FADE, Fonction d'Analyse et de Détection de l'Environnement

multi-focales sont proposés dans la littérature. Ces solutions sont particulièrement utiles à la perception de la scène frontale d'un véhicule équipé. En effet, celle-ci nécessite à la fois une vision «courte» pour détecter les obstacles proches et en situation de dépassement et une vision «longue» pour les véhicules lointains.

Le projet le plus intéressant est celui entrepris par *Dickmanns et al.* [134, 160] pour des applications de détection de la route. Il s'agit d'un capteur de vision, nommé MarVEye (*Multi-focal active/reactive Vehicle Eye*), intégrant 4 caméras, tel que l'illustre la figure 1.20. Les deux premières sont grand angle avec des axes optiques divergents, et offrent à eux deux une vision à 105° de la scène proche. Les autres sont des caméras avec focales longues différentes, offrant des visions à moyenne et longue distances. Ces dernières disposent d'une plateforme pan-tilt (*TaCC* [160]) pour diriger leur champ de vision.

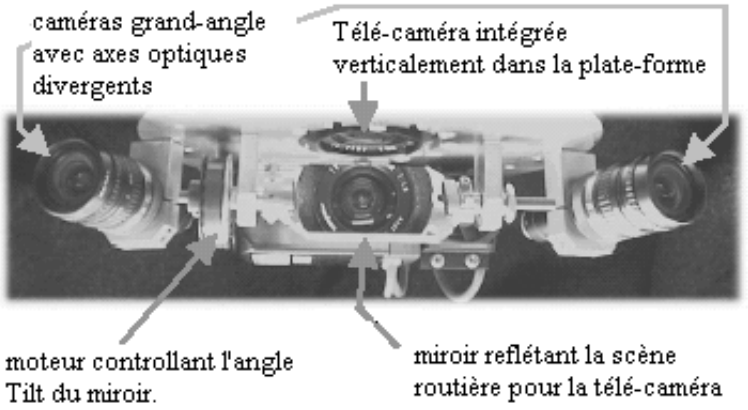


FIG. 1.20 – Configuration du système MarVEye [134, 160]

Le capteur que nous proposons dans la suite s'inscrit dans la même veine, sauf qu'il propose d'utiliser une caméra aussi pilotable en zoom. Dans une application de véhicule intelligent, l'utilisation d'un zoom pourrait permettre :

- d'augmenter le champ de perception de l'environnement du véhicule expérimental. L'augmentation de la focale de la caméra devrait permettre de détecter et de suivre la route et des véhicules obstacles dans des zones lointaines ou mal connues.

- d'améliorer le suivi et la reconnaissance d'un véhicule en particulier, en s'y focalisant.

- d'augmenter la précision des caractéristiques cinématiques acquises à partir de la séquence d'images.

Pour ces raisons, nous pensons développer un capteur de vision combinant une caméra

	Télémètre	RADAR	Ultrasons	GPS	vision stéréo.	vision mono.	vision mono. + PTZ
données 3D	+++	+++	++	++	+	Non	Non
précision des mesures distance	+++	+++	0		+	-	-
précision mesures site et azimut	+++	- -	- - -		+++	+++	+++
résolution distance	+++	++	+		+	-	-
résolution site et azimut	-	- -	- - -		+++	+++	+++
portée distance	$<200m$	$<200m$	$<1m$		$<30m$	$<70\ m$	$<150m$
ouverture angulaire	-	- -	- - -		+	++	+++
mesure directe de la vitesse	Non	Oui (Doppler)	Oui	Oui	Non	Non	Non
positionnement véhicule	relatif	relatif	relatif	absolu	relatif	relatif	relatif
conditions atmosphériques	Temps clair	Tout temps	Tout temps (sauf vent)	Tout temps	Temps clair		
Jour / Nuit	Jour et Nuit				nécessite 2 modes de fonctionnement		
perception voies de circulation	Non			(*)	Oui		
perception obstacles	+++	+++	++	Non	++	+	+
fausses alarmes	+++	- -	- - -		++	-	-
coût	- - -	+	+++	+	+	++	+
encombrement	- - -	+	+++	+	+	++	+
consommation en énergie	- - -	+	+++	+	++	++	++
commentaires	système à balayage mécanique	interférences avec les autres systèmes	capteur de proximité	pertes du signal satellite, (*) avec une cartographie	calibration, algorithmie lourde		

TAB. 1.5 – Bilan sur les caractéristiques des capteurs.

avec zoom, qui pourra se focaliser soit sur une scène lointaine (le bout de la route), soit sur une scène proche (le véhicule le plus dangereux), et une caméra avec grand angle qui fournira une vision globale de la scène au devant du véhicule expérimental

1.4 Notre approche : un capteur de vision double

L'idée originale de notre approche est donc de proposer, en plus d'une caméra grand angle, une caméra commandée en site azimut et en zoom (*Pan-Tilt-Zoom* ou PTZ). En effet, une caméra grand angle présente la contrainte d'une portée limitée (par la focale et la résolution de celui-ci et de la numérisation). Les risques de disparition des autres véhicules (et de la chaussée) sont donc importants surtout lors de virages ou de déclivités de la route.

Ainsi notre capteur de vision permettrait de pallier ces limitations. La caméra commandée en site azimut et en zoom offrirait une vision lointaine dans un champ limité mais orientable, et la caméra grand angle une vision globale de la scène. Nous verrons dans cette partie comment ces avantages peuvent être utilisés dans la détection et le suivi d'obstacles dans un contexte routier ou autoroutier. Nous expliciterons ainsi les intérêts d'un tel capteur mais aussi ses contraintes algorithmiques.

1.4.1 Stratégies

Ce capteur doit permettre la détection et le suivi d'obstacles se situant devant le véhicule expérimental (où le capteur est embarqué). Son rôle est de localiser les obstacles par rapport au véhicule test et aux voies de circulation, et d'estimer leurs attitudes afin d'analyser correctement la scène.

Ce système de vision doit fonctionner en temps réel.

Son domaine d'application est essentiellement l'avertissement du conducteur. En effet, nous avons expliqué dans la section précédente qu'un système de vision monoculaire ne permet pas une localisation longitudinale précise des obstacles sur la route. Or des applications autres, tel qu'un système ACC, demandent *a priori* une mesure avec précision de la distance et de la vitesse. Aussi, dans nos travaux, nous considérons une application de type *Collision Warning*, mais il serait envisageable dans l'avenir de l'étendre à d'autres applications.

La détection des obstacles est essentiellement associée à la caméra grand angle. La vision globale de la scène qu'elle nous fournit doit nous permettre de déterminer quel est le véhicule le plus « dangereux », donc le véhicule auquel il faut porter le plus d'attention. C'est sur ce véhicule que la caméra commandée sera dirigée afin de le suivre (cf. figure 1.21) et d'estimer ses paramètres cinématiques (position, vitesse et direction).

Par ailleurs, la caméra équipée d'un zoom permet d'augmenter la portée du capteur

par rapport à une vision monoculaire grand angle. En effet, un véhicule trop éloigné pourrait ne pas être détecté via la première caméra. Aussi la seconde pourrait prendre le relais et, grâce au zoom, augmenter la résolution des images perçues au loin. Voyons donc plus explicitement les tâches qui sont liées aux deux caméras.

Rôles et intérêts de la caméra grand angle

La caméra grand angle sera utilisée à deux fins essentielles et très étroitement liées : la détection des voies de circulation et celle des obstacles dans les limites du champ de vision de la caméra. L'obstacle le plus dangereux peut être alors déterminé en tenant compte de la distance et de la position par rapport au véhicule équipé et aussi par rapport aux voies de circulation (surtout dans les virages).

Ainsi, c'est à partir des informations fournies par la caméra grand angle que l'on va (ou non) focaliser la caméra commandée sur l'obstacle à suivre. D'où l'importance de garder une vision globale de la scène. Elle permet en effet de détecter tous les changements de situation : apparition ou disparition d'un véhicule, changement de voie d'un véhicule, virage... Tous ces changements ont bien sûr leurs conséquences sur la détermination de l'obstacle le plus dangereux et donc sur la focalisation de la caméra commandée.

Aussi il sera important de lier à la caméra grand angle, un algorithme de détection de la route et des voies de circulation ainsi qu'un autre chargé de la détection des obstacles. Ce dernier devra être capable de détecter simultanément plusieurs véhicules. Par ailleurs, un module de décision devra lui être adjoint (pour repondre à la question : quel obstacle est le plus dangereux ?).

Rôles et intérêts de la caméra commandée

Comme nous l'avons déjà signalé précédemment, la caméra commandée devra alterner entre deux modes de fonctionnement suivant la situation :

- si un ou plusieurs véhicules sont détectés grâce à la caméra à focale fixe, la caméra PTZ doit se focaliser sur l'obstacle le plus dangereux et le suivre.

- si aucun obstacle n'est détecté, elle se focalisera sur le point de fuite de la route et devra y rechercher éventuellement des obstacles.

Ainsi dans le deuxième mode, la caméra PTZ devrait permettre de détecter et de suivre éventuellement un véhicule en dehors du champ de vision et de la portée de la caméra à focale fixe. La portée du système est ainsi accrue par rapport à un système n'intégrant qu'une seule caméra. Son fonctionnement pour la détection sera alors comparable à celui de la caméra grand angle.

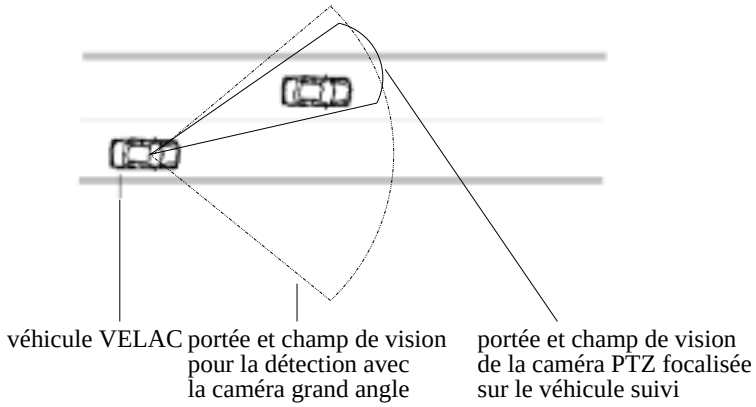


FIG. 1.21 – caméra PTZ utilisée pour le suivi de véhicule.

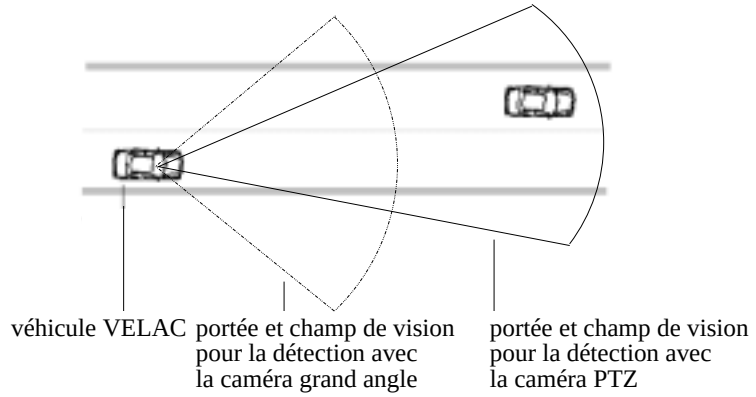


FIG. 1.22 – caméra PTZ utilisée pour augmenter le champ de vision.

Le mode «suivi de véhicule» permettra d'avoir une résolution plus grande et à peu près constante de l'obstacle suivi. Ceci pourrait être extrêmement intéressant pour la reconnaissance de l'obstacle. En effet, la qualité de la résolution (multi-résolution) des objets est un critère important dans les algorithmes de reconnaissance. Par ailleurs, ceci devrait permettre aussi de déterminer plus précisément les paramètres de l'obstacle (position, attitude) en se servant d'un modèle rigoureux de la caméra (notamment du zoom).

1.4.2 Travaux à développer

Ainsi le système qui devra être associé à notre capteur de vision peut se résumer par le schéma de la figure 1.23.

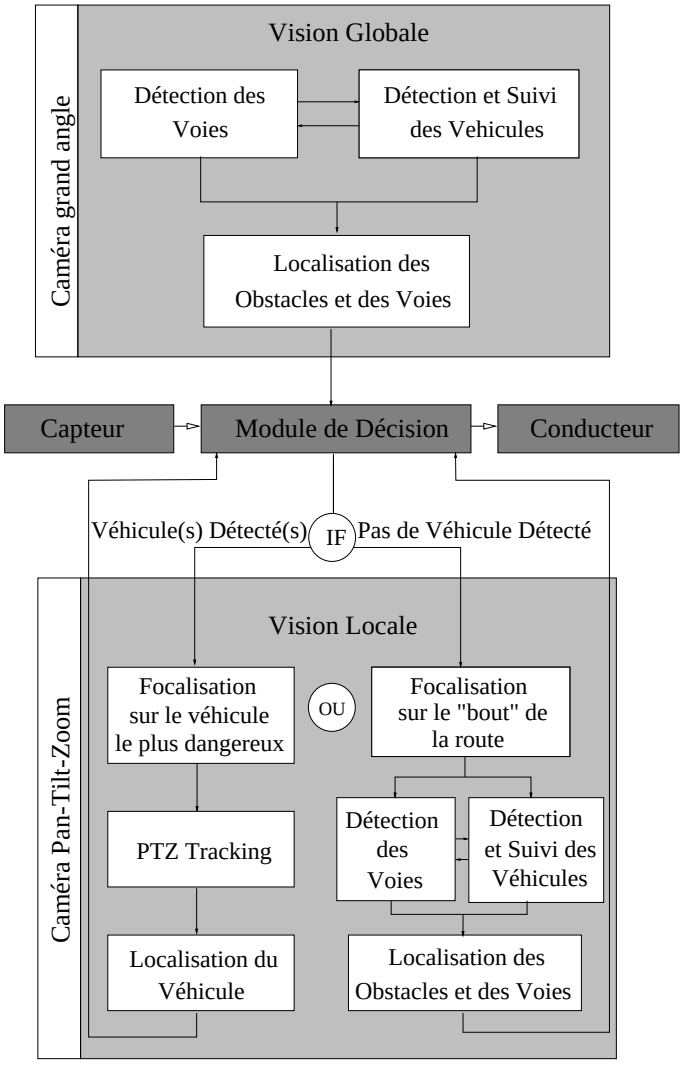


FIG. 1.23 – Synopsis de l'architecture logicielle liée au capteur de vision développé

Comme nous le constatons sur ce schéma, la mise en œuvre d'un tel capteur demande l'étude et la mise en place de nombreux modules. Nous nous sommes intéressés particulièrement dans nos travaux aux problématiques liées aux obstacles. Ainsi, dans les chapitres suivants, nous proposons plusieurs méthodes et algorithmes pour détecter et suivre des véhicules en situation routière, à l'aide de notre capteur.

Tout d'abord, nous allons décrire un algorithme de détection de véhicule par vision monoculaire. Il est destiné avant tout à la caméra grand angle. Mais il pourrait être facilement porté à la caméra PTZ, lorsque celle-ci est focalisée sur la route. Notre préoccupation principale a été la détection d'obstacles relativement distants. Ce qui exclut l'usage de méthodes basées sur le mouvement. En effet, elles ne fonctionnent que pour des obstacles à faibles distances ou en phase de dépassement. Aussi la méthode développée considère la scène comme statique.

Le chapitre suivant sera consacré à la description et à l'expérimentation d'un algorithme de suivi temps réel d'objet original développé au sein du LASMEA. Nous verrons que cet algorithme est particulièrement bien adapté au suivi de véhicules, et permet une estimation assez précise de la cinématique de ceux-ci, en considérant l'hypothèse de route plane et via un filtrage de Kalman. Cet algorithme peut servir aussi bien en vision globale, qu'en vision locale lors de l'asservissement de la caméra PTZ sur un véhicule.

Le dernier chapitre concerne la commande de la caméra PTZ. Nous allons y établir un asservissement de ses positions en angles site et azimut et du zoom de sorte à conserver le véhicule suivi dans l'image. Nous y définissons donc le module nommé «PTZ Tracking» dans le schéma 1.23, essentiel à notre capteur. Ce chapitre sera l'occasion de vérifier la faisabilité et l'utilité de la caméra PTZ pour le suivi d'un véhicule.

A chacun de ces chapitres, un bilan sera établi afin de dégager quelques voies d'investigations et améliorations futures.

Chapitre 2

Détection de véhicules par vision monoculaire

Dans ce chapitre, nous présentons le système de détection de véhicules que nous avons développé. Celui-ci doit répondre à plusieurs conditions.

D'un point de vue de l'application, il s'agit de détecter les véhicules avec une caméra embarquée dans des scènes routières ou autoroutières. Ces véhicules sont supposés circuler dans le même sens de circulation. Bien que nous nous sommes limités aux véhicules de type tourisme, le système doit être générique ; il doit pouvoir être adapté à tout type de véhicule. Par ailleurs, il doit pouvoir fonctionner en temps réel.

Par rapport à la tâche «détection», il s'agit de détecter des vues arrières de véhicule sur des images statiques et achromatiques. Le processus doit permettre une localisation approximative des véhicules détectés par rapport aux voies de circulation. Il doit être relativement robuste. Il n'y a pas besoin de détecter les véhicules sur toutes les images d'une séquence. Cependant, un taux de détection important est indispensable. Par ailleurs, le taux de fausses alarmes doit être faible.

La première partie de ce chapitre est consacrée à un état de l'art de la détection de véhicules dans des scènes statiques. Cet état de l'art nous permet de situer notre propre méthode par rapport à l'existant dans ce domaine. Ensuite, nous explicitons notre approche. Une caractérisation des résultats obtenus sur un grand nombre des images successives est présentée. Elle nous permet de conclure sur les performances de notre approche et sur les perspectives à donner.

2.1 Etat de l'art : détection de véhicules

Dans cette section, nous allons présenter la problématique de la détection d'objets en vision par ordinateur. Puis, nous expliciterons les diverses solutions proposées dans la littérature. Nous aurons à cœur d'illustrer nos propos avec des exemples de travaux, réalisés notamment pour la détection de visages. Ensuite, nous aborderons le problème plus spécifique de la détection d'obstacles routiers par vision embarquée, dans des scènes

routières ou autoroutières statiques.

2.1.1 Problèmatique de la détection d'objet

La détection d'un objet en vision par ordinateur est l'opération consistant à segmenter les pixels de l'image en deux classes : ceux appartenant à la classe objet et ceux à la classe non-objet (ou à l'arrière-plan). C'est une tâche difficile. Il y a deux raisons à cela : d'abord l'extrême variabilité de l'apparence des objets dans l'image, ensuite l'importante quantité d'informations contenues dans une image.

En effet, l'apparence d'un objet dans une image est très variable. Elle va dépendre de la position et de l'orientation de l'objet par rapport à la caméra. Cela va jouer sur sa taille, sa forme et sa texture[1]. Il y a de plus la perte d'informations due à la projection du 3D vers le 2D, qui peut entraîner certaines ambiguïtés. L'intensité d'un pixel dépend aussi de l'éclairage de la scène : de sa position, de sa couleur, de son intensité. De son interaction avec le restant de la scène : ombres, nouvelles sources lumineuses induites par réflexion. La nature même de l'objet peut poser problème : si celui-ci est réfléchissant, il pourra avoir une apparence très changeante selon son environnement, la position de la caméra.

L'autre difficulté de la détection d'objet dans l'image est l'importante quantité de données à traiter. Un objet dans une image peut être composé de plusieurs centaines, voire de milliers, de pixels, et chacun d'eux peut contenir une information importante. Créer un algorithme capable de tenir compte de l'ensemble de ces informations, et de toutes les possibilités résultantes (au maximum, 256^N pour une image en niveaux de gris et contenant N pixels), reste actuellement du domaine de l'utopie.

Ainsi la problèmatique de la détection d'objet est de trouver une représentation la plus réduite possible de l'objet capable de tenir compte de la plus grande variabilité. Cette tâche consistant à trouver la plus petite quantité d'informations offrant la meilleure qualité de discrimination est assez ardue ; au niveau algorithmique, il s'agit souvent de réaliser un compromis entre la rapidité d'exécution et la robustesse.

Parmi les applications en vision par ordinateur requérant un processus de détection d'objet, il est intéressant de se pencher sur celles axées sur les visages. Dans ce cadre, de nombreuses recherches ont été menées, notamment depuis les années 90. Elles répondent à un besoin pour les applications de reconnaissance des visages. En effet, pour cette tâche, les techniques sont très poussées, comme le montre l'état de l'art récemment élaboré par *W. Zhao et al.* [224]. Cependant, dans une scène réaliste, elles ont besoin d'être focalisées sur le visage à reconnaitre. Ainsi, la détection de visages est souvent conçue comme une première étape de la reconnaissance. Deux études récentes [95, 217] dressent des états de l'art sur les techniques mises au point dans ce domaine.

De plus, il s'agit d'une problématique assez similaire à la détection de véhicules :

[1] Notamment si certains détails sont visibles ou pas dans l'image.

environnement non controlé, grande diversité des objets à détecter. Aussi, dans le panorama sur la détection d'objets qui est proposé dans la suite, nous évoquerons souvent des systèmes consacrés à la détection de visages. La figure 2.1 dresse un organigramme des différentes méthodes que nous allons voir.

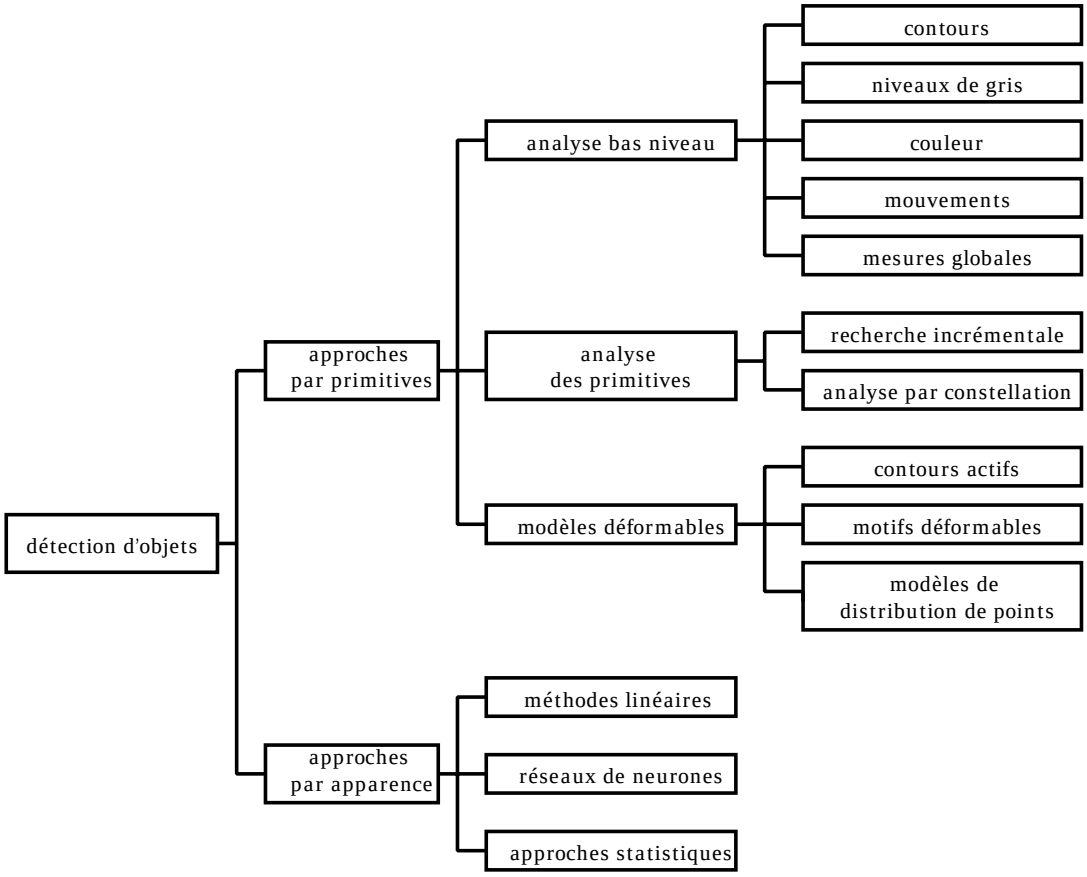


FIG. 2.1 – Les différentes approches en détection d'objets [95, 217]

2.1.2 Panorama des méthodes en détection d'objets par vision

Afin de pouvoir détecter un objet, il faut connaître une information *a priori* sur cet objet. Pour obtenir cette connaissance, deux approches sont distinguées dans la littérature :

- La première considère une connaissance explicite de l'objet, qui aboutit à la constitution d'un modèle de l'objet. Elle suit un schéma classique en détection (et plus généralement en traitement d'image) : des primitives bas niveaux sont extraites de l'image, puis sont analysées par rapport au modèle connu. Ce type de système exploite différentes propriétés de l'objet (apparence, géométrie) à l'aide de mesures de distances, d'angles ou/et de tailles sur les primitives extraites. Ce sont les **approches par primitives**

– La seconde propose une classification directe via une représentation de l'objet selon l'apparence. L'apparence peut se définir comme la forme de la surface d'intensité de l'image. Par exemple, il peut s'agir basiquement du tableau 2D des pixels. Les techniques utilisées, ou **approches via apparence** sont dérivées de celles développées récemment en reconnaissance. Une connaissance implicite de l'objet, est obtenue via un apprentissage sur une base de données.

Les approches par primitives

Les approches par primitives consistent à déterminer explicitement des invariants dans la classe objet, et leurs relations entre eux. La détection de l'objet consistera alors en deux passes :

– extraction des invariants dans l'image,

– analyse selon un modèle *a priori* de l'objet.

A travers cette algorithmie, se dessine un schéma souvent rencontré en vision : analyse bas niveau, et recherche de correspondances avec un modèle. Nous déclinerons donc les paragraphes suivants selon cette schématique. Dans les premiers, nous aborderons rapidement les caractéristiques bas niveaux qui sont généralement exploitées en détection d'objets, ainsi que les analyses «bas niveau» spécifiquement liées à celles-ci. Puis, dans les paragraphes suivants, nous discuterons des techniques d'analyse haut niveau utilisées, essentiellement construites sur du *template matching* avec des modèles prédéfinis ou déformables.

Les invariants

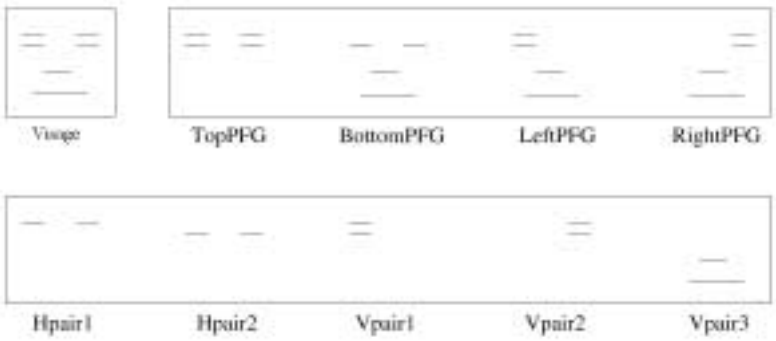
Niveaux de gris Une stratégie simple consiste à lisser une image, afin d'éliminer les détails dans l'image et rechercher une représentation grossière, i.e. à faible résolution, correspondant à l'objet. Pour affiner cette détection, la résolution des zones détectées est alors augmentée pour tenir compte d'autres détails discriminants. La méthode utilisée par *Yang et Haung* [215] en est un bon exemple. Elle consiste à poser l'hypothèse suivante : à faible résolution, un visage est constitué de quatre pixels à niveaux de gris uniformes, entourés de pixels à fortes variations. De semblables hypothèses sont exprimées sur les caractéristiques faciales telles que les yeux et la bouche pour les résolutions plus élevées.

Contours ou segments Les contours ou les segments sont des primitives souvent utilisées en traitement d'image. Les méthodes développées en détection d'objets les utilisant sont très nombreuses,.

Les plus intéressantes, et sans doute les plus abouties, sont celles effectuant une opération dite de *grouping*. Elle consiste à décomposer l'objet à détecter en un ensemble de primitives (étant alors des segments ou des groupes de segments). Puis les distances inter-primitives sont modélisées. Détecter un objet consistera alors en deux étapes : trouver des primitives dans l'image ressemblant à celles de l'objet, puis de déterminer celles qui vérifient le modèle de distances. Cette approche a de nombreux avantages. Elle peut être appliquée à un grand nombre de type d'objets. Elle est assez robuste aux occultations partielles : il n'est pas forcemment nécessaire de trouver toutes les primitives du modèle pour valider la présence de l'objet. Elle peut tenir compte des différentes orientations ou poses de l'objet, soit en projetant un modèle 3D, soit en consistuant un ensemble de modèles 2D (sélectionnés par exemple sur une sphère de vues). Dans les méthodes suivant ce schéma, citons celle développée par *Yow et Cipolla* [219] pour la détection de visage. Ils décomposent un visage en un ensemble de paires de segments (cf. figure 2.1.2). Ces primitives sont mises sous la forme de vecteurs contenant les longueurs des segments, leurs intensités et les variances associées. Ce qui permet, en utilisant une base de données, de calculer pour chaque primitive, sa moyenne et sa covariance dans cet espace vectoriel. Ils obtiennent alors des prototypes pour chaque primitive. Pour la détection, l'image est balayée via une fenètre d'intérêt. Les primitives extraites dans cette fenètre sont alors comparées aux protypes (selon une distance de Mahalanobis). En fonction des distances obtenues et via un seuillage, les primitives sont labélisées, puis groupées selon le modèle des distances inter-primitives. L'identification du visage est ensuite assurée par l'utilisation d'un réseau bayésien.

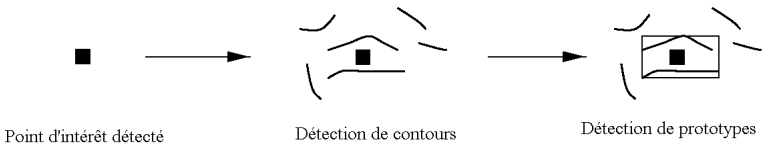
Couleur Le passage dans l'espace couleur se traduit par une augmentation de la dimension de l'espace représentatif des objets. Ainsi pour des objets chromatiques (la peau humaine, amers couleurs,...), cette information est très discriminante. Différents espaces de couleur peuvent être utilisées suivant le type d'objets à détecter. L'espace choisi est souvent celui qui offre les meilleurs résultats, soit en terme de discrimination, soit en terme de rapidité d'exécution. Un des espaces les plus utilisés est le RGB normalisé, puisqu'il demande peu de traitement et qu'il filtre les effets de la luminance. L'essentiel des techniques consiste à étudier l'histogramme des couleurs, soit en fixant des seuils déterminés empiriquement, soit en utilisant des mesures statistiques établies sur de larges bases de données qui modélisent la variation des couleurs pour un type d'objet. Dans ce dernier cas, citons les travaux de *Oliver et al.*[152] qui utilisent une distribution Gaussienne pour représenter la classe couleur de la peau. La distance de Mahalanobis est alors utilisée sur la couleur de chaque pixel comme mesure de vraisemblance avec le modèle, et permet alors une segmentation des images.

Texture Comme la couleur, la détection via la texture consiste essentiellement en une classification des pixels ayant une texture similaire à l'objet à détecter. Plusieurs descripteurs de texture peuvent être employés : les matrices de co-occurrences, de simples sommes et différences d'histogrammes, des transformées en ondelettes [201], ou des mesures statistiques de second ordre (énergie, entropie,...) [53, 116]. Ce type de primitives est souvent utilisé pour identifier des objets non-rigides (peau ou éléments du corps



(a) Modèle du visage et sa décomposition en paires de segments

(b) Prototypes de primitives (sourcils, yeux, nez et bouche)



(c) Processus de détection des primitives autour d'un point d'intérêt

FIG. 2.2 – Modèle d'un visage selon *Yow et Cipolla*

humain, nuages,...).

Mesures globales : la symétrie Certains objets présentent une symétrie naturelle : vue de face d'un visage, piétons, vues arrière ou avant d'un véhicule,... Aussi plusieurs papiers [164, 165, 131] proposent des opérateurs pour détecter des objets présentant une telle caractéristique dans leur apparence. Ceux-ci assignent une valeur à chaque pixel de l'image selon la contribution des pixels environnants. Ces approches aboutissent à la construction d'images de symétrie. D'autres méthodes supposent la symétrie effective dans une zone d'intérêt, et en recherchent le centre. Leur intérêt est la localisation précise de l'objet dans l'image.

Templates Matching

Les techniques de *Templates matching* consistent à relier des éléments indépendament détectés par un modèle soit prédéfini empiriquement ou paramètré par une fonction. Par exemple, la détection d'un visage sera réalisée à partir de celle du contour facial, des yeux, du nez et de la bouche. L'identification d'un visage sera déterminée par un critère (valeur de corrélation) tenant compte globalement de ces éléments.

Cette approche nécessite de tenir compte des variations en échelle, en pose et en forme. Aussi des modèles multi-résolutions, multi-échelles, subdivisés ou déformables sont utilisés.

Modèles prédéfinis L'utilisation d'un modèle prédéfini consiste à modéliser les relations entre plusieurs éléments remarquables de l'objet. La figure 2.3 donne pour l'exemple le modèle défini par *Scassellati* [177] dans son système de localisation de visages. Les relations définies sont de deux sortes : soit géométriques (i.e. les distances relatives entre deux éléments), soit en luminance (i.e. les contrastes entre les moyennes de niveaux de gris calculées dans chaque zone).

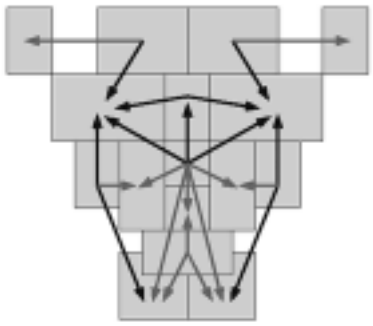


FIG. 2.3 – Exemple d'un modèle prédéfini [177] : il est composé de 16 régions (les rectangles gris) et de 23 relations (les flêches).

Cela peut s'apparenter à une analyse multi-primitives. Souvent cette analyse a lieu de manière séquentielle ou globale (sur une constellation de primitives). Dans les deux

cas, de nombreuses techniques existent. Nous allons en citer quelques unes, celles qui nous paraissent les plus représentatives.

Dans le cas d'une recherche séquentielle ou incrémentale, le degré de confiance dans l'identification d'un élément est augmenté si des éléments voisins sont aussi détectés. Souvent des approches dites de focalisation d'intérêts sont utilisées. Elles consistent à détecter et localiser un ou plusieurs éléments simples de l'objet. Cette détection grossière est alors confirmée et affinée en recherchant selon le modèle géométrique la présence d'autres éléments caractéristiques de l'objet. Une fonction de coût globale est souvent employée pour marquer l'identification de l'objet.

Par exemple, *Jeng et al.* [104] propose un système pour la détection de visage basé sur des mesures anthropométriques. Initialement, ils recherchent les différentes localisations possibles des yeux. La distance séparant chaque paire des yeux possible est utilisée pour se définir des zones de recherches pour les autres éléments du visage (nez, bouche, sourcils). A chaque élément, une valeur de vraisemblance est associée. Ces valeurs sont ensuite combinées suivant une loi linéaire exprimée dans l'équation 2.1, qui attribue un poids différent à chaque élément. Ces pondérations sont évaluées empiriquement.

$$E = 0.5E_{\text{yeux}} + 0.2E_{\text{bouche}} + 0.1E_{\text{sourcil droit}} + 0.1E_{\text{sourcil gauche}} + 0.1E_{\text{nez}} \tag{2.1}$$

D'autres articles proposent l'application d'un modèle global de l'objet dans une image (ou constellation) de primitives élémentaires. Ces modèles représentent souvent la silhouette de l'objet à détecter. Pour la détection de piétons [77], des techniques semblables en deux passes ont été mises au point. Dans un premier temps, des cartes de vraisemblance sont établies par la mise en correspondance de modèles de silhouettes avec une image des contours (basée sur la distance de Haussdorf). Ensuite, l'auteur utilise une classification de type RBF (*Radial Basis Functions*) pour identifier les zones détectées comme étant des piétons ou non.

Modèles déformables Les modèles déformables sont une réponse séduisante à la forte variabilité 3D ou 2D de formes ou d'aspects dans certaines classes d'objets, telles que les visages ou les véhicules. Le principe des modèles déformables est le suivant : le modèle positionné initialement à proximité du motif, va évoluer suivant des mesures images locales (gradients, luminance) pour prendre la forme du motif. Typiquement, ces modèles sont utilisés en suivi d'objets, mais quelques papiers les proposent aussi pour la détection.

En fait, il s'agit essentiellement de palier au problème de la détection des contours dans des images à faibles contrastes [221], de localiser précisément les contours du motif ou d'initialiser un suivi de l'objet.

Trois sortes sont distinguées : les *snakes* ou contours actifs [84], les *deformables templates* [221] et les *points distributed models* (PDM) [48, 125].

Les approches par apparence

Le plus grand défaut des méthodes par primitives est qu'elles sont très sensibles à l'environnement de l'objet. Elles nécessitent souvent, pour être efficace, un arrière plan à l'opposé des primitives utilisées. Par exemple, pour une méthode utilisant les contours, un environnement contenant peu de contours marqués est préférable. Ainsi la robustesse d'une approche par primitives est souvent liée à l'environnement dans lequel la détection doit avoir lieu. Les hypothèses à vérifier en limitent le domaine d'application.

Les techniques basées sur l'apparence répondent dans une certaine mesure à ce problème. La détection de l'objet est traitée comme un problème de reconnaissance d'images. Ces méthodes permettent de contourner les erreurs potentielles dues à une modélisation incomplête ou inappropriée. L'idée est d'utiliser un processus d'apprentissage pour classifier des exemples (ou échantillons) d'images en prototypes, représentant d'une part la classe objet et d'autre part la classe non-objet. Ensuite, décider si cette image contient ou non l'objet, revient à comparer ces prototypes et l'image observée.

La plupart de ces approches nécessite de balayer l'image par fenètre d'intérêt. Ce balayage a pour but d'effectuer une recherche exhaustive dans l'image de l'objet à toutes les positions et échelles possibles. Bien sûr, l'implémentation (taille de la fenêtre, échantillonage, pas d'échelle,...) de cette recherche va dépendre soit de la méthode choisie de classification soit des exigences requises pour le système.

Un système de détection via l'apparence se décompose comme un système de reconnaissance des formes, en deux parties (cf. figure 2.4) :

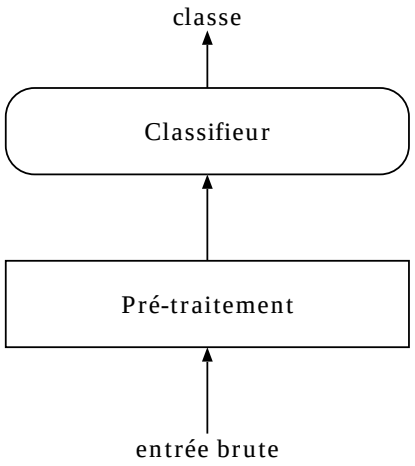


FIG. 2.4 – Synopsis d'un système de reconnaissance des formes

– Un classifieur entraînable, avec des paramètres libres, idéalement choisis pour minimiser un critère d'erreur. L'entrée du classifieur est la sortie du module de prétraitement. En sortie, il fournit la classe à laquelle appartiennent les données fournies

en entrée). En détection, cela consiste soit à utiliser un (ou des) modèle(s) de distribution et à faire une mesure de ressemblance, soit à déterminer une fonction discriminante (i.e. surface de décision, séparation hyperplane, fonction seuil) entre les classes objet et non-objet.

– Un module de prétraitement (*preprocessing*) qui permet l'extraction de traits (*feature extraction*) de l'entrée brute (il s'agit des données dont on veut vérifier l'appartenance à telle ou telle classe). Le prétraitement sert à sélectionner parmi les données brutes, des caractéristiques qui aident spécifiquement à séparer les deux classes. Soit ce prétraitement est appris (c'est ce qui est fait avec des réseaux de neurones multi-couches), soit il utilise des connaissances *a priori* pertinentes. Généralement, ce prétraitement consiste à projeter les images (donc des points dans un espace à N dimensions, où N est le nombre de pixels d'une image) dans un (sous-)espace plus approprié.

Dans cette section, nous verrons quelques unes des principales méthodes de détection par apparence. Nous exposerons d'abord celles linéaires, puis celles non-linéaires (exploitant soit des réseaux de neurones soit des outils statistiques multi-variables).

Méthodes linéaires Ce que nous nommons méthodes linéaires fondent la classification des images sur des mesures de distances entre les images à tester et le(s) sous-espace(s) des images échantillons.

Ce sous-espace est souvent défini à partir d'une Analyse en Composantes Principales (*Principal Component Analysis* ou PCA) afin d'obtenir une représentation de l'objet la plus réduite possible. Cette décomposition est calculée sur les exemples de la classe objet. Elle se base sur un calcul des vecteurs et des valeurs propres de la matrice de covariance. Les vecteurs propres aux valeurs les plus élevés représentent alors les éléments communs aux modèles.

Une autre représentation linéaire est calculée sur les exemples des deux classes, et l'on retiendra alors les vecteurs propres séparant au mieux les deux classes. On parle alors d'Analyse Linéaire Discriminante ou *Linear Discriminant Analysis*.

Une des techniques permettant d'améliorer encore cette réduction de données consiste à considérer plusieurs classes objet et non-objet différentes (ou *clusters*) et de calculer pour chacune, les vecteurs les plus représentatifs, via une analyse discriminante. Par exemple, *Sung et Poggio* [191] classifient leur base de données en 6 *clusters* pour les visages et 6 *clusters* pour les non-visages. Ceux-ci sont approximés par des fonctions Gaussiennes multi-dimensionnelles, avec leurs valeurs moyennes et leurs variances respectives (cf. figure 2.5).

Les mesures de distances pour la classification couramment utilisées sont (cf. figure 2.6) :

– DFFS ou *distance from feature space* qui est la distance entre l'image à tester et

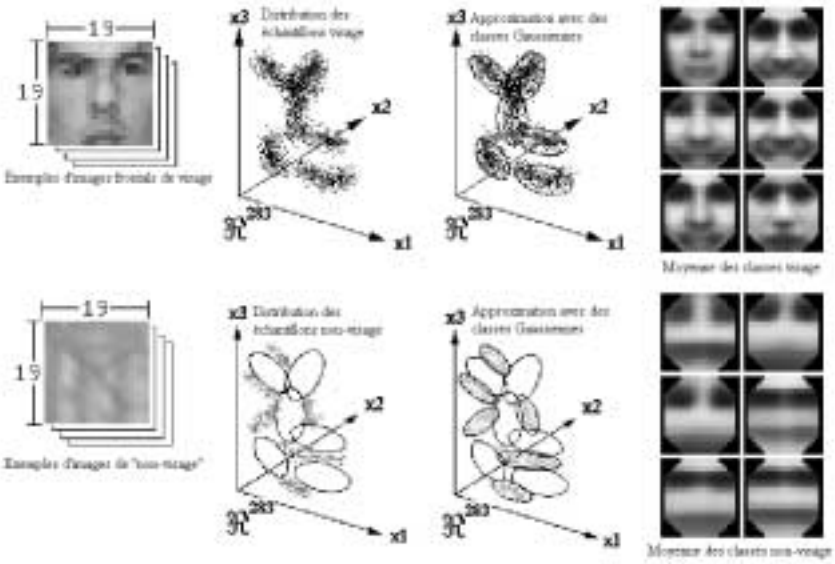


FIG. 2.5 – Les *clusters* visages et non-visages selon *Sung et Poggio*.

un des (ou le) sous-espaces des échantillons.

– DIFS ou *distance in feature space* qui est la distance entre l'image à tester projetée dans le sous-espace des échantillons et une image moyenne des échantillons.

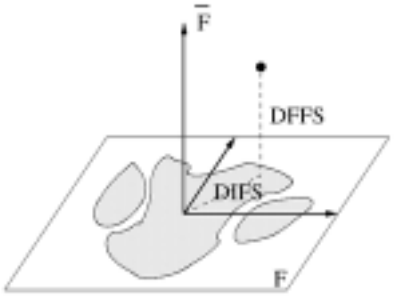


FIG. 2.6 – Décomposition des distances.

Les normes utilisées sont de différentes natures suivant les méthodes développées. Par exemple, *Sung et Poggio* utilisent une distance euclidienne pour la DFFS, et une distance de Mahalanobis pour la DIFS, qui se prêtent bien à leur modèle Gaussien de distribution.

Les différentes distances sont alors soumises à des règles de décision plus ou moins complexes, allant de simples seuils (fixés empiriquement) ou requérant des algorithmes de classification complexes tels que MLP (*Multi-layered Perceptron*) [191].

Réseaux de neurones S'inspirant du fonctionnement du cerveau humain, les réseaux de neurones sont une technique assez séduisante pour les problèmes de vision par ordinateur, en particulier ceux de reconnaissance. Son objectif principal est d'apprendre des classifieurs plus riches que ceux linéaires, et en particulier la transformation non-linéaire. Il existe un grand nombre de modèles de réseaux ; ils peuvent varier selon le nombre de couches, de neurones, l'algorithme d'apprentissage, etc...

Le système mis au point par *Rowley et al.* [170] pour la détection de visage est un bon exemple de ce qu'il est possible de faire. Tel que le montre la figure 2.7, les imagettes, collectées via une fenêtre glissante (parcourant une décomposition en échelles de l'image originale), subissent d'abord un prétraitement destiné à les normaliser. Le système effectue ensuite un tri parmi les réponses de différents réseaux (ayant été appris sur différentes zones de l'image : carrées ou en lignes) par l'intermédiaire d'un nouveau réseau, ou bien, il peut prendre en considération l'ensemble des sorties afin d'achever la détection.

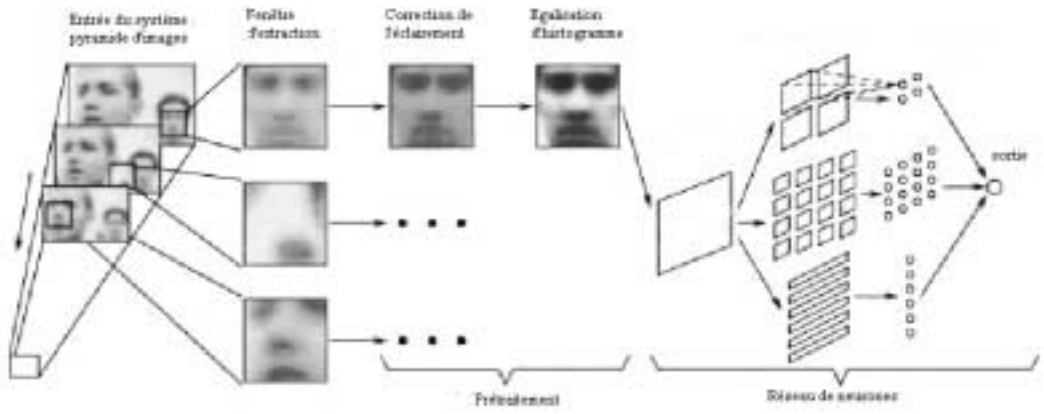


FIG. 2.7 – Le système de détection de visages de *Rowley et al.* [170].

A travers cet exemple, nous voyons que les réseaux de neurones peuvent être employés à différentes tâches de la détection : en prétraitement, en prédétection (localisation de zones d'intérêts) ou dans la classification finale d'une imagette. Ceci montre la grande flexibilité des réseaux de neurones. Cependant cette grande flexibilité est aussi la principale difficulté de cette technique ; du fait du grand nombre des réseaux possibles, déterminer le réseau optimal pour une tâche précise demande une grande connaissance de cette technique et de l'habileté.

Approches statistiques Dans ce paragraphe, nous avons regroupés les autres méthodes utilisant des classifieurs issues des recherches en statistiques. Comme les réseaux de neurones, ils permettent des classifications non-linéaires.

Une des techniques pour modéliser un objet par apprentissage est la modélisation via une chaine de Markov cachée ou *Hidden Markov Model* (HMM). Elle se base sur l'hy-

pothèse que les images peuvent être caractérisées par un processus aléatoire paramétré. Un tel modèle est défini par un certain nombre d'états. La probabilité de passage d'un état à l'autre est calculée lors de l'apprentissage ; chaque échantillon est alors présenté au système comme une séquence d'observations. *Colmanerez et al.* proposent dans [45] d'utiliser cette technique pour créer un modèle de visage et de l'arrière-plan. Il sagit d'observer les transitions entre des couples de pixels. La détection est alors réalisée en calculant le degré de ressemblance avec les modèles.

Schneiderman et al. proposent dans [178, 179] des travaux basés sur la théorie de Bayes. Celle-ci leur permet de présenter un test de ressemblance sous la forme :

$$\frac{P(\text{image}|\text{objet})}{P(\text{image}|\text{non-objet})} > \frac{P(\text{objet})}{P(\text{non-objet})} \tag{2.2}$$

Si la partie gauche de l'inéquation est supérieure à la partie droite, l'image est considérée comme contenant un objet. Différents modèles d'apparence de l'objet sont proposés afin de réduire la quantité d'informations à traiter. Dans [178], ceux-ci sont subdivisés en plusieurs régions d'intérêt. Chacune de ces régions est projetée dans un espace à 12 dimensions (construit par PCA). Dans [179], les éléments visuels sont extraits des images après une transformation en ondelettes de Gabor (utilisée pour décomposer les images dans l'espace, en fréquence et en orientation). Cette méthode a été testée avec succès pour la détection de visages (de profil et de faces) ainsi que de véhicules dans différentes orientations.

Enfin, *Osuna et al.* [154] proposent un système conçu sur la même méthodologie que celui de *Sung et Poggio* [191]. Au lieu d'un classifieur linéaire, ce système utilise une machine à supports vectoriels ou *Support Vector Machine (SVM)*. Nous verrons dans la suite que celui-ci permet une classification non-linéaire des données. Ce même classifieur est plus tard repris par *Papageorgiou et Poggio* [157, 156] pour la détection de piétons et de visages. Nous avons choisi d'utiliser cette méthode dans nos travaux. Nous verrons à la fin de cette section les raisons d'un tel choix.

Discussion sur la détection d'objets

Ainsi, en matière de détection d'objets par vision, deux approches sont distinguées dans la littérature. La première est une *approche par primitives*. De tels systèmes exploitent souvent des primitives simples à extraire et fonctionnent donc en temps réel. La seconde est une *approche selon l'apparence*. Ces techniques récentes, issues de travaux en reconnaissance, s'avèrent très robustes, mais sont souvent très couteuses en temps de calcul car elles requièrent une exploration exhaustive, multi-échelles, multi-résolutions des images.

Dans le cadre de notre application, les deux qualités principales de ces approches sont requises : rapidité et robustesse. Aussi, il nous est apparu intéressant de combiner ces deux approches.

2.1.3 La détection de véhicules en vision embarquée

Dans cette section, nous détaillerons les différentes informations image utilisées pour la détection de véhicules dans des scènes statiques, à partir d'un capteur de vision embarqué. Puis nous présenterons un petit panorama des approches utilisant ces indices de présence et amenant à l'identification, donc à la détection d'un véhicule dans l'image.

Nous retrouverons bien entendu un grand nombre des éléments décrits dans les paragraphes précédents. C'est pourquoi nous seront assez succints ; nous intéressant plus aux informations utilisées qu'aux techniques de traitement d'image.

A partir de ce panorama, nous expliciterons et justifierons la méthode que nous avons développée.

Les informations image liées à un véhicule

Dans la littérature, les informations image utilisés pour la détection par vision de véhicule sont les suivantes :

- la texture,
- la symétrie horizontale,
- la couleur,
- l'ombre portée,
- les segments,
- la route.

Dans les paragraphes qui suivent, ces différents indices seront évoqués et commentés suivant leur exploitation possible, leurs défauts éventuels par rapport à la détection de véhicules.

La texture. La texture est exploitée par [109] pour segmenter l'image en tant que première étape de la détection. Dans ce papier, deux approches sont proposées : une utilisant la mesure de l'entropie, l'autre le calcul de matrices de co-occurrences (réunissant 4 mesures : l'énergie, le contraste, l'entropie et la corrélation). La seconde approche est plus efficace, mais augmente le temps de calcul. L'utilisation de la texture seule entraîne de nombreuses fausses détections.

La symétrie horizontale. Dans les travaux de *Zielke et al.* [225] et de *Broggi et al.* [29, 15] , la symétrie est estimée soit, pour le premier, dans l'image des niveaux de gris, soit dans les images de contours vertical et horizontal pour le second, et utilisée comme indice pour la détection. Ces approches exploitent le fait que l'image d'un véhicule observé dans la vue frontale présente généralement une symétrie suivant un axe vertical.

La couleur. La couleur est très peu utilisée dans les systèmes de vision embarquée, du fait de la trop grande diversité de couleur des véhicules automobiles. Elle peut être cependant utile dans certaines occasions, notamment la nuit, dans des conditions de visibilité réduite ou lors de manœuvres particulières (freinage,...), pour détecter les feux

arrières ou avant des véhicules [20, 189]. D'autres travaux [142, 8] utilisent aussi la couleur (en fait, les images achormatiques fournies par une caméra achromatique munie d'un filtre IR) pour détecter des amers (cf. fig. 1.15) sur des véhicules coopératifs.

Les ombres portées. Les véhicules ont des apparences en forme, en taille et en couleur très diverses. Cependant, une caractéristique leur est commune : leur ombre portée sur la route. Elle est utilisée dans [144, 199, 202, 189]. Dans ces papiers, les zones d'ombres potentielles sont définies comme ayant une intensité nettement plus sombre que celle de la route. Les ombres portées sont aussi nettement plus sombres que celles d'autres objets, tels que les arbres présents aux bords de la route. Aussi, cet indice peut être aussi utilisé dans ces situations. Le principal défaut vient des fausses détections dues aux objets sombres dans l'arrière plan (i.e. bords de la route, véhicule noir,...).

Les segments. D'autres indices sont les segments verticaux et horizontaux. Dans [55], les auteurs utilisent la transformation de Hough généralisée pour identifier les lignes et les colonnes qui peuvent contenir les contours englobants d'un véhicule. Dans les travaux de *Betke et al.* [20], les véhicules situés au loin sont identifiés en utilisant les projetés des contours. Ces projetés indiquent la présence des contours verticaux et horizontaux très prononcés, qui pourraient être liés à une structure rectangulaire. visages. Le principal défaut de cette méthode est qu'elle est dépendante de l'arrière plan (il doit contenir peu de segments) et d'une extraction de contour efficace.

La route La connaissance de la position de la route dans l'image permet de limiter la zone de recherche pour la détection. Par ailleurs, certains auteurs proposent de détecter les obstacles via la détermination de l'espace libre d'évolution, ou *free-driving space*. A l'aide d'algorithmes de morphologie mathématique, *Beucher et al.* [220] segmentent des images de situation routière. Ils déterminent notamment la portion de route au devant du véhicule. Sa mise en correspondance avec un modèle géométrique de type trapèze indiquera la présence d'un véhicule (au niveau de la plus petite base du trapèze), et celle avec un triangle, le contraire. Cette méthode n'est efficace que si la segmentation est de qualité : elle est inefficace en présence d'ombres inopportunes ou d'irrégularités (en luminance) dans le revêtement de la route.

De l'extraction des primitives à l'identification d'un véhicule

Dans la littérature, diverses stratégies sont développées pour utiliser les différents indices visuels que nous avons cités. Nous présentons ici quelques travaux qui sont représentatifs des différentes méthodologies employées.

L'algorithme de détection de véhicules proposé par *Broggi et al* [17, 29] et implementé dans le véhicule expérimental ARGO se décompose de la manière suivante. Premièrement, une zone d'intérêt est identifiée selon la position de la route et les déformations perspectives. Dans cette zone, une analyse des symétries horizontales potentielles est

effectuée sur les niveaux de gris et sur les contours horizontaux et verticaux. Les zones symétriques ayant été ainsi détectées, les coins inférieurs correspondant au bas du véhicule sont recherchés. Enfin, la limite supérieure du véhicule est déterminée, et celui-ci est alors localisé dans une carte 3D obtenue par stéréovision.

Denasi et al. [56] et *Foresti et al* [72] ont des approches similaires d'identification des véhicules. Il s'agit de méthodes de *grouping*. Pour être rapide et robuste, elles sont effectuées sur plusieurs images successives (au cours d'une phase de suivi) et dans des zones d'intérêts. Celles-ci sont désignées soit par analyse des changements dans l'image [72] (cette méthode est appliquée dans le cadre d'une caméra fixe), soit par la détection d'ombres portées sur la route [56].

Betke et al. proposent dans leurs travaux [20] de combiner la détection de plusieurs indices (symétrie, contours, couleur...) en construisant empiriquement une loi linéaire de vraisemblance. A chaque indice, sont associées une estimation de vraisemblance et une pondération reflétant l'importance de l'indice pour l'identification du véhicule. L'identification d'un véhicule est réalisée si ces indices sont persistants, i.e. par l'intégration de cette loi lors du suivi des zones détectées et son seuillage par hystérésis. Par ailleurs, ils distinguent deux zones où détecter les véhicules, le FOC et les zones passantes, et considèrent ces zones suffisantes : les véhicules se trouvant entre ces zones sont supposés avoir été détectés précédemment.

Marinus B. von Leeuwen et Frans C.A. Groen [202] appliquent aussi deux approches différentes selon que l'obstacle (ici, une voiture) est en phase de dépassement ou à distance. Pour les seconds, ils utilisent une combinaison de trois indices. Premièrement, ils sélectionnent des régions à partir des ombres portées. Puis ils y analysent l'entropie et la symétrie horizontale. La décision est basée sur un ensemble de seuils.

Bruno Steux [189] fusionne divers indices détectés (ombres portées, feux arrières, segments verticaux et symétrie) via un réseau bayésien et obtient ainsi l'identification de véhicules. Le choix d'un tel processus est en partie motivé par la possibilité d'ajouter en entrée d'un tel réseau, des informations provenant d'autres types de capteurs, tels que RADAR, télémètre laser,...

Les chercheurs de l'*Institut für Neuroinformatik* de l'Universitée de Ruhr ont appliqué la technique des réseaux de neurones (de type DNN, *Discrete Neural Network*) pour la détection et la classification de vues arrière et frontale de véhicules. Ces algorithmes ont été utilisés et testés sur plusieurs type d'indices (texture [109], symétrie [110], contours [82]). A noter que *Zhao Liang et Charles Thorpe* [130] proposent aussi d'utiliser un réseau de neurone pour la reconnaissance d'objets pré-détectés via un système de stéréovision.

Matthews et al. [144] proposent un processus hybride de détection et de reconnaissance à deux étages. Le premier, servant à désigner des zones d'intérêts, exploite les

lignes horizontales pour la localisation horizontale et les ombres portées par la verticale, ainsi que quelques hypothèses géométriques sur la largeur et la hauteur des véhicules. Des éléments caractéristiques sont ensuite extraits via une ACP, et sont fournis en entrée d'un classificateur de type MLP .

2.1.4 Bilan et choix d'une approche

Dans le panorama précédent, nous pouvons remarquer que les systèmes de vision embarquée adoptent un processus à deux étages. Cette stratégie est essentiellement dictée par la nécessité d'obtenir un système temps réel. Elle consiste à détecter d'abord des zones d'intérêts dans l'image où des véhicules sont susceptibles d'être présents.

Pour *Betke et al.* [20], les ROI sont déduites de la détection de la route ; il s'agit du FOC et des zones passantes. Pour d'autres auteurs [144, 199, 202, 189], elles sont essentiellement déduites via la détection d'ombres portées potentielles.

Le second étage procède à l'identification ou non d'un véhicule pour chaque ROI. Cette identification est souvent obtenue via une approche par primitives. Elle consiste à une mise en correspondance de différents indices (ombres, segments, couleur,...) détectés. Elle peut faire appel à des classifieurs simples reposant sur un ensemble de seuils [20] ou des classifieurs plus complexes comme des réseaux bayésiens [189].

Des approches par apparence sont très rarement exploitées. Celles-ci ont pourtant fait leur preuve, notamment en détection de visages. Le tableau 2.1 présente par exemple une comparaison de différentes approches par apparence, que nous avons cité précédemment. Ces pourcentages de bonnes détections ont été obtenus sur des bases d'images relativement complexes (en terme d'arrière plan et de variabilité des visages). Nous pouvons voir que dans l'ensemble, ces résultats ne permettent pas vraiment de les différencier. Cependant, ceux-ci démontrent la robustesse de telles approches. Aussi, il nous est apparu intéressant d'exploiter une telle approche pour la détection de véhicules dans le cadre de notre application.

Nous avons sélectionné une méthode semblable à celle d'*Osuna et al.*, basée sur un classifieur SVM. Cette méthode a été développée par *Papageorgiou et al* pour la détection d'objets, tels que les visages [157] ou les piétons [156]. Le choix de cette méthode s'explique pour plusieurs raisons :

- D'abord, *Papageorgiou et Poggio* en proposent une adaptation [155] à la détection de vues arrière et avant de véhicules dans des bases de données d'images. L'application visée alors est l'indexation d'images. Les résultats obtenus sur plusieurs types d'image et dans différents type d'environnement (urbain, autoroute, campagne) sont très interessants (cf. annexe A).

- Ensuite, il s'agit d'une méthode générique pouvant être appliquée à divers objets, donc des obstacles de différentes natures. Dans ce mémoire, nous nous sommes

Système de détection de visages	CMU-130	CMU-125	MIT-23	MIT-20
Schneiderman & Kanade-E[a] [179]		94,4% /65		
Schneiderman & Kanade-W[b] [179]		90,2% /110		
Yang et al. -FA [216]		92,3% /82		89,4%/3
Yang et al. -LDA [216]		92,3% /82		91,5%/1
Roth et al. [168]		94,8% /78		94,1% /3
Rowley et al [170]	86% /8		84,5% / 8	
Feraud et al [69]	86% /8		84,5% / 8	
Colmenarez & Huang [45]	93,9% / 8122			
Sung & Poggio [191]			79,9% /5	
Lew & Huijsmans [128]			94,1% /64	
Osona et al. [154]			74,2% /20	
Lin et al. [132]			72,3% /6	
Gu & Li [83]			87,1% /0	

[a] cœfficients de vecteurs propres
[b] cœfficients d'ondelettes

TAB. 2.1 – Résultats des approches en termes de pourcentage de détection correctes et en nombre de fausses détections, sur des bases de données du CMU et de MIT, extrait de [95]

limités aux voitures dites de tourisme (et utilitaires de faible envergure). Mais la méthodologie présentée pourrait très bien être étendue à d'autres obstacles routiers.

- Par ailleurs, les techniques utilisées dans cette méthode sont assez récentes et très en vogue actuellement dans les recherches en détection, en suivi et en reconnaissance d'objets. Aussi il est à espérer que les années, et même les mois, à venir soient très riches en développement dans ces domaines. Nous verrons notamment dans la partie réservée aux perspectives potentielles aux méthodes développées, que certains travaux récents [205, 208] offrent déjà des solutions ou des pistes de recherche intéressantes.

- Enfin, cette méthode s'avère relativement rapide : l'exploitation d'un classifieur SVM rend la méthode développée par *Osuna et al.* [154] 30 fois plus rapide à celle de *Sung et Poggio* [191] pour des résultats comparables, alors que, dans leur construction, elles sont similaires. De plus, suite à une collaboration avec Daimler-Chrysler, *Poggio et al.* proposent leur méthode pour la détection de piétons en temps réel sur des séquences d'images [158]. Le fonctionnement en temps réel est alors obtenu en focalisant l'algorithme grâce à une prédétection via un capteur stéréoscopique.

Dans nos travaux, nous proposons une stratégie similaire à cette dernière, mais adaptée à une vision monoculaire. Il s'agit d'utiliser une détection de primitives simples pour

focaliser l'identification via l'apparence. Nous retrouvons donc le même schéma à deux étages, tel que l'illustre la figure 2.8.

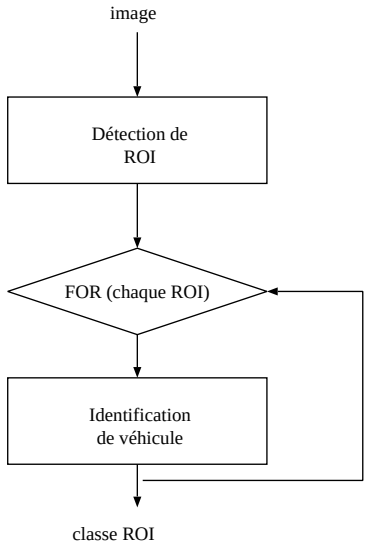


FIG. 2.8 – Processus de détection de véhicules à deux étages

Dans l'ensemble des primitives utilisables, deux nous ont paru particulièrement intéressantes :

- *l'ombre* : elle est présente pour tous les véhicules, dans des conditions normales d'éclairement, et semble facile à détecter. Elle est de plus généralement située sous le véhicule, ce qui permet une localisation du véhicule en ordonnée dans l'image. De plus, si l'on est capable de reconstruire un modèle 3D de la route, la localisation de l'ombre sur la route devra permettre une localisation en profondeur du véhicule potentiel.

- *la symétrie* : elle est aussi généralisable à tous les véhicules et permet une localisation en abscisse.

La combinaison de ces deux primitives, avec une détection et une reconstruction de la route, devraient donc permettre une localisation en coordonnées et en échelle d'un véhicule potentiel dans l'image.

C'est donc une approche, qualifiée d'*hybride*, que nous avons adoptée et qui est décrite dans la section suivante : une prédétection par primitives est utilisée pour focaliser une identification via l'apparence.

2.2 Une approche hybride

Ainsi notre approche se décompose en deux parties :

- des algorithmes dédiés à la détection des ombres sur la route, et de la symétrie axiale,

- une méthode d'identification via l'apparence.

Dans ces méthodes, nous utilisons une modélisation de la scène à analyser : à la fois une modélisation de la route et un ensemble de représentations des véhicules, notamment une décomposition en ondelettes de Haar. Cette dernière est utilisée dans les deux parties de notre approche. Cependant, comme nous le verrons dans la suite, son choix est essentiellement dicté par la méthode d'identification. Aussi, pour des raisons de clarté, il nous est apparu plus judicieux de décrire cette méthode avant d'expliciter les algorithmes servant à la focaliser.

Ainsi, cette section se subdivisera en trois parties. La première explicitera la méthode d'identification de véhicule choisie, en supposant celle-ci focalisée en échelle et en coordonnées sur une zone d'intérêt (que nous nommerons ROI comme *Region Of Interest* dans la suite). Dans la seconde, les algorithmes servant à cette focalisation seront explicités, en commençant par celui permettant la détection des ombres portées, puis par celui détectant la symétrie. Cette seconde partie sera aussi l'occasion d'exposer rapidement la modélisation de la route sélectionnée et l'algorithme servant à sa détection. Enfin, après une schématisation récapilutative de notre approche, nous discuterons des résultats obtenus.

2.2.1 Classification des zones d'intérêts par apparence

Papagergiou et al. proposent donc d'utiliser comme classifieur un classifieur SVM et un prétraitement basé sur la sélection de paramêtres dans une décomposition en ondelettes de l'image de véhicules. C'est dans cet ordre que nous allons expliciter ces deux composantes de la méthode.

Le principe des *Support Vector Machine*

Les Machines à Supports Vectoriels, *Supports Vectors Machines*, sont un processus d'apprentissage mis au point par Vladimir N. Vapnik, que l'on peut considérer comme l'un des pères fondateurs du domaine de l'Intelligence Artificiel, permettant la classification de données en deux classes. Ils consistent à déterminer une fonction limite, séparant au mieux les deux classes. Son principal avantage est qu'il peut être utilisé dans des espaces à grandes dimensions. Par ailleurs, il autorise les techniques dites *Kernel Based Learning Methods*. Ces techniques proposent, dans le cas de données non-séparables, de projeter les données dans des espaces de dimensions supérieures, de façon implicite via

une fonction définie par un noyau ou *kernel*.

Nous allons développer brièvement ces différents points dans les paragraphes qui suivent. Le lecteur intéressé se rapportera à [203, 30], qui sont les références dans le domaine. Nous conseillons aussi la consultation de [171] qui décrit l'implémentation des SVM que nous avons utilisée.

La classification de données Considérons l'espace $\mathbb{R}^N$ des échantillons. Un point dans cet espace sera noté $\mathbf{x} = (x_1, x_2, ..., x_N)$. Comme nous l'avons déjà signalé, dans le cas qui nous intéresse (i.e. la détection), la classification se fait entre deux classes : la classe objet et la classe non-objet. Pour des données séparables, il est possible de définir la fonction de décision f séparant les deux classes sous la forme :

$$f(\mathbf{x}) = \text{signe}(\mathbf{w}.\mathbf{x} + \mathbf{b}) \tag{2.3}$$

Cette fonction définit, via le vecteur $\mathbf{w}$, un hyperplan dans l'espace $\mathbb{R}^N$. Ce type de classificateur est dit linéaire. Plusieurs hyperplans conviennent (cf. fig. 2.9). Parmi eux, il en existe un unique qui permet d'obtenir une marge de séparation maximale entre les deux classes de points (cf. fig. 2.10). La marge est la distance minimale entre la surface de décision et les échantillons $\mathbf{x}_i$ (qui sont supposés être tous correctement classifiés par cette surface de décision). La classe de ces échantillons est notée y_i, avec $y_i = 1$ s'il s'agit d'un objet et $y_i = -1$ sinon.

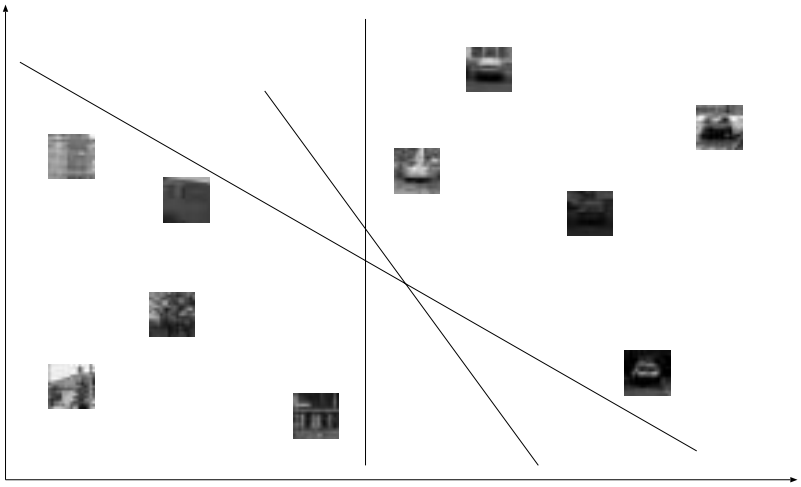


FIG. 2.9 – Plusieurs hyperplans sont possibles pour séparer les classes.

La marge de séparation maximale est donc définie par la formule :

$$\begin{gathered} \mathbf{w}^* = \max_{\mathbf{w},\mathbf{b}}\{\min\{\|\mathbf{x} - \mathbf{x}_i\| : \mathbf{x} \in \mathbb{R}^N, (\mathbf{w}.\mathbf{x}) + \mathbf{b} = 0, i = 1, ..., l\}\} \\ \Leftrightarrow \\ \mathbf{w}^* = \text{argmax}_{\mathbf{w}} M(\mathbf{w}, D), \text{avec } M(\mathbf{w}, D) = \min_i \frac{y_i f(\mathbf{x}_i)}{\|\mathbf{w}\|} \end{gathered} \tag{2.4}$$

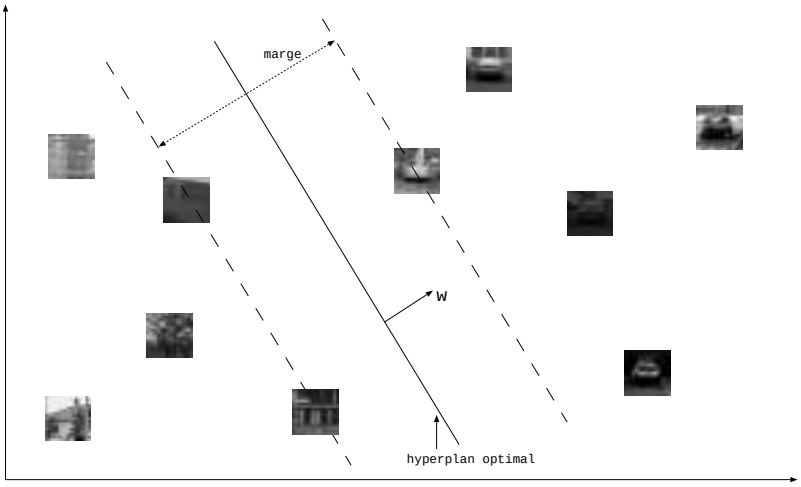


FIG. 2.10 – Hyperplan optimal séparant les deux classes.

La détermination de cet hyperplan optimal peut être obtenue par le processus d'apprentissage SVM.

Points de support On peut montrer que $\mathbf{w}^*$ ne dépend que des $(\mathbf{x}_i, y_i)$ qui sont sur la marge. On appelle ces échantillons des *points de support*. Les SVM vont donc utiliser le fait que la plupart des échantillons ne sont pas des points de support, et que (dans le cas de la fonction de décision linéaire) la solution a forcément la forme d'une combinaison linéaire des exemples,

$$\mathbf{w} = \sum_i y_i \alpha_i \mathbf{x}_i \tag{2.5}$$

où les α_i sont des scalaires. Donc on a forcément $\alpha_i = 0$ pour les i qui ne sont pas des points de support. L'introduction des $y_i \in \{-1, 1\}$ est réalisée pour simplifier les mathématiques et garder les α_i positifs. On peut alors réécrire la fonction de décision 2.3 sous la forme

$$f(\mathbf{x}) = \text{signe}(\mathbf{w}.\mathbf{x} + \mathbf{b}) = \text{signe}(\sum_i y_i \alpha_i (\mathbf{x}_i.\mathbf{x}) + \mathbf{b}) \tag{2.6}$$

Grâce à cette reparamétrisation astucieuse, avec les α_i plutôt qu'avec les $\mathbf{w}$, il est possible de formuler la détermination du plan optimal $\mathbf{w}^*$ sous la forme d'un problème d'optimisation quadratique sous contraintes. A nouveau, nous conseillons la consultation de [203, 30] pour de plus amples détails.

Ce problème consistera donc à maximiser :

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i.\mathbf{x}_j) \tag{2.7}$$

sous les constraintes $\alpha_i = 0$ et $\sum_i y_i \alpha_i = 0$.

Une fois les α obtenus, le biais se déduit par l'expression des conditions de complémentarité de Karush-Kuhn-Tucker :

$$\alpha_i[y_i((\mathbf{w}.\mathbf{x}_i) + \mathbf{b}) - 1] = 0, i = 1, ...l \tag{2.8}$$

Ainsi, pour tout i tel que $\alpha_i \neq 0$, c'est-à-dire tel que $\mathbf{x}_i$ soit un vecteur de support :

$$\mathbf{b} = (1 - \mathbf{w}.\mathbf{x}_i).y_i \tag{2.9}$$

En pratique, ce problème d'optimisation peut être résolu par des méthodes de programmation quadratique classiques. Il existe aussi des implémentations optimales [38].

Le cas des données non-séparables Un développement relativement récent des SVM permet de définir des surfaces de décision non linéaires. Il s'agit des méthodes dites *kernel-methods*. Elles permettent l'extension aux données non-séparables (c'est-à-dire au cas où les échantillons donnés durant l'apprentissage ne peuvent être séparés dans $\mathbb{R}^N$ par un hyperplan). Cela se fait par l'introduction d'une fonction vectorielle ϕ non linéaire. Ceci consiste à projeter les $\mathbf{x}$ dans un espace à plus haute dimension (qui peut-être de dimension infinie),

$$f(\mathbf{x}) = \text{signe}(\mathbf{w}.\phi(\mathbf{x}) + \mathbf{b}) = \text{signe}(\sum_i y_i \alpha_i (\phi(\mathbf{x}_i).\phi(\mathbf{x})) + \mathbf{b}) \tag{2.10}$$

Ces fonctions ϕ sont choisies de sorte que le produit scalaire dans l'espace des ϕ est très peu couteux à faire. Elles s'écrivent sous la forme :

$$f(\mathbf{x}) = \text{signe}(\sum_i y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \mathbf{b}) \tag{2.11}$$

où $K(\mathbf{u}, \mathbf{v})$ est un noyau (d'où l'appellation *kernel methods*), tel que le produit scalaire puisse s'exprimer ainsi : $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u}).\phi(\mathbf{v})$.

Les noyaux couramment utilisés dans la littérature sont :

- $K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}.\mathbf{v} + 1)^d$ i.e., noyau polynomial où $\phi(\mathbf{x})$ est un polynome d'ordre d
- $K(\mathbf{u}, \mathbf{v}) = tanh(a\mathbf{u}.\mathbf{v})$ i.e., noyau sigmoïde où $f(\mathbf{x})$ a un comportement similaire à un MLP
- $K(\mathbf{u}, \mathbf{v}) = e^{-a\|\mathbf{u}-\mathbf{v}\|^2}$, i.e. noyau Gaussien

En introduisant directement un noyau dans l'équation 2.7, on a :

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2.12}$$

avec toujours les mêmes contraintes.

Ceci montre que la projection dans l'espace à plus haute dimension n'est en fait pas réalisée de manière explicite, mais implicitement.

Pour la détection d'objets, un noyau polynomial d'ordre 2 présente des qualités de discrimination suffisante, ainsi qu'un temps de calcul raisonnable [157, 179]. Aussi, c'est ce type de noyau que nous avons choisi.

Le prétraitement : ondelettes de Haar et choix des paramètres

Le choix du prétraitement est essentiel à la bonne marche d'un système de détection par apparence. Il s'agit de sélectionner le sous-espace dans lequel la discrimination entre les objets et les non-objets sera optimale, pour un temps d'exécution minimum.

Le prétraitement utilisé dans nos travaux, se décompose en trois étapes (cf. figure 2.14) :

1. Une mise à l'échelle : elle consiste essentiellement à ré-échantillonner l'image à analyser (ou une zone de l'image) en une imagette 128x128. Les imagettes apprises contenant un véhicule ont été extraites de photographies prises essentiellement dans des parkings (cf. figure 2.15). Elles représentent des vues d'arrières de véhicules. Ces véhicules sont centrés dans ces imagettes de la façon suivante (cf. figure 2.11) :

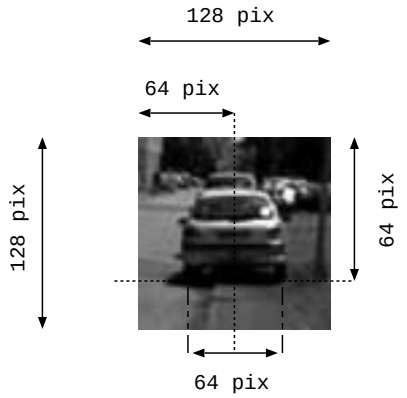


FIG. 2.11 – Exemple d'une imagette 128x128 contenant un véhicule pour la base d'apprentissage de notre système.

   – horizontalement : l'ordonnée du barycentre des interfaces des deux roues avec la route est placée à 96 pixels du haut. Ces interfaces sont données par la sélection de deux points dans l'image (un point pour chaque roue).

- verticalement : l'axe de symétrie donne la position verticale du véhicule dans l'imagette. Lors de l'apprentissage, celui-ci est fourni par la sélection de deux points sur des éléments symétriques facilement distingables dans la partie «basse» du véhicule (par exemple, les feux arrières). Cet axe est placé au centre dans les imagettes de la base d'apprentissage ; à 64 pixels des bords verticaux.
- l'échelle est déterminée en considérant l'écart entre les deux roues arrières normalisé. Pour des voitures de tourisme, cet écart est de l'ordre de 1,5 mètres. Dans les imagettes, cet écart est ramené à 64 pixels.

2. Un changement d'espace de représentation. Il s'agit ici de choisir un (sous-)espace de représentation qui discrimine aux mieux les objets.

   Dans la littérature, les plus utilisés sont basés soit sur une ACP soit sur une décomposition en ondelettes. Dans [179], à propos d'un système de détection de visages, les auteurs remarquent qu'avec un prétraitement utilisant une ACP, les résultats sont meilleurs pour la détection de visages vus de face, alors que la décomposition en ondelettes est plus appropriée à une détection des vues de profil ; ceci s'explique en partie par le fait que beaucoup d'informations nécessaires à la détection d'un profil sont contenues dans les contours séparant le visage et l'arrière plan. Ainsi, pour des objets dont les contours externes sont des informations *a priori* importantes, une décomposition en ondelettes sera conseillée. C'est ce qui est aussi constaté dans [66] pour la détection de piétons.

   Dans le cas des véhicules, il est donc préférable d'utiliser aussi une décomposition en ondelettes, à la manière de [158]. Celle-ci est basée sur les ondelettes de Haar, sauf qu'il s'agit de ne conserver que deux échelles, c'est à dire celles correspondant aux masques 32x32 et 16x16. Un recouvrement des masques (cf. annexe C.5.4) est réalisé afin de rendre cette représentation robuste à des variations d'échelle relativement faibles. Ceci aboutit à la constitution de 6 imagettes d'ondelettes verticales, horizontales et diagonales, présentées dans la figure 2.12.

3. Une sélection de paramètres discriminants. Cette dernière étape consiste à ne sélectionner qu'une partie des cœfficients contenus dans la décomposition en ondelettes, afin que le système puisse fonctionner en temps-réel. Ceci est réalisé en choisissant les cœfficients les plus redondants dans la classe objet, c'est-à-dire ceux qui varient le moins. Les imagettes moyennes des ondelettes sont calculées sur l'ensemble de la base d'apprentissage. Les cœfficients sélectionnés sont les plus éloignés de la valeur moyenne (les plus sombres ou les plus clairs dans l'image 2.13). Nous avons limité le nombre de cœfficients à 58 ; c'est une valeur qui réalise empiriquement un bon compromis entre temps d'exécution et discrimination.

   Nous avons pris soin de ne sélectionner que des cœfficients appartenant à la partie basse des véhicules. Ceci pour deux raisons. D'abord, son apparence change peu avec l'orientation des véhicules par rapport à la caméra embarquée. Ensuite, dans les scènes routières, l'arrière-plan au niveau de cette partie contient assez peu d'éléments pouvant être perturbateurs ; il est alors composé essentiellement de la route.

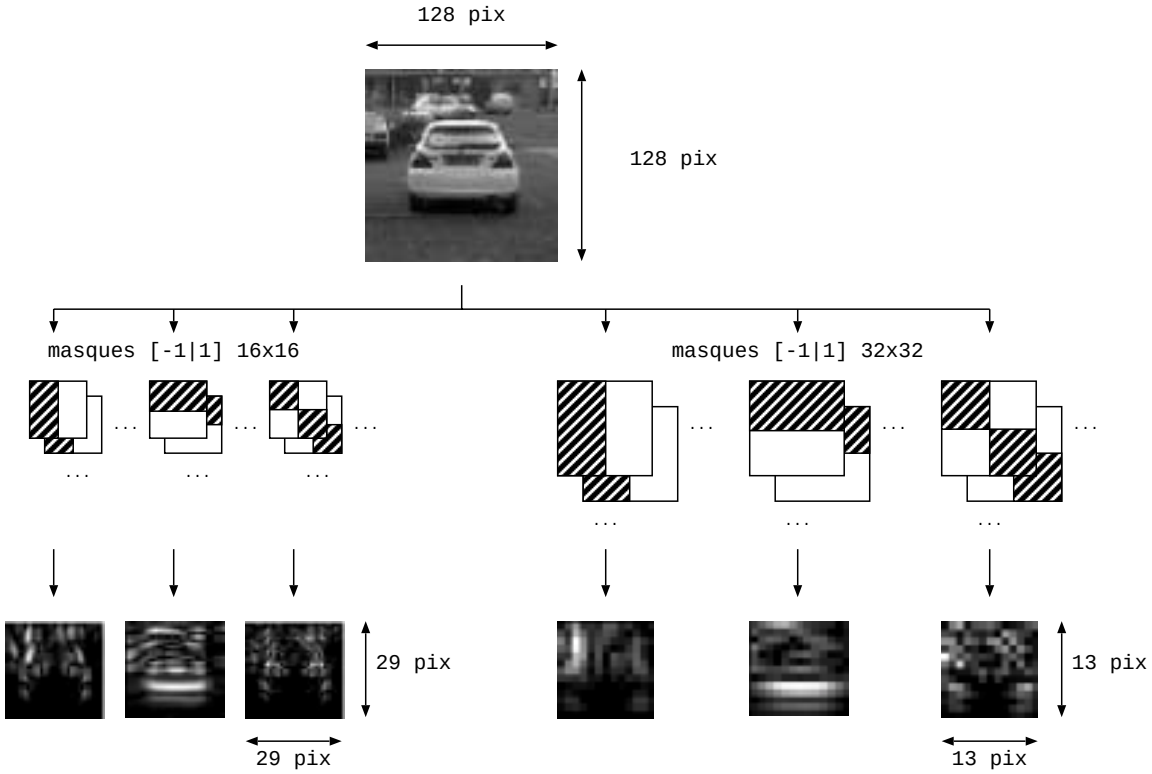


FIG. 2.12 – Décomposition en ondelettes de Haar modifiée d'une imagette contenant un véhicule.

(a) (b)

FIG. 2.13 – Ensemble des valeurs moyennes des cœfficients des ondelettes et cœfficients sélectionnés (encadrés en noir) : (a) image d'un véhicule (b) valeurs moyennes des cœfficents

Dans cette méthode de sélection, il y a un certain *a priori*. Mais nous pouvons voir dans la figure 2.13 qu'elle aboutit à sélectionner des cœfficients dans des zones correspondantes à des zones caractéristiques d'une vue arrière d'un véhicule (zones de luminance uniformes ou à fortes variations). Ceci nous conforte dans notre choix, même si dans l'avenir, il sera intéressant de confronter cette méthode avec d'autres issues de travaux récents (cf. la section 2.4).

Le prétraitement aboutit donc à la constitution d'un vecteur composé des cœfficients sélectionnés sur une décomposition en ondelettes de Haar modifiée de l'imagette 128x128. C'est ce vecteur qui est fourni au classifieur, soit pour l'apprentissage, soit après pour la détection.

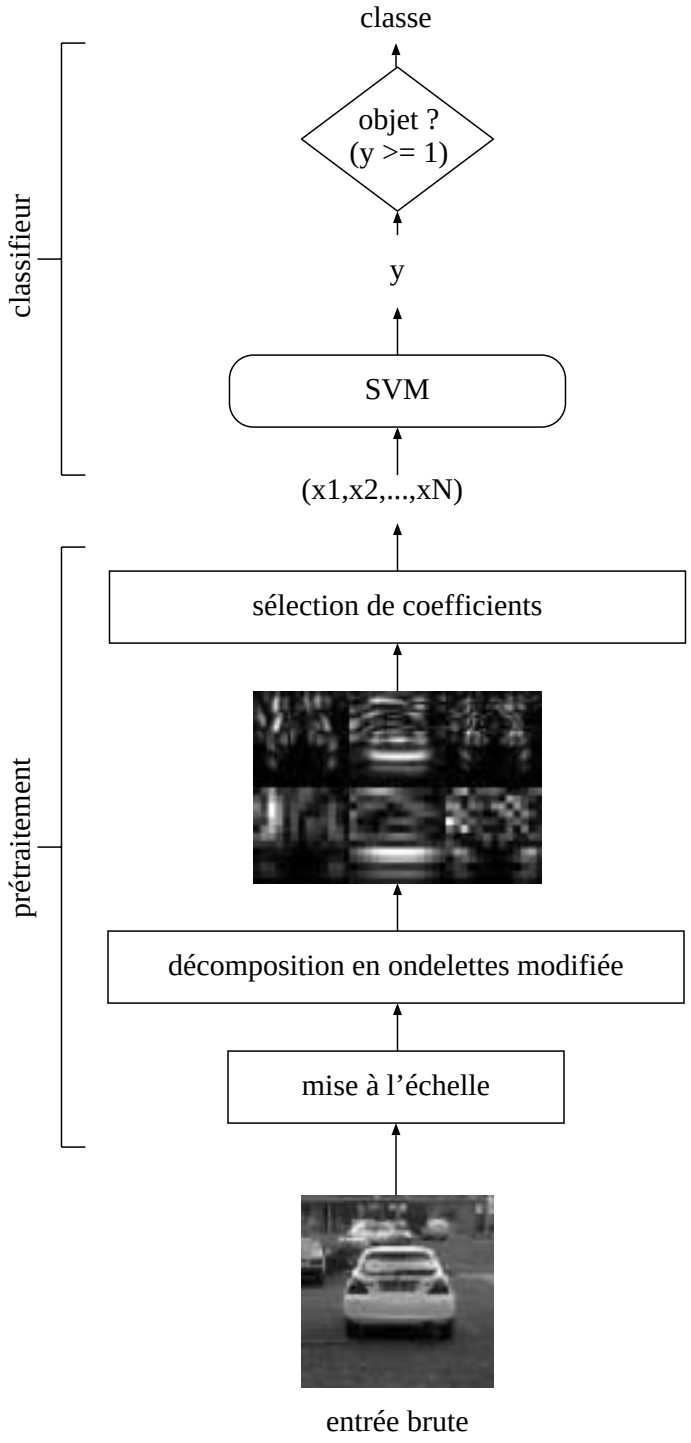


FIG. 2.14 – Synopsis de l'identification de véhicule dans une zone d'intérêt.

Apprentissage et utilisation du classifieur SVM

L'apprentissage du classifieur SVM nécessite de lui fournir des échantillons de la classe objet et de la classe non-objet. Nous avons donc constitué une base d'apprentissage.

FIG. 2.15 – Exemples d'imagettes contenant un véhicule utilisées pour l'apprentissage.

La figure 2.15 donne quelques exemples d'imagettes contenus dans notre base. Celle-ci est composée de plus de 500 images de véhicules, auxquelles il faut ajouter leurs images symétriques ; ce qui nous fait donc plus de 1000 échantillons de véhicules.

La définition de la classe non-objet se fait en ajoutant à la base d'apprentissage des exemples étiquetés non-objet. Ces exemples sont d'abord simplement extraits de manière aléatoire d'images diverses ne contenant aucun véhicule. Cela permet de se définir une base servant à entraîner une première fois le classifieur SVM.

Ensuite, une méthode de *bootstrapping*, mise au point par *Sung et Poggio* [191], est appliquée. Il s'agit d'ajouter des exemples de non-objet à la base ayant été identifiés par le système comme des objets. Elle se déroule en 3 étapes, pouvant être répétées plusieurs fois. La première étape consiste à appliquer le processus de détection sur un ensemble d'images ne contenant pas d'objets. Il peut s'agir de n'importe quelle image. Dans le cadre de notre application, nous avons jugé judicieux de prendre aussi des séquences d'images autoroutières (sans véhicules). Toutes les imagettes identifiées comme des objets sont ensuite ajoutées à la base d'apprentissage, en les reclassant en non-objets. La dernière étape consiste alors à re-entraîner le classifieur avec la base augmentée. L'application de ce procédé permet donc de réduire la marge entre les objets et les non-objets, et donc d'assurer la généralisation du classifieur[2]. Le système sera ainsi plus robuste.

Une fois, le classifieur entraîné, son utilisation est quasiment immédiate pour l'identification d'un véhicule dans une ROI. Comme l'illustre la figure 2.14, celle-ci est obtenue en seuillant la sortie y du classifieur *SVM*. Cette valeur représente la distance normée par rapport au «plan» optimal. Les performances du système en taux de véhicules détectés et en taux de fausses alarmes dépendent du choix de ce seuil. Durant nos expérimentations, nous avons fixé ce seuil à 1. C'est un seuil qui réalise un bon compromis en

[2] voir l'annexe B pour quelques notions sur les processus d'apprentissage.

performance (cf. section 2.3).

2.2.2 Localisation des ROI dans l'image via des primitives simples : *ombres* et *symétrie*

Dans les paragraphes précédents, nous avons explicité le système choisi pour identifier ou non un véhicule dans une ROI. Classiquement, les zones d'intérêt sont extraites des images à analyser par *scanning* à toutes les positions et échelles possibles. Cette méthode est très coûteuse en temps de calcul et limite donc l'application d'approches par apparence à des applications ne nécessitant pas un fonctionnement en temps réel, comme l'indexation d'images.

Dans cette section, nous proposons d'ajouter au préalable un système de prédétection, basé sur des primitives simples et génériques, qui sélectionnera dans l'image un nombre réduits de ROI, celles les plus susceptibles de contenir un véhicule.

Dans la suite, nous décrirons les trois modules que ce système utilise, c'est-à-dire : un module de détection des voies, et de reconstruction de la géométrie 3D de la route, un module de détection de l'ombre et un module de localisation de la symétrie.

Modélisation et détection des voies de circulation

Notre approche nécessite donc la constitution d'un modèle 3D de la route. Celui-ci est obtenu à l'aide d'un algorithme très efficace développé au sein du LASMEA par *Romuald Aufrère et al.* [7] . Cette méthode permet, de par son principe, d'avoir une approche générale, qui s'adapte à tous types de route (marquées et non marquées). Cette méthode permet de reconnaître les bords de la voie de circulation (ou de la chaussée, suivant le contexte) dans l'image fournie par une caméra embarquée dans le véhicule. La reconnaissance est faîte de manière récursive et est guidée par un modèle statistique des bords de la voie dans l'image. Ce modèle intègre les paramètres permettant de localiser le véhicule sur sa voie. Ces paramètres sont remis à jour à l'issue de l'étape de reconnaissance. Le modèle est en fait constitué d'indices représentant l'objet route dans la scène. Quelques résultats de reconnaissance sont présentés sur la figure 2.16.

FIG. 2.16 – Résultats de l'algorithme de reconnaissance de bords de route

A la suite de la reconnaissance, une phase de localisation estime la position et l'orientation du véhicule sur sa voie. L'algorithme est capable de fournir des informations précises sur la position et l'orientation du véhicule mais également sur la largeur et la courbure de la voie.

Cet algorithme présente de nombreuses qualités :

- il est *rapide* : il fonctionne en temps-réel (temps d'éxecution se situant entre 12ms et 56ms (acquisition d'image comprise), selon si le cas est favorable ou non, sur un Pentium III à 800MHz).
- il est *robuste* : il a été testé avec succès en situation réelle de conduite.
- il fournit une *modélisation 3D de la route* et une *localisation du véhicule équipé* (et de sa caméra) par rapport aux voies de circulation.

A partir des informations fournies par cet algorithme et en se confortant à un modèle de route plane (cf. les travaux de *Roland Chapuis* [32, 33]), il est donc possible d'écrire les relations entre un point 3D de coordonnées (X, Y, Z) définies dans un repère «route» glissant (accroché au bord gauche de la route, par rapport au véhicule équipé) et les coordonnées d'un pixel de l'image (supposé appartenant à la route).Les figures 2.17 et 2.18 illustrent le modèle choisi. Les détails de ce modèle et des calculs sont fournis dans l'annexe D.

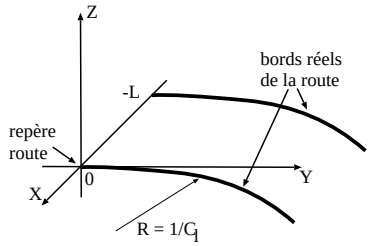


FIG. 2.17 – Représentation des bords de la route.

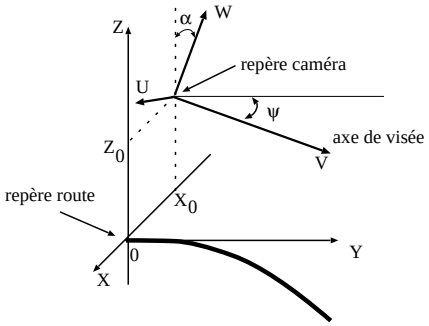


FIG. 2.18 – Passage du repère «route» au repère «caméra».

La projection du point 3D $P_R = (X, Y, 0)^T$ aura donc pour projection dans le plan image le point P_i dont les coordonnées (u, v) sont données par :

$$\begin{aligned} u &= f_u \frac{U}{V} = f_u \frac{X - X_0 - \Psi Y}{Y} \\ v &= f_v \frac{W}{V} = f_v \frac{\alpha Y - Z_0}{Y} \end{aligned} \tag{2.13}$$

avec :

- $f_u = k_u.f$ et $f_v = k_v.f$, f étant la focale de la caméra et k_u et k_v les facteurs d'échelle (pixels/mm) vertical et horizontal.
- Z_0 la hauteur de la caméra.
- X_0 la position latérale du véhicule sur la chaussée.
- Ψ l'angle de cap du véhicule.

– α l'angle d'inclinaison de la caméra.

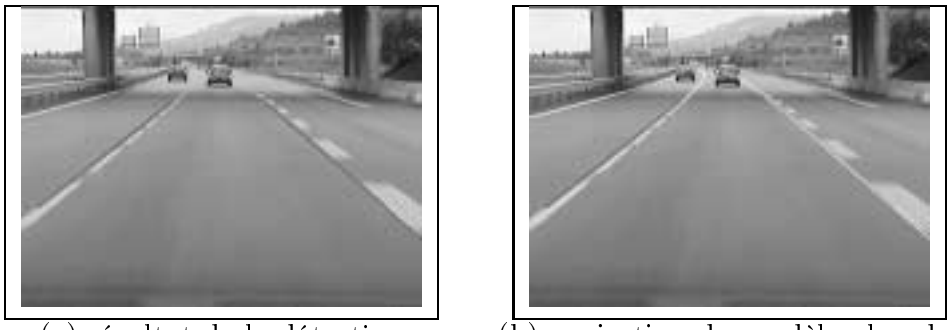
Ces relations permettent dans la suite de reconstruire la géométrie 3D de tout indice détecté sur la route, et ainsi de les confronter à un modèle *a priori*. En particulier, elles sont souvent utilisées dans notre module de recherche des ombres. Par ailleurs, nous pouvons aussi définir approximativement l'échelle et la résolution d'extraction des zones d'intérêt.

De plus, en supposant la courbure de la route localement constante, une relation entre les coordonnées des bords de la route peut être aussi établi :

$$u = f_u \left(-\frac{f_v Z_0}{2(v - f_v\alpha)} C_l + \frac{v - f_v\alpha}{f_v Z_0}(X_0 - \lambda L) - \psi \right) \tag{2.14}$$

avec :
– C_l, la courbure latérale de la chaussée,
– L, la largeur d'une oie de circulation ou de la route,
– λ identifiant les différentes voies ; $\lambda = 0$ pour le bord de la voie à droite du véhicule, $\lambda = -1$ pour le bord gauche,...

Cette relation permet de se définir une zone d'intérêt dans l'image, limitée aux voies de circulation ou à la route, pour la recherche des ombres.



(a) résultat de la détection de voies (courbes noires).

(b) projection du modèle plan des voies dans l'image (courbes blanches).

FIG. 2.19 – Exemple de résultat pour la détection de la route

Recherche des ombres

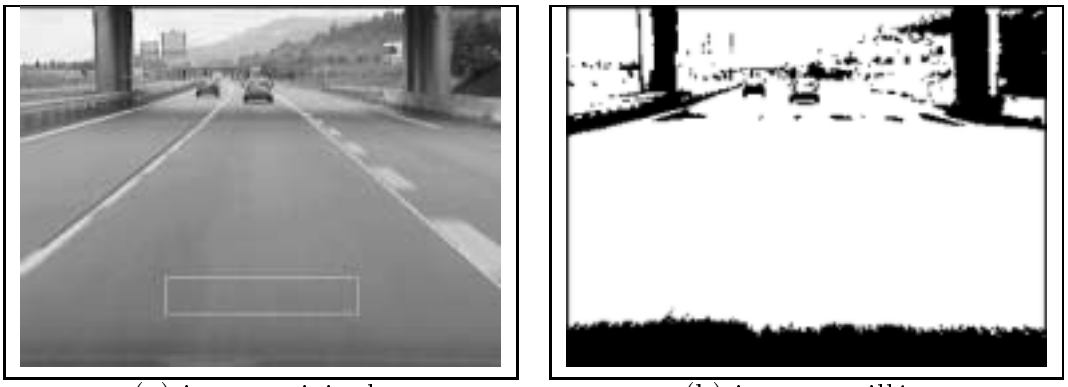
Notre méthode de détection des ombres portées de véhicule sur la route se décompose en plusieurs étapes :
– La première consiste à binariser l'image en niveaux de gris, pour ne retenir que les zones les plus sombres, donc étant potentiellement des ombres. Elle repose sur une modélisation simple de la luminance des pixels de la route, et sur un seuillage de celle-ci.

- Les étapes suivantes sont l'application sur cette image binarisée d'un ensemble de filtres, afin de ne retenir que les zones sombres susceptibles d'être des ombres portées. Ces filtrages sont basés sur un modèle géométrique *a priori* des ombres portées.
- Enfin, une dernière étape consiste simplement à positionner nos zones d'intérêt par rapport à ces ombres potentielles.

Modèle de luminance

Comme nous l'avons vu dans la section 2.1.3, les zones d'ombres potentielles sont définies comme ayant une intensité nettement plus sombre que celle de la route. Leur détection nécessite donc un modèle de luminance des pixels appartenant à la route. Dans la littérature, plusieurs modèles existent en considérant une distribution des niveaux de gris soit gaussienne [144, 199], soit entropique ou tenant compte d'une dérive linéaire selon la position verticale du pixel dans l'image [176]. Cependant la constitution de tels modèles est coûteuse en temps de calcul.

Aussi, nous proposons une approche simple, considérant l'apparence de la route approximativement uniforme en luminance. Au préalable, l'image est lissée par un filtre Gaussien pour assurer au mieux cette hypothèse. L'image binarisée est alors obtenue via un seuillage. Le seuil va correspondre au pixel de plus faible intensité appartenant dans une petite portion placée dans le bas de l'image, c'est-à-dire dans une zone de la route très proche du véhicule équipé où la présence d'un autre véhicule est peu probable. Ce choix permet d'être très réactif aux changements globaux de luminosité de la scène, et reste efficace même en cas de présence d'une ombre parasite (dûe à un élément extérieur à la route, tels qu'un pont ou un arbre) dans la zone choisie. En effet, les ombres portées sont nettement plus sombres que celles d'autres objets, du fait de la proximité des véhicules avec la route.



(a) image originale (b) image seuillée.

FIG. 2.20 – Exemple du résultat d'un seuillage. Dans l'image (a), le rectangle blanc dans le bas de l'image représente le «modèle» de luminance de la route.

Filtrage des ombres

Les filtres appliqués ensuite servent à augmenter la robustesse de notre détection. Le premier utilise la projection du modèle 3D de la route dans l'image, pour éliminer toutes les zones sombres détectées hors de la route. Les autres se servent d'hypothèses géométriques que nous faisons sur la taille d'une ombre portée de véhicule.

Le second filtre analyse la taille des zones (i.e. leur largeur en mètres reconstruite à partir du modèle plan). Cela permet d'éliminer les zones parasites de petites tailles (de l'ordre de quelques centimètres) dues aux irrégularités en luminance de la route (fissures dans le bitume...), et de déterminer les zones de grandes tailles ($> 2m.$).

Ces dernières peuvent être dues à des objets au bord de la route ou à un éclairement de biais sur les véhicules. Elles sont filtrées en opérant un nouveau seuillage adaptatif simple sur chacune d'entre elles, dans l'image de niveaux de gris. Nous avons testé plusieurs méthodes de seuillage adaptatif [173]. Mais celles-ci s'avèrent peu efficaces, en raison du nombre souvent réduit de pixels concernés. Empiriquement, nous avons alors fixé ce seuil comme étant la moyenne des niveaux de gris.

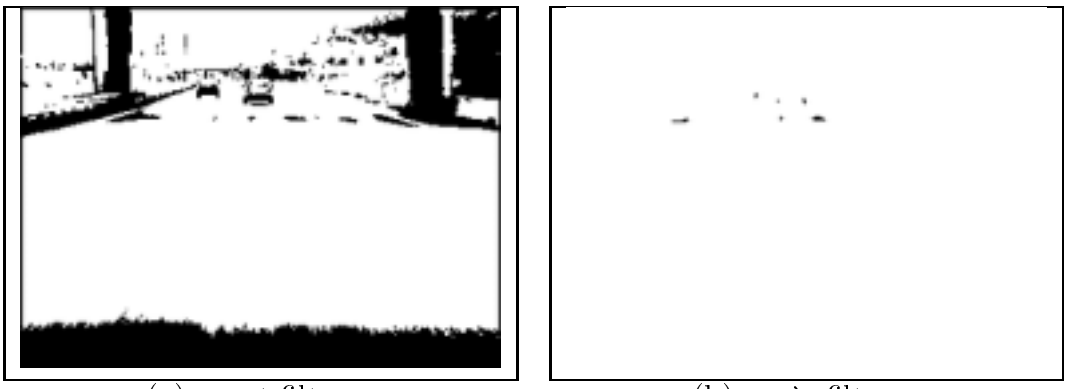
Cette opération permet de ne retenir que leurs parties les plus sombres. Pour un véhicule, ces parties correspondront à la partie d'ombre portée située en dessous du véhicule. En effet, l'intensité d'une ombre dépend de sa distance par rapport à l'objet occultant la lumière. Aussi, nous avons constaté que la partie de l'ombre portée située sous le véhicule est plus sombre que la partie «déportée» (plus éloignée du véhicule). Bien sûr, cette différence est dépendante du gain de la caméra. Or, comme celle-ci est orientée vers la route pour faciliter sa détection, le gain automatique de la caméra dépend essentiellement de la luminance du bitume de la route. Notre hypothèse s'avère donc justifiée.

Enfin, un troisième filtre a été développé pour accroître encore la robustesse de la détection. Après expérimentations, nous obtenions souvent en sortie du processus n'incluant que les deux premiers filtres, des zones d'ombres potentielles réduites aux seuls pixels appartenant aux roues. Ceci est soit du à la dérive assez fréquente de luminance selon la verticale dans l'image, soit issu du deuxième filtre qui n'a retenu que les «roues» plus sombres. Aussi nous avons pensé à un filtre validant les zones sombres distantes horizontalement de $1,5m.$ (qui est l'écart moyen entre les roues sur une voiture de type touristique). Ce filtre permet d'éliminer encore des ombres parasites, et de positionner des ombres artificielles horizontalement au milieu des deux parties «roues» du filtre.

Positionnement et géométrie des zones d'intérêt

Les étapes précédentes aboutissent à la constitution d'une image binaire, dont les zones noires sont des ombres portées potentielles. Celles-ci sont supposées situées sous un véhicule potentiel.

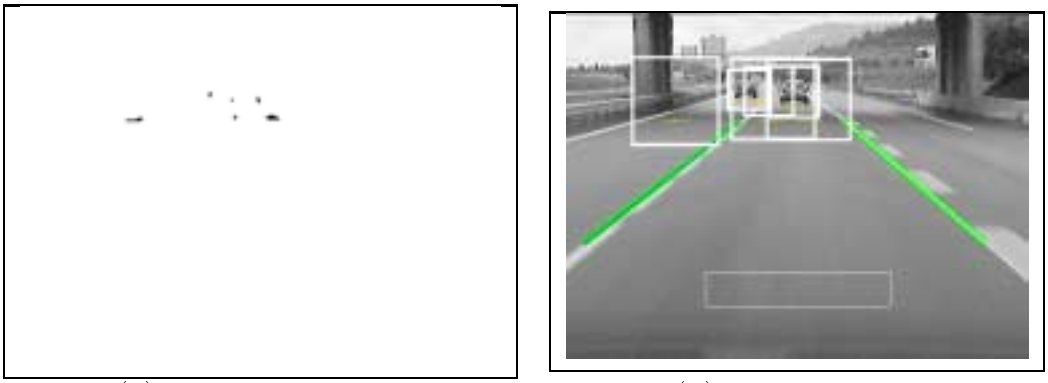
Pour chaque ombre portée potentielle détectée, une ROI carrée est définie. Elle est centrée latéralement sur la moyenne des abscisses des pixels appartenant à l'ombre. Cette



(a) avant filtrage (b) après filtrage

FIG. 2.21 – Exemple du résultat du filtrage : (a) image seuillée, (b) image binaire après filtrage.

position latérale peut s'avérer bien entendu décalée par rapport à celle d'un véhicule situé au dessus de l'ombre. Aussi, nous en avons tenu compte en choisissant une taille verticale importante. La taille géométrique 3D est fixée à $3m$x$3m$.



(a) image binaire des ombres portées potentielles (b) image avec les zones d'intérêt

FIG. 2.22 – Exemple de positionnement de zones d'intérêt, i.e. les carrés blancs dans l'image (c) .

Nous considérons que le pixel le plus bas correspond à la position horizontale de l'interface entre la route et le véhicule. Ceci nous permet d'obtenir une échelle (en m/pixels) pour notre zone d'intérêt. Puis, en se confortant à la résolution des imagettes de la base d'apprentissage, nous pouvons donc extraire les zones d'intérets selon le même modèle.

La détection des ombres permet donc de sélectionner des zones d'intérêts dans l'image, puis d'extraire des imagettes du même format que celles demandées en entrée de notre classifieur. Cependant, si le positionnement horizontal est approximativement correct, ce n'est pas toujours le cas verticalement. Aussi pour centrer notre algorithme sur un véhicule potentiel dans la ROI, nous proposons de calculer l'abscisse de symétrie axiale maximale.

Recherche de la symétrie

Pour calculer la position de l'axe de symétrie maximale dans chaque zone d'intérêt, nous avons utilisé et adapté la méthode de *Zielke et al.*, développée pour le suivi de véhicule [225] et aussi utilisée dans [202].

Cet algorithme permet de mesurer le degré de symétrie $S_L(x_s, w)$ pour chaque point appartenant à une ligne L d'une image I. x_s représente la coordonnée de ce point sur la ligne et w la taille de l'intervalle autour de x_c sur laquelle la symétrie est évaluée. En considérant le fait que toute fonction $G(x)$ (représentant la distribution des niveaux de gris sur une ligne L) peut être décomposée en une somme d'une fonction paire $G_p(x) = \frac{G(x)+G(-x)}{2}$ et d'une fonction impaire $G_i(x) = \frac{G(x)-G(-x)}{2}$, $S(x_s, w)$ est mesuré comme le constraste entre les parties paire et impaire de la distribution, intégrées sur l'intervalle $[x_s - w/2, x_s + w/2]$.

Cette fonction s'obtient en posant la substitution $u = x - x_s$ et en se limitant à l'intervalle donné, on a alors :

$$G_p(x) = E(u, x_s, w) = \begin{cases} \frac{G(x_s+u)+G(x_s-u)}{2} & \text{si } -w/2 \leq u \leq w/2 \\ 0 & \text{sinon} \end{cases} \tag{2.15}$$

$$G_i(x) = O(u, x_s, w) = \begin{cases} \frac{G(x_s+u)-G(x_s-u)}{2} & \text{si } -w/2 \leq u \leq w/2 \\ 0 & \text{sinon} \end{cases} \tag{2.16}$$

Le degré de symétrie pour une ligne L est alors déterminé par :

$$S_L(x_s, w) = \frac{\int_L E_n(u, x_s, w)^2 du - \int_L O(u, x_s, w)^2 du}{\int_L E_n(u, x_s, w)^2 du + \int_L O(u, x_s, w)^2 du} \tag{2.17}$$

où E_n est la fonction normalisée de E, de sorte à avoir une moyenne nulle :

$$E_n(u, x_s, w) = E(u, x_s, w) - \frac{1}{w} \int_{-w/2}^{w/2} E(v, x_s, w) dv \tag{2.18}$$

Ainsi écrite, S_L sera égale à 1 pour une parfaite symétrie, 0 pour une asymétrie et -1 pour une parfaite anti-symétrie.

Dans notre application, nous voulons évaluer la symétrie d'une imagette. Nous proposons donc d'intégrer cette fonction sur plusieurs lignes comprises dans $[y_{min}, y_{max}]$ de cette imagette. La somme obtenue est bien entendu normalisée par le nombre de lignes intégrées. Nous obtenons alors une évaluation $S(x_s, w)$ de la symétrie sur chaque colonne d'une portion d'image. L'abscisse $x_{\text{sym}}(w)$ de l'axe de symétrie maximale sera alors définie par :

$$x_{\text{sym}}(w) = \{x_s / S(x_s, w) = \max_x[S(x, w)]\} \tag{2.19}$$

En fixant w à la taille approximative d'un véhicule (équivalente à $1.8m$) dans l'imagette, nous serions alors capable de déterminer l'axe de symétrie d'un véhicule potentiellement présent dans l'imagette.

Cependant, la détermination de cet axe sert à centrer verticalement l'algorithme de détection par apparence. Or nous avons vu que celui-ci fonctionne avec une représentation en ondelettes de Haar modifiées. Aussi il nous a semblé plus judicieux d'évaluer la position de l'axe de symétrie par rapport à cette représentation : sur une de ses échelles (i.e. celle obtenue avec le masque 16x16), plutôt que dans l'imagette.

Ce choix présente plusieurs autres avantages :

- Tout d'abord, il y a moins d'éléments dans la distribution des ondelettes que de pixels dans l'imagette. Le temps de calcul est donc moindre.
- Les bruits ou les asymétries locales sur les véhicules (dûs à des reflets, de petits éléments non-symétriques sur la carroserie) sont filtrés. Le calcul du degré de symétrie est donc plus robuste.

Taille des ondelettes	29x29
Nombre d'imagettes testées	521
Position moyenne de l'axe de symétrie (en cœfs)	13,8
Positions extrêmes (en cœfs)	[9-19]
Temps de calcul moyen	0,05 ms.
Temps extrêmes (en ms)	[0.045-0,134]

TAB. 2.2 – Résultats de l'estimation de la position de l'axe de symétrie sur la base d'images de véhicules.

Afin de vérifier l'efficacité de cette méthode de recherche de l'axe de symétrie, nous l'avons testé sur l'ensemble des imagettes de notre base de véhicules. Nous obtenons les résultats donnés dans le tableau 2.2. Ces tests ont été réalisés avec un Pentium III cadencé à 800Mhz. Nous pouvons voir que le calcul est extrêmement rapide. En moyenne, les résultats sont corrects. Ils tournent généralement autour de 14 pixels (qui est la position «théorique» de l'axe de symétrie, introduite lors de la création de la base) à 1 ou 2 pixels près. Les dérives plus importantes sont dues essentiellement à une orientation biaisée des véhicules par rapport à la caméra ou à des images de mauvaises qualités (saturation des niveaux de gris). La figure 2.23 donnent quelques exemples extrêmes de ces cas.

2.2.3 Organisation générale du processus de détection

Dans les sections précédentes, nous avons explicité les différents modules utilisés dans notre méthode de détection. La figure 2.24 illustre comment ils sont agencés pour permettre la détection de véhicule dans l'image. Nous pouvons le décomposer en trois parties principales.

La première partie, nommée «détection de la route», permet de modéliser la géométrie de la route. Cette géométrie sert notamment pour la détection des ombres. Elle est

FIG. 2.23 – Exemples d'imagettes où l'estimation de l'axe de symétrie est la plus «biaisée» : la ligne pleine représente l'axe estimé et la ligne en pointillée l'axe de symétrie selon l'initialisation «utilisateur» de l'imagette.

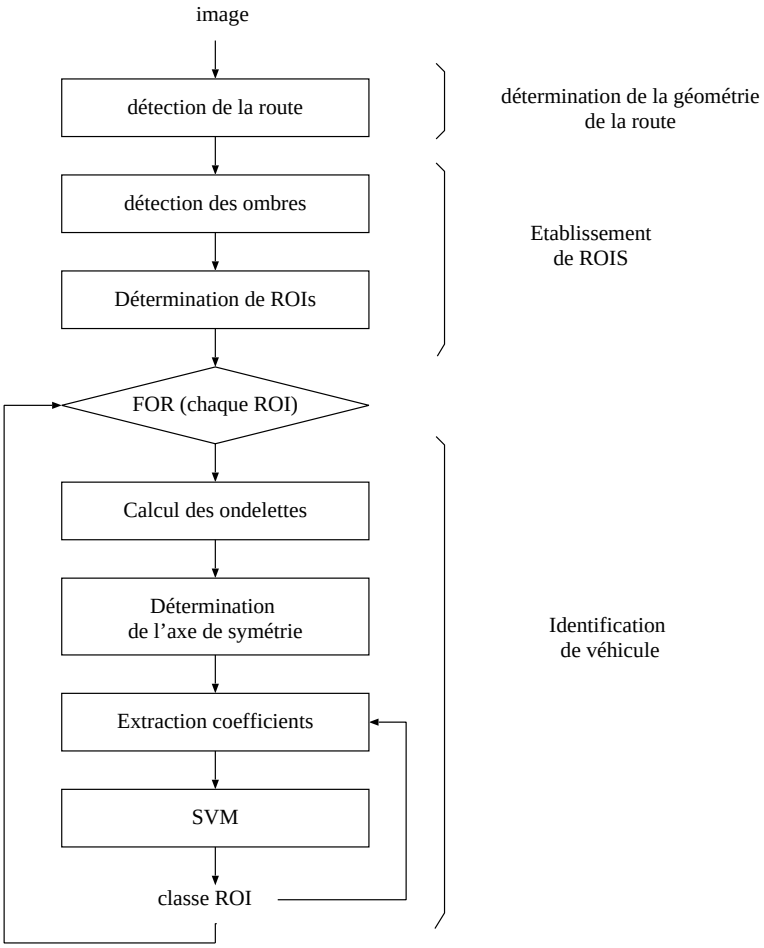


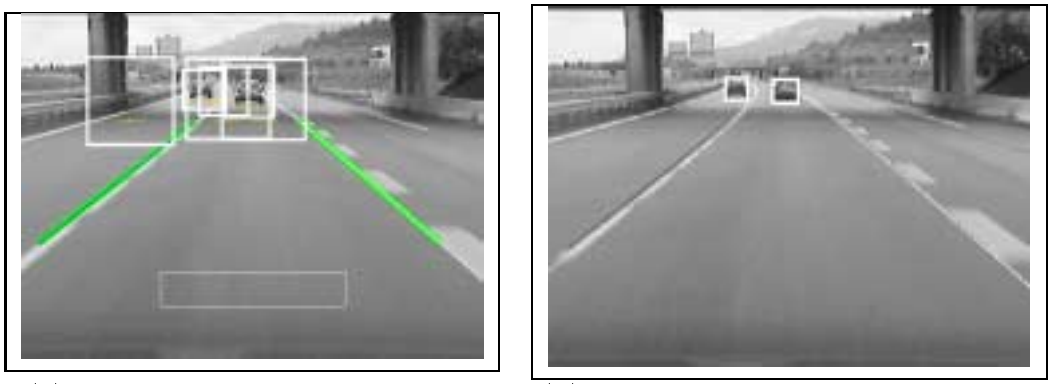
FIG. 2.24 – Synopsis de l'algorithme de détection.

utilisée pour faire correspondre les ombres détectées avec le modèle d'ombres portées sur la route. Elle est aussi nécessaire à l'extraction des ROI. Enfin, elle permet d'estimer la localisation des objets détectés sur la route. Cette localisation est réalisée en considérant la position horizontale de l'ombre détectée et la position de l'axe de symétrie déterminé.

La seconde partie, intitulée «établissement de ROI», se base essentiellement sur la dé-

tection des ombres portées. Elle permet de focaliser la suite de l'algorithme sur certaines parties de l'image.

La troisième partie s'exécute uniquement sur les ROI déterminées dans l'image. Elle est réalisée de manière hiérarchique suivant la distance estimée de celles-ci par rapport au véhicule équipé, c'est-à-dire que les ROIs sont traitées en commençant par celles qui sont les plus proches. En cas de détection d'un véhicule dans une ROI, nous ne traitons pas les autres ROIs détectées dans la même zone de l'image. Cette organisation permet donc d'éliminer certaines fausses détections d'ombres portées, souvent dues à des véhicules sombres. Nous pouvons aussi remarquer que la détermination de la symétrie est réalisée après l'extraction de l'imagette correspondant à la ROI et le calcul des ondelettes associées, pour les raisons que nous avons signalées dans la section 2.2.2. Enfin, pour chaque imagette, la détection selon l'apparence est exécutée autour de l'axe de symétrie détecté. Pour augmenter la robustesse de notre méthodologie, les cœfficients nécessaires ne sont pas extraits à une seule position dans l'imagette. En effet, nous considérons le fait que des erreurs sont plausibles dans la détermination de cette position. Aussi, nous effectuons un parcours en spirale, en nous éloignant de cette position, en relevant et testant à chaque position un jeu de cœfficients. Si un jeu est classé comme étant un véhicule, ce parcours est arrêté. Au maximum, 25 positions seront testées pour une imagette, si aucun véhicule n'est détecté.



(a) image avec les zones d'intérêt (b) image avec les véhicules détectés

FIG. 2.25 – Exemple de résultats de la classification des ROIs.

Dans la suite, nous allons donner quelques autres résultats caractérisant les performances de ce système.

2.3 Résultats obtenus

Nous avons testé notre méthode sur plusieurs séquences d'images acquises avec le véhicule VELAC sur des routes de type autoroutières à deux voies de circulation. Elles représentent un total avoisinant 30000 images de taille 640x480 pix. Elles ont été prises dans des conditions atmophériques favorables à un capteur de vision (de jour, ni pluie, ni

brouillard). La caméra utilisée est placée à une hauteur de $1.5m$, au niveau du rétroviseur intérieur. Elle est inclinée vers le bas d'environ 6°. La focale lors de ces expérimentations est de l'ordre de 2000 pixels (environ $10mm$). L'angle d'ouverture est de 40°. Ces caractéristiques ont été choisies afin de faciliter la détection de la route, tout en offrant une vision assez globale du reste de la scène.

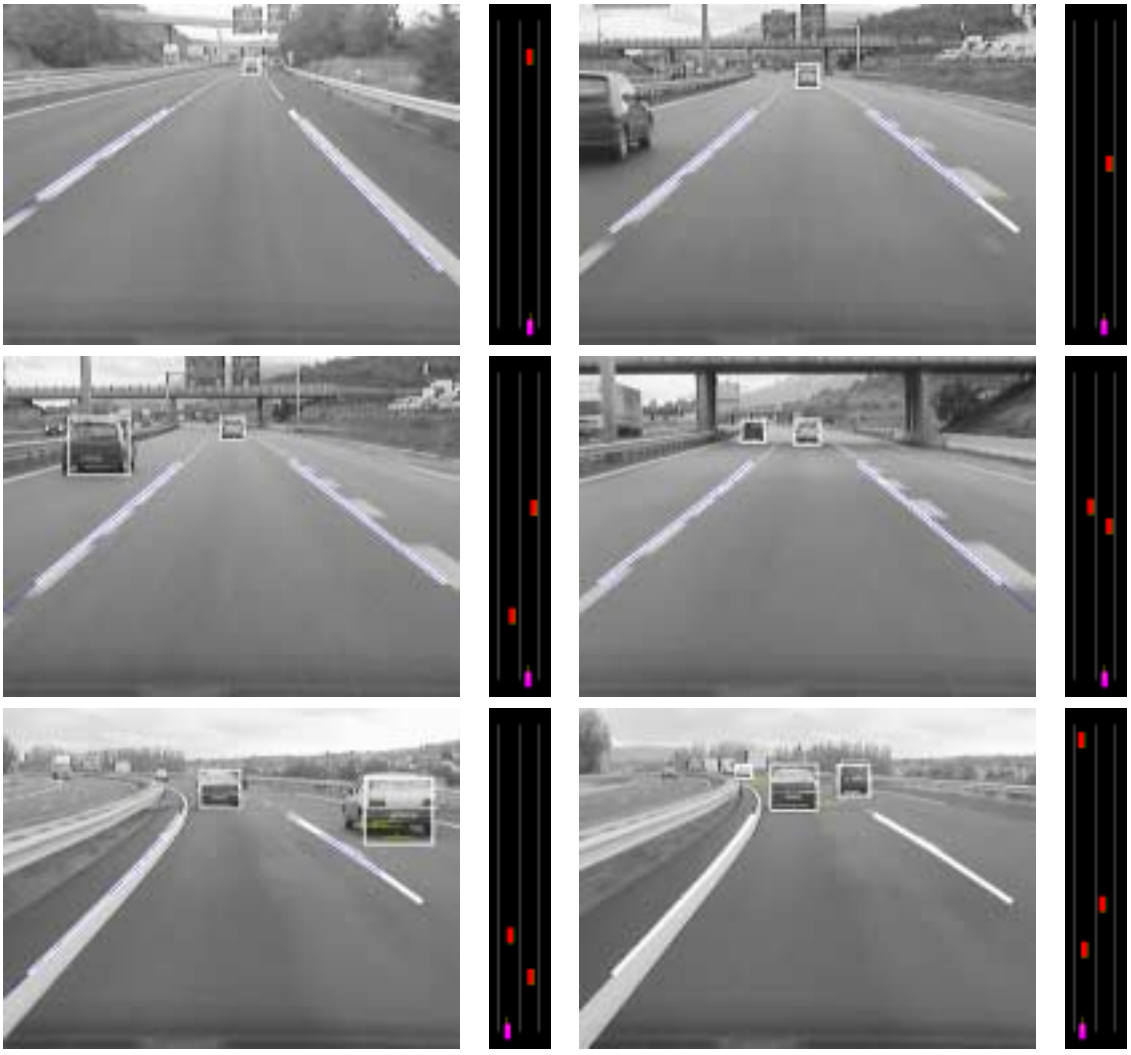


FIG. 2.26 – Exemples de détection de véhicule. A gauche, les rectangles blancs représentent les véhicules détectés. A droite, les véhicules détectés et le véhicule VELAC (en bas) sont localisés par rapport aux voies de circulation (entre 0 et $100m$).

La figure 2.26 présente quelques résultats obtenus sur ces séquences. Les expérimentations réalisées ont permis de vérifier la robustesse de notre approche : tous les véhicules dits de tourisme, présents à une distance comprise entre 20 et $100m$, sont détectés avec un taux de réussite de l'ordre de 0.8. Le taux de fausses alarmes est de l'ordre de 1%.

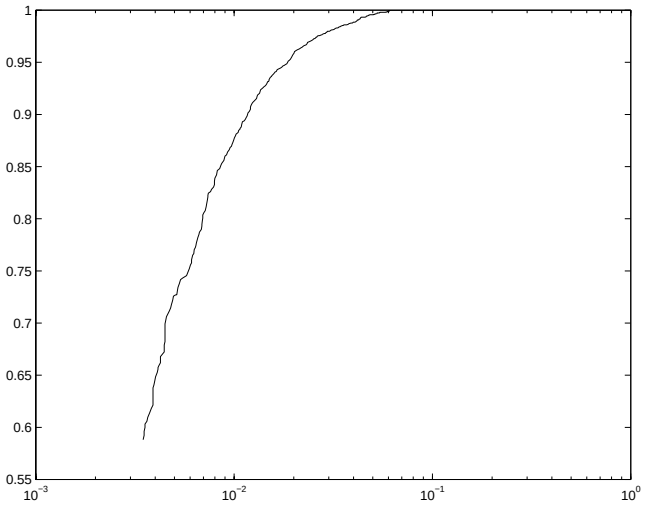


FIG. 2.27 – Caractérisation du processus d'identification de véhicule via une courbe COR (présentant le taux de bonnes détections en fonction du taux de fausses alarmes).

Afin de caractériser le processus d'identification, nous avons calculé une courbe COR ou caractéristique opérationnelle du récepteur. Elle est paramétrée par le seuil évoqué dans la section 2.2.1. Elle présente le pourcentage de bonnes détections en fonction du pourcentage de fausses alarmes. Avec un seuil fixé à 1, nous observons que le pourcentage de fausses alarmes est de l'orde de 1% et celui de bonnes détections, i.e. le nombre d'imagettes contenant un véhicule ayant été bien classées sur le nombre d'imagettes extraites, est de l'ordre de 0.88. Nous avons conservé cette valeur pour la suite.

Les courbes 2.28 et 2.29 présentent les différents taux en fonction de la distance estimée. Nous rappelons que cette distance est calculée suivant la localisation des ombres portées sur la route et le modèle de route plane. Pour réaliser ces courbes, nous avons sélectionné toutes les détections réalisées autour de plusieurs hauteurs dans l'image. Puis pour chacune de ses hauteurs, nous avons calculé la distance estimée équivalente.

La figure 2.28 présente l'évolution du taux de fausses alarmes en fonction de la distance. Nous pouvons la décomposer en trois parties.

- En dessous de $15m$, aucun véhicule n'est détecté. Nos séquences ont été prises en effet sur des voies autoroutières. Pour des raisons de sécurité évidentes, aucun véhicule ne s'est trouvé à cette distance sur la même voie de circulation. Quant aux véhicules en situation de dépassement (sur les voies latérales), nous sommes ici limités par l'angle de vue de la caméra.
- De 15 à $20m$, le taux de fausses alarmes est assez important, avec une pointe à 13%. Elles sont déclenchées par les véhicules en situation de dépassement. Ceux-ci se présentent de biais. Dans ce cas, la détection de symétrie est souvent erronée. Ce qui influe sur la position horizontale de la zone d'extraction de cœfficients. Or les ondelettes horizontales sont prédominantes en nombre dans notre sélection et

sont peu dépendantes de la position horizontale (tant qu'elles sont au niveau du véhicule). C'est donc probablement une source importante d'erreur pour le classifieur SVM. Pour y remédier, il faudrait soit améliorer la détection de la symétrie, soit ré-entraîner le classifieur SVM pour tenir compte de ces cas.

– Puis, entre 20 et $200m$, il croît régulièrement de moins de 0.01 à plus de 0.02. Cette évolution est sans doute due à la diminution de la résolution des imagettes extraites de l'image selon la distance.

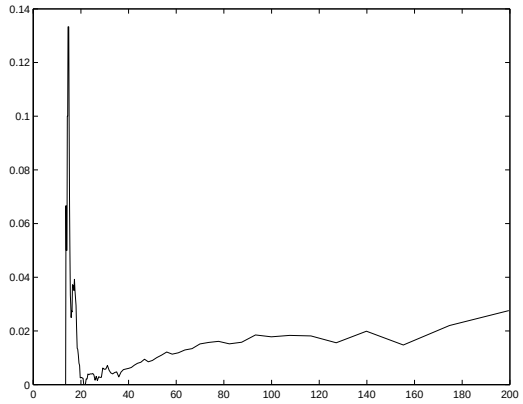


FIG. 2.28 – Taux de fausses alarmes en fonction de la distance par rapport au véhicule embarqué.

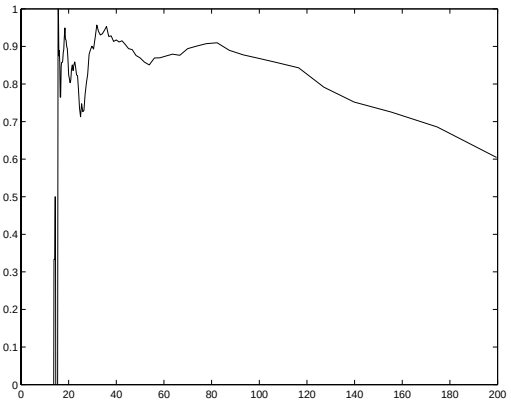


FIG. 2.29 – Taux de bonnes détections en fonction de la distance par rapport au véhicule embarqué.

La figure 2.29 présente l'évolution du taux de bonnes détections en fonction de la distance. Nous pouvons faire des observations assez semblables. Nous pouvons voir notamment que ce taux diminue nettement lorsque la distance estimée est supérieure à $100m$. Les véhicules présents à ces distances apparaissent dans l'image à des résolutions trop faibles pour être identifiés correctement par notre système.

Caractérisons à présent notre algorithme en terme de temps de calcul. C'est en effet un paramètre important pour une application de type véhicule intelligent. Et comme nous l'avons vu dans les sections précédentes, il a été un des critères les plus déterminants dans les différents choix que nous avons émis pour le développement de notre approche.

	Temps par image (en ms)			Temps par ROI (en ms)		
	Min.	Moyen	Max.	Min.	Moyen	Max.
Détection de la route	12	15	56	12	15	56
Etablissement des ROIs	40	53	90	40	43	90
Nombre de ROIs	0	1.7	9	0	1	1
Identification de véhicules	0	17	160	0	10	50
Total	52	87	310	52	58	196

TAB. 2.3 – Temps de calcul sur l'architecture de type PC.

Le tableau 2.3 présente les temps de calcul obtenus sur une architecture de type

PC (Pentium III cadencé à 800 Mhz). Le temps d'exécution pour une image dépend essentiellement du nombre de ROIs à traiter. Les temps obtenus permettent d'envisager une exploitation en temps réel, d'autant plus qu'ici nous n'avons travaillé que sur des scènes statiques. L'intégration d'un module de suivi des cibles détectées dans les images précédentes permettrait de diminuer le nombre de ROIs par image. Tout véhicule étant suivi élimine de ce fait dans l'image, une zone à traiter. Seules les autres ROIs sont à traiter.

2.4 Conclusions

Ce chapitre a présenté une méthode hybride permettant la détection de véhicules dans des scènes routières et autoroutières, via une caméra achromatique embarquée. Une prédétection via des primitives simples, ombres portées et symétrie axiale, focalise une identification selon une approche par apparence. Ce processus permet de répondre aux deux qualités principales requises par un système de détection : rapidité et robustesse. Par ailleurs, celle-ci a été conçue dans un esprit de généricité. En effet, bien que testée uniquement pour les véhicules de tourisme, rien ne semble empêcher son portage pour la détection de tout autre type de véhicule. Il suffirait de considérer plusieurs modèles d'ombres portées (différenciées par leur largeur dans l'image) et d'apprendre plusieurs classifieurs SVM, un par classe de véhicules.

Les résultats obtenus sur un grand nombre d'images successives sont globalement satisfaisants. La méthode développée réponds aux exigences que nous nous sommes fixées. Tous les véhicules sont détectés et localisés par rapport aux voies de circulation. Cette localisation est approximative, mais elle semble suffisante pour une application dédiée à l'information du conducteur.

Dans l'avenir, nous pensons améliorer la robustesse de la méthode proposée.

L'algorithme de détection des véhicules est par exemple très dépendant de la reconnaissance des voies de circulation. Or celle-ci peut être perturbée par la présence même de ces véhicules. Une coopération plus poussée entre ces deux algorithmes permettrait d'améliorer leur fonctionnement.

D'autre part, des développements récents dans le domaine de la classification par apparence [205, 208] permettraient d'améliorer encore le processus d'identification. Le premier [205] propose un processus en cascade de détection de visage, permettant d'affiner à fur et à mesure l'identification des zones testées en intégrant de plus en plus d'éléments. Le second [208] proposé par *V. N. Vapnik* décrit une méthode rigoureuse pour sélectionner les paramètres les plus judicieux pour l'apprentissage d'un classifieur SVM. La combinaison de ces deux méthodes permettraient sans doute d'accélérer la phase d'identification des véhicules. Les zones d'intérêt pourraient éventuellement être étendues à toute la route : la détection des ombres portées et de la symétrie fixeraient alors simplement un ordre de priorité dans le traitement de ces zones.

Enfin, afin d'améliorer la localisation, il est nécessaire d'adjoindre au système un mo-

dule de suivi. Nous proposons dans le chapitre suivant, une méthode permettant cette fonction.

Chapitre 3

Suivi de véhicule par vision monoculaire

Dans ce chapitre, nous allons présenter une méthode de suivi d'objets par vision monoculaire. Celle-ci sera appliquée au suivi de véhicule obstacle. Ce suivi est destiné à la localisation de l'obstacle et à la détermination de ses vitesses, notamment de la vitesse longitudinale relative. La connaissance de ces informations est essentielle à un système d'aide à la conduite. C'est en fonction de celles-ci que le système décidera s'il doit ou non avertir le conducteur.

Ce chapitre est organisé en trois sections. La première dresse un brief état de l'art du suivi d'objets par vision monoculaire. Il permet de positionner l'approche choisie par rapport aux différentes techniques développées récemment pour le suivi de véhicules. La seconde section expose en détails la méthode choisie. La troisième présente quelques uns des résultats obtenus pour le suivi de véhicule dans des séquences d'images. L'extraction des caractéristiques cinématiques via un filtre de Kalman permet de caractériser cette méthode par rapport à l'application.

3.1 Etat de l'art : suivi de véhicules

Dans cette section, nous allons d'abord formaliser le problème du suivi d'objets par vision. Puis nous réaliserons un brief panorama du suivi de véhicules par vision monoculaire. Nous nous sommes particulièrement intéressé à celles ayant application dans l'aide à la conduite.

3.1.1 Suivi d'objet

Le suivi en vision par ordinateur a pour objectif d'estimer l'évolution de l'état d'un objet suivant le temps. Cet état peut, par exemple, être composé de ses caractéristiques cinématiques. Son estimation est réalisée à partir de mesures (bruitées) sur une séquence d'image. Suivre un objet dans l'image revient donc à optimiser l'estimation du mouve-

ment pour qu'elle corresponde au mieux aux mesures réalisées dans l'image.

Elle nécessite la mise en place de deux modèles. D'abord un *modèle de mouvement* qui décrit l'évolution de l'état dans le temps. Ensuite, un *modèle de mesures* qui relie les mesures dans les images avec l'état.

Algorithmie du suivi

D'un point de vue algorithmique, le processus permettant cette estimation est cyclique. Il peut se décomposer en plusieurs étapes successives : la prédiction, la mesure, l'observation, la validation et enfin l'estimation.

1. La *prédiction* calcule la position la plus probable de la cible dans l'image courante. Cette étape fait appel notamment à la connaissance des états dans les images précédentes ou d'un état initial déterminé via un processus de détection. Dans le premier cas, la prédiction est réalisée selon le modèle de mouvement choisi, ainsi que d'un modèle d'incertitudes.

2. La *mesure* consiste à estimer une ou plusieurs propriétés dans les images de la séquence, à la position prédite (ou autour). Ces propriétés sont propres à la représentation ou signature(s) visuelle(s) de l'objet choisie dans les méthodes de suivi développées.

3. L'*observation* consiste à déterminer la position de l'objet à partir de la mesure réalisée précédemment. Cette étape nécessite donc la définition d'un modèle de mesures ; il s'agit de déterminer la position optimisant un ou plusieurs critères de mesures. A noter que, dans certains travaux comme [51], les deux étapes «mesure» et «observation» sont regroupées en une seule, nommée alors uniquement observation.

4. La *validation* examine la validité de la position estimée de l'objet. Ce processus peut utiliser des mesures dans l'image, la position prédite précédemment ou des connaissances externes liées à l'application visée [1].

5. L'*estimation* conclue ce cycle en fournissant une estimation de l'état de l'objet, ainsi que les incertitudes éventuelles sur celui-ci. Cette mise à jour tient compte de

[1] Pour illustrer ce propos, prenons l'exemple du suivi de véhicule : certaines positions, manœuvres ou mouvements sont impossibles ou peu probables ; une observation supposant un tel cas peut être alors considérée comme erronée. Par exemple, dans [142], un système de suivi de véhicules coopératifs, munis de trois amers visuels (cf. figure 1.15), est proposé. Dans ce système, l'estimation de la pose des véhicules entraîne deux solutions, dont une considère le véhicule retourné sur le toît. Cette observation est bien entendue considérée comme abérrante.

ou des observations réalisées (si elles sont validées) réalisées dans les cycles précédents et du modèle de mouvement choisi. Dans certains systèmes [26], l'observation tient lieu d'estimation, la position estimée correspondant à la position observée et validée. Sinon, deux techniques sont principalement employées dans la littérature : les *filtres de Kalman* ou *à particules*.

Dans les paragraphes qui suivent, nous allons revenir sur quelques notions qui se sont dégagées de cette revue algorithmique : les modèles de mouvement, les techniques d'estimation et enfin les modèles de mesure. Cela revient à décrire en premier lieu l'ensemble de variables (en fait les paramètres du modèle de mouvement) que l'on veut estimer, puis les outils utilisés pour cette estimation et enfin le type de mesures sur lesquelles ces outils sont appliqués.

Modèles de mouvement

Le choix du *modèle de mouvement* dans les processus de suivi est crucial. D'une manière générale, la précision de l'estimation du champ de mouvement dépends du nombre de paramètres. Cependant, un nombre excessif peut entraîner des instabilités numériques [188], un coût en temps de calcul élevé ou une trop grande sensibilité aux bruits [24].

On peut distinguer deux approches.

Dans la première approche, le modèle est exprimé sous la forme d'une **transformation dans l'espace 3D**. Le processus de suivi incluera alors un calcul de pose 3D de l'objet d'après les mesures effectuées dans l'image. L'estimation de la transformation sera réalisée selon ses paramètres.

Pour un objet rigide, il y a six paramètres à estimer : 3 paramètres pour la translation et 3 pour la rotation de l'objet. La transformation est souvent donnée par rapport à un repère de référence fixe (ou lié au véhicule équipé). Pour illustrer ceci, considérons la figure 3.1 qui représente un repère R_{obj} lié à l'objet par rapport à un repère R_{ref}.

Soit un point $\mathbf{P}_{obj}$ de l'objet de coordonnées $(X_{ref}, Y_{ref}, Z_{ref})$ dans le repère de référence. La transformation, induite par le mouvement de l'objet, telle que $\mathbf{P}_{obj} \rightarrow \mathbf{P}'_{obj}$, peut s'exprimer sous la forme de la relation matricielle suivante (dans le cas d'un objet rigide) :

$$\begin{pmatrix} X'_{ref} \\ Y'_{ref} \\ Z'_{ref} \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{ref} \\ Y_{ref} \\ Z_{ref} \\ 1 \end{pmatrix} \tag{3.1}$$

Les éléments $r_{ij}\|_{j=1..3}^{i=1..3}$ paramètrent les rotations et les $t_i\|^{i=1..3}$ les translations. Par exemple, dans une représentation Eulérienne, où (α, β, γ) sont les angles et (T_x, T_y, T_z) les translations suivant les axes, on a :

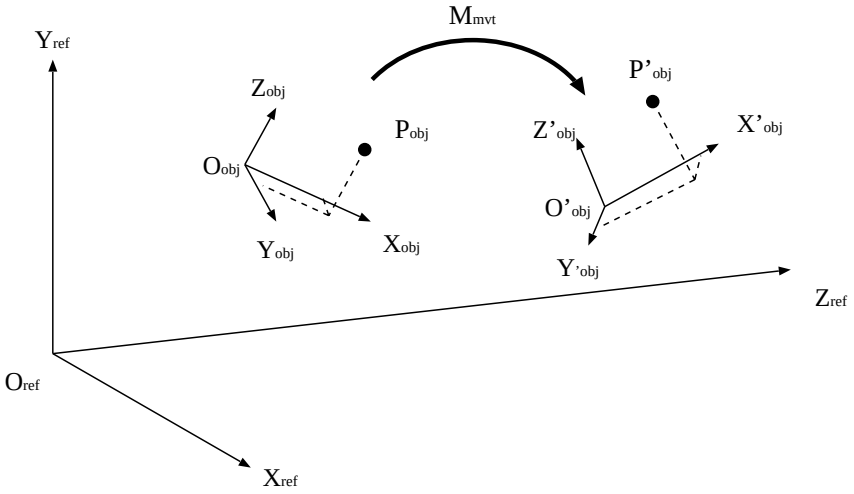


FIG. 3.1 – Transformation 3D d'un objet.

$$\mathbf{M}_{mvt} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbf{M}_{rot}\mathbf{M}_{transl} \tag{3.2}$$

avec :

$$\mathbf{M}_{rot} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & \cos\gamma & \sin\gamma & 0 \\ 0 & -\sin\gamma & \cos\gamma & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

$$\mathbf{M}_{transl} = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

Le modèle géométrique est ensuite reprojeté dans les images, selon le modèle de la caméra choisi (orthographique, projectif,...), pour déterminer la position des mesures image à effectuer lors de l'observation.

L'autre approche consiste à modéliser le **mouvement 2D apparent** de l'objet. Les paramètres estimés seront donc des paramètres décrivant la trajectoire 2D d'un motif (lié à l'objet). A noter que cette approche n'empêche pas de déterminer la transformation 3D via un calcul de pose, si un modèle géométrique 3D de l'objet est connu. Mais contrairement à l'approche précédente, ce calcul n'intervient pas à proprement parlé dans le processus de suivi.

Dans un modèle de mouvement 2D, les paramètres sont souvent regroupés dans un *vecteur de paramètres* noté μ. Ce vecteur décrit la transformation $\mathbf{p} \rightarrow \mathbf{p'}$ d'un point $\mathbf{p}$ de coordonnées (x, y) dans un repère lié au motif, en un point $\mathbf{p'}$ de coordonnées (x', y'). Dans la littérature [13, 190], plusieurs modèles paramétriques sont présentés : des modèles linéaires ou non-linéaires. Les modèles linéaires de déplacement sont les plus couramment utilisés. Ils peuvent s'écrire sous une forme d'une matrice homogène paramètrée $\mathbf{F}(\mu)$, telle que :

$$\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix} = \mathbf{F}(\mu) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.5}$$

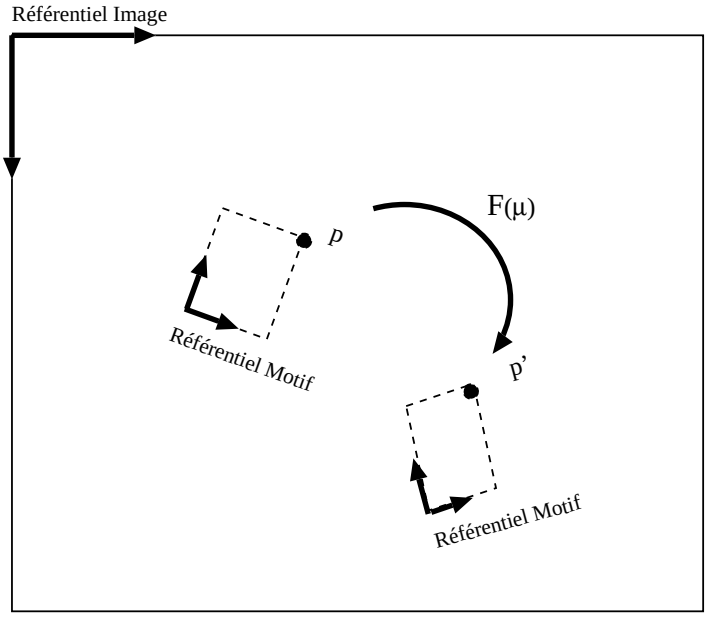


FIG. 3.2 – Transformation 2D d'un objet.

De manière hiérarchique, on peut définir :

– La *translation* qui se paramétre avec deux paramètres (t_x, t_y) tels que :

$$\mathbf{F}(\mu) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

– La *«similarité»* (translation, rotation planaire et changement d'échelle) se paramètrant avec (t_x, t_y) pour la translation, θ pour la rotation planaire et ρ le changement d'échelle.

$$\mathbf{F}(\mu) = \begin{bmatrix} \rho cos(\theta) & \rho sin(\theta) & t_x \\ -\rho sin(\theta) & \rho cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

– La transformation *affine* intègre six paramètres :

$$\mathbf{F}(\mu) = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

– L'*homographie* est un modèle considérant 8 paramètres : $\mu = (a, b, c, d, e, f, g, h)$.

$$\mathbf{F}(\mu) = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \tag{3.9}$$

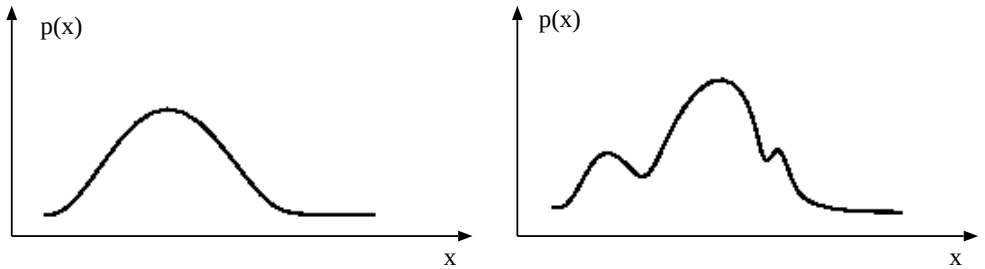
Les transformations telles que la translation, la «similarité» et l'affinité considèrent un modèle orthographique de la caméra ; l'homographie considère elle un modèle projectif.

Filtrage ou estimation de l'état

Pour chaque image, les paramètres du mouvement (et donc l'état de l'objet) sont donc calculés à partir de mesures dans l'image. Ces mesures sont naturellement bruitées. Aussi l'évolution de l'état obtenue par la seule observation de la position de l'objet est elle-même bruitée.

Dans certaines applications [26], ce bruit n'est pas génant. Il s'agit souvent d'applications requérant l'estimation d'un état limitée à la position d'un motif dans l'image et ne comprenant pas ses caractéristiques dynamiques. Pour des applications nécessitant ces dernières, il est indispensable d'opérer à un filtrage. Il doit permettre de connaître les états estimés ainsi que les incertitudes sur cette estimation. Dans une formulation bayésienne [6], il s'agit de déterminer une densité de probabilité. Nous pouvons citer deux techniques employées dans les systèmes de suivi.

La technique la plus utilisée est le *filtrage de Kalman* [111]. Il considère la fonction de probabilité comme une Gaussienne. L'estimation de l'état sera alors souvent donnée via un vecteur moyen et sa matrice de covariance associée.



(a) filtre de Kalman (b) filtre à particules

FIG. 3.3 – Exemples de représentations de la densité de probabilté [103].

Pour des problèmes de suivi non-linéaires et non-Gaussiens, il existe des techniques basées sur des algorithmes Bayésiens, nommés *filtres à particules* [6, 103]. Ils reposent sur une caractérisation de la fonction de probabilité selon la méthode de Monte-Carlo, c'est-à-dire de la représenter par un très grand nombre d'échantillons tirés au hasard et pondérés. Les estimés sont alors calculés à partir de ces échantillons et de leurs poids associés. Ces filtres permettent d'avoir des densités de probabilité multi-modales et non-Gaussiennes (cf. figure 3.3).

Ces différentes techniques permettent donc un filtrage de l'état et de ses dynamiques. Et aussi de déterminer (une ou) des zones de mesures et d'observation dont (la ou) les positions et les dimensions sont déterminées selon l'état prédit ainsi que l'incertitude associée[2]. Ceci nous amène à évoquer les *modèles de mesures*, liés à la représentation de l'objet ; via des caractéristiques locales ou un modèle global du motif (ou de l'objet) à suivre.

Modèles de mesures

En effet, il faut distinguer deux types d'approches pour les modèles de mesures dans la littérature.

La première est basée sur une représentation de l'objet par des *primitives visuelles locales*, telles que des points [90], des segments [106], des contours [102] ou des régions (de petites tailles).

Ces primitives sont suivies indépendamment d'image en image. Les outils employés pour leur mise en correspondance sont nombreux. Ils reposent sur une optimisation d'une fonction de coût (ou d'énergie) au niveau de chaque primitive. Il s'agit souvent de techniques d'analyse d'image simples et rapides, que l'on peut qualifier de bas niveau.

Bhanu et Burger [21] utilisent par exemple l'équation de flot optique pour identifier et suivre des points d'intérêts. Le choix du plus proche voisin est aussi une technique couramment utilisé. Ce choix est alors validé par des mesures de corrélation ou par cohérence temporelle des trajectoires [195].

La position de l'objet est ensuite déterminée par appariement de ces primitives locales, dans l'image ou dans l'espace 3D [133]. Dans le cas d'un suivi uniquement dans l'image, des modèles approximatifs de l'objet et de mouvement 2D sont utilisés ; il est alors souvent nécessaire d'introduire une composante déformable pour s'adapter aux déformations non-linéaires des projections de l'objet (dues notamment aux effets de perspectives). Pour le suivi 3D, un modèle géométrique 3D de l'objet est nécessaire. Ce qui limite la méthode aux objets connus *a priori*.

Ces approches s'avèrent souvent robustes aux occultations partielles. En effet, les appariements réalisés prennent souvent en compte la possibilité d'avoir des correspondances manquantes ou erronées. Seules celles présentant un fort taux de confiance (par

[2] Dans le cas où ces filtres sont appliqués sur des paramètres 3D, une projection de cet état est bien sûr requise.

rapport aux mesures de corrélation ou au modèle de mouvement [92, 93]) sont alors utilisées.

Les autres approches considèrent l'objet (ou le motif) dans sa totalité, via un *modèle global* de celui-ci. Elles sont notamment utilisées pour le suivi des motifs complexes où l'extraction et la mise en correspondance de primitives locales est difficile. Ces méthodes ont ainsi souvent l'avantage d'être plus génériques que celles exploitant des primitives locales.

Elles s'appuient sur la mesure de la similarité entre un motif de référence (souvent extrait de la première image de la séquence où il est détecté) et la partie observée de l'image. La position de l'objet est déduite par optimisation de ce critère global.

Il existe différents critères de similarité. Le plus cité est la somme des carrés des différences (SSD ou *Sum of Squared Difference*). Plus récemment, certains auteurs proposent des critères basés sur une fonction d'énergie calculée à partir d'une décomposition en ondelettes [70], sur la sortie d'un classifieur SVM [9]. Les algorithmes travaillent soit directement sur les valeurs des pixels éventuellement ré-échantillonnés soit sur des sous-espaces représentatifs, souvent obtenus par ACP [198, 146].

D'autres auteurs proposent d'optimiser des critères non plus par rapport à l'image de l'objet, mais à son histogramme couleur [46] ou à un estimé de la distribution spatiale des couleurs [60]. L'usage d'un algorithme de *Mean-Shift* sur la rétroprojection des variations de l'histogramme permet dans [46] de suivre des objets déformables, tels que des piétons dans une station de métro, un maillot de footballeur américan,...

Une autre technique très utilisée pour les objets déformables sont les contours actifs [84, 99, 221], qui consistent à minimiser une énergie calculée sur le contour de l'objet. Cette énergie intégre souvent des paramètres images (par exemple, les gradients) et de formes.

La limitation principale des approches globales est leur manque de résistance au regard des occultations. *Black et Jepson* [23] ont surmonté cette limitation en reconstruisant les parties occultées. Ils remplacent la norme quadratique généralement utilisée pour construire l'approximation de l'image dans l'espace propre, par une norme d'erreur robuste. Cette reconstruction revient à une minimisation d'une fonction non linéaire, optimisée en utilisant une méthode de descente de gradient simple. Ils utilisent la même stratégie pour trouver la transformation paramétrique alignant le motif sur l'image. Cet algorithme de optimisation est malheureusement lent et ne tolère que des petits mouvements. Des travaux similaires reposant sur l'utilisation d'espaces propres ont été réalisés par *Shree K. Nayar et al.* [149] et *K. Deguchi et al.* [192] pour le suivi d'objets et du positionnement d'un robot par vision.

Historiquement, l'optimisation était effectuée via une exploration exhaustive des positions potentielles du motif. Mais cette stratégie n'est pas effectivement applicable dans le cas de transformations autres que des translations 2D, qui nécessitent des espaces de paramètres de dimensions supérieures. Aussi des méthodes plus récentes posent le problème comme une minimisation de fonctions non linéaires. Elles utilisent alors des algorithmes de type Newton-Raphson ou Levenberg-Marquard [70].

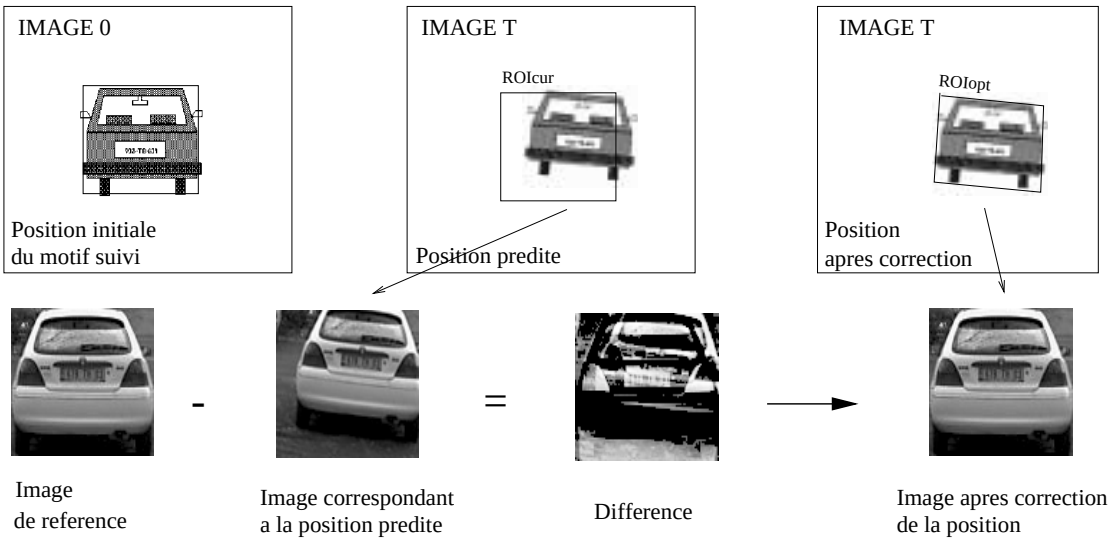


FIG. 3.4 – Suivi par différence d'images.

Cette exploration, qu'elle soit exhaustive ou guidée par un algorithme d'optimisation non-linéaire, s'avère généralement l'étape du processus de suivi la plus coûteuse en temps de calcul.

De nouvelles méthodes ont été proposées récemment qui permettent de se libérer de cette phase exploratrice. Il s'agit d'approximer les variations des paramètres comme une fonction linéaire d'une image de différence (la différence entre l'image de référence et l'image courante - cf. figure 3.4). Cette approche est efficace car le mouvement peut être facilement déduit de l'image de différence. *Cootes, Edwards et Taylor* [47] l'utilisent pour estimer dynamiquement les paramètres d'un modèle de visage en se basant sur l'apparence. Quelques travaux [81, 180] utilisent cette approche avec des transformations projectives. Dans le même esprit, *Hager et Belhumeur* [86] et *Jurie et Dhome* [107, 108] proposent deux approches similaires de suivi efficaces et rapides.

Dans nos travaux, nous avons appliqué au problème du suivi de véhicule l'algorithme de *Jurie et Dhome* développé au LASMEA. Avant d'expliciter plus en détails celui-ci et de décrire les résultats obtenus, dressons un panorama des différentes approches développées par la communauté scientifique pour le suivi de véhicules. Ce panorama nous permettra de situer nos propres travaux avec l'existant dans ce domaine.

3.1.2 Panorama du suivi de véhicules

Un grand nombre d'approches de suivi d'objets ont été appliquées au suivi de véhicules. Deux catégories peuvent être distinguées :

– les approches basées sur le **mouvement** ou *Motion-based* proposent de regrouper les vecteurs exprimant le mouvement durant le temps. Ces vecteurs peuvent être calculés sur des points [187] ou des régions [163].

– les approches basées sur un **modèle** ou *Model-based* qui utilisent des représentations 2D ou 3D du véhicule. Les modèles peuvent être un rectangle [20], des formes 2D [109, 153, 59] ou des modèles 3D [85, 169] des véhicules.

Voyons à présent quelques exemples récents de ces approches, ainsi que de celles qui les combinent.

En 1994, *Christopher E. Smith et al.* [186] proposent d'utiliser un algorithme de SSD, basé sur le flot optique pour suivre un motif de petite taille sur un véhicule. Il s'agit de déterminer les translations minimisant la mesure SSD. Pour éviter les minima locaux, la surface de la réponse SSD est filtrée en la considérant de forme parabolique. Cette approche simple n'est efficace que si le véhicule suivi reste à une distance quasi-constante du véhicule équipé, puisque le modèle de mouvement ne considère pas de changement d'échelle.

Le système ASSET-2 (*A Scene Segmenter Establishing Tracking*) [187] proposé par *Stephen M. Smith* se base sur l'appariement de coins dans des images successives pour déterminer des vecteurs 2D de flot optique. Ces vecteurs sont ensuite regroupés selon un critère de ressemblance (une mesure de distance dans l'espace vectoriel).

Récemment, *Volker Rehrmann* [163] a proposé une méthode pour mettre en correspondance des régions, segmentées au sens de la couleur, entre plusieurs images successives. Le suivi d'un véhicule est alors obtenu en regroupant les régions connexes ayant des trajectoires 2D similaires.

Certains travaux analysent à la fois les trajectoires et leurs positions dans l'image ou dans l'espace 3D. Pour *Coifman et al.* [41], dans un système de surveillance du trafic, les coins d'un même véhicule doivent suivre des trajectoires similaires et leurs distances deux à deux doivent être spatiallement constantes. *Enkelmann et al.* [209] proposent aussi de regrouper des vecteurs de flot optique, obtenus localement par dérivation des niveaux de gris, en tenant compte d'un modèle 3D (réduit à une simple boite englobante) des véhicules.

D'autres utilisent la détection du mouvement pour initialiser un modèle de forme sur l'objet à suivre. Ces travaux concernent souvent des applications de surveillance du trafic, c'est-à-dire utilisant des caméras stationnaires. Le mouvement des objets à suivre est alors relativement facile à segmenter par rapport à une scène fixe. *Koller et al.* [120] présentent par exemple un système de suivi de contour, basé sur les intensités des gradients limitrophes de véhicule et sur une estimation affine du mouvement de ceux-ci. Le contour à suivre est alors initialisé dans des zones où le mouvement est détecté par un processus de *background substraction*.

Haag et Nagel [85] proposent d'initier un suivi de véhicule en calculant le champ de flot optique sur une première image. Les vecteurs de flot optique similaires et voisins, qui sont supposés appartenir au même objet, sont regroupés dans le plan image, puis reprojeté dans l'espace (dans un plan parallèle à la route) connaissant les paramètres de la caméra. A partir de leur position moyenne, de leur orientation et de leur amplitude, ces vecteurs permettent d'estimer l'état (position, orientation et vitesses) initial de véhicule dans la scène 3D. Cet état est alors affiné en mettant en correspondance des segments extraits de l'image avec un modèle 3D polyhédrique reprojeté selon l'état initial déterminé. Dans [169], les mêmes auteurs proposent une amélioration du système en intégrant dans leur modèle 3D des véhicules, l'ombre portée sur la route.

Plusieurs méthodes se basent sur la persistence de certaines propriétés image ou sur le *matching* de modèles de forme dans des zones d'intérêt. Ces zones d'intérêts sont souvent déterminées via un filtre temporel, essentiellement de Kalman.

Ainsi certains systèmes se limitent à vérifier la persistence de certaines propriétés image des véhicules. Ces propriétés sont généralement les mêmes que celles qui ont servi à la détection et à la localisation des véhicules, c'est-à-dire la symétrie [225, 110], la texture [109], la couleur [98],...

Dans [55], *Dellaert et al.* recherchent les rectangles englobant les véhicules et les suit le long de la séquence. Le processus de *matching* se décompose en deux étapes. Premièrement, une transformée de Hough est appliquée à l'image de gradient pour sélectionner des lignes et des colonnes pouvant contenir potentiellement les bords droite, gauche, haut et bas du véhicule. Puis, les rectangles englobants potentiels sont testés. Celui qui minimise une fonction de ressemblance Bayésienne est choisi. Cette approche a deux principaux défauts. Le premier est qu'elle considère les contours verticaux d'un véhicule modélisable par deux droites, ce qui n'est pas toujours vérifié. Ensuite, cet algorithme peut être fortement perturbé par les contours d'autres véhicules ou d'éléments de l'infrastructure environnante.

Aussi, d'autres auteurs proposent des modèles des contours d'un véhicule plus complexes. *van Leeuwen et al.* présentent dans [127] des modèles de contours composés de plusieurs segments. La méthode utilise une image de distance (*Distance Transform*) à la manière de celles développées par *Kalinke* [88], *Gravila* [78] et *Olson* [153]. La distance utilisée alors est la distance au sens de *Chamfer* qui consiste à intégrer sur les modèles de forme, les valeurs des pixels dans l'image de distance. La figure 3.5 illustre le processus pour le *matching* des modèles de contours verticaux dans une ROI. Par ailleurs, ils proposent pour les modèles de contours horizontaux, un processus dynamique permettant de les modifier au cours du temps. Ils gèrent ainsi les possibles variations d'aspect de tels contours, dues aux changements d'échelle et de luminosité sur les séquences.

Ce panorama nous permet de constater que ces systèmes de suivi sont souvent basés sur une représentation du véhicule selon des primitives visuelles locales. La contrainte

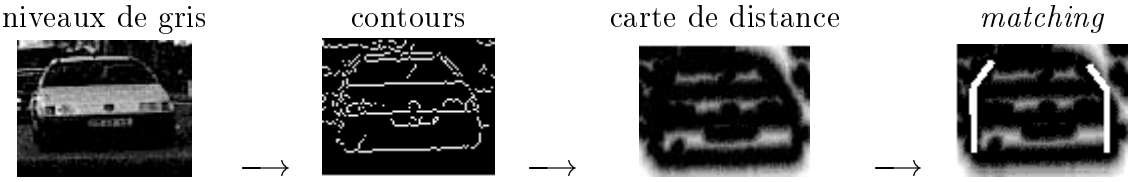


FIG. 3.5 – Processus de *matching* de modèles de forme sur une carte de distance [127]

de temps réel explique ceci. En effet, la plupart des processus utilisant des modèles globaux d'objets sont relativement coûteux en temps. Ils demandent en effet des processus d'optimisation sur un grand nombre de données.

Dans la suite de ce chapitre, nous proposons d'utiliser pour le suivi de véhicule, une méthode basée sur une représentation globale du véhicule, sous la forme d'un motif 2D (vue arrière détectée du véhicule). Cette méthode permet un suivi en temps réel.

3.2 Suivi 2D d'un véhicule

Dans cette section, nous allons d'abord poser quelques définitions. Il s'agit de formaliser la problèmatique du suivi d'un motif. Avec les notations obtenues, nous pourrons alors décrire les modèles de mesure développées par *Hager et al.* et par *Jurie et Dhome*. Nous évoquerons notamment un comparatif entre ces deux approches réalisés dans [107]. Comme nous le verrons, le modèle de *Jurie et Dhome* s'avère justifié sur un plus grand domaine de variations. L'algorithmie permettant l'utilisation de ce modèle est décrit à la suite.

Puis nous proposerons d'utiliser une décomposition en ondelettes de Haar pour la représentation du motif à suivre plutôt qu'un échantillonage de l'image de niveaux de gris. Par ailleurs, nous évoquerons le modèle de mouvement qui nous a semblé le mieux approprié au suivi de véhicules. Enfin, un filtre de Kalman permettant l'extraction des caractéristiques cinématiques sera exposé.

3.2.1 Le suivi de motif : quelques définitions

Dans la suite, nous adoptons les notations introduites par *Hager et al.* dans [86].

Définition d'un motif

Un motif se définit par son apparence, c'est-à-dire par la forme de la surface d'intensité de l'image dans la région qu'il occupe . Cette surface peut s'écrire sous la forme d'une fonction I dans l'espace tri-dimensionnel XYI, telle que $I(\mathbf{x}, t)$ est la valeur de l'intensité lumineuse au point de coordonnées $\mathbf{x} = (x, y)$ dans une image acquise au temps t.

Une région de l'image peut se décomposer en un ensemble de N points. Posons $\mathcal{R} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$ l'ensemble des coordonnées de ces N points. Le motif dans cette

région sera alors décrit par l'ensemble $(\mathcal{R}, \mathbf{I}(\mathcal{R}, t))$. $\mathbf{I}(\mathcal{R}, t)$ correspond à la modélisation de la surface d'intensité dans la région, telle que :

$$\mathbf{I}(\mathcal{R}, t) = (I(\mathbf{x}_1, t), I(\mathbf{x}_2, t), \cdots, I(\mathbf{x}_N, t)) \tag{3.10}$$

Supposons le motif à suivre détecté dans une première image, c'est-à-dire au temps initial t_0. Il sera donc défini par $(\mathcal{R}, \mathbf{I}(\mathcal{R}, t_0))$. Dans la suite, $\mathbf{I}(\mathcal{R}, t_0))$ sera appelé *modèle de référence.*

Modélisation du mouvement d'un motif

Considérons μ le vecteur des paramètres du mouvement déjà évoqué dans la section 3.1.1. De manière générale, posons $\mu(t) = (\mu_1(t), \mu_2(t), \ldots, \mu_n(t))$, où n est le nombre de paramètres. Le *modèle de mouvement paramétrique* de la transformation d'un point $\mathbf{x}$ sera alors noté $\mathbf{f}(\mathbf{x}; \mu(t))$. Cette fonction sera supposée différentiable.

La position d'un motif au temps t peut donc être estimée en connaissant sa position initiale au temps t_0 et en déterminant l'ensemble $\mathbf{f}(\mathcal{R}; \mu(t))$ des transformations, tel que :

$$\mathbf{f}(\mathcal{R}; \mu(t)) = (\mathbf{f}(\mathbf{x}_1; \mu(t)), \mathbf{f}(\mathbf{x}_2; \mu(t)), \ldots, \mathbf{f}(\mathbf{x}_N; \mu(\mathbf{t}))) \tag{3.11}$$

Dans la suite, nous supposerons $N > n$.

La position initiale peut s'écrire sous la forme $\mu(t_0)$, que nous noterons μ_0^*. Le sigle * indique qu'il s'agit de la position réelle (ou exacte) du motif. Ainsi μ représentera l'estimation de la valeur réelle μ^*.

Traduction de l'opération «suivi d'un motif»

Avec les notations données précédemment, suivre le motif consiste donc à estimer le vecteur $\mu(t)$ tel que $\mathbf{I}(\mathbf{f}(\mathcal{R}; \mu(t)), t) = \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu_0^*), t_0)$. Dans une approche globale, cela peut être obtenu en minimisant au sens des moindres carrés la fonction :

$$O(\mu(t)) = \|\mathbf{I}(\mathbf{f}(\mathcal{R}; \mu_0^*), t_0) - \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu(t)), t)\| \tag{3.12}$$

Nous reconnaissons sous cette formulation, un critère de similarité. Comme nous l'avons déjà signalé dans la section 3.1.1, sa minimisation peut être obtenue par une exploration exhaustive ou via un algorithme d'optimisation de type Levenberg-Marquard [23].

Hager et Belhumeur [86] et *Jurie et Dhome* [107] proposent de calculer directement la valeur de $\mu(t + \tau)$ à partir de la différence de niveaux de gris entre deux images successives. τ est le temps entre ces deux images successives. Ils considèrent qu'il existe une relation $\mathbf{A}(t + \tau)$ telle que :

$$\mu(t+\tau) = \mu(t) + \mathbf{A}(t+\tau)\left(\mathbf{I}(\mathbf{f}(\mathcal{R};\mu_0^*),t_0) - \mathbf{I}(\mathbf{f}(\mathcal{R};\mu(t)),t+\tau)\right), \tag{3.13}$$

Ainsi, la variation du modèle paramétrique du motif peut être déduite, en écrivant :

$$\delta\mu(t+\tau) = \mathbf{A}(t+\tau)\delta\mathbf{i}(t+\tau) \tag{3.14}$$

avec $\delta\mathbf{i}(t+\tau) = \mathbf{I}(\mathbf{f}(\mathcal{R};\mu_0^*,t_0) - \mathbf{I}(\mathbf{f}(\mathcal{R};\mu(t)),t+\tau)$ et $\delta\mu(t+\tau) = \mu(t+\tau) - \mu(t)$.

La relation $\mathbf{A}(t+\tau)$ proposée dans ces deux groupes d'auteurs représente donc le *modèle de mesure* dans ces algorithmes. Ils en proposent deux approches différentes pour en estimer des approximations linéaires.

3.2.2 Approximations de la relation $\mathbf{A}(t+\tau)$

La première approche [86] utilise l'image Jacobienne pour leur approximation, notée alors $\mathbf{A}_j(t+\tau)$. La seconde [107] propose une approximation par hyperplans. Elle sera notée $\mathbf{A}_h(t+\tau)$.

Approximation Jacobienne (AJ)

Pour des raisons de lisibilité, les notations seront simplifiées dans la suite : $\mathbf{I}(\mathbf{f}(\mathcal{R};\mu(t)),t)$ est indiquée par $\mathbf{I}(\mu,t)$.

Cette approximation repose sur l'hypothèse que les composantes τ et $\delta\mu$ sont petites. Il est alors possible de linéariser le problème en développant $\mathbf{I}(\mu+\delta\mu,t+\tau)$ en une série de Taylor par rapport à μ et t :

$$\begin{aligned}\mathbf{I}(\mu+\delta\mu,t+\tau) &= \mathbf{I}(\mu,t) + \tfrac{\partial\mathbf{I}(\mu,t)}{\partial\mu}\delta\mu + \tfrac{\partial\mathbf{I}(\mu,t)}{\partial t}\tau + h.o.t. \\ &= \mathbf{I}(\mu,t) + \mathbf{I}_\mu(\mu,t)\delta\mu + \mathbf{I}_t(\mu,t)\tau + h.o.t.\end{aligned} \tag{3.15}$$

où *h.o.t.* sont les autres termes d'ordre supérieur du développement. Ils peuvent être et sont négligés. Par ailleurs, il est possible de supposer $\mathbf{I}_t(\mu,t) \simeq \frac{\mathbf{I}(\mu,t+\tau)-\mathbf{I}(\mu,t)}{\tau}$.

Avec ces hypothèses et en écrivant $\delta\mathbf{i} = \mathbf{I}(\mu+\delta\mu,t+\tau) - \mathbf{I}(\mu,t+\tau)$, l'équation 3.15 devient :

$$\delta\mathbf{i}(t) = \mathbf{M}(\mu,t)\delta\mu. \tag{3.16}$$

où $\mathbf{I}_\mu(\mu,t) = \mathbf{M}(\mu,t)$ est la matrice Jacobienne.

L'examen des équations 3.16 et 3.14 montrent que la matrice $\mathbf{A}$ peut donc s'approximer comme la pseudo-inverse de $\mathbf{M}$, telle que :

$$\mathbf{A}_j(t) = (\mathbf{M}^t(\mu,t)\mathbf{M}(\mu,t))^{-1}\mathbf{M}^t(\mu,t), \tag{3.17}$$

Nous pouvons alors écrire :

$$\delta\mu = (\mathbf{M}^t(\mu,t)\mathbf{M}(\mu,t))^{-1}\mathbf{M}^t(\mu,t)\delta\mathbf{i} = \mathbf{A}_j(t)\delta\mathbf{i} \tag{3.18}$$

Cette expression suppose que $\delta\mathbf{i} = \mathbf{I}(\mu_0^*, t_0) - \mathbf{I}(\mu, t+\tau)$, donc que $\mathbf{I}(\mu+\delta\mu, t+\delta\tau) = \mathbf{I}(\mu_0^*, t_0)$. Ceci est réalisée si on considère le motif correctement localisé après la correction du vecteur de paramètres du mouvement $\delta\mu$. Cette correction consiste en la mise à jour de $\mu(t+\tau)$. Elle est réalisée en écrivant $\mu(t+\tau) = \mu(t) + \delta\mu(t+\tau)$.

Ainsi, l'équation 3.18 relie la différence $\delta\mathbf{i}$ entre l'image dans la région courante et le modèle de référence de la cible avec un déplacement $\delta\mu$. Ce déplacement aligne la région sur la cible, ce qui assure donc son suivi.

Ceci implique l'évaluation de la matrice $\mathbf{M}$ à chaque itération. En l'exprimant comme une fonction du gradient du modèle de référence, il est possible d'obtenir l'expression de la matrice $\mathbf{A}_j$ en quelques lignes de calcul [86].

Approximation Hyperplane (AH)

Jurie et Dhome [107] propose une interprétation différente du calcul de $\mathbf{A}$. L'approximation alors obtenue sera notée $\mathbf{A}_h$, afin de faire le distingo avec $\mathbf{A}_j$ précédemment employée.

Cette approche consiste à modéliser l'équation 3.14 par n hyperplans. En effet, si elle est considérée comme une relation matricielle, elle peut s'écrire de la manière suivante :

$$\begin{aligned}
0 &= (a_{11}, ..., a_{1N}, -1)(\delta i_1, ..., \delta i_N, \delta\mu_1)^T \\
&... \\
0 &= (a_{i1}, ..., a_{iN}, -1)(\delta i_1, ..., \delta i_N, \delta\mu_i)^T \\
&... \\
0 &= (a_{n1}, ..., a_{nN}, -1)(\delta i_1, ..., \delta i_N, \delta\mu_n)^T
\end{aligned} \tag{3.19}$$

Les éléments $a_{11}, ..., a_{1N}$ de la matrice désignent alors les cœfficients de n hyperplans.

Ceux-ci peuvent être estimés durant une phase d'apprentissage. Il s'agit de construire pour chaque paramètre, un système de N_p équations à N inconnues, avec $N_p > N$. Puis de résoudre les n systèmes d'équations obtenus via une estimation au sens des moindres carrées.

Ces systèmes d'équations sont déterminés en collectant N_p couples $(\delta\mathbf{i}^k, \delta\mu^k), k \in [1, N_p]$, où $\delta\mu^k$ est le déplacement associé à une différence $\mathbf{i}^k$ observée. L'obtention de ces N_p couples est réalisée en considérant la dualité du problème ; dans la première image (où le motif à suivre est détecté), l'introduction d'un $\delta\mu_0$ sur la position μ_0^* du modèle de référence permet l'observation de la modification $\delta\mathbf{i} = \mathbf{I}(\mathcal{R}, \mu_0^*) - \mathbf{I}(\mathcal{R}, \mu_0{}')$. En répétant de telles «perturbations» N_p fois, il est alors possible de construire les systèmes d'équations.

Avec ces équations, il s'agit donc de déterminer $\mathbf{A}_h$ telle que : $\sum_{k=1}^{k=N_p}(\delta\mu^k - \mathbf{A}_h\delta\mathbf{i}^k)^2$ soit minimale. Cela est réalisé par une résolution au sens des moindres carrés. Posons $\mathbf{H} = (\delta\mathbf{i}^1, \ldots, \delta\mathbf{i}^{N_p})$ et $\mathbf{Y} = (\delta\mu^1, \ldots, \delta\mu^{N_p})$. $\mathbf{A}_h$ peut être alors approximée par :

$$\mathbf{A}_h = (\mathbf{H}^t\mathbf{H})^{-1}\mathbf{H}^t\mathbf{Y} \tag{3.20}$$

Comparaison des deux approches

Une comparaison des deux approches a été réalisée à la fois sur des images de synthése et sur des images réelles, avec différents modèles de mouvement [107]. Les résultats observés démontrent que l'AH est supérieure que l'AJ.

La différence entre ces deux approches est due au fait que l'AJ réalise des approximations au niveau de chaque point du motif, puis calcule la relation $\mathbf{A}_j$, alors que l'AH tient compte directement de l'ensemble des points. Le calcul de la Jacobienne $\mathbf{M}$ suppose en effet qu'au niveau de chaque point, les niveaux de gris sont des combinaisons linéaires des paramètres du mouvement. Cette hypothèse, qui n'est pas confirmée dans les images réelles, n'est pas nécessaire dans le cas de l'AH.

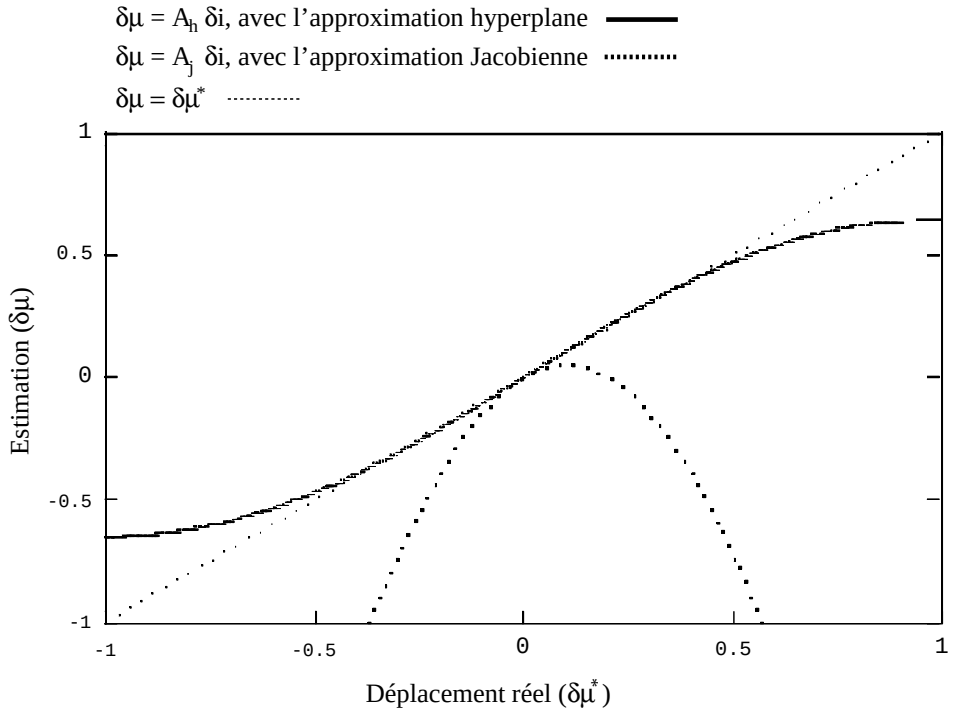


FIG. 3.6 – Comparaison entre l'approximation Jacobienne et l'approximation par hyperplans.

Reprenons l'exemple d'une simple application numérique considérant une ligne image ayant des valeurs de niveaux de gris correspondant à la fonction : $I(x) = exp(-\frac{x^2}{2})$. La

région à suivre est $\mathcal{R} = (x_1, x_2) = (-0.1, 0.0)$ lors d'une simple translation ($\mu = t_x$). La phase d'apprentissage consiste à produire 1000 perturbations aléatoires sur t_x, en utilisant une loi uniforme dans un intervalle $[-.25, +.25]$, donnant des couples de valeurs de δt_x et δi. A partir de ces couples, les différentes matrices sont déterminées ; $\mathbf{A}_j = (11.05, 1.06)$ et $\mathbf{A}_h = (13.42, -13.41)$.

Afin d'évaluer la meilleure modélisation dans ce cas, plusieurs perturbations $\delta\mu^*$ sont à nouveau réalisées dans l'intervalle $[-1., +1.]$. Les valeurs de $\delta\mathbf{i}$ correspondantes sont relevées et multipliées aux différentes matrices. Ceci permet de comparer les $\delta\mu^*$ introduits et ceux $\delta\mu$ estimés par les deux approches. La figure 3.6 représente $\delta\mu$ en fonction $\delta\mu^*$. Dans ce cas, l'approximation par hyperplans donne de meilleurs résultats que l'approximation Jacobienne.

De semblables expérimentations ont été menées sur des images réelles statiques ou en mouvement. Elles ont abouti au même type de résultats.

3.2.3 Description de l'algorithmie du suivi

Ainsi, l'approximation par hyperplans s'avère meilleure que l'approximation Jacobienne. Cependant, prise sous la forme proposée dans la section 3.2.2, elle ne semble pas une solution réaliste pour le suivi de motif. En effet, elle implique à chaque nouvelle image, le calcul de la matrice $\mathbf{A}_h$, c'est-à-dire une minimisation au sens des moindres carrés sur un grand nombre de données.

La matrice telle qu'elle est décrite précédemment dépend des paramètres de μ. Son estimation est réalisée en calculant une relation linéaire entre un ensemble de différences de niveaux de gris et une correction des paramètres μ. Cette relation est calculée autour de μ_0^* et n'est pas *a priori* valable pour les autres valeurs de μ.

En fait, grâce à un changement de référentiel astucieux, il est possible d'établir cette relation pour n'importe quelle valeur de μ, sans avoir à recalculer la matrice. L'astuce consiste à exprimer le mouvement des points formant le motif, non pas par rapport au *référentiel image* mais par rapport à un référentiel lié au motif, appelé *référentiel région* (cf. figures 3.8 et 3.9).

Voyons comment se formalise cette astuce lors des phases d'apprentissage et de suivi du motif. Pour des raisons de clarté, nous allons considérer le motif comme un échantillonage des niveaux de gris d'une image dans une région. Plus tard, nous proposerons une autre représentation du motif, basée sur une décomposition en ondelettes de Haar (cf. annexe C).

Phase d'apprentissage

Dans cette partie, nous allons exprimer dans le *référentiel région* la transformation d'un point lié au motif lors d'une perturbation, en fonction des transformations reliant les coordonnées $\mathbf{u}$ de ce point liées au *référentiel image* et celles $\mathbf{x}$ liées au *référentiel région*.

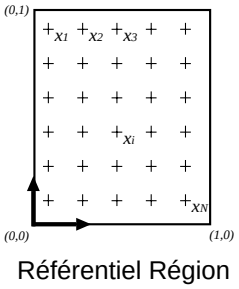


FIG. 3.7 – *Référentiel Région* associé au motif

Considérons donc un référentiel local appelé *référentiel région*. Ce référentiel est attaché à la région $\mathcal{R} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$. La fonction $\mathbf{f}(\mathbf{x}; \mu)$ définit le passage du *référentiel région* au *référentiel image* telle que :

$$\mathbf{x} = (x, y) \rightarrow \mathbf{u} = (u, v) = \mathbf{f}(\mathbf{x}; \mu) \tag{3.21}$$

Dans la première image, la détection du motif à suivre permet de connaître l'ensemble des correspondances entre les points dans le *référentiel région* et les points dans le *référentiel image*. Le calcul de μ_0^*, tel que $\mathbf{f}(\mathbf{x}, \mu_0^*)$ aligne le référentiel région sur le motif, est alors possible.

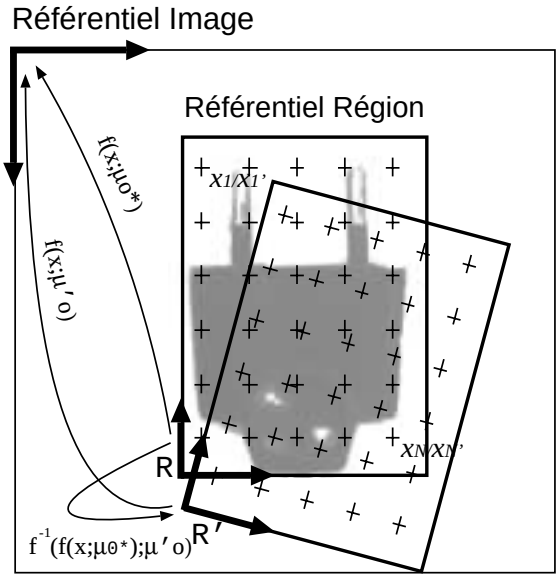


FIG. 3.8 – Perturbation lors de la phase d'apprentissage

La phase d'apprentissage consiste à produire de petites perturbations aléatoires $\delta\mu$ autour de μ_0^*, telles que ${\mu_0}' = \mu_0^* + \delta\mu$. Comme le montre la figure 3.8, une perturbation paramétrée par μ_0' déplace la région $\mathcal{R}$ en une région $\mathcal{R}'$ via une transformation que nous allons exprimer dans le *référentiel région* (lié à la région $\mathcal{R}'$).

Considérons $\mathbf{x}'$ les coordonnées d'un point de la région $\mathcal{R}$ exprimées dans le *référentiel région* lié à la région $\mathcal{R}'$. On a donc ses coordonnées $\mathbf{u}$ dans le référentiel image telles que :

$$\mathbf{u} = \mathbf{f}(\mathbf{x}'; \mu_0') \tag{3.22}$$

Ce point est aussi défini par :

$$\mathbf{u} = \mathbf{f}(\mathbf{x}; \mu_0^*) \tag{3.23}$$

En supposant $\mathbf{f}$ inversible et en considérant les deux précédentes équations, la transformation $\mathbf{x} \rightarrow \mathbf{x}'$ caractérisant le déplacement de $\mathcal{R}$ vers $\mathcal{R}'$ s'écrit dans le *référentiel région* :

$$\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu_0^*); \mu_0') \tag{3.24}$$

Ainsi, durant la phase d'apprentissage, un ensemble de N_p perturbations $\delta\mu$ sont produites dans le but d'obtenir un modèle linéaire donnant $\delta\mu = \mathbf{A}_h \delta\mathbf{i}$, avec $\delta\mathbf{i} = \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu_0^*)) - \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu_0'))$ le changement de luminance entraîné par un $\delta\mu$.

Par conséquent, le déplacement de la région peut être exprimé dans le *référentiel région*, tel que les points $\mathbf{x}'$ soient définis par :

$$\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu_0^*); \mu_0^* + \mathbf{A}_h \delta\mathbf{i}) \tag{3.25}$$

Ce déplacement n'est seulement valable qu'à proximité de μ_0^*.

Phase de suivi

Au début du suivi, une prédiction des paramètres est connue (en pratique, cette prédiction consiste à considérer le motif au même emplcament que dans l'image précédente) et est notée μ'. Le suivi consiste dans l'estimation de μ tel que :

$$\mathbf{I}(\mathbf{f}(\mathcal{R}; \mu), t) = \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu_0^*), t_0)$$

Nous allons noter $I_0(\mathbf{f}(\mathcal{R}; \mu_0^*)) = \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu_0^*), t_0)$ et supprimer des écritures l'expression du temps t afin de faciliter la lecture.

En calculant :

$$\delta\mu = \mathbf{A_h}\delta\mathbf{i} = \mathbf{A_h}[\mathbf{I}_0(\mathbf{f}(\mathcal{R}; \mu_0^*)) - \mathbf{I}(\mathbf{f}(\mathcal{R}; \mu'))] \tag{3.26}$$

nous obtiendrons une perturbation $\delta\mu$ qui aurait produite une différence $\delta\mathbf{i}$ si le vecteur de paramètres à estimer avait été μ_0^*. Dans ce cas, une position $\mathbf{x}$ de la région est transformée en $\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu'); \mu_0^*)$ avec $\mu' = \mu_0^* + \delta\mu$.

La transformation qui est à déterminer est celle qui transforme $\mathbf{x}$ en $\mathbf{u}$: $\mathbf{u} = \mathbf{f}(\mathbf{x}, \mu)$. Comme décrite sur la figure 3.9, l'utilisation de l'équation $\mathbf{x}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu_0^*); \mu_0')$ dans la relation $\mathbf{u} = \mathbf{f}(\mathbf{x}', \mu')$ donne :

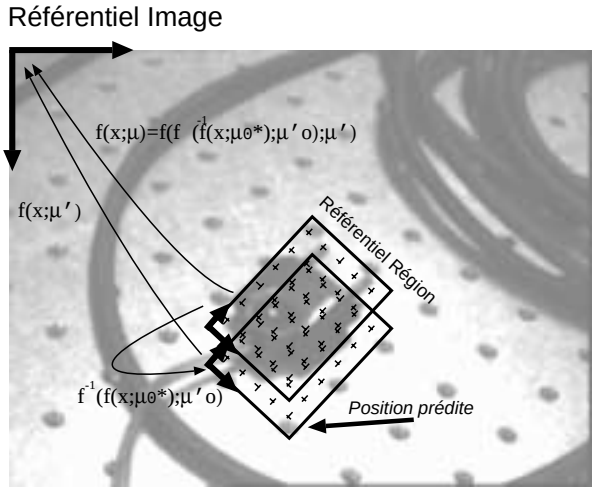


FIG. 3.9 – Phase de suivi

$$\mathbf{u} = \mathbf{f}(\mathbf{x}'; \mu') = \mathbf{f}(\mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu_0^*); \mu_0'); \mu') \tag{3.27}$$

Cette équation est essentielle pour le suivi : elle montre que la transformation permettant d'aligner la région sur la cible au cours du temps, peut être exprimée avec une prédiction μ' et une perturbation locale $\delta\mu$. Cette dernière est obtenue en calculant la différence $\delta\mathbf{i} = \mathbf{I}(\mathbf{f}(\mathcal{R}, \mu_0^*), t_0) - \mathbf{I}(\mathbf{f}(\mathcal{R}, \mu'), t)$. Nous rappelons que l'équation (3.14) est $\mu_0' = \mu_0^* + \mathbf{A}_h \delta\mathbf{i}$.

Le principe du suivi est donc de déterminer la correction de la région dans le *référentiel région*, en agissant comme si les paramètres étaient μ_0^*. Puis, en appliquant la transformation paramétrée par μ' sur la position corrigée $\mathbf{f}^{-1}(\mathbf{f}(\mathbf{x}; \mu_0^*)$ dans le *référentiel région*, la position du motif dans le *référentiel image* est obtenue.

3.2.4 Modèle de mouvement

Application aux modèles linéaires de mouvement

Dans les sections précédentes, la fonction $\mathbf{f}$ définit un modèle de mouvement 2D. Nous avons vu dans la section 3.1.1 qu'il existe différents modèles, dont des modèles linéaires. Ces derniers peuvent s'exprimer sous la forme d'une matrice homogène paramétrée par μ, telle que :

$$\mathbf{f}(\mathbf{x}; \mu) = \mathbf{F}(\mu)\mathbf{x} \tag{3.28}$$

Ainsi, dans le cas de modèles linéaires de mouvement, tels que ceux présentés dans la section 3.1.1, l'équation (3.27) devient :

$$\mathbf{F}(\mu) = \mathbf{F}(\mu')\mathbf{F}^{-1}(\mu_0{}')\mathbf{F}(\mu_0^*) \tag{3.29}$$

où $\mathbf{F}(\mu_0{}') = \mathbf{F}(\mu_0^* + \mathbf{A}_h \delta\mathbf{i})$. $\mathbf{F}(\mu')$ est la transformation obtenue à l'image précédente. $\mathbf{F}(\mu_0^*)$ est calculée à partir de la détection de la région dans l'image initiale. La matrice $\mathbf{A}_h$ est donc obtenue lors d'une phase d'apprentissage. Enfin, $\delta\mu$ est donné par l'équation (3.26).

Dans [107, 108], plusieurs types de modèles de mouvement ont été testés avec succès pour le suivi de motifs plans texturés :
- translation planaire, rotation planaire et changement d'échelle,
- transformation affine,
- transformation homographique.

Le choix du modèle dépends du type d'application visé.

Application au suivi de véhicules

Pour le suivi de vue arrière d'un véhicule sur des voies routières ou autoroutières, le modèle de mouvement 2D apparent le plus souvent utilisé dans la littérature [20, 55, 187, 110, 127, 9] est un mouvement combinant translations planaires et changement d'échelle.

Ce modèle s'exprime sous la forme matricielle suivante :

$$\mathbf{F}(\mu) = \begin{bmatrix} \rho & 0 & t_x \\ 0 & \rho & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.30}$$

avec $\mu = (t_x, t_y, \rho)$, où (t_x, t_y) sont les translations et ρ le changement d'échelle.

Nous avons également choisi ce modèle. Ce choix s'explique pour plusieurs raisons.

Dans l'application visée qui est le suivi de véhicule sur des voies autoroutières, les rotations (dues au tangage et au roulis) et les transformations perspectives d'un motif lié à la vue arrière d'un véhicule sont assez faibles, et peuvent donc être négligées. Ce modèle est donc suffisant à la description du mouvement d'un véhicule dans des scènes autoroutières.

Par ailleurs, il est intéressant de limiter la dimension du paramètre μ au minimum possible. Le choix d'un nombre restreint de paramètres rends en effet l'algorithme plus rapide et plus robuste :

- Réduire la dimension du vecteur $\delta\mu$ à estimer revient à réduire celle de la matrice $\mathbf{A}_h$. Son approximation et son application sont donc moins coûteux en temps de calcul.

- L'estimation de $\delta\mu$ se fait par différenciation de motifs, selon l'équation 3.26. Plus la dimension des paramètres est grande, plus les risques d'ambiguïtés introduit par des différences de motifs proches sont importants. A contrario, en réduisant le nombre de paramètres, la robustesse de l'algorithme aux petites occultations et

au bruit est augmentée.

3.2.5 Représentation du motif par des ondelettes

Dans les paragraphes précédents, nous avons considéré une représentation du motif par sa surface d'intensité $\mathbf{I}$, échantillonnée suivant un ensemble $\mathcal{R}$ de points sélectionnés dans le motif. En pratique, l'ensemble des coordonnées de ces points peut être déduits de celles de quatre points encadrant le motif (dans le cas d'un modèle linéaire de mouvement). Ces quatre points désignent une région d'intérêt dans l'image. Durant le suivi, l'estimation de leurs coordonnées est réalisée via celle de μ.

Dans [107, 108], les auteurs proposent de sélectionner 100 points dans des zones à fort gradient, tout en conservant une distribution relativement uniforme de ces points sur la ROI.

Ici nous proposons d'utiliser une représentation en ondelettes de Haar du motif. Le choix de cette représentation est principalement motivé par le fait qu'une vue arrière de véhicule est assez faiblement texturée. Or l'algorithme de suivi proposé est surtout efficace pour des surfaces très texturées, où les alentours des points échantillonnés ont des variations en luminance remarquables. Il le sera d'autant plus si ces variations sont régulières. La décomposition multi-échelles des ondelettes de Haar permet d'obtenir de telles variations pour les cœfficients des ondelettes, puisqu'on considère les variations d'intégrations de voisinages de points.

Par ailleurs, la représentation via la décomposition en ondelettes est une représentation plus globale du motif que celle via l'échantillonage de la luminance.

Nous allons montrer dans un premier temps que ces représentations sont *a priori* équivalentes. Puis nous donnerons un exemple montrant que la représentation en ondelettes est meilleure que celle en luminance pour le suivi d'un motif appartenant à une vue arrière de véhicule.

Equivalence des deux représentations

Le passage d'une surface d'intensité $\mathbf{I}$ régulièrement échantillonnée à sa transformée en ondelettes peut s'écrire sous la forme d'une fonction $\mathcal{H}$ inversible, telle que :

$$\mathbf{H} = \mathcal{H}(\mathbf{I}) \Leftrightarrow \mathcal{H}^{-1}(\mathbf{H}) = \mathbf{I} \tag{3.31}$$

où $\mathbf{H}$ est l'ensemble des cœfficients de la décomposition ondelettes.

Comme le calcul des cœfficients des ondelettes de Haar s'apparente à des sommations et des soustractions des luminances aux points échantillonnés, la fonction $\mathcal{H}$ peut s'écrire sous la forme d'une matrice, que nous continuons de noter $\mathcal{H}$ par facilité d'écriture.

En introduisant la relation 3.31 dans l'équation 3.26 qui caractérise le principe de l'algorithme de suivi, nous obtenons alors :

$$\begin{array}{c} \delta\mu = \mathbf{A_h}\delta\mathbf{i} = \mathbf{A_h}[\mathbf{I}_0(\mathbf{f}(\mathcal{R};\mu_0^*)) - \mathbf{I}(\mathbf{f}(\mathcal{R};\mu'))] \\ \Leftrightarrow \\ \delta\mu = \mathbf{A_h}\mathcal{H}^{-1}\delta\mathbf{h} = \mathbf{A_h}[\mathcal{H}^{-1}\mathbf{H}_0(\mathbf{f}(\mathcal{R};\mu_0^*)) - \mathcal{H}^{-1}\mathbf{H}(\mathbf{f}(\mathcal{R};\mu'))] \end{array} \tag{3.32}$$

avec donc $\delta\mathbf{h} = [\mathbf{H}_0(\mathbf{f}(\mathcal{R};\mu_0^*)) - \mathbf{H}(\mathbf{f}(\mathcal{R};\mu'))]$ la différence entre la décomposition de Haar du motif de référence avec celle du motif courant.

Ainsi, la représentation en ondelettes de Haar est équivalente à une représentation en luminance du motif. Nous pouvons donc poser une matrice $\mathbf{W}_h$, telle que :

$$\delta\mu = \mathbf{W_h}\delta\mathbf{h} \tag{3.33}$$

Cette matrice $\mathbf{W}_h$ est évaluée par apprentissage, en calculant les cœfficients en ondelettes après chaque perturbation de la *région* $\mathcal{R}$.

Comme pour la représentation en luminance, le fonctionnement en temps réel n'est possible que si on considère une sélection des cœfficients. Ici, nous utilisons les premières échelles de manière décroissante (c'est-à-dire en commençant par les cœfficients calculés via des masques des tailles maximales vers ceux de plus petites tailles).

Cette sélection fait la différence entre les deux représentations. Dans une représentation en luminance, cette sélection est ponctuelle sur le motif. Dans une représentation en ondelettes, la sélection prends mieux en compte la globalité du motif.

Comparaison des deux représentations

Nous avons testé les deux représentations sur différents motifs. Ici nous représentons un des résultats obtenus pour une image fixe. Il est représentatif des autres résultats obtenus en fixe ou en dynamique, et de l'application.

Le motif est sélectionné manuellement sur la partie basse de la vue arrière d'un véhicule (cf. figure 3.10). Sur cette figure, les croix représentent 63 points sélectionnés aléatoirement. Pour le calcul des ondelettes, nous avons réchantillonné ce motif pour obtenir une imagette 32x32 et nous lui appliquons des masques de taille décroissante (de 32x32 à 8x8), ce qui nous fait un total de 63 cœfficients de Haar (3 pour les masques de taille 32x32, 12 pour les masques 16x16 et 48 pour les masques 8x8). Nous obtenons donc deux vecteurs $\delta\mathbf{i}$ et $\delta\mathbf{h}$ de même dimension, représentatifs du motif.

Une série de 1000 perturbations aléatoires du rectangle de référence est réalisée pour estimer les matrices $\mathbf{A}_h$ et $\mathbf{W}_h$. Les variations des paramètres de perturbation sont de 6 %. A chaque itération, les niveaux de gris correspondants aux points échantillonnées et les cœfficients de Haar sont recalculés. A l'aide des jeux de données obtenus, nous

FIG. 3.10 – Motif sélectionné sur une vue arrière d'un véhicule.

pouvons alors estimer les matrices $\mathbf{A}_h$ et $\mathbf{W}_h$.

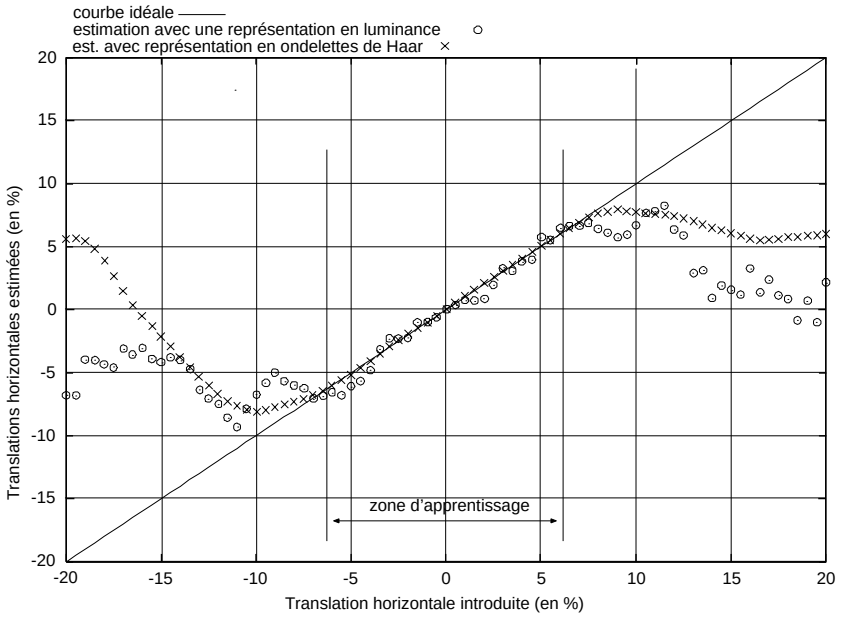


FIG. 3.11 – Comparaison des représentations du motif en niveaux de gris (les ronds) et en ondelettes de Haar (les croix) pour la translation horizontale (t_x).

Afin d'évaluer la meilleure représentation, nous avons appliqué une méthodologie semblable à celle utilisée à la section 3.2.2 pour comparer les approches par Jacobienne et par hyperplans. Plusieurs perturbations $\delta\mu^*$ son à nouveau réalisées avec des variations maximales de l'ordre de 6 %. Les valeurs de $\delta\mathbf{i}$ et $\delta\mathbf{h}$ sont relevées et multipliées à leurs matrices respectives. Ceci permet de comparer les $\delta\mu^*$ introduits et ceux estimés par les deux approches. Les figures 3.11, 3.12 et 3.13 représentent respectivement les

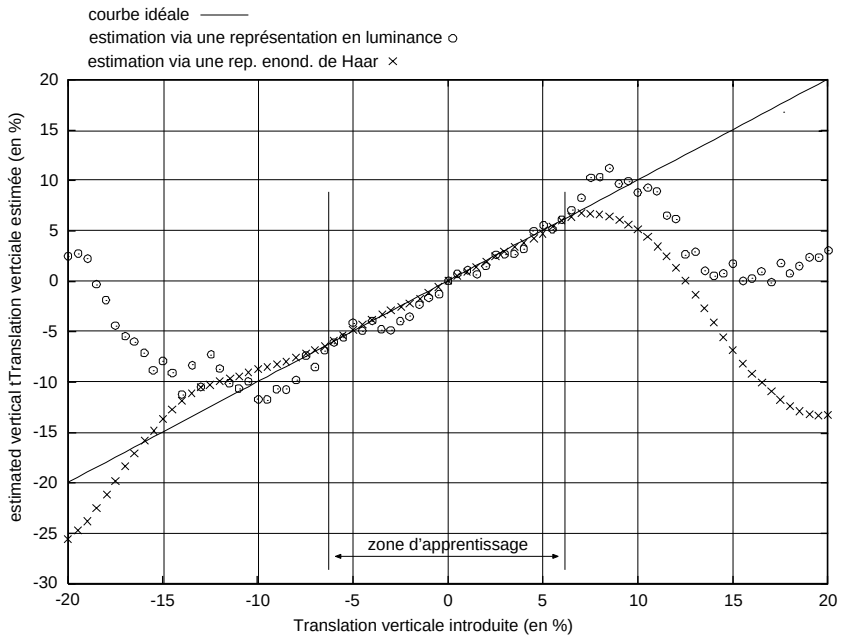


FIG. 3.12 – Comparaison des représentations du motif en niveaux de gris (les ronds) et en ondelettes de Haar (les croix) pour la translation verticale (t_y).

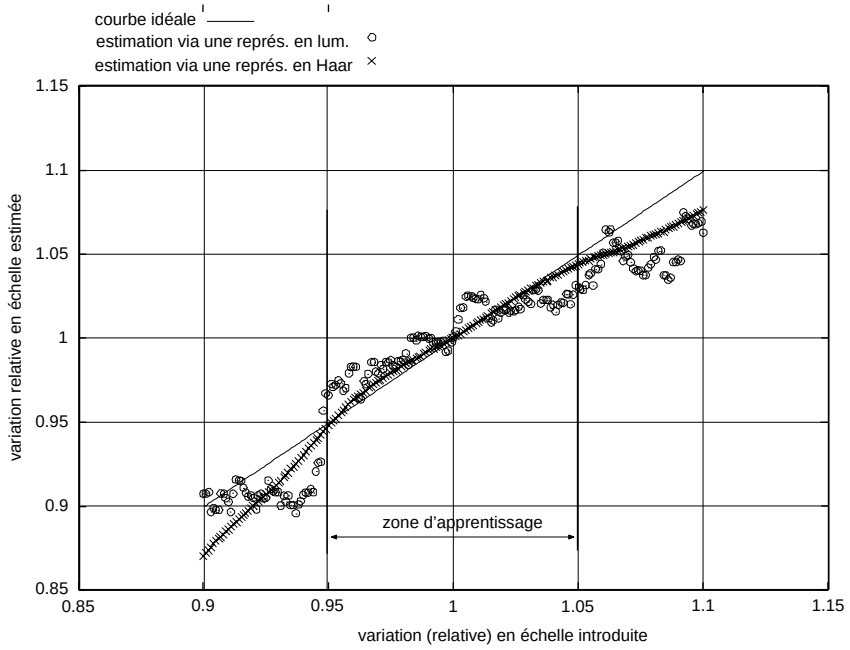


FIG. 3.13 – Comparaison des représentations du motif en niveaux de gris (les ronds) et en ondelettes de Haar (les croix) pour le facteur d'échelle (ρ).

translations t_x et t_y et le changement d'échelle ρ en fonction de t_x^*, t_y^* et ρ^*.

On remarque que les courbes correspondant au motif décomposé en ondelettes sont

plus régulières et, dans la zone d'apprentissage, plus proches de la courbe idéale ($\delta\mu = \delta\mu^*$). Cela montre que, pour un même nombre d'éléments, la représentation en ondelettes de Haar s'avère plus robuste que celle via un échantillonage de la luminance.

Résultats pour le suivi de véhicule et discussion

FIG. 3.14 – Résultats de suivi d'un véhicule avec une représentation en ondelettes de Haar, sur une séquence d'image.

Se basant sur ces résultats, nous avons montré qu'il est possible de suivre un véhicule dans des séquences d'images avec seulement 12 cœfficients de Haar (ceux correspondant aux masques de taille 16x16) aussi bien qu'avec une représentation en luminance utilisant 100 points échantillonnés.

La figure 3.14 montre un exemple de résultats obtenus. Le motif à suivre est initialisé sur une détection réalisée avec l'algorithme du chapitre 2. Le motif correspond à la partie basse du véhicule. Nous voyons que l'utilisation d'ondelettes de Haar permet de suivre un véhicule avec très peu d'éléments.

Cependant, le gain en temps du fait de la dimension réduite des vecteurs utilisés est perdu dans le calcul des cœfficients des ondelettes, notamment lors de la phase d'apprentissage. Aussi, dans les résultats présentés dans la section suivante et dans la suite de nos travaux, nous utilisons la représentation selon les niveaux de gris échantillonnés.

3.3 Résultats sur des séquences d'images

Nous avons testé cette méthode de suivi sur plusieurs véhicules de type différents dans des séquences d'images. Ces séquences ont été enregistrées à bord du véhicule expérimental VELAC. Les résultats présentés dans la suite sont caractéristiques des nombreux essais réalisés.

3.3.1 Application de l'algorithme au suivi de véhicule

L'apprentissage est réalisé sur une série de 1000 perturbations aléatoires. Les amplitudes de ces perturbations ont été fixées à 5 % de la taille de la région pour les variations en échelle et 11 % pour les translations. Ces valeurs autorisent des vitesses relatives longitudinale de plus de 130 $km.h^{-1}$ et latérale de 25 $km.h^{-1}$, ainsi que des perturbations angulaires supérieures à 1 $°.s^{-1}$.

Le motif est représenté par une centaine de points échantillonnés de sorte à avoir une répartition spatiale approximativement uniforme. En pratique la région est découpée en 10x10 zones. Dans chacune de ces zones, plusieurs points sont tirés aléatoirement. Le point présentant une valeur de gradient maximale dans chaque zone, est alors sélectionné.

Pour les séquences 3.15 et 3.16, la caméra est placée de la même façon que décrite dans le chapitre 2. La focale est fixée à 1000 *pixels*. Celles-ci montrent différents véhicules de types et de «couleurs» différents dans différentes situations réelles de conduite.

De plus, les exemples présentés ici reflètent deux types différents d'apprentissage. Dans la figure 3.15, le motif est initialisé sur un véhicule situé assez loin (au delà de 40 m). Dans la figure 3.16, le motif est initialisé sur un véhicule proche (à 20 m).

Nous avons constaté que l'algorithme choisi permet de suivre dans l'image de tels véhicules pendant plusieurs minutes. Il est générique : il permet de suivre n'importe quel type de véhicule, après que la vue arrière lui correspondant ait été apprise. Il est robuste aux translations et aux changements d'échelle. Au travers des deux exemples donnés, il est particulièrement remarquable que le suivi soit assuré quelque soit la position longitudinale du véhicule suivi (dans une fourchette comprises entre 15 et 50 m, pour une focale de 1000 *pixels*) lors de l'apprentissage. Ainsi que le motif soit de taille réduite ou importante, son suivi est assurée dans la suite, même lors de changement d'échelle important. Il s'avère, par ailleurs, assez peu sensible au bruit, aux petites tâches spéculaires, ainsi qu'aux occultations de petites tailles, du fait du modèle de mouvement considéré. Une normalisation des niveaux de gris permet de plus de la rendre résistant aux changements globaux de luminance (passage de ponts,...).

En terme de temps de calcul, l'algorithme s'avère très efficace. En phase de suivi, il ne consiste qu'en une simple multiplication de matrice. Il fonctionne donc en temps réel : à moins de 3 ms par cycle (sur un Pentium III à 800 MHz). Seule la phase d'apprentissage s'avère relativement longue. Il faut en effet 300 ms pour apprendre un motif contenant une centaine de points d'échantillonnage. L'essentiel du temps de calcul est consacré à

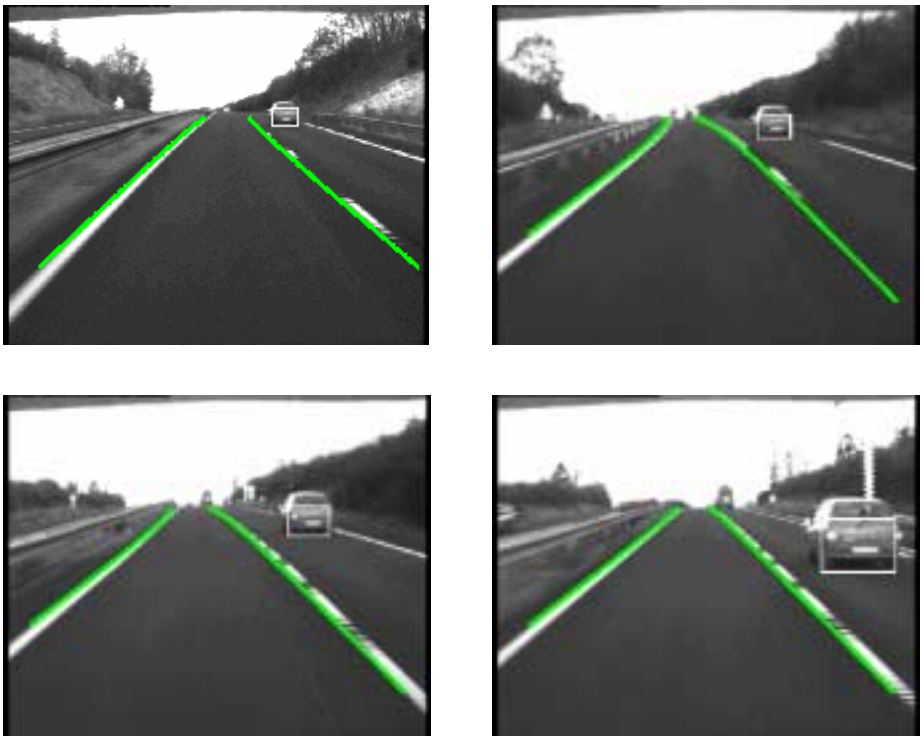


FIG. 3.15 – Résultats de suivi d'un véhicule «gris» sur une séquence d'images.

FIG. 3.16 – Résultats de suivi d'un véhicule «blanc» sur une séquence d'images.

l'approximation aux moindres carrés.

Celle-ci dépends essentiellement de la dimension du vecteur représentatif du motif. Nous avons montré qu'une représentation en ondelettes plutôt que par échantillonage des niveaux de gris permet de réduire sensiblement la dimension du vecteur. La seule difficulté à appliquer cette représentation en ondelettes provient du temps de calcul des ondelettes. Cependant, un article récent [205] propose une méthode (cf. annexe C.5.3) qui permettrait de contourner cet écueil.

Aussi nous pensons que la phase d'apprentissage n'est pas un réel problème à l'application future de cet algorithme pour le suivi de véhicules. D'autant plus qu'en phase de suivi, celui-ci s'avère extrêmement rapide. Il est plus rapide que la fréquence d'acquisition d'image. Il serait donc envisageable de construire un système enregistrant les images de la séquence dans une mémoire provisoire durant la phase d'apprentissage. Puis, cette dernière achevée, la séquence mémorisée sera rejouée à fréquence élevée : le retard sur le déroulement réel de la scène pourra être ainsi compensé.

3.3.2 Extraction des caractéristiques cinématiques

Pour estimer les caractéristiques cinématiques 3D du véhicule suivi, nous utilisons (comme pour la détection) l'hypothèse du monde plan, ainsi qu'un filtre de Kalman [111]. Ce filtre n'est ici utilisé qu'à des fins de filtrage et non de prédiction. Il permet essentiellement d'estimer les vitesses du véhicule suivi. Il s'inspire des travaux dans [142] sur le suivi de véhicules coopératifs. Nous invitons le lecteur à consulter cet ouvrage pour connaître ses différentes caractéristiques que nous avons utilisées et adaptées à un monde plan.

Nous rappelons que l'interprétation de la scène comme un monde plan repose sur deux postulats de base :

- la route à l'avant du véhicule est supposée plane.

- tout obstacle présent dans la scène est en contact avec le sol.

Cette hypothèse permet d'estimer les positions du véhicule, en supposant que la position du motif sur l'image du véhicule est constante. Dans notre cas, le motif est sélectionné de sorte que la base de celui-ci soit confondue avec l'interface entre les roues du véhicule et la route. Ainsi l'extraction des coordonnées 3D $(X, Y, Z = 0)$ du véhicule suivi sur la route est quasi-immédiate en considérant les équations 2.13 du modèle de route données dans la section 2.2.2 et l'annexe D :

$$\begin{aligned} Y &= \frac{Z_0}{\frac{v_s}{f_v} - \alpha} \\ X &= X_0 + \Psi Y + Y \frac{u_s}{f_u} \end{aligned} \tag{3.34}$$

où (u_s, v_s) sont les coordonnées image du barycentre de la base du motif rectangulaire suivi.

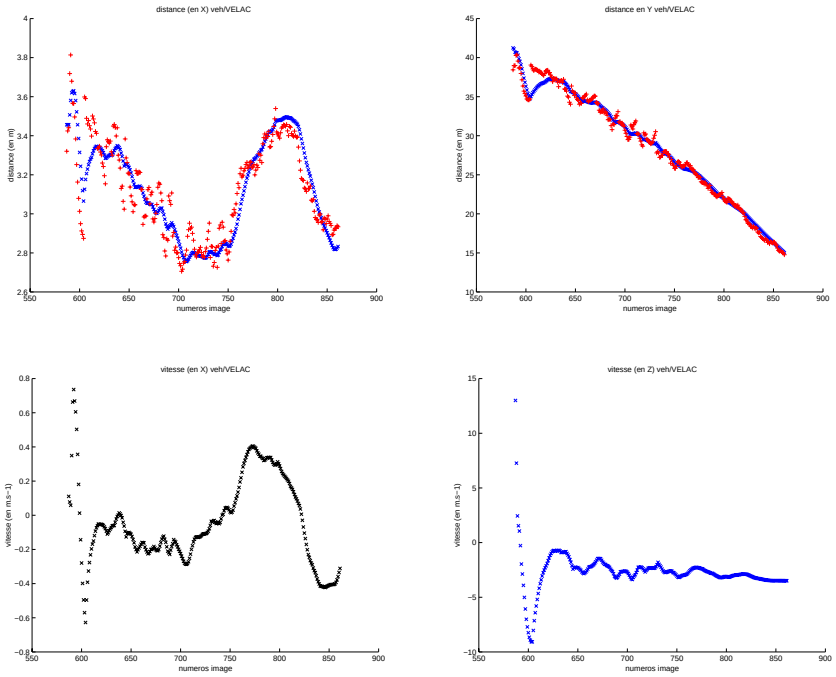


FIG. 3.17 – Estimation des paramètres cinématiques relatifs pour le véhicule «gris» de la séquence 3.15 : les '+' représentent l'état mesuré et les 'x' l'état estimé.

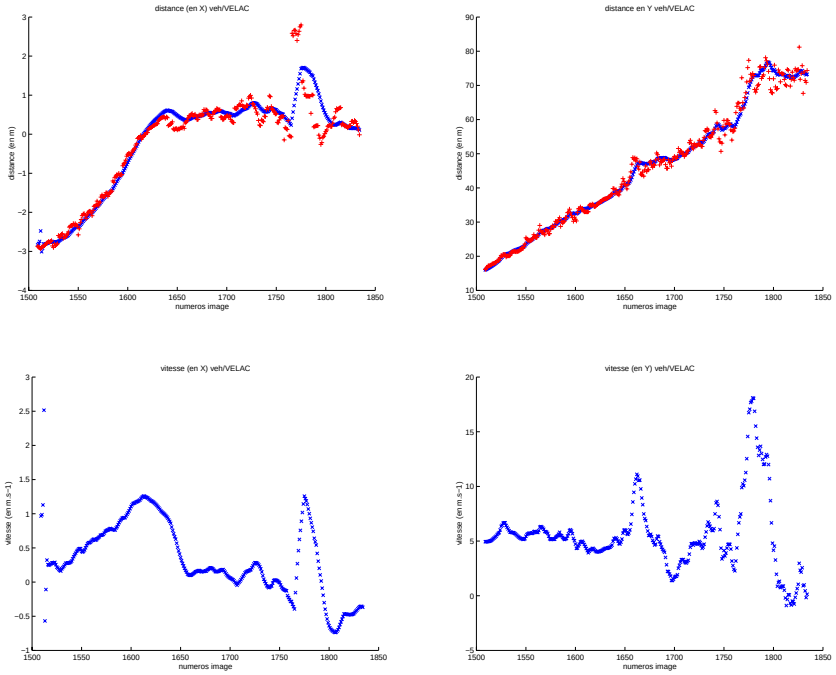


FIG. 3.18 – Estimation des paramètres cinématiques relatifs pour le véhicule «blanc» de la séquence 3.16 : les '+' représentent l'état mesuré et les 'x' l'état estimé.

Le vecteur d'état considéré est limité aux positions 3D du véhicule et aux vitesses :

$$\mathbf{X}_{k/k} = (X, Y, V_X, V_Y)^T \tag{3.35}$$

Nous n'incluons pas :

- la position en Z (ainsi que la vitesse associée) du fait de l'hypothèse d'un monde plan : $Z = 0$,
- les rotations. Du fait de l'algorithme de suivi, nous ne disposons d'aucune mesure permettant leur mise à jour. Par ailleurs, l'expérience acquise au LASMEA dans le cadre du suivi de véhicules coopératifs [142] nous a appris que l'estimation et l'utilisation de ces angles ne s'avèrent pas primordiales pour une tâche de suivi de véhicules obstacles sur voies autoroutières, car ceux-ci sont souvent très faibles, donc négligeables.

Le choix d'utiliser un filtre de Kalman sur des données 3D plutôt que sur des données 2D s'explique d'abord par l'expérience déjà acquise dans ce type de filtre au laboratoire. Ensuite, l'utilisation de données 2D est plus hasardeuse quant au réglage du filtre, car il ne repose sur aucune information dont la pertinence peut être *a priori* vérifiée.

Les figures 3.17 et 3.18 montrent les résultats obtenus pour les séquences présentées aux figures 3.15 et 3.16.

Au cours de nos expérimentations illustrées par ces figures, nous avons constaté que l'estimation des paramètres cinématiques via cette méthode s'avère satisfaisante pour des véhicules à une distance inférieure à 50 m. Au delà de cette distance, l'estimation de l'état devient peu fiable : les mesures sont alors trop bruitées. Des variations d'une dizaine de mètres d'une image à l'autre sont même observables pour la distance longitudinale dans la figure 3.18 pour un véhicule se situant au delà de 60 m.

Ce bruit s'explique par la taille du motif : sa résolution est alors trop faible par rapport à celle du motif de référence.

3.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode permettant de suivre dans l'image des vues arrières de véhicules dans des situations réelles de conduite. Nous avons proposé deux types de représentations du motif à suivre (sélectionné sur le véhicule à suivre) : une basée sur un échantillonnage des niveaux de gris, l'autre sur une décomposition en ondelettes.

La seconde s'avère plus robuste que la première. Son coût algorithmique induit par la décomposition en ondelettes est à l'heure actuelle encore trop important pour son implantation au sein d'un système d'aide à la conduite. Cependant, une optimisation de ce calcul est envisageable dans l'avenir.

De plus, cette représentation permettrait d'envisager des développements futurs in-

téressants. En effet, il s'agit d'une représentation commune avec la méthode de détection présentée dans le chapitre précédent. Une uniformisation en une formalisation commune paraît concevable. L'utilisation du même outil pour l'apprentissage est aussi possible. Un des avantages pourrait être dans l'utilisation d'un apprentissage hors-ligne pour le suivi, et donc un gain de temps dans son initialisation.

Par ailleurs, il serait aussi envisageable de rendre les méthodes plus robustes aux occultations partielles [52].

En conclusion, la représentation par échantillonnage par niveaux de gris peut sous sa forme actuelle être implantée dans des systèmes d'aide à la conduite. L'algorithme de suivi présente alors des résultats satisfaisants. Un filtrage de Kalman permet d'obtenir les caractéristiques cinématiques du véhicule suivi à des distances relativement courtes : entre 15 et 50 m (avec une focale de 1000 $pixels$).

Or, selon la loi, lorsque deux véhicules se suivent, le conducteur du second doit maintenir une distance de sécurité suffisante pour pouvoir éviter une collision en cas de ralentissement brusque ou d'arrêt subit du véhicule qui le précède. Cette distance est d'autant plus grande que la vitesse est plus élevée. Elle correspond à la distance parcourue par le véhicule pendant un délai d'au moins deux secondes. Le tableau 3.1 ci-dessous dresse un indicatif de ces distances.

50 km/h	28 m
90 km/h	50 m
110 km/h	62 m
130 km/h	73 m

TAB. 3.1 – Tableau indicatif de la distance de sécurité minimale en fonction de la vitesse.

Ainsi, pour une application sur voies autoroutières où les véhicules circulent souvent au dessus de 90 km/h (la vitesse minimale sur autoroutes est de 70 km/h), un système avec une focale courte est inadapté au suivi de véhicules.

Afin de suivre un véhicule sur voies autoroutières, il est donc nécessaire d'utiliser une caméra à focale plus longue.

Chapitre 4

Asservissement de la caméra PTZ

Dans le chapitre précédent, nous avons vu que la méthode de suivi de véhicule obstacle pose problème lorsque la distance entre la caméra et le véhicule s'accroit. En effet, au delà d'une distance relativement courte ($\sim$50 m), une reconstruction de la position (en particulier de la distance) et de la vitesse d'un véhicule est peu fiable. La raison principale est que la caméra dispose d'un objectif grand angle afin d'avoir une vision globale de la scène avant, et ainsi détecter de nouveaux véhicules entrants. Ce type de caméra permet donc une capture large de la scène avant, avec une bonne qualité d'image pour des véhicules à faibles distances. En contrepartie, pour des objets à plus grande distance, leur taille dans l'image (et donc leur résolution) devient alors faible, ce qui est un handicap sérieux pour le suivi de véhicule obstacle.

Aussi, nous proposons d'utiliser, en plus de la caméra fixe à grand angle, une caméra munie d'un zoom qui permettra d'avoir une capture de l'obstacle avec une meilleure résolution.

Cependant, augmenter le zoom d'une caméra revient à réduire son angle de vision. Pour pouvoir suivre un véhicule sans que celui-ci ne sorte de l'image, il est donc nécessaire de faire varier la position du champ de vision de la caméra avec zoom, en particulier ses angles en site et azimut.

Par ailleurs, prendre simplement une caméra munie d'une focale longue mais fixe poserait aussi des problèmes de sortie de l'image, lorsque le véhicule suivi se rapprocherait de la caméra embarquée. Le motif suivi pourrait alors devenir plus grand que l'image ou, si son déplacement est relativement important, il pourrait sortir du champ de vision d'une image à l'autre. Il est donc important de contrôler aussi le zoom en fonction de la taille du motif suivi.

Ainsi suivre un véhicule obstacle avec une caméra avec zoom nécessite de commander l'orientation en angles site et azimut de son axe optique et de piloter son zoom, de sorte à conserver le motif suivi dans l'image.

Dans ce chapitre, nous proposons une méthodologie pour arriver à cette fin. Elle définit la notion de «*PTZ tracking*». Il s'agit de l'asservissement en angles site et azimut

et en zoom d'une caméra Pan-Tilt-Zoom, afin de maintenir le motif suivi, ici la vue arrière d'un véhicule, au centre de l'image et à une taille quasi-constante. Cette notion est à rapprocher de celle de «*zoom tracking*» définie dans [68], où seul le zoom est asservi, afin de conserver la taille du motif quasi-constante.

Nous proposons ici d'utiliser pour la commande de la caméra PTZ, une approche référencée capteur, c'est-à-dire de commander les actionneurs (pan, tilt et zoom) de la caméra à partir des informations directement fournies par le capteur. Ces informations seront bien entendu extraites via le même algorithme de suivi d'objets, proposé dans le chapitre précédent. Celles-ci étant ici de type visuelles, on parle alors d'asservissement visuel.

Aussi, après une brève introduction sur les utilisations et les travaux concernant les caméras avec zoom, nous proposons dans ce chapitre de dresser rapidement un panorama de l'asservissement visuel. Nous y énumérons les diverses méthodes possibles. Nous expliciterons notamment le formalisme de fonction de tâche. Dans ce cadre, nous rappellerons la définition d'une loi de commande proportionelle, générallement utilisée en asservissement visuel.

Puis, nous aborderons le problème du contrôle de la caméra PTZ. Nous le traduirons dans le formalisme énoncé ; ce qui nous permettra d'écrire une loi de commande simple, sur le modèle défini. Plusieurs résultats de simulation démontront que cette loi est adaptée au domaine du suivi de véhicule obstacle. Une autre série d'essais, effectués en laboratoire, donnera ensuite des éléments de discussion quant à l'apport et aux inconvénients éventuels du zoom pour l'estimation de la distance du véhicule par rapport à la caméra embarquée, et donc au véhicule expérimental. Enfin, avant de conclure et d'énoncer quelques perspectives à ces travaux, quelques démonstrations légitimeront notre méthode pour notre application dans des situations réelles de conduite.

4.1 De l'utilisation d'une caméra avec zoom en vision par ordinateur

Depuis quelques années, plusieurs travaux s'intéressent à l'utilisation de caméra avec zoom dans le cadre d'applications de vision par ordinateur, notamment pour la détection ou le suivi d'objets. En effet, contrôler le zoom d'une caméra permet d'augmenter la perception visuelle d'un objet dans son environnement. Son apport peut se décomposer de la manière suivante :

- La précision, la sensitivité et la précision d'un capteur visuel dépendent directement de sa focale. Aussi le contrôle de la focale permet d'obtenir pour l'image, les caractéristiques requises directement, sans avoir à recourir à des techniques de restoration d'images souvent très coûteuses en temps. Cette stratégie permet d'obtenir directement des images de qualité suffisante.

- Pour une tâche dédiée, il serait aussi possible de déterminer la distance focale optimale au sens d'un compromis entre la précision des données obtenues et la robustesse des algorithmes. Bien sûr, suivant la tâche, il faut savoir définir une stratégie appropriée. Par exemple, comme nous l'avons déjà signalé, en suivi, il est souvent nécessaire d'avoir un champ de vue suffisamment important.

- Une image d'une grande résolution peut s'avérer plus simple qu'une vue générale (à faible résolution) de la scène, lorsqu'on veut se focaliser sur un détail. La complexité de la scène peut donc être contrôler sans perdre les performances du signal d'entrée, comme lors d'un ré-enchantillonnage. il est ainsi possible de se focaliser :
  1. sur un objet mobile (comme c'est le cas dans notre application),
  2. sur un détail qui doit être observé avec plus de précision ou
  3. sur une partie inconnue de la scène.

- Zoomer permet de réaliser de la stéréoscopie axiale en combinant deux images avec différentes focales. Cette méthode, utilisant conjointement le système de mise au point et une analyse du mouvement, peut permettre de calculer des informations 3D sur la scène observée avec uniquement une caméra monoculaire et sans avoir à la déplacer.

Dans ce cadre, nous pouvons citer plusieurs études qui ont été menées sur les caméras munies de zoom et de leur possibles utilisations. Citons d'abord, les travaux menés dans [129, 211, 63] pour ce qui est de la modélisation et de la calibration d'une caméra avec zoom. D'autres travaux [91] ont été menés aussi sur l'auto-calibration de ce type de capteurs. Ensuite ceux [135, 126, 159, 68] menées pour reconstruire la profondeur d'une scène statique en utilisant le zoom. D'autres [206, 44, 145, 181, 182, 93] étudient plus les conséquences de l'utilisation d'une caméra avec zoom sur la robustesse d'algorithmes. *Cahn von Seelen et al.* [206] proposent, par exemple, un système de vision active qui permet de suivre une cible malgré des changements de focale. Il est basé sur une méthode de corrélation adaptative, qui renouvelle le motif à corréler en fonction des variations de la focale. Ces variations sont mesurées par un processus externe (les contrôleurs de la caméra) à l'algorithme. *Hayman et al.* [93] proposent un système similaire, mais fondé sur le suivi de plusieurs éléments sur le même motif. La résistance aux variations de zoom est obtenue en éliminant les éléments incertains (qui ne vérifient plus le modèle géométrique de l'objet ou dont le critère de corrélation est trop faible) et en sélectionnant d'autres de manière dynamique. Enfin, l'utilisation potentielle d'une caméra avec zoom pour des applications d'asservissement visuel ont été étudiées dans [65, 68, 97, 101].

Comme nous le voyons, l'utilisation potentielle d'une caméra avec zoom dans le domaine de la vision par ordinateur a été le sujet de plusieurs recherches.

Dans la suite, nous allons exploiter quelques uns des résultats de ces travaux, notamment au niveau de la modélisation d'une caméra avec zoom. Ils vont nous permettre de définir pour notre caméra PTZ, une loi de commande par asservissement visuel adaptée à notre application de suivi d'un véhicule obstacle.

4.2 Asservissement visuel : un bref état de l'art

Le problème qui nous intéresse peut être résolu en le formulant sous la forme d'une commande automatique (ou asservie) d'un robot en utilisant les informations recueillies par un capteur fixé sur ce robot.

Dans notre cas, le robot est constitué des actionneurs contrôlant les positions angulaires de la caméra et de son zoom. Le capteur en question est naturellement la caméra.

Aussi notre problème peut se résumer à un problème d'*asservissement visuel*. Il s'agit donc de réaliser notre tâche robotique à partir des informations visuelles extraites des images (via un algorithme de traitement d'image ; ici, il s'agit de l'algorithme de suivi d'objets présenté dans le chapitre précédent) fournies par notre caméra.

Avant d'expliciter cette tâche et de la formuler, et afin de mieux l'appréhender, nous présentons dans cette section, un bref état de l'art des techniques d'asservissement visuel. Nous verrons les principaux schémas de commande dans ce domaine. Puis nous définirons l'approche *fonction de tâche*, que nous allons utiliser pour la commande de notre caméra.

Nous nous attacherons à exposer les avantages et les inconvénients de chaque approche, en la référant si nécessaire à notre application.

4.2.1 Les principaux schémas de commande

Différentes approches de l'asservissement visuel sont décrites dans la littérature. Elles peuvent classées suivant trois critères définis par *Sanderson et Weiss* [175] :

- la liaison de la caméra avec le robot,
- l'utilisation ou non d'une boucle interne,
- l'espace de contrôle.

Dans les paragraphes suivants, nous allons détailler brièvement cette classification.

La liaison de la caméra avec le robot

Ces systèmes peuvent être d'abord distingués par la position des caméras suivant que :

- la caméra est déportée. Elle ne possède alors aucune liaison mécanique avec le robot. Elle peut observer soit uniquement la scène (c'est-à-dire le ou les objets sur lesquels l'asservissement du robot est effectué), soit le robot [71], ou bien sûr les deux à la fois [96].

– la caméra est embarquée. Une telle configuration est appelée *Eye in Hand*. Elle est alors placée sur l'organe terminal du robot, et observe alors la scène. C'est la configuration la plus répandue [100] et qui corresponds à notre application.

L'utilisation d'une boucle interne

La commande d'un robot nécessite de connaître son état aux articulations. En asservissement visuel, deux approches sont considérées.

Soit le système considère l'existence d'une boucle interne. La consigne fournie par le système de vision est alors traduite par le contrôleur du robot dans l'espace articulaire. Une boucle fermée interne, utilisant le modèle dynamique du robot, stabilise le système. Le processus de vision et le système robotique sont alors clairement disjoints, ce qui permet une grande portabilité et simplicité des processus de commande par vision.

D'autres travaux utilisent un système de vision pour obtenir directement une estimation de l'état du robot aux articulations, se substituant à ses contrôleurs. La boucle interne est alors rendue inutile : la stabilisation du système est obtenue uniquement en utilisant le système de vision. Ce type de schéma est peu usité en raison de l'influence importante des perturbations sur l'estimation temps réel de l'état. Elle est utilisée cependant dans certains travaux, comme ceux de *Gangloff et al.* [76] pour l'asservissement rapide d'un robot à 6 degrés de liberté.

L'espace de contrôle

Trois grandes classes de méthode peuvent par ailleurs se distinguer suivant le type d'informations visuelles considérées pour la commande :

– les asservissements visuels en situation, ou **asservissement 3D** [212, 194]. La situation courante de la caméra, et donc de la pose du robot, vis-à-vis de l'objet cible est obtenue en interprétant les informations visuelles extraites de l'image. Le déplacement du robot est alors effectué en corrigeant la différence entre l'évaluation de la pose actuelle et celle de la pose à atteindre. En pratique, l'étape de reconstruction des informations tridimensionelles entraîne une forte sensibilité aux bruits de mesure. Cela peut être très penalisant près de la convergence. Par ailleurs, durant l'asservissement, l'observabilité de la scène ne peut être forcément garantie.

– les **asservissements basés image**, ou dits **2D**, où la loi de commande est exprimée directement dans l'espace capteur. Ils ont pour objectifs d'atteindre un motif dans l'image et non plus à contrôler une situation entre la caméra (ou le robot) et l'objet.
Différents types d'informations extraites de l'image peuvent être utilisés. La plupart des travaux dans ce domaine sont basés sur l'utilisation de points dans les images [34]. Dans certains cas, ces primitives peuvent être problèmatiques [35]. Aussi d'autres peuvent être utilisées, comme des droites [34, 1], des cylindres ou

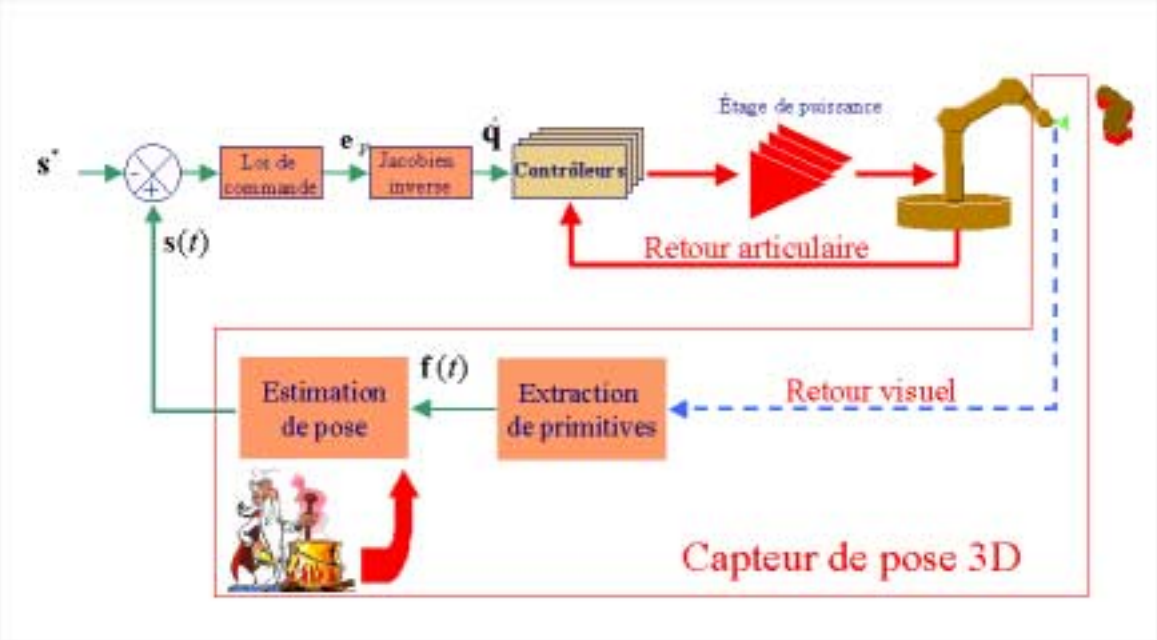


FIG. 4.1 – Schéma d'un asservissement 3D.

des ellipses [34, 114]. Pour appréhender des objets plus complexes, des informations visuelles telles les moments centraux des surfaces [207, 36], les descripteurs de Fourier ou la signature polaire [42] sont aussi utilisés. Dans le cas où des primitives visuelles peuvent être difficilement être extraites de l'image, des informations dynamiques pourront être choisies [50].
Le comportement du système commandé dépendra fortement du choix des informations visuelles, de leur nombre et de leur configuration [35], mais également du point de vue à partir duquel elles sont observées[183]. Enfin, signalons que la convergence de ce type de commande peut être observé même en présence d'importantes erreurs de modélisation, lorsque le déplacement à effectuer n'est pas très important. Cependant, ces travaux ont été étendus à des déplacements plus conséquents grâce aux résultats de *Y. Mézouar et al.* [147] sur la planification et le suivi de trajectoires.

– les **asservissements hybrides**, dans lesquels s'insèrent les travaux d'*Enzio Malis et al.* [138] sur le **2D 1/2**, où la loi de commande utilise à la fois des informations 2D et des informations 3D, afin d'améliorer les performances et la robustesse du système. Un tel schéma de commande intègre donc à la fois un contrôle de la trajectoire de la caméra dans son espace de travail et un contrôle de la trajectoire de certaines primitives dans l'image. Ceci permet d'augmenter la probabilité que l'objet cible reste dans le champ de vision de la caméra. Cependant, si par cette méthode, on assure la présence d'un certain nombre de primitives visuelles, l'observabilité de la tâche n'est pas garantie forcément (c'est-à-dire qu'un nombre suffisant de primitives visuelles reste dans l'image pour estimer les paramètres 3D).

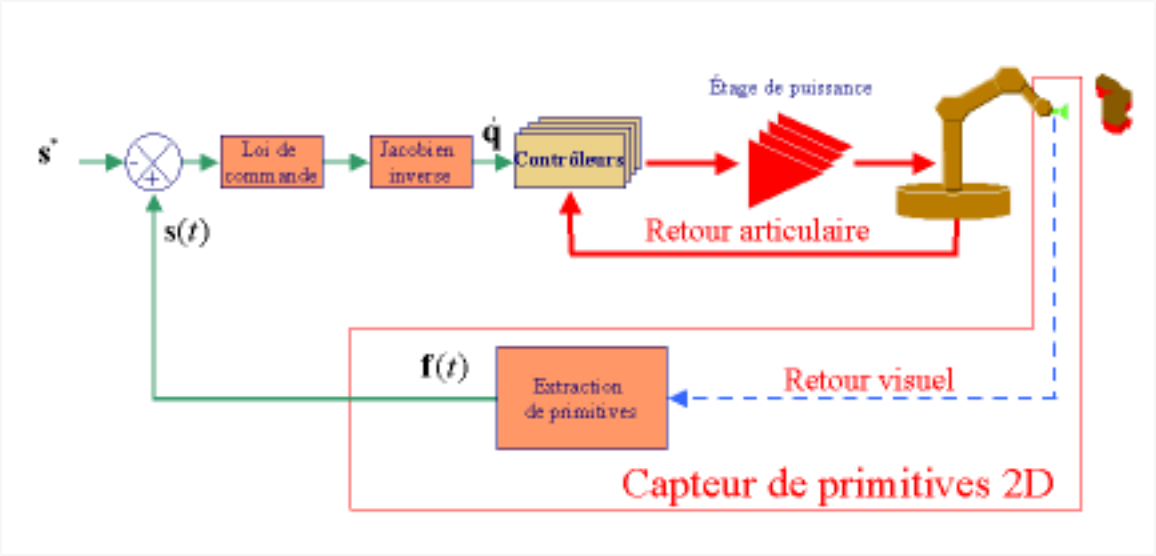


FIG. 4.2 – Schéma d'un asservissement 2D.

Quel asservissement choisir ?

D'après la classification donnée précédemment, étudions ici quel asservissement choisir par rapport à notre application :

- Il est assez évident que dans notre cas, la caméra est embarquée sur une platine contrôlée en site et azimut.

- La commande de notre caméra PTZ sera assurée via une boucle interne. C'est la solution la plus simple, mais aussi la plus portable.

- Enfin, l'objectif de notre asservissement est de conserver un motif visuel dans l'image. Comme nous le verrons dans la suite, la configuration de notre application est assez simple : les trajectoires dans l'image ne posent pas de problème complexe de sorties du champ de vision. Un asservissement visuel 2D semble alors le mieux approprié vis-à-vis de cet objectif.

4.2.2 La commande en asservissement visuel

Dans le domaine de l'asservissement visuel, plusieurs types de commande ont été appliqués. Les approches les plus courantes sont celles utilisant une commande de type proportionnelle, permettant d'assurer une décroissance exponentielle d'une erreur définie [34, 143]. C'est ce type de commande que nous nous proposons d'utiliser.

Ces dernières utilisent le formalisme de la fonction de tâche définie par *Claude Samson et al.* en 1991 [174]. Aussi dans les paragraphes qui suivent, nous décrirons ce formalisme, ainsi que l'établissement d'une commande en vitesse, en négligeant la dynamique du robot, permettant une décroissance exponentielle d'une erreur choisie.

Comme nous l'avons dit, nous allons utiliser un asservissement visuel 2D ; aussi, nous allons aborder ce formalisme uniquement dans ce cadre.

La fonction de tâche

Le principe de ce formalisme consiste à réaliser une tâche robotique par la régulation (à zéro) sur un horizon temporel limité d'une fonction d'erreur $\mathbf{e}(\mathbf{q},t)$ de vecteur configuration $\mathbf{q}$ (représentant les coordonnées articulaires du robot) et de variable temporelle t . Cette fonction d'erreur de classe C^2 dite, fonction de tâche, s'exprime sous la forme :

$$\mathbf{e}(\mathbf{q},t) = \mathbf{C}.(\mathbf{s}(\mathbf{q},t) - \mathbf{s}^*(t)) \tag{4.1}$$

où :
- $\mathbf{C}$ est la matrice de combinaison permettant de prendre en compte un nombre d'informations visuelles supérieur au nombre de degrés de liberté à commander.
- $\mathbf{s}(\mathbf{q},t)$ est le vecteur de mesure, représentant l'ensemble des informations visuelles pour réaliser la tâche.
- $\mathbf{s}^*(t)$ est la consigne, c'est-à-dire la trajectoire idéale devant être suivie dans l'espace image.

Le problème général de régulation de la fonction de tâche $\mathbf{e}$ est bien posé si celle-ci possède certaines propriétés. D'une part, l'existence d'une trajectoire idéale et unique de $\mathbf{q}$ doit être garantie. D'autre part, le jacobien

$$\mathbf{J}_e = \frac{\partial \mathbf{e}}{\partial \mathbf{q}} \tag{4.2}$$

doit être régulier autour de cette trajectoire, c'est-à-dire que son déterminant soit non nul. C'est la propriété d'admissibilité de la fonction de tâche.

Via ce formalisme, il est aussi possible de définir une tâche secondaire, qui utilisera les degrés de liberté non contraints par la tâche principale (définie par l'équation 4.1). Cette tâche secondaire peut consister à éviter les singularités du robot [140], les butées articulaires [37], les occultations [141] ou à contourner les obstacles [16].

La définition d'une tâche secondaire n'est possible que si des degrés de liberté du robot ne sont pas utilisés pour la réalisation de la tâche principale. Dans notre application, nous verrons que la tâche principale utilise tous les degrès de liberté de notre robot.

Réalisation d'une commande en vitesse

La réalisation d'une commande en vitesse est posée comme un problème dual. Dans un premier temps, l'influence d'une commande $\dot{\mathbf{q}}$ (le robot est considéré comme un intégrateur parfait) sur $\mathbf{e}$, et donc sur les primitives visuelles $\mathbf{s}$, est exprimée sous la forme d'une relation. Dans un second temps, cette relation est «inversée» pour définir une loi de commande selon la variation $\dot{\mathbf{s}}$ des primitives visuelles.

Générallement, cette loi de commande est exprimée sous la forme d'un torseur cinématique $\mathbf{T}_c$ à appliquer à la caméra.

Ainsi, ce processus passe par l'expression de la dérivée temporelle de $\mathbf{e}$:

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t} \tag{4.3}$$

En considérant $\mathbf{r}$ la pose de la caméra, celle-ci devient :

$$\dot{\mathbf{e}} = \frac{\partial \mathbf{e}}{\partial \mathbf{r}}\frac{\partial \mathbf{r}}{\partial \mathbf{q}}\dot{\mathbf{q}} + \frac{\partial \mathbf{e}}{\partial t} \tag{4.4}$$

Dans cette équation, deux termes sont particulièrement remarquables.

- En premier lieu, le terme $\frac{\partial \mathbf{r}}{\partial \mathbf{q}}$ exprime le passage entre le torseur cinématique de la caméra $\mathbf{T}_c = (V_X, V_Y, V_Z, \Omega_X, \Omega_Y, \Omega_Z)^T = \dot{\mathbf{r}}$ et la vitesse articulaire $\dot{q}$ de la caméra. Celui-ci est caractérisé par une matrice $\mathbf{J}$:

  $$\mathbf{T}_c = \mathbf{J}\dot{\mathbf{q}} \tag{4.5}$$

  Cette matrice peut être décomposée en un produit de deux jacobiens supposés connus et inversibles :
  - le jacobien liant le torseur cinématique à la vitesse de l'effecteur ; celui-ci peut être obtenu lors d'une phase de calibration [167, 2],
  - le jacobien exprimant le passage entre la vitesse de l'effecteur du robot et la vitesse articulaire du robot ; c'est la matrice jacobienne du robot.

- En second lieu, comme nous l'avons déjà signalé, le jacobien de la tâche $\mathbf{J}_t$ est défini par le terme $\frac{\partial \mathbf{e}}{\partial \mathbf{r}}$. Si la matrice de combinaison $\mathbf{C}$ ne dépend pas explicitement de $\mathbf{r}$, celui-ci peut aussi s'écrire sous la forme :

  $$\mathbf{J}_t = \mathbf{C}\mathbf{L}_\mathbf{s} \tag{4.6}$$

  où $\mathbf{L}_\mathbf{s} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}}$. Elle exprime le passage entre la vitesse de la caméra et les variations du vecteur de mesure $\mathbf{s}$. Dans le cas où $\mathbf{s}$ est composée de primitives visuelles, elle est connue sous le nom de matrice d'interaction.

Au final, nous pouvons écrire que :

$$\dot{\mathbf{e}} = \mathbf{J}_t \mathbf{T}_c + \frac{\partial \mathbf{e}}{\partial t} \tag{4.7}$$

Lors de la régulation de la fonction de tâche, il est possible d'obtenir sa décroissance exponentielle de constante de temps λ, en considérant uniquement la cinématique du robot. Cela peut s'écrire sous la forme suivante :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{4.8}$$

En considérant les deux dernières équations 4.8 et 4.7, la loi de commande recherchée pour cette régulation est donc :

$$\mathbf{T}_c = -\mathbf{J}_t{}^{+}(\lambda \mathbf{e} + \frac{\partial \mathbf{e}}{\partial t}) \tag{4.9}$$

En pratique, seules des approximations (distinguées par un chapeau) du jacobien de tâche et de $\frac{\partial \mathbf{e}}{\partial t}$ peuvent etre utilisées dans la relation précédente. La commande effective est donc :

$$\mathbf{T}_c = -\widehat{\mathbf{J}}_t{}^{+}(\lambda \mathbf{e} + \widehat{\frac{\partial \mathbf{e}}{\partial t}}) \tag{4.10}$$

En introduisant cette dernière relation dans 4.7, on aboutit à :

$$\dot{\mathbf{e}} = -\mathbf{J}_t \widehat{\mathbf{J}}_t{}^{+}(\lambda \mathbf{e} + \widehat{\frac{\partial \mathbf{e}}{\partial t}}) + \frac{\partial \mathbf{e}}{\partial t} \tag{4.11}$$

Ce qui permet d'obtenir la condition suffisante de décroissance suivante :

$$\mathbf{J}_t \widehat{\mathbf{J}}_t{}^{+} > 0 \Leftrightarrow \mathbf{C}\mathbf{L}_\mathbf{s}(\mathbf{C}\widehat{\mathbf{L}_\mathbf{s}})^{+} > 0 \tag{4.12}$$

Un choix simple de $\mathbf{C}$ égale à $\mathbf{L_s}^{+}$ permet de satisfaire cette condition.

A présent que ce formalisme est défini, nous pouvons l'appliquer à notre robot, c'est-à-dire à la caméra PTZ.

4.3 Asservissement visuel 2D de la caméra PTZ

Dans notre application, le robot est une caméra PTZ. Il possède donc 3 degrès de liberté :

- l'angle ω_X en site,
- l'angle ω_Y en azimut,
- la focale f.

Remarque : en fait, un quatrième degrè de liberté serait envisageable. Il s'agit de la mise en point du zoom. Cependant, dans notre application, l'objet se trouvant loin de la caméra, nous considérons la mise au point à l'infini, donc fixe.

Dans cette section, nous allons donc d'abord discuter de la modélisation de notre robot. Puis nous traduirons notre application en terme d'asservissement 2D visuel, ce qui nous amènera à définir la primitive à asservir (i.e. un segment). Pour l'établissement de la commande, nous discuterons de la mise en équation de la Jacobienne Image $\mathbf{L_s}$ pour un point, dans le cadre d'une caméra avec zoom. Enfin, nous l'adapterons à notre application et exprimerons la loi de commande utilisée.

4.3.1 Modélisation de la caméra PTZ

Notre robot peut être vu comme l'intégration d'une caméra avec zoom sur une platine Pan-Tilt. Nous allons donc aborder sa modélisation en ce sens, en disposant les axes de rotation par rapport à la caméra, puis en discutant du modèle optique approprié pour une caméra avec zoom.

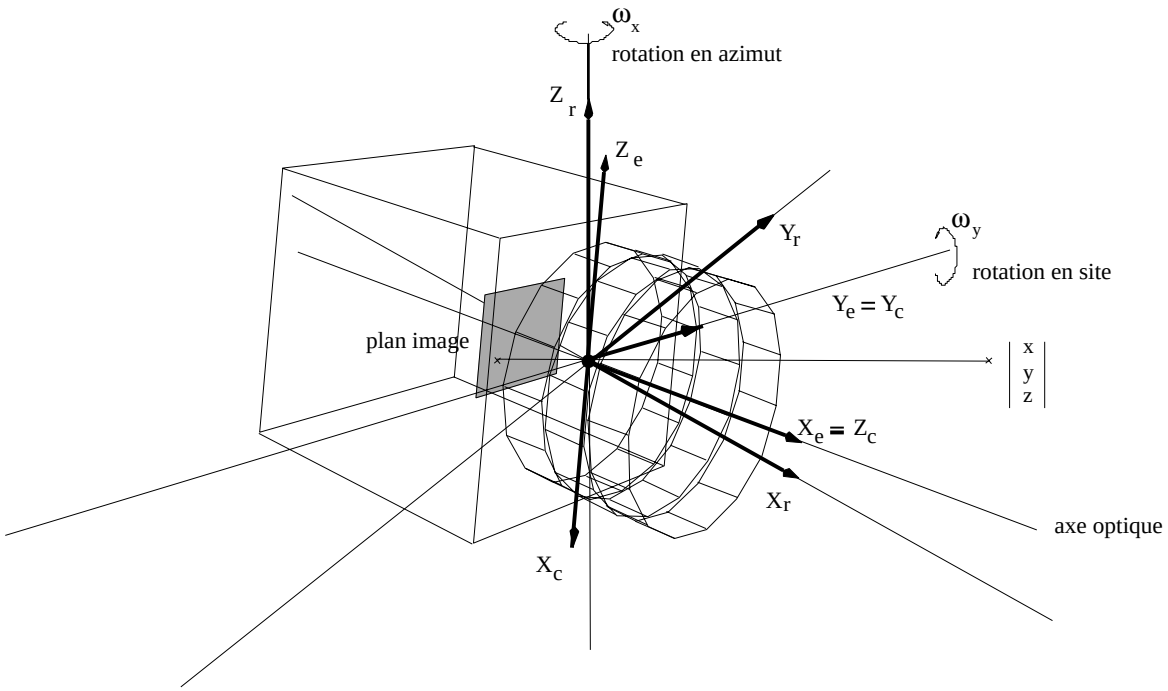


FIG. 4.3 – Modélisation du robot.

Pour la modélisation de la caméra posée sur la platine, nous avons posé l'hypothèse que l'origine du repère de la caméra est confondu avec celui du repère de la platine. Cette hypothèse est acceptable au vu de la caméra utilisée, qui dispose la caméra au centre des deux moteurs électro-magnétiques contrôlant les axes de la platine (cf. figure 4.4). Ceci permet de considérer les commandes des articulations en Pan et en Tilt confondues avec le couple (Ω_X,Ω_Y) du torseur cinématique de la caméra.

FIG. 4.4 – Vue de la caméra SONY EVI-G21

Classiquement, le modèle optique de caméra utilisée en asservissement visuel est le modèle *pinhole*. Cependant, des travaux, notamment en calibration [126, 129], ont montré que ce modèle est trop simple pour expliquer le comportement d'une caméra avec zoom. D'après ces mêmes travaux, le modèle convenant mieux est le modèle optique à lentilles épaisses. Il représente le système optique par deux plans appelés plans principaux, tel que le montre la figure 4.5. Ignorant les effets du diaphragme, ce modèle est équivalent au modèle *pinhole* sauf qu'il considère en plus un mouvement virtuel selon l'axe optique dû au changement de focale. Comme *a priori* ce mouvement pourrait avoir des conséquences sur la commande de notre caméra, il nous a semblé nécessaire d'utiliser ce modèle pour déterminer l'expression de la Jacobienne Image pour un point, puis de discuter de la justesse de ce modèle par rapport à notre application. C'est ce que nous allons faire dans la suite.

4.3.2 Expression de la Jacobienne Image pour un point

Avant de continuer, il nous faut définir quelques notations (cf. schéma 4.5) :

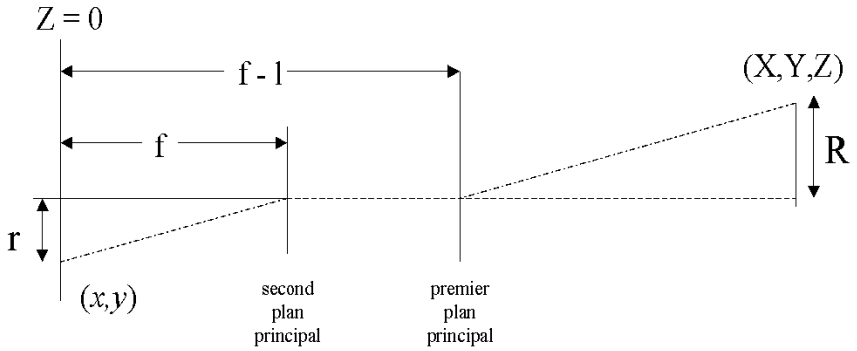


FIG. 4.5 – Modèle d'une caméra avec zoom : modèle optique à lentilles épaisses.

- f_a et f_b sont les focales associées aux deux plans principaux du modèle optique.

- (X, Y, Z) sont les coordonnées d'un point P de l'espace et (x, y) celles de son projeté p dans le plan image ($Z = 0$).

- f est bien entendue la focale du système (c'est-à-dire la distance entre le second plan principal et le plan image) et l est la distance entre les plans principaux (à noter que l est défini comme négatif, si le second plan est plus proche que le premier). La distance l est définie par :

$$l = f_a + f_b - \frac{f_a f_b}{f} \tag{4.13}$$

Avec ces notations, nous allons chercher l'expression de la Jacobienne Image $\mathbf{L}_{\mathbf{s}_p}$ qui relie les variations $\dot{\mathbf{s}_p} = (\dot{x}, \dot{y})^T$ du point p au torseur cinématique $\mathbf{T}_c = (V_X, V_Y, V_Z, \Omega_X, \Omega_Y, \Omega_Z, \dot{f})^T$ de la caméra. Comme nous le voyons, nous avons augmenté la taille du vecteur cinématique par rapport à sa définition donnée précédemment, afin de tenir compte de la variation de la focale ; dans un esprit de simplification, nous conservons cependant la même notation pour le torseur ainsi que pour la pose $\mathbf{r}$ (à laquelle est donc ajoutée f).

Nous avons donc :

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}_p}{\partial \mathbf{r}} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} + \frac{\partial \mathbf{s}}{\partial t} = \mathbf{L}_{\mathbf{s}_p} \mathbf{T}_c + \frac{\partial \mathbf{s}_p}{\partial t} \tag{4.14}$$

Pour déterminer cette matrice, nous suivons le même raisonnement que dans [68]. Il s'agit d'un raisonnement similaire à ceux que l'on trouve classiquement en asservissement visuel, sauf qu'il tient compte du changement de la focale.

Il consiste d'abord à exprimer la vélocité $\frac{d}{dt}\mathbf{P}$ du point P (en considérant celui-ci immobile : $\frac{\partial \mathbf{s}_p}{\partial t} = 0$) par rapport à la caméra, tel que :

$$\frac{d}{dt}\mathbf{P} = -\mathbf{t} - \mathbf{\Omega} \wedge \mathbf{r} \tag{4.15}$$

avec $\mathbf{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^T$ et $\mathbf{t} = (V_X, V_Y, V_Z)^T$; ce qui en composantes, donne :

$$\begin{aligned} \dot{X} &= -V_X - \Omega_Y Z + \Omega_Z Y \\ \dot{Y} &= -V_Y - \Omega_Z X + \Omega_X Z \\ \dot{Z} &= -V_X Z - \Omega_X Y + \Omega_Y X \end{aligned} \tag{4.16}$$

En introduisant dans ces équations la projection perspective, définie par :

$$x = \frac{Xf}{Z+l-f}; y = \frac{Yf}{Z+l-f} \tag{4.17}$$

la vitesse $(\dot{x}, \dot{y})^T$ du point p est alors obtenue :

$$\begin{aligned} \dot{x} = \tfrac{-V_X f + xV_Z}{Z+l-f} + \tfrac{1}{f}\left(\Omega_X xy - \Omega_Y\left(x^2 + \tfrac{Zf^2}{Z+l-f}\right) + \Omega_Z fy\right) \\ + \tfrac{\dot{f}x}{f}\left(1 + \tfrac{f^2 - f_a f_b}{f(Z+l-f)}\right) \\ \dot{y} = \tfrac{-V_Y f + yV_Z}{Z+l-f} + \tfrac{1}{f}\left(\Omega_X\left(y^2 + \tfrac{Zf^2}{Z+l-f}\right) - \Omega_Y xy + \Omega_Z fx\right) \\ + \tfrac{\dot{f}y}{f}\left(1 + \tfrac{f^2 - f_a f_b}{f(Z+l-f)}\right) \end{aligned} \tag{4.18}$$

Ce qui peut s'exprimer sous la forme matricielle :

$$\begin{aligned} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{f}{Z+l-f} & 0 & \frac{x}{Z+l-f} \\ 0 & -\frac{f}{Z+l-f} & \frac{y}{Z+l-f} \end{pmatrix} \\ \begin{pmatrix} \frac{xy}{f} & -\frac{1}{f}(x^2 + \frac{Zf^2}{Z+l-f}) & y & \frac{x}{f}\left(1 + \frac{f^2 - f_a f_b}{f(Z+l-f)}\right) \\ \frac{1}{f}(y^2 + \frac{Zf^2}{Z+l-f}) & -\frac{xy}{f} & x & \frac{y}{f}\left(1 + \frac{f^2 - f_a f_b}{f(Z+l-f)}\right) \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \\ = \mathbf{L}_{\mathbf{s}_p} \mathbf{T}_c \end{aligned} \tag{4.19}$$

Nous avons donc exprimé la Jacobienne Image en tenant compte d'un modèle à lentilles épaisses de la caméra. Dans cette matrice, la différence avec celle que nous pourrions déterminer avec le modèle *pinhole* se situe au niveau des fractions $\frac{1}{Z+l-f}$, $\frac{f^2 - f_a f_b}{f(Z+l-f)}$ et $\frac{Z}{Z+l-f}$.

Considérons ces termes par rapport à l'application. Pour cela, étudions leurs variations suivant Z et f. Celles-ci sont données par les figures 4.6, 4.7 et 4.8. Nous pouvons constater que les deux premiers termes tendent très vite vers 0 et le troisième vers 1 lorsque la distance Z augmente ; ceci quelque soit la focale située dans une plage comprenant celle correspondant à la caméra que nous utilisons.

Or, dans notre domaine d'application, nous considérons que l'objet suivi est à une distance relativement éloignée de la caméra. Sur autoroute, nous avons vu à la fin du chapitre 3 que les distances de sécurité minimales sont supérieures à $40m$. A cette distance, les écarts des fractions par rapport à leurs limites (pour Z à l'infini) peuvent être considérés comme négligeables. Aussi il est possible de les remplacer par leurs limites dans l'expression de la Jacobienne Image.

Ceci revient à considérer que le modèle optique du zoom, pour notre application, se simplifie à un modèle *pinhole*. Ceci vérifie l'observation faîte dans [126] : «Pour la

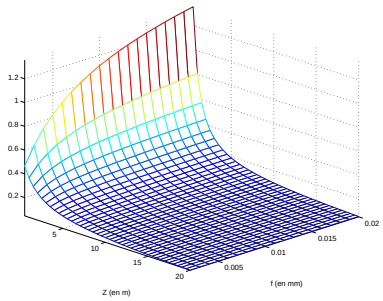


FIG. 4.6 – Variation de $\frac{1}{Z+l-f}$ en fonction de la distance et de la focale.

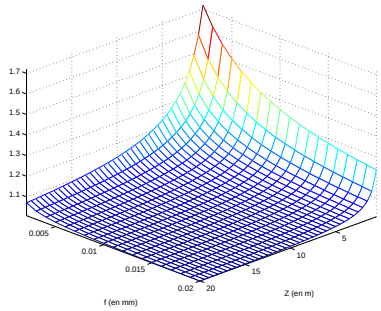


FIG. 4.7 – Variation de $\frac{f^2-f_a f_b}{f(Z+l-f)}$ en fonction de la distance et de la focale.

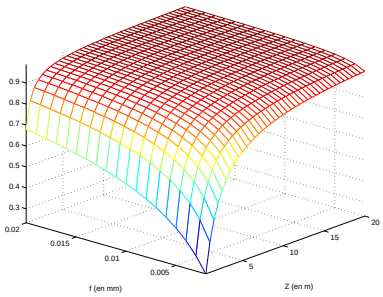


FIG. 4.8 – Variation de $\frac{Z}{Z+l-f}$ en fonction de la distance et de la focale.

plupart des applications, le modèle sténopé pourra être suffisant si l'interprétation qu'on en fait se réfère au modèle épais et que la distance entre l'objet et l'image ne soit plus considérée constante».

Ainsi, après passage dans un repère image (lié au tableau des pixels) classique, l'expression de la relation entre la vitesse du point p et le torseur cinématique augmentée prend la forme suivante :

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -\frac{f_u}{Z} & 0 & \frac{u}{Z} & \frac{uv}{f_v} & -(f_u+\frac{u^2}{f_u}) & v\frac{f_u}{f_v} & \frac{u}{f} \\ 0 & -\frac{f_v}{Z} & \frac{v}{Z} & (f_v+\frac{v^2}{f_v}) & -\frac{uv}{f_u} & -u\frac{f_v}{f_u} & \frac{v}{f} \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \tag{4.20}$$

où, si on considère les facteurs d'échelle k_u et k_v (qui représentent les dimensions d'un pixel), $k_u f_u = f$, $k_v f_v = f$, $k_u u = x$ et $k_v v = y$.

Cette matrice obtenue, nous pouvons passer à l'asservissement à proprement parler de notre caméra PTZ.

4.3.3 Asservissement de la caméra PTZ sur le motif suivi

Comme nous l'avons dit dans la partie consacrée au formalisme de la fonction de tâche, l'asservissement de la caméra PTZ demande que nous choisissions d'abord la primitive image à asservir, ainsi que la consigne, en fonction de la tâche désirée. Ensuite, nous devons exprimer la Jacobienne Image associée. Enfin, la loi de commande peut être définie.

Nous allons développer ces différents points dans les paragraphes qui suivent.

Choix de la primitive et de la consigne

Nous avons déjà signalé que notre caméra PTZ a 3 degrés de liberté. Une solution simple est donc de l'asservir sur un segment (non parallèle à l'axe optique), composé de 2 points. Nous obtenons ainsi suffisamment de paramêtres pour l'asservir.

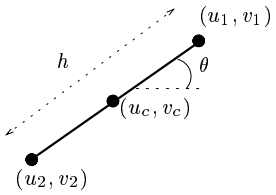


FIG. 4.9 – Repérage du segment centré dans l'image.

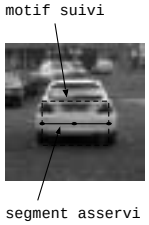


FIG. 4.10 – Position du segment asservi sur le motif suivi

Ce segment doit être défini sur le motif suivi afin de réaliser la tâche voulue. Nous rappelons que celle-ci est de centrer le motif suivi, en lui conservant une résolution quasi-constante. Comme nous l'avons vu dans le chapitre précédent, ce motif est de forme rectangulaire. Aussi, si nous plaçons le segment sur le motif comme indiqué dans la figure 4.10.

La tâche à effectuer se traduit donc par :

– centrer le segment dans l'image

– conserver constante la taille h du segment

Dans le formalisme de la fonction de tâche, ceci s'écrit sous la forme :

$$\mathbf{s} = \begin{pmatrix} u_c \\ v_c \\ h \end{pmatrix} \tag{4.21}$$

avec la consigne suivante :

$$\mathbf{s}^* = \begin{pmatrix} 0 \\ 0 \\ h^* \end{pmatrix} \tag{4.22}$$

où h^* est la taille désirée du segment (donc du motif) dans l'image.

Etablissement de la matrice jacobienne associée

Avec un repérage centré de ce segment (cf. figure 4.9), nous obtenons la relation suivante (cf. annexe E) :

$$\begin{pmatrix} \dot{u_c} \\ \dot{v_c} \\ \dot{h} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\delta_2 f_u & 0 & \delta_2 u_c - \delta_1 \frac{h cos\theta}{4} & \frac{u_c v_c}{f_v} + h^2 \frac{cos\theta sin\theta}{4 f_v} \\ 0 & -\delta_2 f_v & \delta_2 v_c - \delta_1 \frac{h sin\theta}{4} & f_v + \frac{{v_c}^2}{f_v} + \frac{h^2 sin^2\theta}{4 f_v} \\ \delta_1 f_u cos\theta & \delta_1 f_v sin\theta & \delta_2 h - \delta_1 (u_c cos\theta + v_c sin\theta) & \frac{h(u_c cos\theta sin\theta + v_c(1+sin^2\theta))}{f_v} \\ -\delta_1 \frac{f_u sin\theta}{h} & \delta_1 \frac{f_v cos\theta}{h} & \delta_1 \frac{(u_c sin\theta - v_c cos\theta)}{h} & \frac{-u_c sin^2\theta + v_c cos\theta sin\theta}{f_v} \end{pmatrix}$$

$$\begin{pmatrix} -(f_u + \frac{{u_c}^2}{f_u} + \frac{h^2 cos^2\theta}{4 f_u}) & v_c \frac{f_u}{f_v} & \frac{u_c}{f} \\ -\frac{u_c v_c}{f_u} - h^2 \frac{cos\theta sin\theta}{4 f_u} & -u_c \frac{f_v}{f_u} & \frac{v_c}{f} \\ -\frac{h}{f_u}(u_c(1+cos^2\theta) + v_c cos\theta sin\theta) & h cos\theta sin\theta (\frac{f_u}{f_v} - \frac{f_v}{f_u}) & \frac{h}{f} \\ \frac{1}{f_u}(u_c cos\theta sin\theta - v_c cos^2\theta) & -\frac{f_u}{f_v} sin^2\theta - \frac{f_v}{f_u} cos^2\theta & 0 \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \quad (4.23)$$

En nous limitant au vecteur de commande $\mathbf{T}_c$, défini par $\mathbf{T}_c = (\Omega_x, \Omega_y, \dot{f})^T$, nous pouvons donc exprimer la Jacobienne Image pour notre segment :

$$\mathbf{L}_s = \begin{pmatrix} \frac{u_c v_c}{f_v} + h^2 \frac{cos\theta sin\theta}{4 f_v} & -(f_u + \frac{{u_c}^2}{f_u} + \frac{h^2 cos^2\theta}{4 f_u}) & \frac{u_c}{f} \\ f_v + \frac{{v_c}^2}{f_v} + \frac{h^2 sin^2\theta}{4 f_v} & -\frac{u_c v_c}{f_u} - h^2 \frac{cos\theta sin\theta}{4 f_u} & \frac{v_c}{f} \\ \frac{h(u_c cos\theta sin\theta + v_c(1+sin^2\theta))}{f_v} & -\frac{h}{f_u}(u_c(1+cos^2\theta) + v_c cos\theta sin\theta) & \frac{h}{f} \end{pmatrix} \quad (4.24)$$

Comme nous l'avons dit dans la section 4.2.2, il est d'usage en asservissement d'utiliser une estimation de cette matrice pour construire la loi de commande, pour peu que la convergence de la commande soit vérifiée par après (soit expérimentalement, soit théoriquement). Cette simplification facilite ensuite son inversion.

Nous allons donc formuler quelques hypothèses dans ce but.

- La première est courante en asservissement visuel 2D. Elle consiste à considérer que l'asservissement se fait dans un voisinage proche de l'équilibre : $\mathbf{s} \simeq \mathbf{s}^*$. D'où $(u_c, v_c, h) \simeq (0, 0, h^*)$.

- La seconde considère que l'on prends le segment de sorte que $\theta = 0$. Ceci se justifie d'abord parce que, dans des conditions réelles de conduite, cette valeur varie dans un voisinage proche de zéro. Ensuite, cela s'explique par modèle de mouvement utilisé dans l'algorithme de suivi de motif (cf. la section 3.2.4). En effet, celui-ci ne tient pas compte des rotations du motif. Donc, comme nous déduisons la position du segment sur celle du motif, l'angle θ ne peut varier au cours du temps.

- La troisième consiste à supposer que $f_{£} \gg \frac{{h^*}^2}{f_{£}}$ (avec £ = u ou v). Cette hypothèse est réaliste car les focales en pixels de notre caméra sont supérieures à $1000 pixels$ et h^* est choisie de l'ordre de la centaine. Ce qui fait un rapport entre ces deux

valeurs de l'ordre de 0.1, qui passe après l'élévation au carré à 0.01.

Sous ces hypothèses, nous avons donc :

$$\widehat{\mathbf{L}_{s^*}} = \begin{pmatrix} 0 & -f_u & 0 \\ f_v & 0 & 0 \\ 0 & 0 & \frac{h^*}{f} \end{pmatrix} \tag{4.25}$$

Nous pouvons remarquer que, du fait des hypothèses pré-citées, les termes de couplage entre les articulations de notre «robot» sont négligés. La commande sera donc totalement découplée, notamment entre le zoom et la platine. Ce découplage est essentiellement du à notre volonté de centrer le motif. Ce découplage a été aussi utilisé par Flavien Huynh [101] dans ses travaux à propos d'une configuration similaire à la nôtre (où l'objectif est simplement d'agrandir la taille du motif suivi d'un facteur 3).

Si ce n'était pas le cas, nous aurions dû adopter une autre approche consistant à corriger l'effet du zoom sur la position des points non-centrés. Dans la littérature, deux approches permettent de gérer cette situation. La première [97] propose de corriger cette influence en la supposant proportionelle à l'écart entre la position de la focale et celle voulue. La seconde approche [65] testée en simulation par Espiau, consiste à utiliser le formalisme de la tâche secondaire pour réguler la focale en fonction du périmêtre d'un motif polygonal.

4.3.4 Expression et application de la loi de commande

Rappelons que, d'après les équations 4.6 et 4.9 , une loi de commande proportionnelle peut s'écrire sous la forme :

$$\mathbf{T}_c = \widehat{\mathbf{L}_{s^*}}^{+}(-\lambda.(\mathbf{s}-\mathbf{s}^*) - \frac{\widehat{\partial \mathbf{s}}}{\partial t}) \tag{4.26}$$

où $\widehat{\mathbf{L}_{s^*}}^{+}$ est la pseudo-inverse de l'estimation de la Jacobienne Image, qui est trivialement :

$$\widehat{\mathbf{L}_{s^*}}^{+} = \begin{pmatrix} 0 & \frac{1}{f_v} & 0 \\ -\frac{1}{f_u} & 0 & 0 \\ 0 & 0 & \frac{f}{h^*} \end{pmatrix} \tag{4.27}$$

Afin de compléter la loi de commande, il reste à déterminer le terme $\frac{\widehat{\partial \mathbf{s}}}{\partial t}$, qui représente l'estimation de la contribution de la dynamique de la cible. Il existe différentes méthodes pour estimer ce terme.

Cependant, après simulations et expérimentations, nous avons choisi de négliger ce terme et de poser : $\frac{\widehat{\partial \mathbf{s}}}{\partial t} \simeq 0$. Ce choix s'explique par l'application considérée.

En effet, pour le suivi d'un véhicule avec une caméra PTZ, la dynamique de la cible s'avère négligeable la plupart du temps par rapport au bruit observé sur la position

du motif dans l'image. Les seuls moments où cette dynamique pourrait entraîner un décalage avec la position d'équilibre correspondent à de manœuvres de changements de voies du véhicule obstacle ou du véhicule expérimental, ou lorsque la vitesse relative entre le véhicule obstacle et le véhicule expérimental est importante (lors d'un dépassement, par exemple).

Dans les deux cas, il s'agit souvent de manœuvres de courtes durées. Les erreurs entre la position d'équilibre et la position de la cible dans l'image restent ainsi faibles. Sitôt la manœuvre finie, ces écarts sont rapidement résorbés.

Si la manœuvre est de plus longue durée :

- soit le suivi du véhicule avec la caméra PTZ n'est pas utile. Ce serait, par exemple, le cas d'un véhicule s'éloignant à grande vitesse ou quittant l'autoroute (via une sortie). Dans les deux cas, le véhicule suivi quitte la zone de perception du capteur (la caméra PTZ arrive en butée),
- soit la situation nécessite plutôt une réaction au niveau de la commande du véhicule expérimental, donc une réaction du conducteur. C'est le cas d'un véhicule s'approchant à grande vitesse.

Afin de valider notre approche, nous avons réalisé plusieurs simulations et expérimentations. Celles-ci sont développées dans les sections suivantes. Tout d'abord, nous présentons plusieurs résultats de simulations qui nous ont permis de valider la convergence de notre loi de commande. Puis, nous avons testé l'influence des variations de focale d'une caméra sur l'estimation de l'état d'un objet suivi. Enfin, nous présentons quelques essais de suivi de véhicule avec une caméra PTZ réalisés en situations réelles de conduite.

4.4 Résultats de simulations

Afin de valider l'approche développée, nous avons réalisé un simulateur intégrant :

- une cible carrée de 4 points, de largeur et de longueur égales à $1m$. Comme pour un véhicule, elle est mise en contact perpendiculairement à un plan représentant la route,
- la caméra PTZ. Elle est placée à une hauteur de $1.5m$ par rapport au plan-route. Nous avons tenu compte d'un modèle réaliste de la caméra EVI-G21 utilisée, c'est-à-dire notamment des limitations et des quantifications en vitesses du zoom. Les zooms des caméras sont en effet des mécanismes complexes ne permettant pas d'obtenir une gamme et des amplitudes de vitesses élevées.

A l'aide de ce simulateur, nous avons réalisé diverses simulations. Nous en présentons ici quelques échantillons, caractéristiques des autres simulations obtenues. Pour ceux-ci, nous avons placé la cible à une distance de $40m$ par rapport à l'origine (c'est-à-dire par rapport au référentiel lié au véhicule).

La consigne s^* est fixée à $(0, 0, 100)$.

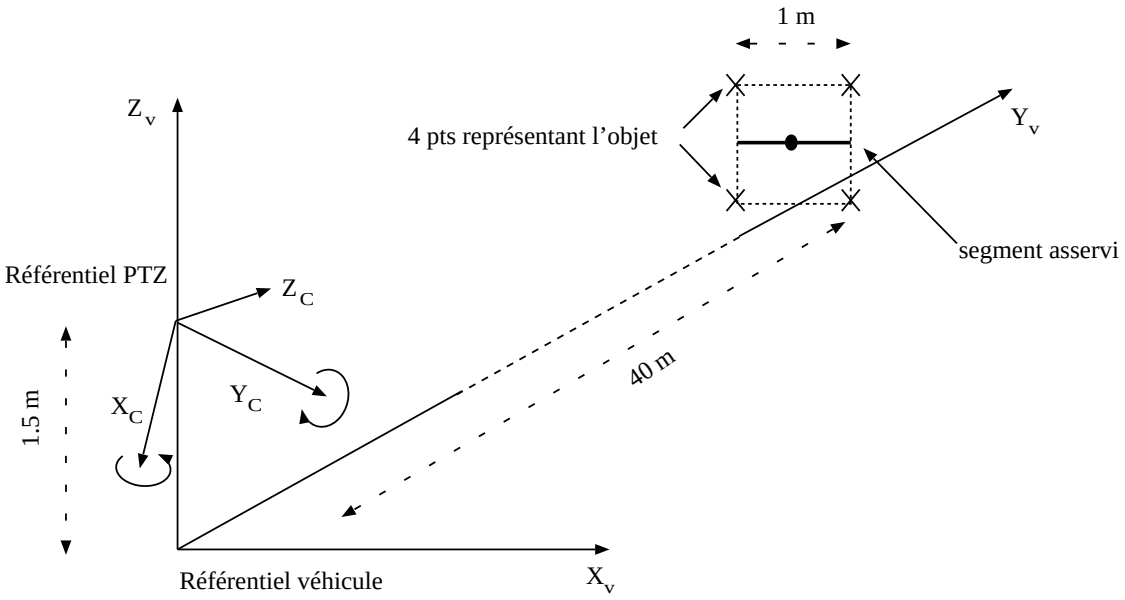


FIG. 4.11 – Schématisation du contexte simulé.

Les figures 4.12 et 4.13 démontrent la convergence et la robustesse de la loi de commande choisie, respectivement par rapport au bruit et par rapport à l'éloignement de la position finale désirée.

Pour chacune de ces figures, les courbes de la première colonne montrent l'évolution des 4 points dans l'image dans les différents cas considérés. Les courbes de la seconde, l'évolution des erreurs en pixels sur les coordonnées du centre du segment. La dernière colonne donne l'évolution des erreurs en pixels sur la longueur du segment.

Pour la figure 4.12, des bruits aléatoires de type Gaussien à différentes intensités ont été introduits à tous les niveaux du système : entrées et sorties de la caméra PTZ, vibrations du véhicule expérimental, erreurs de mesures image,... Les angles initiaux de la caméra sont de l'ordre de 5° par rapport à l'orientation du véhicule. La focale initiale est de 1000 pixels.

Les figures de la première ligne correspondent à un système sans bruits. Celles de la deuxième ligne à des bruits d'amplitudes réalistes, déterminées selon l'expérience acquise au laboratoire. Les mesures image sont bruitées avec un bruit Gaussien d'écart-type $\sigma_{\text{image}} = 0.5$ pixels qui est directement rajouté aux coordonnées des 4 points. Nous avons tenu compte de vibrations dans les orientations du véhicule expérimental selon un bruit Gaussien d'écart-type $\sigma_{\text{vib.}} = 0.01$ rad. (soit environ 1 deg.). Au niveau du vecteur commande, nous avons admis des erreurs aléatoires de l'ordre de 5% ($=\text{e}_{\mathbf{T}_c}$) pour les vitesses en focale et en angles.

Pour la dernière ligne, ces valeurs ont été accrues exagéremment ($\sigma_{\text{image}} = 2.$ pixels, $\sigma_{\text{vib.}} = 0.03$ rad.,$\text{e}_{\mathbf{T}_c} = 15\%$). Il s'agit d'une situation irréaliste ; même dans ce cas, la loi de commande converge.

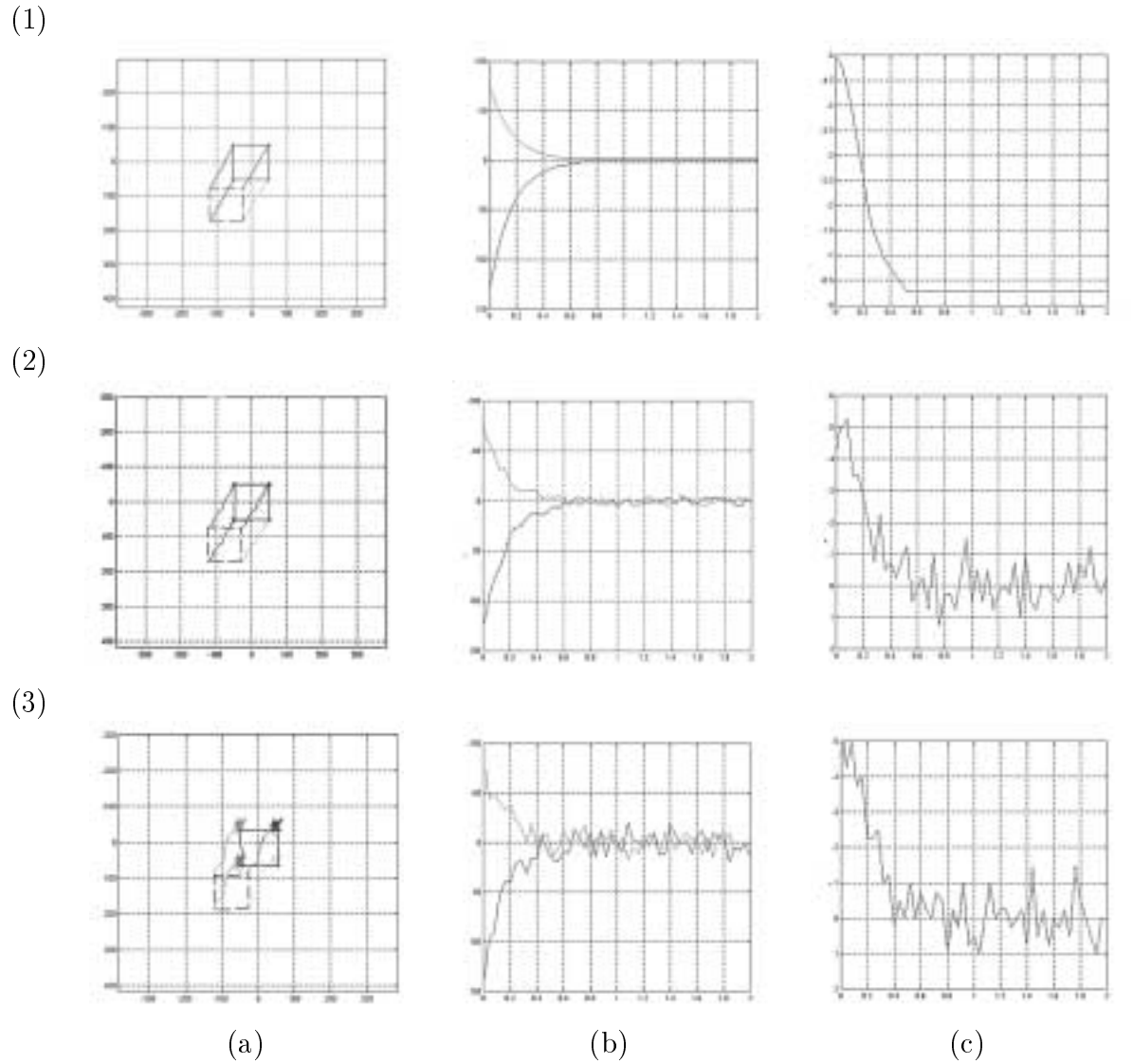


FIG. 4.12 – Robustesse de la loi de commande par rapport au bruit : (1) sans bruit, (2) avec un bruit moyen, (3) avec un bruit important ; (a) Evolution des 4 points (encadrant le motif à suivre ; la position initiale du motif est représentée par le quadrilatère en trait discontinu, et la position finale en trait continu) dans le plan image, (b) Evolution au cours de temps (en sec.) des erreurs en pixels sur les coordonnées du barycentre , (c) Evolution au cours du temps (en seconde) de l'erreur en pixels sur la longueur h de l'axe.

Pour tester la robustesse de la loi de commande par rapport à l'éloignement de la position initiale par rapport à la position finale, nous avons réalisé des simulations dans le même esprit, en considérant des situations de plus en plus éloignées, jusqu'à nous placer dans une situation irréaliste (où la position initiale de l'objet par rapport à la caméra PTZ le situe en dehors de l'image). Nous remarquons que la loi de commande converge. Nous pouvons remarquer cependant que la convergence au niveau de la longueur du segment n'est plus exponentielle lorsque le facteur d'échelle entre la position initiale du

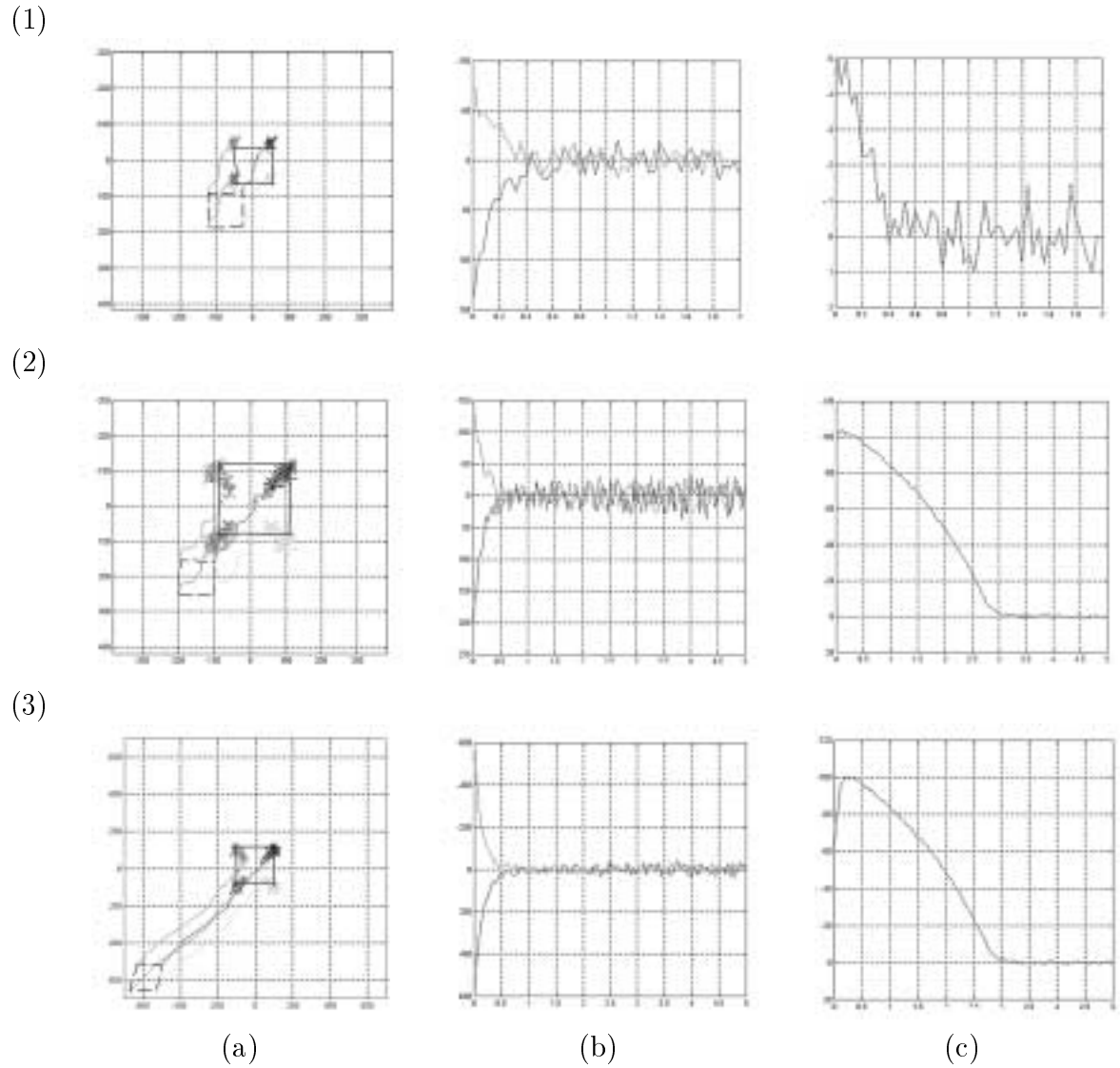


FIG. 4.13 – Robustesse de la loi de commande par rapport à la position initiale de l'objet par rapport à la caméra : (1) proche de la position finale désirée, (2) loin, (3) très loin (en dehors de l'image) ; (a) Evolution au cours du temps des 4 points (représentant le motif à suivre) dans le plan image, (b) Evolution au cours du temps (en sec) des erreurs en pixels sur les coordonnées du barycentre, (c) Evolution au cours du temps (en sec) de l'erreur en pixels sur la longueur h de l'axe.

segment et sa position finale est importante : ce phénomène est du à la saturation en vitesse du changement de focale.

D'une manière générale, ce simulateur nous a permis de valider la loi de commande choisie par rapport à l'application. Dans les nombreux cas que nous avons testé, celle-ci converge.

4.5 Influence des variations du zoom

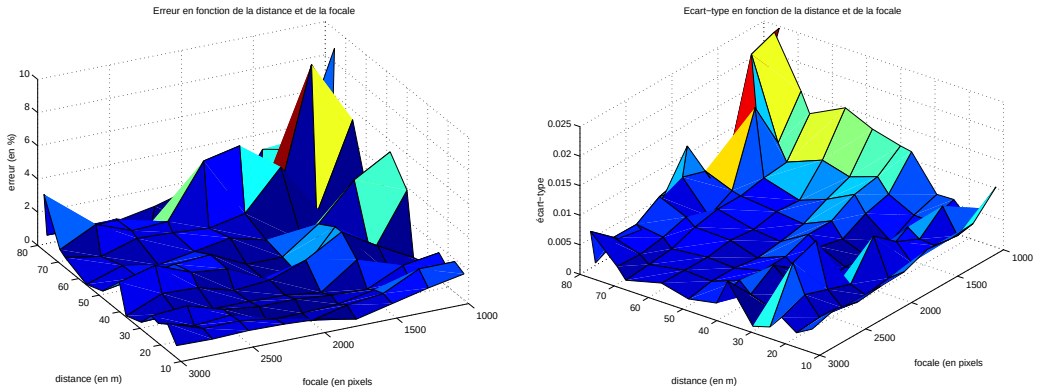
Dans la section 4.3, nous avons abordé plusieurs fois la problèmatique de la modèlisation du zoom. Nous avons constaté que la caméra pouvait être approximée à un modèle *pinhole*. Pour cela, il ne faut plus considérer constante la distance entre l'objet et l'image. Pour la commande, il s'est avéré que cette considération est négligeable au vu de l'application visée.

Cependant, il nous est apparu important de vérifier l'influence des variations du zoom sur l'estimation de la distance et de la vitesse relatives entre la caméra pilotée et un objet en mouvement.

4.5.1 Influence du zoom sur l'estimation de la distance

Afin d'estimer l'influence du zoom dans le contexte de notre application, nous avons placé une cible connue à différentes distances de la caméra avec zoom. L'algorithme de suivi a été alors initialisé sur un motif texturé de la même taille que celui que l'on suit sur un véhicule (1x1.5 m). L'estimation de la distance a été obtenue via l'algorithme itérative d'estimation de pose de Dementhon, décrit dans [151].

Nous avons testé une plage de distance allant de 10 à 80 m et de focale de 1000 à 3000 pixels. Cette plage correspond à celle de notre application. La mise au point de la caméra a été reglée alors à l'infini.



(a) Erreur (b) Ecart-type

FIG. 4.14 – Influence du zoom sur l'estimation de la pose : (a) distance estimée, (b) erreur sur la distance estimée et (c) écart-type (sur plusieurs mesures) en fonction de la distance et de la focale.

La figure 4.14 illustre les résultats obtenus en terme de d'erreur moyenne sur la distance reconstruite et d'écart-type.

Nous pouvons remarquer que les distances sont bien reconstruites (avec une erreur inférieure à 2 %) sauf lorsque les distances augmentent et que la focale reste faible. L'er-

reur moyenne monte jusqu'à 10 % à 70 m avec une focale à 1000 $pixels$. Ceci s'explique naturellement par la diminution de la taille du motif dans l'image.

De même, l'écart-type sur les mesures réalisées à ces distances et ces focales augmente de façon significative. Ceci signifie que l'algorithme de suivi est alors moins précis, du fait de la plus faible résolution du motif dans l'image.

4.5.2 Influence du zoom sur l'estimation de la vitesse

Pour cette deuxième évaluation, nous avons installé une cible, représentant une photographie d'une vue arrière de véhicule à une échelle 1/10, sur l'effecteur d'un robot cartésien (cf. figure 4.15).

La vitesse de translation maximum de ce robot est de $1m.s^{-1}$. Ce qui, par le rapport d'échelle entre la photographie et le véhicule réel, représente une vitesse maximale de $36km.h^{-1}$ pour un véhicule réel. Durant nos expérimentations, la distance initiale entre la cible et la caméra PTZ est d'approximativement $1m$, ce qui représente une distance «réelle» de $10m$. Le débattement maximum du robot est de $1.5m$; Dans nos expérimentations, nous l'avons limité à environ $1.4m$.

Un motif quadrilatère (dont la géométrie est connue) est suivi sur cette vue arrière de véhicule avec l'algorithme décrit dans le chapitre précédent.

FIG. 4.15 – La caméra EVI (à droite) et la cible (à gauche) installée sur l'effecteur du robot cartésien

Les expérimentations ont donc consisté à faire varier la distance entre $1m$ et $2.4m$ à vitesse constante. Sur cette plage, la mise au point automatique de la caméra a été utilisée afin d'éviter les erreurs d'estimation due à la défocalisation. Différentes vitesses ont été testées, puis comparées avec les vitesses reconstruites à l'aide d'un filtre de Kalman.

Le tableau 4.16 illustre les résultats obtenus. Il donne les vitesses moyennes estimées via caméra à focale fixe et via la caméra à focale variable, sur l'ensemble de la distance parcourue (entre $1m$ et $2.4m$), ainsi que sur trois plages de distances différentes : entre $1m$ et $1.5m$, entre $1.5m$ et $2m$, et entre $2m$ et $2.4m$. Les chiffres entre parenthèses représentent les variances associées aux différentes moyennes.

D'après ce tableau, nous pouvons constater que la variation de la focale n'a que peu d'influences sur l'estimation de la vitesse. Les vitesses estimées avec la caméra à focale

Vitesse "robot" $m.s^{-1}$	Vitesses estimées $m.s^{-1}$							
	[1-2.4] m		[1-1.5] m		[1.5-2.0] m		[2.0-2.4] m	
	c.f.f.	c.f.p.	c.f.f.	c.f.p.	c.f.f.	c.f.p.	c.f.f.	c.f.p.
0.1	0.11 (0.04)	0.10 (0.03)	0.12 (0.03)	0.12 (0.04)	0.1 (0.03)	0.1 (0.02)	0.12 (0.03)	0.09 (0.017)
0.2	0.21 (0.05)	0.20 (0.04)	0.24 (0.04)	0.20 (0.06)	0.23 (0.06)	0.22 (0.03)	0.18 (0.02)	0.19 (0.017)
0.4	0.46 (0.11)	0.44 (0.06)	0.44 (0.05)	0.49 (0.06)	0.46 (0.03)	0.43 (0.03)	0.49 (0.04)	0.39 (0.03)
0.6	0.62 (0.17)	0.57 (0.1)	0.64 (0.14)	0.70 (0.14)	0.73 (0.01)	0.68 (0.08)	0.65 (0.2)	0.64 (0.03)
1.0	1.06 (0.03)	0.99 (0.03)	0.91 (0.06)	0.89 (0.07)	1.09 (0.03)	1.06 (0.03)	1.18 (0.08)	1.03 (0.017)

FIG. 4.16 – Vitesses moyennes estimées via la caméra à focale fixe ou variable (c.f.f. = caméra à focale fixée et c.f.p. = caméra à focale pilotée) ; les variances associées à ces moyennes sont mises entre parenthèses.

variable sont similaires à celles estimées à l'aide de la caméra à focale fixe. Nous pouvons observer, d'après les écart-types, que l'estimation de la vitesse sur la dernière plage de distance est légéremment plus stable avec la caméra à focale variable. Cette observation est en accord avec le fait que l'estimation de la distance est moins bruitée avec la caméra à focale pilotée qu'avec la caméra à focale fixe, lorsque la taille du motif dans l'image diminue.

4.5.3 Discussion

D'une manière générale, nous pouvons conclure de ces expérimentations l'approximation de la caméra avec zoom à un modèle *pinhole* est aussi valide pour l'estimation de la distance.

De plus, les résultats obtenus sont meilleurs avec une caméra à focale pilotée qu'avec une caméra à focale fixe. Ceci s'explique par le fait que le motif suivi conserve une taille quasi-constante dans l'image, donc une résolution optimale.

Cependant, il nous faut signaler que ces résultats sont dépendants de la caméra utilisée. Il n'est pas certain d'obtenir des résultats similaires avec d'autres caméras avec zoom.

4.6 Résultats en situation réelle de conduite

Dans cette dernière section, nous donnons quelques résultats obtenus en situations réelles de conduite. Les séquences présentées valident notre approche de suivi de véhicule

par la caméra PTZ.

4.6.1 Architecture matérielle

Elles ont été obtenues en intégrant dans le véhicule VELAC du laboratoire, deux caméras : une grand angle à focale fixe et la caméra PTZ. Ces deux caméras sont placées au niveau du rétroviseur (cf. figure 4.17). Ce montage permet d'illustrer le champ de perception qui pourra être obtenu avec le capteur proposé dans le chapitre 1 lors d'un suivi de véhicule par la caméra PTZ.

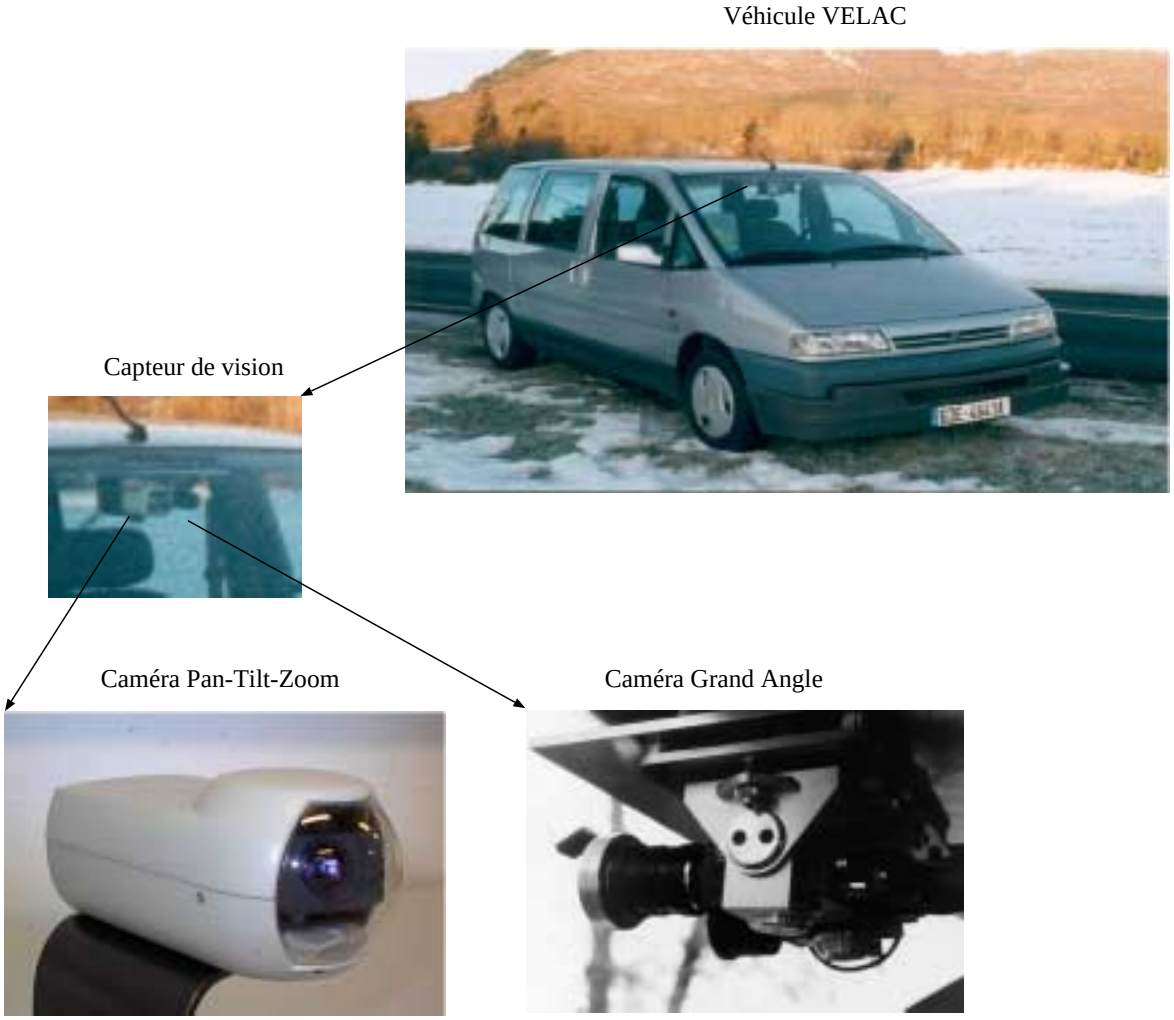


FIG. 4.17 – Capteur de vision embarqué dans le véhicule VELAC.

La caméra grand angle est une caméra analogique. Sa focale est fixée à $4.5mm$. La caméra PTZ est une caméra SONY EVI-G21. La détermination des paramètres intrinsèques de cette caméra est présentée dans l'annexe F. Le tableau ci-dessous en rappelle les caractéristiques générales :

focales	4.5 - 13.5mm (×3 zoom) 1000 - 3000 pixels
pan angle	±30°
tilt angle	±15°

L'ensemble des opérations nécessaires au suivi d'un véhicule par la caméra PTZ, c'est-à-dire le suivi du motif et le calcul de la loi de commande est assurée par un PC de type Pentium III $800MHz$ sous Linux. Cet ordinateur est alimenté par un alternateur spécifique, couplé à deux batteries et un onduleur (24V vers 220V). Cet ensemble est installé au sein du véhicule VELAC.

4.6.2 Extraits de séquences

Dans cette section, nous allons présenter quelques extraits de séquences enregistrées à bord de VELAC.

Dans les figures exposées, l'image à gauche est celle acquise par la caméra grand angle et celle à droite celle par la caméra PTZ asservie sur un motif sélectionné (manuellement) sur un véhicule, tel que l'illustre la figure 4.10. Différents types de véhicule ont été suivis avec succès.

L'asservissement de la caméra est assuré à une fréquence de $80ms$, c'est-à-dire à une image sur deux. L'essentiel de ce temps est consacré à la communication entre le PC et la caméra via une liaison série RS-232, limitée à 9600 bauds.

L'image acquise par la caméra PTZ a une taille de 728x576 pixels. La consigne en taille pour le segment est de 64 pixels. Ainsi le zoom est hors butées, donc actif, pour un véhicule suivi à une distance comprise entre 25 et 70 $m.$ environ.

Le premier extrait (cf. figure 4.18) permet d'illustrer l'action du zoom en situation réelle de conduite. Lorsque notre véhicule s'approche du véhicule suivi, l'image de celui-ci est conservée à une taille constante et supérieure à celle acquise via la caméra grand angle.

L'extrait de la figure 4.19 montre un véhicule suivi de type «Polo». Cet extrait permet d'illustrer à nouveau l'effet de l'asservissement du zoom, ainsi que des angles en site et azimut. De plus, nous pouvons constater la robustesse de l'algorithme aux changements globaux de luminosité lors d'un passage d'un pont.

Dans cet extrait, nous pouvons constater aussi qu'il y a une erreur de suivi de la caméra en angle azimuth. Cela se traduit dans l'image par un décalage latéral par rapport au centre de l'ordre d'une trentaine de pixels au maximum. Pour un véhicule se situant aux environs de 40 m, cela représente une erreur en angle inférieure à 0.5°. Du fait que nous avons choisi un gain assez faible afin d'éviter des oscillations sur la commande, cette faible erreur en angle situe la consigne en vitesse angulaire en dessous du point de fonctionnement de l'actionneur. Par ailleurs, cette erreur est accrue dans la séquence présentée (en particulier dans la dernière image) par le fait que nous avons négligé la contribution de la dynamique de la cible.

L'extrait de la figure 4.20 illustre la généricité de l'algorithme de suivi choisi puisqu'il permet de suivre un véhicule autre qu'une voiture. De plus, cet extrait montre que la ca-

FIG. 4.18 – Extraits de séquences de suivi d'une «406 Peugeot» avec la caméra PTZ.

FIG. 4.19 – Extraits de séquences de suivi d'une «Polo» avec la caméra PTZ.

méra PTZ permet aussi d'accroître le champ de perception visuelle latérale, notamment lorsque le véhicule équipé double le véhicule suivi. Lors de cette manœuvre, on voit dans l'image du bas que le véhicule suivi est sur le point de disparaître du champ de vision de la caméra fixe, alors que la platine Pan-Tilt permet de le conserver dans celui de la caméra PTZ.

FIG. 4.20 – Extraits de séquences de suivi d'un «camion» avec la caméra PTZ.

FIG. 4.21 – Extraits de séquences de suivi d'une «R5 blanche» avec la caméra PTZ.

Le dernier extrait (cf. figure 4.21) démontre enfin que la caméra grand angle reste indispensable. En effet, dans cette séquence où la caméra PTZ est focalisée sur une Renault 5 blanche, on peut remarquer que la caméra grand angle permet de visualiser la présence d'autres véhicules apparaissant sur la voie de gauche. Ceux-ci pourraient très bien venir s'intercaler entre le véhicule équipé et le véhicule suivi. La caméra grand angle permettrait donc de détecter une telle manœuvre.

Tous ces résultats valident notre capteur et notre approche par rapport à diverses situations réelles de conduite. Dans toutes ces situations, l'algorithme de suivi ainsi que l'asservissement de la caméra PTZ permettent de suivre un véhicule dans l'image, avec une résolution supérieure à celle fournie par la caméra grand angle.

4.7 Discussion

Au travers de ces différents résultats, nous pouvons constater que le système proposé permet de suivre avec la caméra PTZ, tout type de véhicule à une taille, donc à une

résolution, quasi-constante.

Les essais en laboratoire montrent que l'asservissement du zoom n'apporte qu'un gain relativement faible dans la précision de l'estimation de la position et de la vitesse d'un véhicule, lorsque celui-ci est à une distance relativement faible du véhicule asservi.

Cependant, la mesure de la distance nécessite la connaissance de données géométriques précises sur le véhicule suivi. Dans les essais réalisés en laboratoire, ceux-ci étaient fournis. Ainsi, dans un cas réel, il sera important de pouvoir reconnaître le type de véhicule suivi. Cette reconnaissance apportera les données géométriques nécessaires : connaître le type du véhicule permettrait d'en connaître les dimensions exactes. Or, comme pour l'algorithme de détection présenté dans le chapitre 2, les processus de reconnaissance sont très sensibles à la résolution des objets dans l'image. Ainsi, conserver la même résolution à l'objet via notre système, devrait permettre dans l'avenir de définir une méthode de reconnaissance et d'extraction de données géométriques fiable.

Par ailleurs, l'usage de la caméra PTZ permet de suivre des véhicules au delà de 50 m. La figure 4.22 montre en effet un extrait d'une séquence où un véhicule de type «AX rouge» est suivi au delà de cette limite.

Les figures 4.23 et 4.24 donnent un exemple de l'estimation de la distance et de la vitesse, réalisée sur la séquence où le véhicule suivi est une AX (cf. figure 4.22). Elles donnent l'estimation de la distance et de la vitesse relative entre le véhicule VELAC et le véhicule suivi.

La distance est mesurée en considérant l'espacement entre les roues égal à 1.5 m et en considérant le motif asservi initialisé entre les roues comme dans la figure 4.10. Ces hypothèses introduisent certes un biais proportionnel dans les mesures. Mais ce biais peut être considéré acceptable dans l'application visée. Pour avertir le conducteur, connaître la distance exacte n'est pas nécessaire. Il est par contre vital que cette estimation soit peu bruitée.

Dans le cas de la caméra grand angle (cf. chapitre 3), nous avons vu qu'au delà de 50 m, la mesure de la distance pouvait varier d'une dizaine de mètres d'une image à l'autre, ce qui rends l'estimation de la distance (et de la vitesse) peu fiable.

Dans le cas d'un suivi avec la caméra PTZ, les mesures sont certes biaisées du fait de nos hypothèses, mais peu bruitées. Le suivi d'un véhicule obstacle à des distances supérieures (de 60 à 140 m) est, aux vues des courbes présentées, envisageable avec notre capteur. A noter qu'à ces distances, l'actionneur du zoom est en butée maximale : cet exemple illustre donc plus exactement le comportement que pourrait avoir notre capteur dans un mode «suivi de route», explicité dans le chapitre 1.

4.8 Conclusion

Dans ce chapitre, nous avons proposé une méthode afin d'asservir les actionneurs d'une caméra PTZ sur un motif, afin de maintenir celui-ci au centre de l'image et à une taille quasi-constante. Celle-ci s'appuie sur le formalisme de fonction de tâche pour

FIG. 4.22 – Extraits de séquences de suivi d'une «AX rouge» avec la caméra PTZ.

définir une loi de commande référencée image.

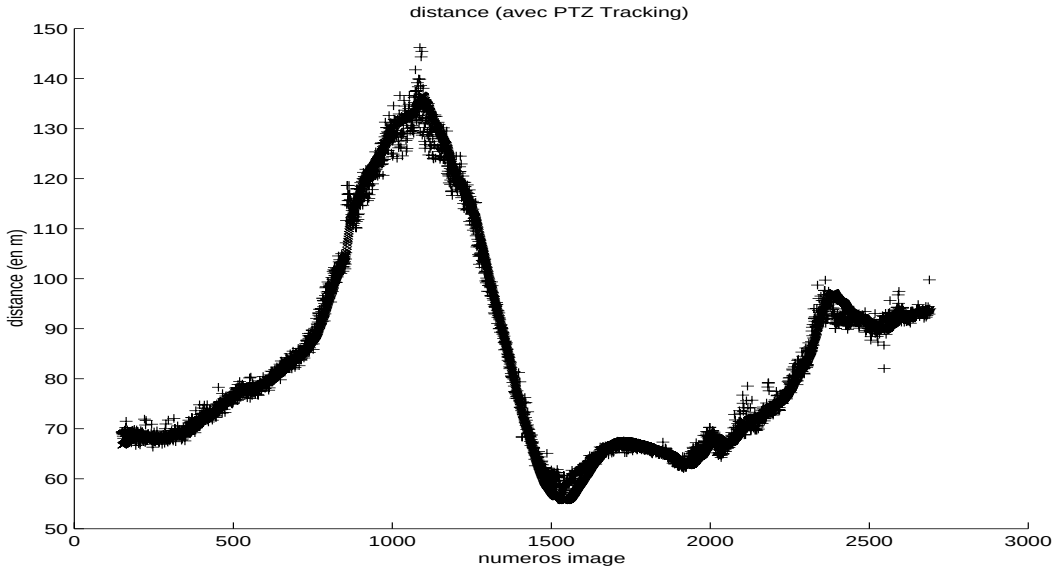


FIG. 4.23 – Estimation de la distance (suivant Y) relative entre le véhicule suivi et le véhicule VELAC ; la distance mesurée est notée par les '+' et celle estimée par les 'x'.

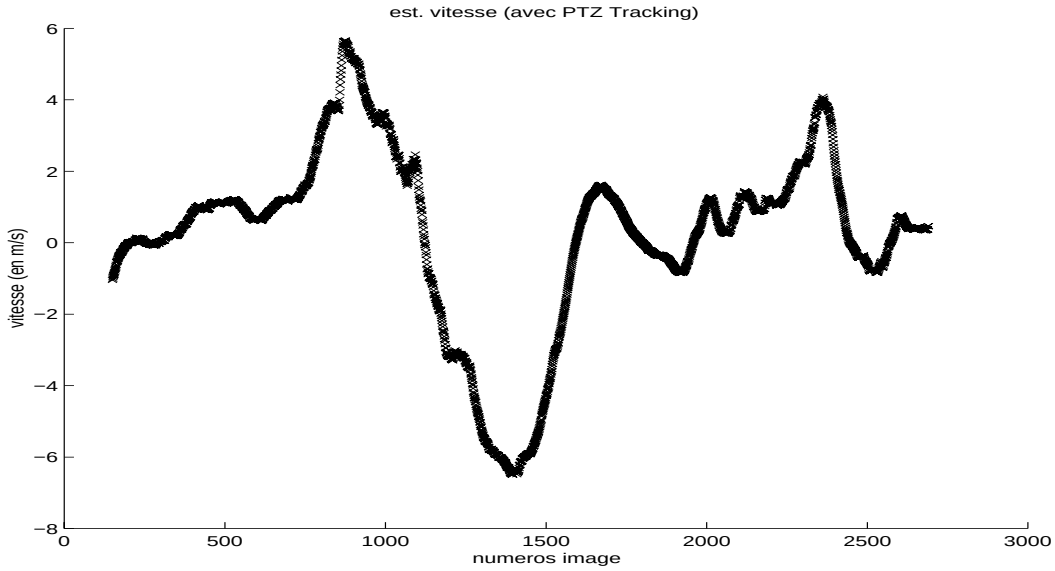


FIG. 4.24 – Estimation de la vitesse (suivant Y) relative entre le véhicule suivi et le véhicule VELAC.

Nous avons adaptée cette méthode au suivi d'un véhicule obstacle dans l'image. Nous utilisons notamment les informations visuelles fournies par l'algorithme de suivi vu dans le chapitre 3.

Plusieurs résultats présentés démontrent la validitée de notre approche pour le suivi d'un véhicule à des distances allant entre 15 m et 140 m, avec une caméra PTZ pilotée (dont la focale peut varier de 1000 à 3000 *pixels* environ).

Nous avons notamment exposé plusieurs extraits de séquences acquises en situations réelles de conduite (sur voies autoroutières) présentant le suivi d'un véhicule automobile par la caméra PTZ. Simultanément, la même scène a été capturée via une caméra grand angle. Ce montage illustre le champ de vision potentiel du capteur de vision double proposé dans le chapitre 1 dans un fonctionnement «suivi de véhicule». Les résultats présentés permettent ainsi de légimitiser et d'apprécier les capacités d'un tel capteur pour un système d'aide à la conduite.

En ce qui concerne l'asservissement de la caméra PTZ, plusieurs perspectives peuvent être envisagées. Tout d'abord signalons les travaux récents de *Ezio Malis et al.* [136, 137] qui concernent aussi l'asservissement d'une caméra Pan-Tilt-Zoom sur un ensemble de points [136] ou sur des droites [137]. Les auteurs proposent une approche permettant de rendre l'asservissement indépendant des paramètres intrinsèques de la caméra (en particulier de la valeur de la focale). Ensuite, l'asservissement de la caméra sur les paramètres de la route afin de permettre un «suivi de la route» reste à étudier. Enfin, nous avons proposé ici une méthode permettant de suivre un seul véhicule avec une résolution optimale. Cependant, dans une scène routière, il peut être intéressant de suivre un ensemble de véhicules (par exemple, deux véhicules sur deux voies différentes). Pour cela, il faudrait déterminer des paramètres appropriés. L'utilisation des moments centraux des surfaces pourrait être envisagée [207, 36].

Conclusion

Les travaux présentés dans ce mémoire sont des contributions à l'élaboration d'un système de vision pour un dispositif d'aide à la conduite. L'objectif d'un tel dispositif est la prévention ou l'évitement d'une situation potentiellement dangereuse. Ceci peut être réalisé en avertissant le conducteur par un signal sonore, visuel ou mécanique (vibration du volant).

Le non-respect des distances de sécurité minimales entre les véhicules est une importante cause d'accidents. Aussi la prévention et l'évitement d'une collision frontale nécessite l'estimation de cette distance. Durant cette thèse, nous nous sommes intéressés aux développements nécessaires qui permettraient l'accomplissement de cette tâche via un capteur de vision.

Dans ce cadre, l'état de l'art sur les systèmes de perception extéroceptive dans les véhicules intelligents, présenté dans un premier temps, permet de souligner l'importance des problématiques liées à la détection et au suivi des obstacles routiers, ainsi qu'à leur localisation sur les voies de circulation. Parmi les capteurs utilisés dans ce domaine, seuls les capteurs de type caméra sont capables de renvoyer des informations à la fois sur les obstacles et sur les voies de circulation. De plus, il s'agit d'une technologie peu coûteuse et flexible, son champ de perception étant essentiellement lié à sa focale. Nous proposons donc d'utiliser un capteur combinant une caméra à focale courte et une caméra Pan Tilt Zoom, commandée en angle site et azimut et pilotée en zoom. La première permet une vision globale de la scène frontale et la seconde une capture locale mais avec une plus grande résolution. L'utilisation d'un tel capteur pour la détection et le suivi de véhicules nécessite le développement de nombreuses méthodes adaptées. Les contributions de cette thèse se situent dans la définition et l'application de telles méthodes et algorithmes pour détecter et suivre des véhicules en situation routière.

La première application de ce capteur concerne la détection de véhicules dans la scène qui peuvent être jugés comme potentiellement dangereux. Celle-ci a deux principales exigences : la robustesse et la rapidité d'exécution. Nous avons donc cherché à développer une méthode de détection de véhicules alliant ces deux qualités. Elle est construite de manière hybride : une approche rapide par extraction de primitives, ombres portées et symétrie, permet de focaliser sur un nombre restreint de zones d'intérêts, une identification robuste selon l'apparence. Cette dernière méthode de classification est basée sur une représentation en ondelettes de Haar des vues arrières de véhicules et sur l'utilisation d'un processus d'apprentissage par supports vectoriels. Les résultats présentés montrent

que cette méthode permet de détecter et de localiser approximativement dans la scène frontale, des voitures avec des taux de bonnes détections et de fausses alarmes satisfaisants. Par ailleurs, cette méthode ne nécessite aucune coopération de la part des autres véhicules et, bien que développée et testée uniquement pour détecter des vues arrières de véhicules de tourisme, elle peut très bien être étendue à toute la gamme des véhicules routiers.

La seconde application de ce capteur concerne le suivi de véhicules dans la scène afin d'estimer les caractéristiques cinématiques du mouvement relatif entre la cible et le véhicule équipé. Les algorithmes de suivi classiques proposés dans la littérature s'exécutent en deux étapes : une phase de prédiction suivie d'une phase d'exploration autour de cette prédiction. La seconde phase est très coûteuse en temps de calcul. Aussi nous avons utilisé une méthode développée au laboratoire par *Jurie et Dhome*, qui permet de s'affranchir de cette seconde phase. Les résultats obtenus sur séquences d'images montrent que cet algorithme présente des avantages tels que la généricité de l'approche, le fonctionnement en temps réel et une grande précision dans l'estimation des déplacements. Cette précision permet notamment d'obtenir de bons résultats pour l'estimation des positions et des vitesses relatives du véhicule obstacle, à des distances allant de 15 à 50 m avec une caméra à focale courte.

Afin d'assurer le suivi d'un véhicule à des distances plus importantes, il est nécessaire d'utiliser une focale plus longue. Dans cette optique, un asservissement des actionneurs de la caméra PTZ a été étudié. En utilisant une approche référencée capteur, nous avons montré qu'il est possible de commander ses positions en angles et en zoom, à partir des informations visuelles fournies par l'algorithme de suivi précédemment présenté, de sorte à conserver le motif suivi (i.e. la vue arrière du véhicule) au centre de l'image, et ce à une taille quasi-constante. Les résultats en situation réelle de conduite ont montré que cet asservissement permet d'obtenir en permanence une image du véhicule suivi avec une meilleure résolution qu'avec une caméra à focale fixe et grand angle. De plus, la portée du capteur est alors accrue jusqu'aux environs de 140 m.

Dans l'ensemble, les travaux menés démontrent la faisabilité de notre système de perception pour la détection et le suivi d'obstacles routiers dans la scène frontale en situation réelle, et ceci :

- de façon autonome, c'est-à-dire sans aucune coopération de la part de l'infrastructure et des autres véhicules,
- en temps réel,
- de manière générique,
- avec une portée et un champ de vision adaptés à une application d'aide à la conduite.

Cependant un large champ d'investigation reste encore à explorer pour la mise en œuvre de notre capteur.

En premier lieu, la coopération entre les deux caméras est à définir et à étudier plus précisemment. Ceci nécessite un étalonnage précis de l'ensemble du capteur. Dans une telle optique, une méthode a été développée récemment au laboratoire. Son utilisation dans un avenir proche est envisagée. Cette coopération pourrait être bénéfique à la fois au niveau des algorithmes de détection et de suivi, et à la focalisation de la caméra PTZ sur un véhicule ou sur l'horizon de la route.

En détection, le processus d'identification peut être amélioré selon deux axes :
- Mise en place d'un processus en cascade permettant d'affiner l'identification des zones testées.
- Une sélection des coefficients selon leur pertinence pour le classifieur SVM.

Nous attendons de ces deux améliorations une détection des obstacles encore plus rapide et robuste.

Pour le suivi, deux points méritent d'être améliorés compte tenu de l'application :
- Diminuer le temps de calcul réservé à l'initialisation.
- Améliorer la robustesse du suivi par rapport aux occultations partielles.

L'intégration de travaux récents en détection et en reconnaissance via des représentations selon l'apparence est une piste de recherche qui, à notre avis, est riche en promesses. Cela représente bien sûr un travail conséquent.

Pour l'asservissement de la caméra PTZ, la solution développée, bien qu'efficace pour notre application, demeure basique. Beaucoup d'améliorations peuvent y être apportées :
- Tenir compte de la dynamique de la cible, permettrait d'étendre l'approche à d'autres types d'objets que les véhicules obstacles sur voies autoroutières : par exemple, le suivi de panneaux de signalisation ou de piétons dans un contexte urbain.
- Il serait intéressant par ailleurs de comparer l'approche développée dans cette thèse avec celle récemment développée par *Ezio Malis et al.* Comme nous l'avons déjà dit, ce dernier propose en effet un asservissement visuel indépendant des paramètres intrinsèques de la caméra (donc de la focale).
- Une autre voie d'investigation concerne l'étude de nouveaux paramètres visuels visant à l'asservissement de la caméra PTZ sur la route ou sur un ensemble d'obstacles.

Par ailleurs, il serait important dans l'avenir de pouvoir évaluer les performances exactes du système. L'utilisation d'autres capteurs (par exemple, des DGPS) permettraient d'obtenir une vérité terrain et d'ainsi évaluer la pertinence des estimations réalisées, notamment des caractéristiques cinématiques du véhicule suivi.

Enfin, au delà de l'application d'aide à la conduite, les travaux menés sur ce capteur pourraient être généralisés à la navigation d'autres types de robots ou de véhicules (aériens, sous-marins).

Annexe A

Exemples de résultats obtenus par *Papageorgiou et al.*

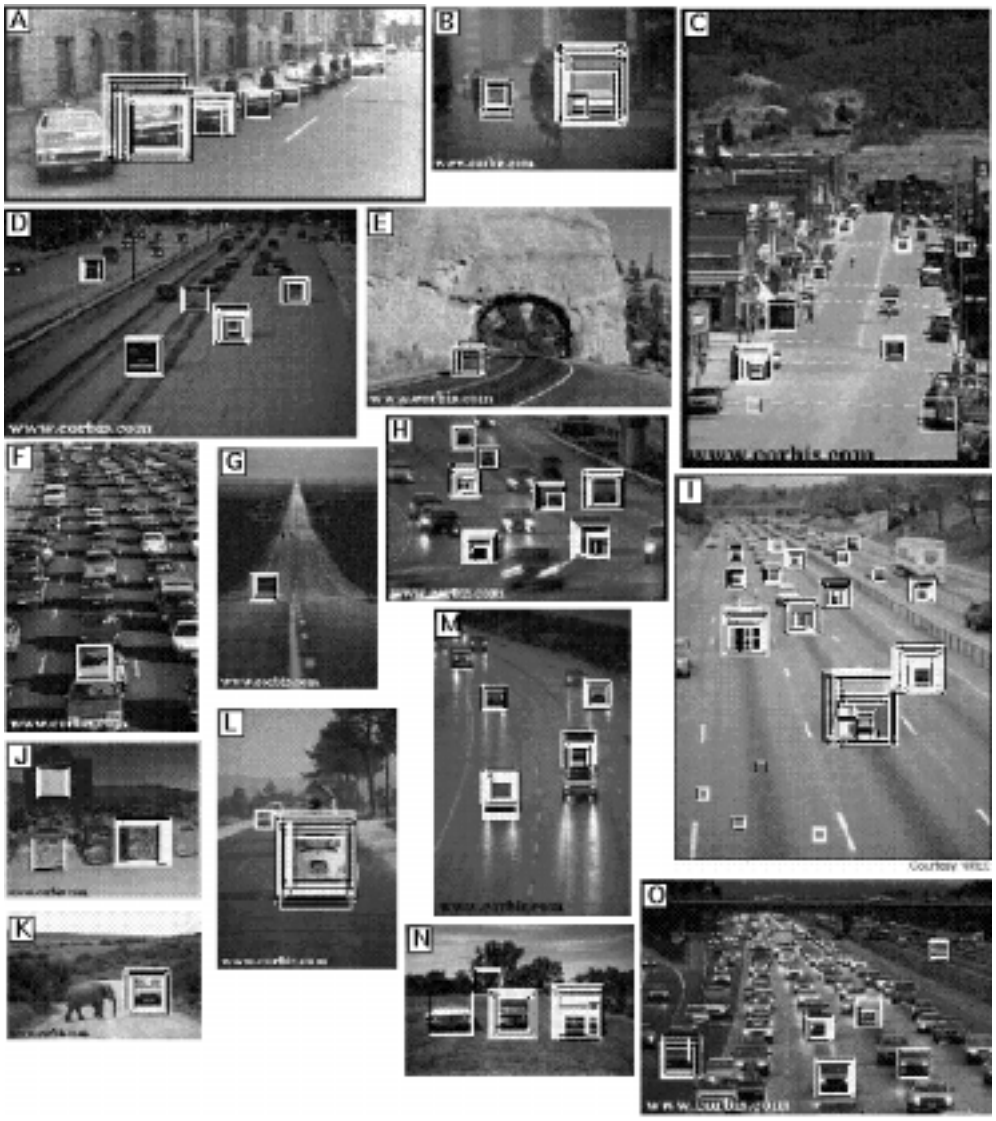


FIG. A.1 – Résultats extraits de [158]

Annexe B

Quelques notions sur les processus d'apprentissage.

De nombreux écrits existent en ce qui concerne l'apprentissage automatique. L'approche qui en est faite ici s'inspire avant tout de Vladimir N. Vapnik. Il décrit le problème d'apprentissage à partir des trois constituants suivants :

- un *générateur* (G) de vecteurs $x \in \mathbb{R}^N$, dont la distribution est fixée, mais inconnue *à priori*. Dans la suite, cet ensemble de vecteurs générés sera noté $D_l = (x_1, x_2, ..., x_l)$.

- un *superviseur* (S) qui associe une sortie y à chaque entrée x.

- une *machine à apprendre* (M) dépendant de paramètres donnés, induisant une fonction $f \in \mathbb{F}$, ensemble de fonctions, sur les entrées x.

Le but est que cette fonction f donne des valeurs les plus proches possibles de la décision du superviseur , i.e. minimiser le risque de se tromper. Un schéma de ce principe d'apprentissage est donné figure B.1.

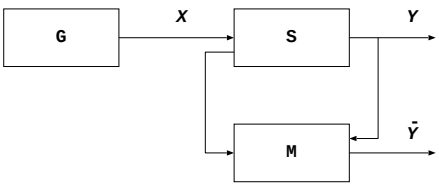


FIG. B.1 – Processus d'apprentissage automatisé selon V.N. Vapnik.

Il est important de remarquer que, du fait de la présence du superviseur, la machine est dans sa phase d'apprentissage. Ainsi les valeurs y fournies par le superviseur sont également des entrées de la machine. Avec les vecteurs x, elles forment un jeu d'exemples d'entraînement.

La machine est destinée à fonctionner en dehors de tout contrôle de la part du superviseur : c'est la phase d'utilisation réelle.

Dans la théorie de l'apprentissage, on distingue plusieurs types d'erreurs :

- le risque empirique qui mesure l'erreur ou perte ou coût moyen pour un f particulier sur un ensemble D_l :

$$\widehat{R}(f, D_l) = \frac{1}{l} \sum_{i=1}^{l} L(x_i, f) \tag{B.1}$$

  avec L la fonction de coût définie par : $L(x, f) = 0 si f(x) = y$ et 1 sinon.
- le risque espéré ou erreur de généralisation qui mesure l'erreur espérée avec f sur un exemple tiré au hasard dans $\mathbb{R}^N$:

$$R(f, D_l) = \frac{1}{l} \sum_{i=1}^{l} L(x_i, f) \tag{B.2}$$

  On ne peut calculer cette quantité, mais c'est celle qui nous intéresse vraiment.

Comme nous l'avons déjà dit, le principe de minimisation du risque empirique est à la base d'une grande quantité d'algorithmes d'apprentissage, i.e. on cherche :

$$f^*(D_l) = argmin_{f \in \mathbb{F}} \widehat{R}(f, D_l). \tag{B.3}$$

D'où est définie l'erreur d'apprentissage comme la plus petite valeur moyenne de perte sur les exemples d'apprentissage D_l obtenue en choisissant $f^*(D_l)$ dans $\mathbb{F}$:

$$\text{erreur d'apprentissage} = \widehat{R}(f^*(D_l), D_l) = min_{f \in \mathbb{F}} \widehat{R}(f, D_l) \tag{B.4}$$

Selon la théorie de l'apprentissage, sous certaines conditions sur $\mathbb{F}$, il est possible de borner la différence entre l'erreur de généralisation et l'erreur d'apprentissage. Un des moyens de contrôler ainsi la généralisation est, pour les problèmes de classification, est le ratio entre la marge autour de la surface de classification et le diamètre de l'hypershpère qui englobe les vecteurs d'entrée de l'ensemble d'apprentissage. Plus ce ratio est grand et plus est petite est la différence entre l'erreur d'apprentissage et l'erreur de généralisation. Les classificateurs réalisant cette condition sont nommés classificateurs à marge optimale. C'est le cas des SVM.

Annexe C

Théorie des ondelettes et base de Haar

La décomposition en ondelettes [139] est un outil très intéressant en traitement d'image du fait de son effet décorrélatoire sur les pixels de l'image, de la concentration de l'énergie sur un nombre réduit de coefficients, de son approche multi-échelles et multi-résolutions, et de sa décomposition fréquentielle.

C.1 Ondelettes continues

On peut désigner comme ondelettes réelles, une fonction de $\Re$ dans $\Re$ dont la transformée de Fourier vérifie certaines conditions de régularité et de localisation. Il en résulte diverses propriétés, dont la localisation de la fonction elle-même (i.e. négligeabilité en dehors d'un intervalle compact), et une moyenne nulle (d'où le nom d'ondelettes, en raison des oscillations assurant cette propriété). Les propriétés des ondelettes en traitement du signal sont à la fois proches et complémentaires de celles des fonctions sinusoïdales à la base de l'analyse de Fourier. On peut ainsi décomposer tout signal (suffisamment régulier) sur des bases d'ondelettes obtenues par dilatation et translation d'une seule et même ondelette, dite ondelette mère. La théorie de la décomposition en ondelettes permet même d'exhiber des bases d'ondelettes orthonormales, à support compact, etc.

Soit une fonction Ψ appartenant a $L^2(\Re)$ et $TF(\Psi)$ sa transformée de Fourier satisfaisant la condition d'admissibilité :

$$C = \int_{\Re} \frac{|TF(\Psi(v))|^2}{|v|} dv < \infty \tag{C.1}$$

Alors Ψ est appelée une ondelette mère. Et on appelle transformée en ondelette, la transformation intégrale qui à toute fonction f appartenant à $L^2(\Re)$ fait correspondre la fonction $W_f(a,b)$ définie par :

$$\begin{array}{rl} W_f(a,b) & =< f, \Psi_{ab} > \\ & = |a|^{-1/2} \int_{\Re} f(t) \Psi(\frac{t-b}{a}) dt \end{array} \tag{C.2}$$

avec $b \in \Re$ et $a \in \Re^*$

A partir de l'ondelette mère, on construit toute une famille d'ondelettes $\Psi(\frac{t-b}{a})$ obtenues par dilatation (coefficient a) et translation (coefficient b) de Ψ. On désignera chaque élément de cette famille par :

$$\Psi_{ab}(t) = |a|^{-1/2}\Psi(\frac{t-b}{a}) \tag{C.3}$$

On a par ailleurs, la condition suivante :

$$TF(\Psi(0)) \leftrightarrow \int \Psi(t)dt = 0 \tag{C.4}$$

Si $\int \Psi(t)dt = 0$, la fonction $\Psi(t)$ oscille et s'amortit, d'où le nom d'ondelette.

C.2 Inversion de la transformée en ondelettes

Il faut tout d'abord remarquer qu'en traitement du signal, on ne considère généralement que des fréquences positives. Si le centre est aussi positif, on peut ne considérer que des valeurs positives du paramètre de dilatation : dans la reconstitution de f à partir des ondelettes, on a utilisé seulement les valeurs de $a > 0$ dans $W_f(a,b) =< f, \Psi_{ab} >$. La condition d'admissibilité devient alors : $C/2 < 0$.

On a alors, le théorème suivant :

Soit Ψ une ondelette mère satisfaisant la condition d'admissibilité :

$$\frac{C}{2} = \int_0^\infty \frac{|\Psi(v)|^2}{v}dv \tag{C.5}$$

Alors pour tout f, g appartenant à $L^2(\Re)$.

$$\frac{C}{2} < f, g >= \int_0^\infty \frac{da}{a^2}\int_\Re W_f(a,b)W_g(a,b)db \tag{C.6}$$

Et pour tout f appartenant à $L^2(\Re)$ et t appartenant à $\Re$ où f est continue, on a la formule d'inversion :

$$f(t) = \frac{2}{C}\int_0^\infty \frac{da}{a^2}\int_\Re W_f(a,b)\Psi_{ab}(t)db \tag{C.7}$$

C.3 Transformations en ondelettes discrètes

Contrairement à la transformation de Fourier, la transformation en ondelettes continues ne se prête pas aisément à des calculs analytiques simples, même dans les cas où l'ondelette mère et le signal à analyser sont de formes simples. Le calcul numérique de transformations devient indispensable et de ce fait, on est amené à discrétiser le problème. De plus, certains problèmes se trouvent déjà sous forme discrétisée, de par la nature numérique du signal à analyser. Dans le cas continu, on a considéré les ondelettes

de la forme :

$\Psi_{ab}(t) = |a|^{-1/2}\Psi(\frac{t-b}{a}), b \in \Re \text{et} a \in \Re^*$.
Avec la condition d'admissibilité :

$C = \int_{\Re} \frac{|TF(\Psi(v))|^2}{|v|} dv < \infty$
Nous choisissons pour un pas de dilatation $a_0 > 1$. On désigne également un paramètre de translation $b_0 > 0$. On peut alors choisir des valeurs fixes : $a_m = a_0^m$ et $b_n = nb_0.a_0^m$ pour $(m,\ n)$ éléments de $\mathbb{Z}^2$. Le réseau $\{(a_m, b_n), (m, n)$ éléments de$\mathbb{Z}^2\}$ forme une grille dyadique.

La famille d'ondelettes discrétisées

$$\begin{aligned} \Psi_{mn}(t) &= a_0^{-m/2}\Psi(\tfrac{t-nb_0 a_0^m}{a_0^m}) \\ &= a_0^{-m/2}\Psi(a_0^{-m}t - nb_0) \end{aligned} \tag{C.8}$$

où m et n sont éléments de $\mathbb{Z}$ définit la transformation en ondelettes discrètes par la formule :

$$\begin{aligned} W_f(m,n) &= < \Psi_{mn}, f > \\ &= a_0^{-m/2} \int_{\Re} f(t)\Psi(a_0^{-m}t - nb_0)dt \end{aligned} \tag{C.9}$$

Généralement, on prend $a_0 = 2$, $b_0 = 1$.

C.4 Construction d'une analyse multirésolution

C.4.1 Principe de l'analyse espace - échelle ou analyse multirésolution

Une analyse multirésolution consiste à utiliser une gamme très étendue d'échelles pour effectuer l'analyse d'un signal. Le signal sera à une échelle donnée, remplacé par l'approximation la plus adéquate que l'on puisse tracer à cette échelle. L'analyse multirésolution calcule ce qui diffère d'une échelle à l'autre, c'est-à-dire les détails qui permettent, à partir d'une approximation à une certaine échelle, d'accéder à une approximation à une échelle plus petite [139]. Mathématiquement, l'analyse multirésolution s'introduit de la façon suivante :

$$F_n = \sum_{j=-\infty}^{n} f_j \tag{C.10}$$

avec :

$$F_n = \sum_{k=-\infty}^{+\infty} < f(t), \Psi_{jk}(t) > \Psi_{jk}(t) \tag{C.11}$$

Cette approximation, qui est la projection de $f(t)$ sur un espace V_n, tend vers $f(t)$ quand $n \to \infty$. f_j représente alors le détail que l'on doit ajouter à F_j pour obtenir $F_j + 1$, approximation plus fine.

Définition formelle

Par définition, une analyse multirésolution à la résolution j est définie par l'application d'un opérateur linéaire A_j ; $A_j f \in V_j$, V_j est un sous espace vectoriel de l'espace V (espace d'approximation) satisfaisant :

- $0... \subset V_2 \subset V_1 \subset V_0 \subset V_{-1}... \subset V_{j-1} \subset V_{j-2}...L^2(\Re)$
- $\cap_{j \in \mathbb{Z}} V_j = 0$
- $\cup_{j \in \mathbb{Z}} V_j = 0 = L^2(\Re)$ (est dense dans $L^2(\Re)$)

Il faut à présent trouver une base engendrant V_j.

Soit la fonction $\phi(t)$ appartenant à $L^2(\Re)$ telle que $\phi(t-k)$, $k \in \mathbb{Z}$ soit une base orthonormée dans $L^2(\Re)$. La fonction $\phi(t)$ est appelée "fonction d'échelle" ou fonction d'interpolation. Soit V_0 le sous-espace vectoriel engendré par $\phi(t-k)$, $k \in Z$, V_j est défini à partir de V_0 par simple changement d'échelle : $f(t) \in V_0 \Leftrightarrow f(2^{-j}t) \in V_j$ pour toute fonction f de $L^2(\Re)$.

La fonction générant une multirésolution est une fonction échelle $\phi \in L^2(\Re)$. Dans le cas dyadique, la fonction échelle est définie par $\phi_{jk}(t) = 2^{-j/2}\phi(2^{-j}t_k)$, $k \in Z$. Le projecteur est alors $A_j f = \sum < f, \phi_{jk} > \phi_{jk}$, ou le produit scalaire $a_k^j =< f, \phi_{jk} >$ représente l'approximation du signal à l'échelle j.

Pour compléter cette analyse, nous avons besoin d'un espace complémentaire W_j (espace des détails), orthogonal à V_j. l'ensemble des W_j, couvre tout l'espace $L^2(\Re)$, et les W_j sont orthogonaux deux à deux.

Les espaces V_j et W_j sont liés entre eux par la relation $V_{j-1} = V_j \oplus W_j$.

Les espaces d'approximations V_j et les espaces de détails W_j étant orthogonaux, toute l'information du signal est conservée. Par conséquent, on pourra reconstruire exactement le signal. La fonction générant W_j, est une ondelette Φ. L'opérateur linéaire est alors $D_j f = \sum < f, \Phi_{jk} > \Phi_{jk}$, et le produit scalaire $d_k^j =< f, \Phi_{jk} >$ représente les détails du signal à l'échelle j.

$$\Phi_{jk}(t) = 2^{-j/2}\Phi(2^{-j}t - k) \tag{C.12}$$

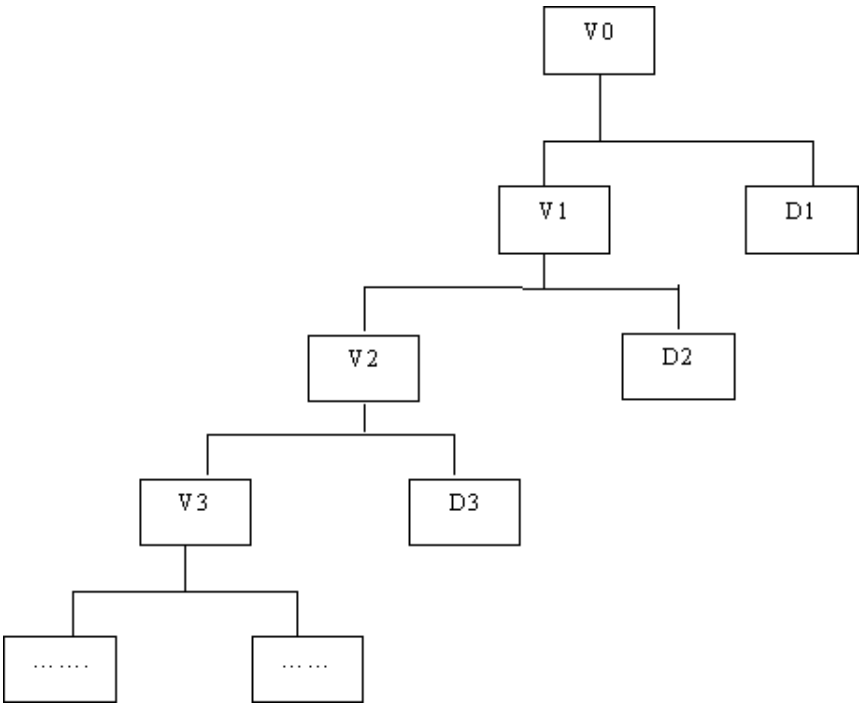


FIG. C.1 – Schéma de l'approche multirésolution : $A_{j+1}f = A_jf + D_jf$

C.5 Ondelette de Haar

C.5.1 Ondelette 1D

Parmi les ondelettes les plus connues et que nous avons utilisé sont celles de Haar présentées en 1910. Il a démontré une fonction simple qui peut être utilisée pour générer une base orthonormale dans $L^2(\Re)$.

Comme une base pour l'espace vectoriel V_j, on utilise la fonction d'échelle (cf. figure C.2) :

$\phi_{jk}(t) = 2^{-j/2}\Phi(2^{-j}t_k), k = 0, ..., 2^j - 1.$

Où :

$$\Phi(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & \text{sinon} \end{cases} \tag{C.13}$$

On définit le sous-espace vectoriel des ondelettes W_j qui est le complément orthogonal des deux sous espaces consécutifs $V_{j+1} = V_j \oplus W_j$. W_j est généré par les fonctions de base :

$\Phi_{jk}(t) = 2^{-j/2}\Phi(2^{-j}t - k)$

Où, dans le cas d'ondelettes de Haar (cf. figure C.3), :

$$\Phi(t) = \begin{cases} 1 & 0 \leq t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{sinon} \end{cases} \tag{C.14}$$

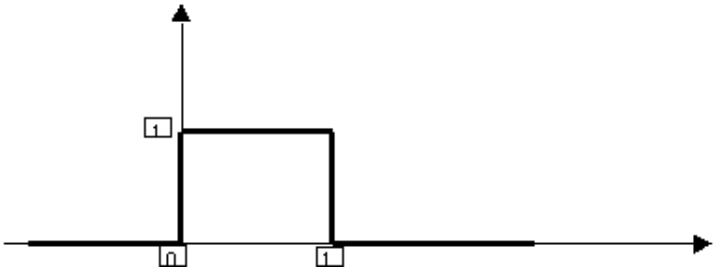


FIG. C.2 – Fonction d'échelle de Haar 1D

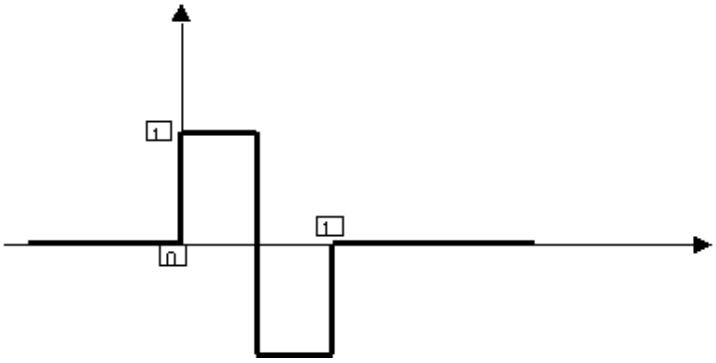


FIG. C.3 – Fonction d'ondelette de Haar 1D

C.5.2 Ondelettes 2D

L'extension des ondelettes au cas 2D se fait en prenant le produit tensoriel de deux ondelettes 1D. Le résultat est trois types d'ondelettes montrés à la figure C.4. Le premier type d'ondelette est le produit tensoriel d'ondelette avec la fonction d'échelle, $\Phi(x, y) = \Phi(x) \otimes \phi(y)$; cette ondelette encode la différence de l'intensité moyenne le long d'une bordure vertical, on référence sa valeur par les coefficients verticaux. Similairement, le produit tensoriel de la fonction d'échelle avec l'ondelette, $\Phi(x, y) = \phi(y) \otimes \Phi(x)$, donne les coefficients horizontaux, et ondelette par ondelette $\Phi(x, y) = \Phi(x) \otimes \Phi(y)$, donne les coefficients diagonaux.

C.5.3 Calcul des ondelettes par la méthode de l'image intégrale

Les ondelettes peuvent être calculées rapidement en utilisant une représentation intermédiaire pour l'image qui est appelée l'image intégrale [205]. L'image intégrale à x, y contient la somme des pixels au dessus et à gauche de x, y, inclus :

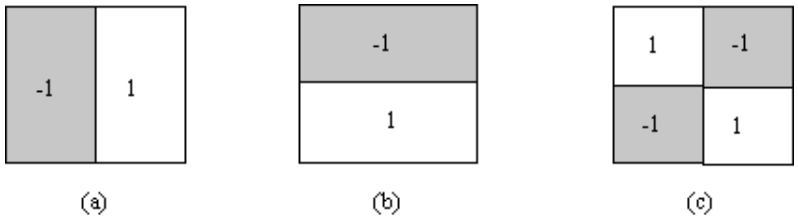


FIG. C.4 – Ondelettes de Haar 2D : (a) vertical, (b) horizontal et (c) diagonal

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \tag{C.15}$$

où $ii(x,y)$ est l'image intégrale et $i(x,y)$ est l'image originale. Utilisant les formules de récurrence suivantes :

$$\begin{aligned} s(x,y) &= s(x,y-1) + i(x,y) \\ ii(x,y) &= ii(x-1,y) + s(x,y) \end{aligned} \tag{C.16}$$

où $s(x,y)$ est la somme cumulative d'une ligne. On pose que : $s(x,-1) = 0$ et $ii(-1,y) = 0$.

Utilisant l'image intégrale, n'importe quelle somme d'une zone rectangulaire peut être calculée en quatre vecteurs référence (cf. figure C.5).

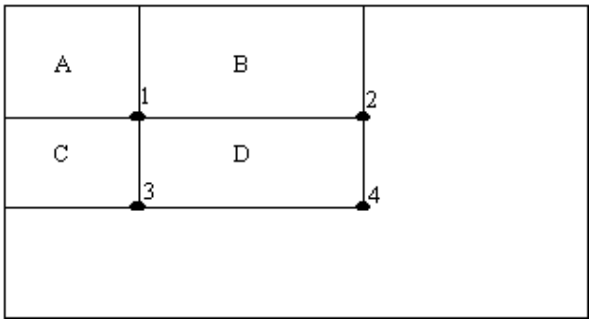


FIG. C.5 – La valeur de l'image intégrale en 1, est la somme des pixels dans le rectangle A. Celle en 2, est A+B, en 3, est A+C et en 4, est A+B+C+D. La somme dans D est donc 4+1-(2+3)

C.5.4 Transformée d'ondelettes dense

Dans certaines applications comme la détection, une base standard de Haar n'est pas suffisamment dense. Dans le cas 1D, la distance entre deux ondelettes voisines à l'échelle n est 2^n. Pour une meilleure résolution spatiale, on a besoin d'un ensemble de fonctions de base redondantes, ou un dictionnaire sur-complet, où la distance entre deux ondelettes à l'échelle n est $\frac{1}{4}2^n$. On appelle ça un dictionnaire de quadruple densité.

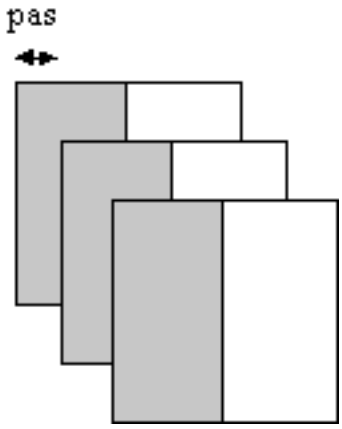


FIG. C.6 – densité Quadruple de base de Haar 2D

Annexe D

Modèle 3D de route plane / véhicule

Nous utilisons, dans nos algorithmes de détection et suivi de véhicules, un modèle 3D de l'ensemble route/véhicule établi dans [32]. Ce modèle est défini selon certaines hypothèses. Tout d'abord, la route est considérée localement plane dans l'image (aucune courbure verticale n'est prise en compte avec ce type de modélisation). D'autre part, la courbure de la route est supposée localement constante. Ainsi, les bords de la route peuvent être représentés par deux arcs de cercle de courbure égale sur une même image.

Le schéma de la figure D.1 illustre la définition de ce modèle de route.

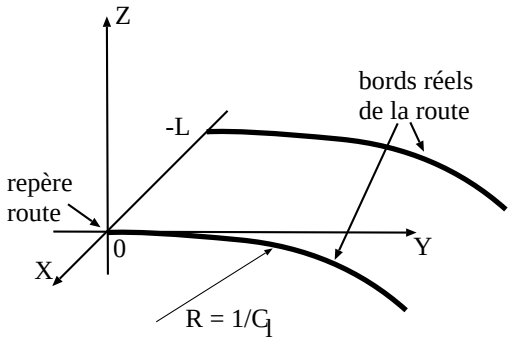


FIG. D.1 – Représentation des bords de la route

L'équation caractéristique de cette modélisation 3D, en considérant que le monde est plan devant le véhicule et que la route présente une courbure latérale constante dans une image, est de la forme suivante :

$$X \simeq \frac{C_l Y^2}{2} + \lambda L \qquad \text{et} \qquad Z = 0$$

avec $\lambda = 0$ pour identifier le bord droit de la voie et $\lambda = -1$ pour le bord gauche.

La projection des bords dans l'image doit tenir compte du fait que les repères de la scène et de la caméra ne sont pas identiques.

Les positions relatives du repère "route" et du repère "caméra" sont représentées sur la figure D.2.

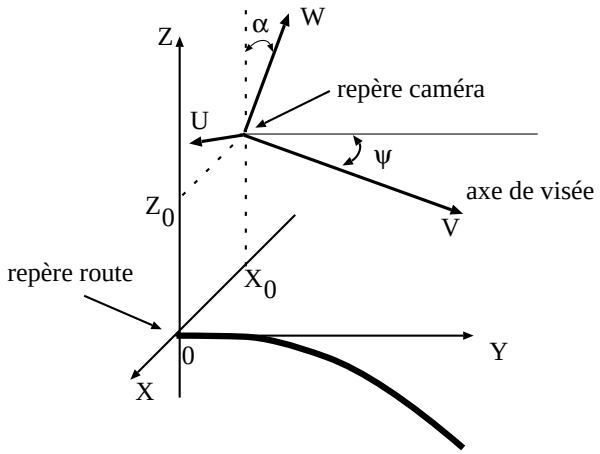


FIG. D.2 – Passage du repère "route" au repère "caméra"

Les paramètres utilisés pour cette modélisation sont les suivants :
- $f_u = f/d_u$ et $f_v = f/d_v$ sont les paramètres intrinsèques de la caméra supposés connus par un calibrage préalable avec la constante f qui représente la focale de la caméra,
- d_u et d_v : largeur et hauteur d'un pixel pour la matrice CCD de la caméra,
- Z_0 : hauteur de la caméra,
- α : angle d'inclinaison de la caméra,
- X_0 : position latérale du véhicule sur la chaussée,
- C_l : courbure latérale de la chaussée,
- L : largeur d'une voie de circulation ou de la route,
- ψ : angle de cap du véhicule.

Le repère associé à la caméra a donc son origine au point de coordonnées $(X = X_0, Y = 0, Z = Z_0)$.
Par conséquent, un point 3D $\underline{P}_R = (X, Y, Z)^t$ défini dans le repère de la route, en tenant compte des angles α et ψ ainsi que de la translation $\underline{T} = (X_0, 0, Z_0)^t$, a pour coordonnées $\underline{P}_{Rc}$ dans le repère de la caméra : $(U, V, W)^t$

$$\underline{P}_{Rc} = \mathbf{R}_\psi \mathbf{R}_\alpha \underline{\mathbf{P_R}} - \underline{\mathbf{T}} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y \\ Z - Z_0 \end{pmatrix}$$

En faisant l'approximation que les angles α et ψ sont faibles, nous avons :

$$\underline{P}_{Rc} \approx \begin{pmatrix} 1 & -\psi & 0 \\ \psi & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y \\ Z - Z_0 \end{pmatrix}$$

soit :

$$\underline{P}_{Rc} \approx \begin{pmatrix} 1 & -\psi & 0 \\ \psi & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - \alpha Z + \alpha Z_0 \\ \alpha Y + Z - Z_0 \end{pmatrix}$$

Ce qui donne, après développement et pour $Z = 0$ (hauteur de la route) :

$$\underline{P}_{Rc} = \begin{pmatrix} X - X_0 - \psi(Y + \alpha Z_0) \\ \psi(X - X_0) + Y + \alpha Z_0 \\ \alpha Y - Z_0 \end{pmatrix} \approx \begin{pmatrix} X - X_0 - \psi Y \\ Y \\ \alpha Y - Z_0 \end{pmatrix} = \begin{pmatrix} U \\ V \\ W \end{pmatrix}$$

Les approximations choisies supposent que Y est suffisamment grand devant les grandeurs αZ_0 et $\psi(X - X_0)$. Ces hypothèses peuvent être considérées comme toujours respectées car les valeurs admissibles pour ces paramètres dans les différents contextes envisagés sont : $3m < Y < 100m$, $-0.01m^{-1} < C_l < 0.01m$, $0.5m < Z_0 < 2m$ et $-5m < X_0 < 5m$.

Le point 3D $\underline{P}_R = (X, Y, Y)^t$ a pour projection dans le plan image le point $\underline{P}_i = (u, v)^t$ dont les coordonnées u et v sont données par :

$$u = f_u \frac{U}{V} = f_u \frac{X - X_0 - \psi Y}{Y} \tag{D.1}$$

et

$$v = f_v \frac{W}{V} = f_v \frac{\alpha Y - Z_0}{Y} \tag{D.2}$$

D'après la relation D.2, nous avons :

$$Y(v - f_v \alpha) = -f_v Z_0 \Rightarrow Y = -\frac{f_v Z_0}{v - f_v \alpha} \tag{D.3}$$

et d'après D.1 :

$$u = f_u \left(\frac{X}{Y} - \frac{X_0}{Y} - \psi \right) \tag{D.4}$$

avec $X \simeq C_l Y^2/2 + \lambda L$, l'équation (D.4) donne :

$$u = f_u \left(-\frac{f_v Z_0}{2(v - f_v \alpha)} C_l + \frac{v - f_v \alpha}{f_v Z_0} (X_0 - \lambda L) - \psi \right) \tag{D.5}$$

L'angle de roulis θ (angle de rotation autour de l'axe Y) est supposé négligeable dans ces différentes équations car on montre que son influence se compense avec le terme X_0. En effet, cet angle se caractérise dans l'équation (D.5) par le terme $\theta(\frac{v - f_v \alpha}{f_v})$.

Ce modèle donne donc les abscisses u_{ig} et u_{id} des bords de la route, respectivement gauche et droit, dans l'image en fonction des ordonnées fixes v_i selon les équations de deux hyperboles de la forme suivante :

$$u_{ig} = f_u \left(-\frac{f_v Z_0}{2(v_i - f_v \alpha)} C_l + \frac{v_i - f_v \alpha}{f_v Z_0} (X_0 + L) - \psi \right) \tag{D.6}$$

$$u_{id} = f_u \left(-\frac{f_v Z_0}{2(v_i - f_v \alpha)} C_l + \frac{v_i - f_v \alpha}{f_v Z_0} X_0 - \psi \right) \tag{D.7}$$

Annexe E

Asservissement visuel avec un segment centré

Établissons d'abord la matrice d'interaction pour une primitive point en considérant le torseur cinématique augmenté suivant :

$$\tilde{T}_c = \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \tag{E.1}$$

La relation est :

$$\begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} -\frac{f_u}{Z} & 0 & \frac{u}{Z} & \frac{uv}{f_v} & -(f_u + \frac{u^2}{f_u}) & v\frac{f_u}{f_v} & \frac{u}{f} \\ 0 & -\frac{f_v}{Z} & \frac{v}{Z} & (f_v + \frac{v^2}{f_v}) & -\frac{uv}{f_u} & -u\frac{f_v}{f_u} & \frac{v}{f} \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \tag{E.2}$$

Établissons la matrice d'interaction pour une primitive segment. Plusieurs représentations de ce segment sont possibles. Nous avons choisi de le représenter par son centre de gravité dont les coordonnées seront notées (u_c, v_c) , sa longueur h et son angle θ par rapportá l'horizontale. Les coordonnées des points extrêmes P_1 et P_2 seront notées (u_1, v_1) et (u_2, v_2) (cf. figure E.1).

Nous avons donc les relations suivantes :
– Coordonnées de P_1 : $u_1 = u_c + \frac{h}{2}cos\theta$, $v_1 = v_c + \frac{h}{2}sin\theta$

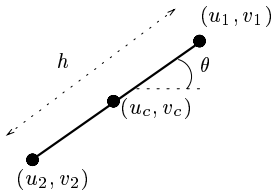


FIG. E.1 – Repérage du segment centré dans l'image.

– Coordonnées de P_2 : $u_2 = u_c - \frac{h}{2}cos\theta$, $v_2 = v_c - \frac{h}{2}sin\theta$
– Angle : $\theta = arctan(\frac{v_1 - v_2}{u_1 - u_2})$
– Longueur : $h = \sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2}$
Ce qui nous permet d'établir la relation suivante :

$$\begin{aligned}
\dot{\mathbf{s}} &= \begin{pmatrix} \dot{u_c} \\ \dot{v_c} \\ \dot{h} \\ \dot{\theta} \end{pmatrix} \\
&= \begin{pmatrix} \frac{\partial u_c}{\partial u_1} & \frac{\partial u_c}{\partial v_1} & \frac{\partial u_c}{\partial u_2} & \frac{\partial u_c}{\partial v_2} \\ \frac{\partial v_c}{\partial u_1} & \frac{\partial v_c}{\partial v_1} & \frac{\partial v_c}{\partial u_2} & \frac{\partial v_c}{\partial v_2} \\ \frac{\partial h}{\partial u_1} & \frac{\partial h}{\partial v_1} & \frac{\partial h}{\partial u_2} & \frac{\partial h}{\partial v_2} \\ \frac{\partial \theta}{\partial u_1} & \frac{\partial \theta}{\partial v_1} & \frac{\partial \theta}{\partial u_2} & \frac{\partial \theta}{\partial v_2} \end{pmatrix} \begin{pmatrix} \dot{u_1} \\ \dot{v_1} \\ \dot{u_2} \\ \dot{v_2} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{u_1 - u_2}{h} & \frac{v_1 - v_2}{h} & -\frac{u_1 - u_2}{h} & -\frac{v_1 - v_2}{h} \\ -\frac{v_1 - v_2}{h^2} & \frac{u_1 - u_2}{h^2} & \frac{v_1 - v_2}{h^2} & -\frac{u_1 - u_2}{h^2} \end{pmatrix} \begin{pmatrix} \dot{u_1} \\ \dot{v_1} \\ \dot{u_2} \\ \dot{v_2} \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ cos\theta & sin\theta & -cos\theta & -sin\theta \\ -\frac{sin\theta}{h} & \frac{cos\theta}{h} & \frac{sin\theta}{h} & -\frac{cos\theta}{h} \end{pmatrix} \begin{pmatrix} \dot{u_1} \\ \dot{v_1} \\ \dot{u_2} \\ \dot{v_2} \end{pmatrix}
\end{aligned} \tag{E.3}$$

Or, l'équation E.2 nous donne pour nos deux points :

$$\begin{pmatrix} \dot{u_1} \\ \dot{v_1} \\ \dot{u_2} \\ \dot{v_2} \end{pmatrix} = \begin{pmatrix} -\frac{f_u}{Z_1} & 0 & \frac{u_1}{Z_1} & \frac{u_1 v_1}{f_v} & -(f_u + \frac{u_1^2}{f_u}) & v_1 \frac{f_u}{f_v} & \frac{u_1}{f} \\ 0 & -\frac{f_v}{Z_1} & \frac{v_1}{Z_1} & (f_v + \frac{v_1^2}{f_v}) & -\frac{u_1 v_1}{f_u} & -u_1 \frac{f_v}{f_u} & \frac{v_1}{f} \\ -\frac{f_u}{Z_2} & 0 & \frac{u_2}{Z_2} & \frac{u_2 v_2}{f_v} & -(f_u + \frac{u_2^2}{f_u}) & v_2 \frac{f_u}{f_v} & \frac{u_2}{f} \\ 0 & -\frac{f_v}{Z_2} & \frac{v_2}{Z_2} & (f_v + \frac{v_2^2}{f_v}) & -\frac{u_2 v_2}{f_u} & -u_2 \frac{f_v}{f_u} & \frac{v_2}{f} \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \tag{E.4}$$

ce qui nous permet d'établir l'équation suivante :

$$\begin{pmatrix} \dot{u_c} \\ \dot{v_c} \\ \dot{h} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\delta_2 f_u & 0 & \delta_2 u_c - \delta_1 \frac{h cos\theta}{4} & \frac{u_c v_c}{f_v} + h^2 \frac{cos\theta sin\theta}{4 f_v} \\ 0 & -\delta_2 f_v & \delta_2 v_c - \delta_1 \frac{h sin\theta}{4} & f_v + \frac{{v_c}^2}{f_v} + \frac{h^2 sin^2\theta}{4 f_v} \\ \delta_1 f_u cos\theta & \delta_1 f_v sin\theta & \delta_2 h - \delta_1 (u_c cos\theta + v_c sin\theta) & \frac{h(u_c cos\theta sin\theta + v_c(1+sin^2\theta))}{f_v} \\ -\delta_1 \frac{f_u sin\theta}{h} & \delta_1 \frac{f_v cos\theta}{h} & \delta_1 \frac{(u_c sin\theta - v_c cos\theta)}{h} & \frac{-u_c sin^2\theta + v_c cos\theta sin\theta}{f_v} \end{pmatrix}$$

$$\begin{pmatrix} -(f_u + \frac{{u_c}^2}{f_u} + \frac{h^2 cos^2\theta}{4 f_u}) & v_c \frac{f_u}{f_v} & \frac{u_c}{f} \\ -\frac{u_c v_c}{f_u} - h^2 \frac{cos\theta sin\theta}{4 f_u} & -u_c \frac{f_v}{f_u} & \frac{v_c}{f} \\ -\frac{h}{f_u}(u_c(1+cos^2\theta) + v_c cos\theta sin\theta) & h cos\theta sin\theta (\frac{f_u}{f_v} - \frac{f_v}{f_u}) & \frac{h}{f} \\ \frac{1}{f_u}(u_c cos\theta sin\theta - v_c cos^2\theta) & -\frac{f_u}{f_v} sin^2\theta - \frac{f_v}{f_u} cos^2\theta & 0 \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \quad \text{(E.5)}$$

avec $\delta_1 = \frac{Z_1 - Z_2}{Z_1 Z_2}$ et $\delta_2 = \frac{Z_1 + Z_2}{2 Z_1 Z_2}$.

Cette équation nous donne à l'équilibre $(u_c, v_c, h) = (0, 0, h^*)^T$:

$$\begin{pmatrix} \dot{u_c} \\ \dot{v_c} \\ \dot{h} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\delta_2 f_u & 0 & -\delta_1 \frac{h^* cos\theta}{4} & h^{*2} \frac{cos\theta sin\theta}{4 f_v} \\ 0 & -\delta_2 f_v & -\delta_1 \frac{h^* sin\theta}{4} & f_v + \frac{h^{*2} sin^2\theta}{4 f_v} \\ \delta_1 f_u cos\theta & \delta_1 f_v sin\theta & \delta_2 h^* & 0 \\ -\delta_1 \frac{f_u sin\theta}{h^*} & \delta_1 \frac{f_v cos\theta}{h^*} & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} -(f_u + \frac{h^{*2} cos^2\theta}{4 f_u}) & 0 & 0 \\ -h^{*2} \frac{cos\theta sin\theta}{4 f_u} & 0 & 0 \\ 0 & h^* cos\theta sin\theta (\frac{f_u}{f_v} - \frac{f_v}{f_u}) & \frac{h^*}{f} \\ 0 & -\frac{f_u}{f_v} sin^2\theta - \frac{f_v}{f_u} cos^2\theta & 0 \end{pmatrix} \cdot \begin{pmatrix} V_X \\ V_Y \\ V_Z \\ \Omega_X \\ \Omega_Y \\ \Omega_Z \\ \dot{f} \end{pmatrix} \quad \text{(E.6)}$$

Annexe F

Modèle et calibration du zoom

Afin de pouvoir utiliser une caméra muni d'un zoom, il est important de bien connaître ses caractéristiques intrinsèques. Pour cela une phase de calibrage est nécessaire. Pour cela nous utilisons les techniques développées dans [126, 129].

F.0.5 Modélisation d'une caméra avec zoom

Elles se réfèrent à une modélisation de la caméra par un système optique à lentille épaisse. De ce modèle, nous pouvons écrire la relation de conjugaison (F.1) et l'expression du cercle de flou (F.2).

$$\frac{1}{f_z} = \frac{1}{u} - \frac{1}{w} \tag{F.1}$$

$$d = D f_f (\frac{1}{f_z} - \frac{1}{u} - \frac{1}{f_f}) \tag{F.2}$$

Quelques hypothèses réalistes :

- que l'objet est relativement lointain, i.e $Z \gg t$ et $Z \gg f_z$
- mise au point à peu près correcte : $D \gg d$

nous donnent :

$$d = D f_f (\frac{1}{f_z} - \frac{1}{u} - \frac{1}{f_f}) \tag{F.3}$$

Par ailleurs, nous avons :

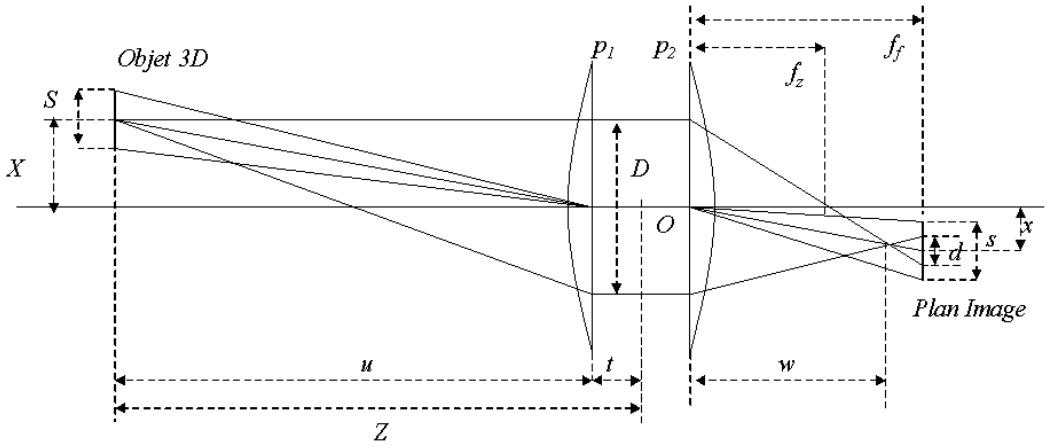
$$\frac{x}{f_z} = K \frac{X}{Z} \tag{F.4}$$

avec K le grandissement angulaire tel que :

$$K = \frac{K_a f_z + K_b}{K_c f_z + K_d} \tag{F.5}$$

$$\begin{aligned} focus &= g_f f_f + o_f \\ zoom &= g_z f_z + o_z \end{aligned} \tag{F.6}$$

En combinant les équations (F.3), (F.4), (F.5) et (F.6), en posant $x = f_g.X/Z$ et avec un développment de Taylor au premier ordre, nous obtenons alors un modèle



- O le centre optique de la lentille,
- p_1 et p_2 les plans principaux,
- f_z la **distance focale** et f_f le **focus** (distance entre le plan p_2 et le plan rétinien) exprimés en mètre,
- u la distance de l'objet 3D au plan principal p_1, t la distance de p_1 au centre optique, et $Z = t+u$,
- w la distance du plan p_2 au plan image,
- D le diamètre de l 'ouverture de la lentille et d le diamètre du cercle flou,
- S la taille de l 'objet 3D et s la taille de son image.

FIG. F.1 – Géométrie d'un système optique à lentille épaisse

linéaire du premier ordre pour la focale globale en fonction des données zoom et focus du construteur :

$$f_g = \delta.zoom + \beta.focus + \gamma \tag{F.7}$$

D'où, nous obtenons les relations suivantes :

$$\begin{aligned} \alpha_u &= k_u(zoom + \beta'.focus + \gamma') \\ \alpha_v &= k_v(zoom + \beta'.focus + \gamma') \end{aligned} \tag{F.8}$$

Par ailleurs, nous supposons l'invariance de k_u, k_v et θ. Cette hypothèse est vérifiée par la calibration.

La calibration consiste essentiellement à utiliser un modèle sténopé à différentes positions du zoom et du focus. Pour cela une mire de calibrage est utilisée. Elle comporte 18 points dont les positions relatives sont connues précisement dans son plan.

La détermination du point principal est la phase la plus importante du calibrage, puisque l'utilisation du modèle sténopé pour le calcul des autres caractéristiques nécessite sa connaissance. Elle repose sur la démonstration que le vecteur passant par le centre

optique d'une caméra et le projeté du point de fuite d'un ensemble de droites parallèles, est colinéaire au vecteur directeur de ces droites. Pour une séquence de zoom, le déplacement virtuel de l'objet se fait le long de l'axe optique. Par conséqunet, le projeté du point de fuite n'est rien d'autre que le point (u_0, u_0). Le calcul des autres caractéristiques permet de calculer et de vérifier notre modèle linéaire.

F.0.6 Application à la caméra EVI-G21

Dans notre application, nous utilisons actuellement une caméra EVI-G21 de SONY (cf. fig. F.2). Elle a la particularité d'intégrer la platine site et azimut et la caméra pilotée en zoom. Elle permet notamment de supposer que le centre optique est confondu avec le centre des rotations de la platine.

FIG. F.2 – camera EVI-G21

Les techniques de calibrage décrites ci-dessus permettent d'en évaluer la position(u_0,v_0) du point principal et les longueurs (en pixels) des focales. Ces longueurs sont fonctions des valeurs en *zoom* et en *focus* fournies par la caméra, suivant la relation linéaire : $a_* \cdot focus + b_* \cdot zoom + c_* \cdot f_* + d_* = 0$, $(* = u, v)$.

u_0	379.97 ± 0.05 pixels
v_0	295.53 ± 0.05 pixels

-	$f_u(pixels)$	$f_v(pixels)$
a_*	-0.05049	-0.07326
b_*	0.1208	0.1207
c_*	-0.9914	-0.9900
d_*	1028	1055

FIG. F.3 – Résultats du calibrage du zoom de la caméra EVI-G21

Bibliographie

[1] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. In *Proceedings of the International Conference on Robotics and Automation, San Francisco, USA*, pages 2070–2075, April 2000.

[2] N. Andreff, R. Horaud, and B. Espiau. Robot hand-eye calibration using structure from motion. *International Journal of Robotics Research*, 20(3) :228–248, March 2001.

[3] D. Angelin. Satellites brideurs. *Télérama*, (2755) :24, 30 october 2002.

[4] S. Araki, T. Matsuoka, N. Yokoyo, and H. Takemura. Real-time tracking of multiple moving object contours in a moving camera sequence. *IEICE Transactions on Information and Systems*, E86-D (7) :1583–1591, 2000.

[5] M. Aron, B. Biecheler, and J.-F. Peytalin. Temps intervéhiculaire sur autoroute. *RTS (Recherche, transports, sécurité)*, 64, july-September 1999.

[6] Arulampalam, Maskell, Gordon, and Clapp. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing, Special Issue on Monte Carlo Methods*, 50(2) :174–188, February 2002.

[7] R. Aufrère. *Reconnaissance et suivi de route par vision artificielle, application à l'aide à la conduite*. PhD thesis, Ecole doctorale SPI, LASMEA, Université Blaise Pascal, Clermont-Ferrand, France, June 2001.

[8] R. Aufrère, F. Marmoiton, R. Chapuis, F. Collange, and J. P. Dérutin. Détection de route et suivi de véhicules par vision pour l'acc. *Traitement du Signal*, 17 :233–247, 2000.

[9] S. Avidan. Support vector tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, Hawaii, December 2001.

[10] J. Bandoh and K. Aoyama. Traffic concepts in asia (e.g. tokyo) : Its developed by japanese police. In *Proceeding of the World Engineers'Convention : Professional Congress Mobility*, Hannover, Germany, June 2000. Verein Deutscher Ingenieure.

[11] P. H. Batavia, D. A. Pomerleau, and C. E. Thorpe. Detecting overtaking vehicles using implicit optical flow. In *Proceedings IEEE Intelligent Transportation Systems Conference*, pages 729–734, Boston, MA, 1997.

[12] J. C. Becker and A. Simon. Sensor and Navigation Data Fusion for an Autonomous Vehicle. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 156–161, Detroit, USA, october 2000.

[13] M. Ben-Ezra, S. Peleg, and M. Werman. Real-time motion analysis with linear programming. *Computer Vision and Image Understanding*, 78(1) :32–52, April 2000.

[14] A. Bensrhair, M. Bertozzi, A. Broggi, A. Fascioli, S. Mousset, and G. Touliminet. Stereo vision-based feature extraction for vehicle detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, June 2002.

[15] A. Bensrhair, M. Bertozzi, A. Broggi, P. Miché, S. Mousset, and G. Touliminet. A cooperative approach to vision-based vehicle detection. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 209–214, Oakland, Canada, August 2001.

[16] F. Berry, P. Martinet, and J. Gallice. Real time visual servoing around a complex object. *IEICE Transactions on Information and systems, Special issue on Machine Vision Applications, IEICE'2000*, pages 1358–1368, 2000.

[17] M. Bertozzi and A. Broggi. GOLD : a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7(1) :62–81, january 1998.

[18] M. Bertozzi, A. Broggi, and A. Fascioli. An extension to the inverse perspective to handle non flat roads. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 305–310, Stuttgart, Germany, October 1998.

[19] M. Bertozzi, A. Broggi, A. Fascioli, and G. Conte. Stereo-vision system performance analysis. *Enabling Technologies for the PRASSI Autonomous Robot*, pages 68–73, january 2002.

[20] M. Betke, E. Haritaoglu, and L. S. Davis. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications*, 12(2) :69–83, September 2000.

[21] B. Bhanu and W. Burger. Qualitative understanding of scene dynamics for moving robots. *International Journal in Robotics Research*, 9(6) :74–90, 1990.

[22] R. Bishop. A survey of intelligent vehicle applications worldwide. In *Proceedings IEEE International Conference on Intelligent Vehicles*, pages 25–30, Dearborn, USA, october 2000.

[23] M. J. Black and A. D. Jepson. Eigentracking : Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1) :63–84, 1998.

[24] M. Bober and J. Kittler. A hough transform based hierarchical algorithm for motion segmentation and estimation. In *International Workshop on Time -Varying Image Processing and Moving Object Recognition*, pages 335–342, Florence, Italy, july 1993.

[25] S. Bohrer, T. Zielke, and V. Freiburg. An integrated obstacle detection framework for intelligent cruise control on motorways. In *Proceeding Intelligent Vehicles '95 Symposium*, pages 276–281, Detroit, USA, 1995.

[26] F. Bérard. *Vision par ordinateur pour l'interaction homme-machine fortement couplé*. PhD thesis, Université Joseph Fourier Grenoble I, Grenoble, France, 1999.

[27] A. Broggi, M. Bertozzi, and A. Fascioli. The 2000km test of the ARGO vision-based autonomous vehicle. *IEEE Intelligent Systems*, pages 55–64, January-February 1999.

[28] A. Broggi, M. Bertozzi, and A. Fascioli. Self-calibration of a stereo vision system for automotive applications. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 4, pages 3698–3703, Seoul, Korea, may 2001.

[29] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi. Visual perception of obstacles and vehicles for platonning. *IEEE Transactions on Intelligent Transportation Systems*, 1(3) :164–176, 2000.

[30] C. Burges. A tutorial on support vector machines for pattern recognition. In e. Usama Fayyad, editor, *Data Mining and Knowledge Discovery*, pages 1–43, 1998.

[31] J. C. Burie and J. G. Postaire. Enhancement of the road safety with a stereovision system based on linear cameras. In *IEEE Intelligent Vehicle Symposium*, Tokyo, Japan, september 1996.

[32] R. Chapuis. *Suivi de primitives image, application à la conduite automatique sur route*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France, January 1991.

[33] R. Chapuis, A. Potelle, J. Brame, and F. Chausse. Real time vehicle trajectory supervision on highway. *International Journal of Robotics Research*, 14(6) :531–542, 1995.

[34] F. Chaumette. *La relation vision commande : théorie et application à des tâches robotiques*. PhD thesis, IRISA/INRIA, Rennes, France, July 1990.

[35] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control*, 237 :66–78, 1998.

[36] F. Chaumette. A first step towards visual servoing using image moments. In *IEEE International Conference on Intelligent Robots and Systems*, pages 378–383, Lausanne, Switzerland, October 2002.

[37] F. Chaumette and E. Marchand. A redundancy-based iterative approach for avoiding joint limits : Application to visual servoing. *IEEE Transactions on Robotics and Automation*, 17(5) :719–730, october 2001.

[38] N. Christianini and T. Showe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[39] X. Clady, C. Blanc, L. Trassoudaine, D. Aubert, F. L. Coat, G. Yahiaoui, F. Nashashibi, Abdelaziz, and S. Mousset. Etat de l'art en détection d'obstacles par perception. Technical Report PSI-INSA/R1.3-v1.0, ARCOS, PSI-INSA de Rouen, 2002.

[40] I. Cohen and G. Medioni. Detecting and tracking moving objects for video surveillance. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, volume II, pages 319–325, Fort Collins (CO), USA, June 1999.

[41] B. Coifman, D. Beymer, P. Mclauchlan, and J. Malik. A realtime computer vision system for vehicle tracking and traffic surveillance. *Transportation Research*, 4, august 1998.

[42] C. Collewet. *Contributions à l'élargissement du champ applicatif des asservissements visuels 2D.* PhD thesis, Université de Rennes 1, IRISA/INRIA, Rennes, France, february 1999.

[43] J. Collins. International view of mobolity : The american perspective. In *Proceeding of the World Engineers'Convention : Professional Congress Mobility*, pages 13–21, Hannover, Germany, June 2000.

[44] R. Collins and Y. Tsin. Calibration of an outdoor active camera system. In *Proceedings IEEE Computer Vision and Pattern Recognition (CVPR '99)*, pages 528–534, June 1999.

[45] A. Colmenarez and T. Huang. Face detection with information based maximum discrimination. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 782–787, 1997.

[46] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, SC, June 13-15 2000.

[47] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appareance models. In *Proceedings IEEE European Conference on Computer Vision*, volume 2, pages 484–498, 1998.

[48] T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper, and J. Graham. Building and using flexible models incorporating grey-level information. In *Proceedings IEEE International Conference on Computer Vision*, pages 242–246, Berlin, Germany, 1993.

[49] L. Cordesses, P. Martinet, B. Thuilot, and M. Berducat. Gps based control of a land vehicle. In *International Symposium on Automation and Robotics in Construction*, pages 41–46, Madrid, Spain, September 1999.

[50] A. Cretual and F. Chaumette. Dynamic stabilization of a pan and tilt camera for submarine image visualization. *Computer Vision and Image Understanding*, 79(1) :47–65, 2000.

[51] J. Crowley and J. Bedrune. Integration and control of reactive visual processes. In *Proceedings European Conference on Computer Vision*, pages 47–58, Stockholm, Sweeden, may 1994.

[52] T. K. D. Cremers and C. Schnörr. Nonlinear shape statistics in mumford-shah based segmentation. In S. L. A. Heyden et al. (Ed.), editor, *Proceedings 7th European Conf. on Computer Vision*, volume 2351, pages 93–108, Copenhagen, June 2002.

[53] Y. Dai and Y. Nakano. Face-texture model based on sgld and its application. *Pattern Recognition*, 29 :1007–1017, 1996.

[54] P. Daviet and M. Parent. Platooning for small public urban vehicles. In *fourth International Symposium on Experimental Robotics*, pages 345–354, Stanford, California, USA, July 1995.

[55] F. Dellaert and C. Thorpe. Robust car tracking using kalman filtering and bayesian templates. In *Proceedings of SPIE : Intelligent Transportation Systems*, volume 3207, pages 72–83, Pittsburg (PA), USA, October 1997.

[56] S. Denasi and G. Quaglia. Early obstacle detection using region segmentation and model based edge grouping. In *Proceedings of IEEE International Conference on Intelligent Vehicles*, pages 257–262, Stuttgart, Germany, october 1998. IV'98.

[57] E. D. Dickmanns. The development of the sense of vision for ground vehicles over the last decade. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[58] K. Dietmayer, J. Sparbert, and D. Streller. Object tracking in traffic scenes from range images. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2001, IV 2001*, pages 25–30, Tokyo, Japan, june 2001.

[59] M. Dubuisson, S. Lakshmanan, and A. Jain. Vehicle segmentation and classication using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3) :293–308, March 1996.

[60] A. Elgammal, R. Duraiswami, and L. S. Davis. Efficient kernel density estimation using the fast gauss transform with applications to segmentation and tracking, December 2001.

[61] A. Elouardi, S. Bouaziz, and R. Reynaud. Evaluation of an artificial cmos retina sensor for tracking system. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[62] T. Emura and M. Kumagai. A non-scanning ultrasonic sensor for driver assistant systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[63] R. Enciso, T. Viéville, and O. Faugeras. Approximation du changement de focale et de mise au point par une transformation affine à trois paramètres. *Traitement du Signal*, 11(5) :361–372, 1994.

[64] W. Enkelmann. Obstacle detection by evaluation of optical flow fields from image sequences. *Image and Vision Computing (UK)*, 9(3), 1991.

[65] B. Espiau. Visual servoing with zoom control. Technical Report 2613, INRIA, July 1995.

[66] T. Evgeniou, M. Pontil, C. Papageorgiou, and T. Poggio. Image representations and feature selection for multimedia database search. *IEEE Transactions in Knowledge and Data Engineering*, pages 346–356, may 2002.

[67] Y. Fang, I. Masaki, and B. Horn. Distance Range Based Segmentation in Intelligent Transportation System : Fusion of Radar And Binocular Stereo. In *Proceedings IEEE Intelligent Vehicles Symposium 2001*, pages 171–176, Tokyo, Japan, june 2001.

[68] J. Fayman, O. Sudarsky, and E. Rivlin. Zoom tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 2783–2788, Leuven, Belgium, May 1998. ICRA'98.

[69] R. Feraud, O. Bernier, J.-E. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1) :42–53, may 2001.

[70] R. Feris, V. Krueger, and R. J. Cesar. Efficient real-time face tracking in wavelet subspace. In *International Conference on Computer Vision, Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems*, pages 113–118, Vancouver, Canada, 2001.

[71] G. Flandin and E. M. F. Chaumette. Eye-in-hand / eye-to-hand cooperation for visual servoing. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 3, pages 2741–2746, San Francisco, USA, april 2000.

[72] G. L. Foresti, V. Murino, and C. S. Regazzoni. Vehicle recognition and tracking from road image sequences. *IEEE Transactions on Vehicular Technology*, 48(1) :301–318, january 1999.

[73] U. Franke. Real-Time Stereo Vision for Urban Traffic Scene Understanding. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 273–278, Detroit, USA, Oct. 2000.

[74] U. Franke, F. Bottiger, Z. Zomoter, and D. Seeberger. Truck platooning in mixed traffic. In *Proceedings Intelligent Vehicles'95 Symposium*, pages 1–6, Detroit, U.S.A., 1995.

[75] K. Fürstanberg, V. Willhoeft, and K. Dietmayer. New sensor for 360°vehicle surveillance-innovative approach to stop and go, lane assistance and pedestrian recognition. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2001, IV 2001*, pages 157–161, Tokyo, Japan, june 2001.

[76] J. Gangloff and M. de Mathelin. High speed visual servoing of a dof manipulator using multivariable predictive control. In *IEEE International Conference on Robotics and Automation*, pages 3236–3242, San Francisco, California, U.S.A., april 2000.

[77] D. Gavrila. Pedestrian detection from a moving vehicle. In *Proceedings IEEE European Conference on Computer Vision*, pages 37–49, Dublin, Germany, 2000. ECCV'00.

[78] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proceedings IEEE International Conference on Computer vision*, pages 87–93, Kerkyra, Greece, september 1999.

[79] D. M. gavrila, M. Kunert, and U. Lages. A multi-sensor approach for the protection of vulnerable traffic participants - the protector project. In *Proceedings of the IEEE Instrumentation and Measurement Technology Conference*, volume 3, pages 2044–2048, Budapest, Hongary, 2001.

[80] A. Gern, U. Franke, and P. Levi. Advanced Lane recognition - fusing vision and radar. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 45–51, Detroit, USA, Oct. 2000.

[81] M. Gleicher. Projective registration with difference decomposition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 331–337, San Juan, Puerto Rico, june 1997.

[82] C. Goerick, D. Noll, and M. Werner. Artificial neural networks in real time car detection and tracking applications. In *Proccedings International Conference on Engeneering Applications of Neural Networks (EANN'95)*, pages 335–343, Helsinki, april 1995.

[83] Q. Gu. and S. Li. Combining feature optimization into neural network based face recognition. In *Proceedings of The 15th International Conference on Pattern Recognition*, Barcelona, Catalonia, Spain, September 2000.

[84] S. R. Gunn and M. S. Nixon. A robust snake implementation; a dual active contour. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 19(1) :63–68, 1997.

[85] M. Haag and H.-H. Nagel. Beginning a transition from a local to a more global point of view in model-based vehicle tracking. In *Proceedings European Conference on Computer Vision*, volume 1, pages 812–827, Freiburg, Germany, 1998.

[86] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 20(10) :1025–1039, 1998.

[87] J. A. Hancock. *Laser Intensity-Based Obstacle Detection and Tracking*. PhD thesis, Carnegie Mellon university, The Robotics Institute, Pittsburg, Pennsylvania, january 1999.

[88] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, and W. von Seelen. An image processing system for driver assistance. In *IEEE International Conference on Intelligent Vehicles 1998*, pages 481–486, Stuttgart, Germany, october 1998. IV'98.

[89] M. Hariyama, T. Takeuchi, and M. Kameyama. Reliable Stereo Matching for Highly-Safe Intelligent Vehicles and Its VLSI Implementation. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 128–133, Detroit, USA, Oct. 2000.

[90] C. Harris and M. Stephens. A combined corner and edge detector. In *the 4th Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

[91] E. Hayman, L. de Agapito, I. D. Reid, and D. W. Murray. The role of self-calibration in euclidean reconstruction from two rotating and zooming cameras. In *Proceedings European Conference on Computer Vision*, Dublin, Germany, 2000.

[92] E. Hayman, I. D. Reid, and D. W. Murray. Zooming while tracking using affine transfer. In *Proceedings of the British Machine Vision Conference*, pages 395–404, September 1996.

[93] E. Hayman, T. Thorhallson, and D. W. Murray. Zoom-invariant tracking using points and lines in affine views : an application of the affine multifocal tensors. In *Proceedings of the International Conference on Computer Vision*, pages 269–275, Corfu, Greece, September 1999.

[94] J. K. Hedrick. The technology of automated highway systems. In *Proceeding of the World Engineers'Convention : Professional Congress Mobility*, pages 199–215, Hannover, Germany, June 2000.

[95] E. Hejmas and B. K. Low. Face detection : A survey. *Computer Vision and Image Understanding*, 83(3) :236–274, september 2001.

[96] R. Horaud, F. Dornaika, and B. Espiau. Visually guided object grasping. *IEEE Transactions on Robotics and Automation*, 4(14) :525–532, aout 1998.

[97] K. Hosoda, H. Moriyama, and M. Asada. Visual servoing utilizing zoom mechanism. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 178–183, Nagoya, Japan, May 1995. ICRA'95.

[98] Z. Hu and K. Uchimura. Tracking Cycle : A New Concept for Simultaneously Tracking of Multiple Moving Objects in a Typical Traffic Scene. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 233–239, Detroit, USA, Oct. 2000.

[99] C. Huang and C. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12) :1435–1444, 1992.

[100] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5) :651–670, October 1996.

[101] F. Huynh. *Vision-based manipulation for supervision and grasping tasks on a static or mobile object*. PhD thesis, Paul Sabatier University, Toulouse, France, 1998.

[102] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proceedings European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[103] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1) :5–28, 1998.

[104] S.-H. Jeng, H. Y. M. Yao, C. C. Han, M. Y. Chern, and Y. T. Liu. Facial feature detection using geometrical face model : An efficient approach. *Pattern Recognition*, 31(3) :273–282, 1998.

[105] S. Jouannin. *Association et fusion de données : application au suivi et à la localisation d'obstacles par RADAR à bord d'un véhicule routier intelligent*. PhD thesis, Ecole doctorale SPI, Laboratoire des sciences et Matériaux pour l'Electronique, et d'Automatique, Clermont Ferrand, France, 1999.

[106] F. Jurie. Solution of the simultaneous pose and correspondence problem using gaussian error model. *Computer Vision and Image Understanding*, 73(3) :357–373, March 1999.

[107] F. Jurie and M. Dhome. Un algorithme efficace de suivi d'objets dans des séquences d'images. In *Congrès francophone RFIA*, volume 1, pages 537–546, Paris, February 2000.

[108] F. Jurie and M. Dhome. A simple and efficient template matching algorithm. In *IEEE International Conference on Computer Vision*, pages 544–549, Vancouver, Canada, July 2001.

[109] T. Kalinke, C. Tzomakas, and W. v. Seelen. A texture-based object detection and an adaptive model-based classification. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 143–148, Stuttgart, Germany, October 1998.

[110] T. Kalinke and W. v. Seelen. A neural network for symmetry-based object detection and tracking. In *Mustererkennung 1996*, pages 627–634, 1996.

[111] R. E. Kalman. A new approach to linear filtering and prediction problems. In *Transactions ASME*, volume 82, pages 34–45, 1973.

[112] T. Kato, T. Tanizaki, T. Ishii, H. Tanaka, and Y. Takimoto. 76 GHz high performance radar sensor featuring fine step scanning mechanism utilizing NRD technology. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2001, IV 2001*, pages 163–170, Tokyo, Japan, june 2001.

[113] S. Kawamata, N. Ito, S. Katahara, and M. Aoki. Precise position and speed detection from slit camera image of road surface marks. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[114] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette. Visual servoing in robotics scheme using a camera/laser-stripe sensor. *IEEE Transactions on Robotics and Automation*, 12(5) :743–750, October 1996.

[115] D. Khadraoui, R. Rouveure, C. Debain, P. Martinet, P. Bonton, and J. Gallice. Vision based control in driving assistance of agricultural vehicles. *International Journal of Robotics Research*, 17(10) :1040–1054, October 1998.

[116] W. Kim and C. Brislawn. Plume detection and tracking in video data. Technical report, Los Alamos National Laboratory, 2000.

[117] A. Kirchner and C. Ameling. Integrated Obstacle and Road Tracking using a Laser Scanner. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 162–167, Detroit, USA, Oct. 2000.

[118] C. Knoeppel, B. Michaelis, and A. Schanz. Robust Vehicle Detection at Large Distance Using Low Resolution Cameras. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 267–272, Detroit, USA, Oct. 2000.

[119] D. Koller, T. Luong, and J. Malik. Binocular stereopsis and lane marker flow for vehicle navigation : lateral and longitudinal control. Technical Report UCB/CSD-94-804, California Institut of Technology, march 1994.

[120] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *IEEE European Conference on Computer Vision*, pages 186–196, Stockholm, Sweeden, may 1994.

[121] H. Kuroda, S. Kuragaki, T. Minowa, and K. Nakamura. An adaptative cruise control using a millimeter wave radar. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 168–172, Stuttgart, Germany, October 1998.

[122] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection in stereo vision on non flat road geometry through v-disparity representation. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[123] D. Langer. *An Integrated MMW Radar System for Outdoor Navigation*. PhD thesis, Carnegie Mellon university, The Robotics Institute, Pittsburg, Pennsylvania, USA, 1997.

[124] J. Langheim. CARSENSE-New environment sensing for advanced driver assistance systems. In *Proceedings IEEE Intelligent Vehicles Symposium 2001*, pages 89–94, Tokyo, Japan, june 2001.

[125] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions Pattern Analysis and Machine Intelligent*, 19(7) :743–756, 1997.

[126] J. Lavest, G. Rives, and M. Dhome. 3d-reconstruction by zooming. *IEEE Transactions on Robotics and Automation*, 2(9) :196–207, April 1993.

[127] J. V. Leuven, M. B. V. Leeuwen, and F. C. A. Groen. Real-time vehicle tracking in image sequences. In *Proceedings IEEE Instrumentation and Measurement Conference*, Budapest, Hungary, 2001.

[128] M. Lew and N. Huijsmans. Information theory and face detection. In *Proceedings of the International Conference on Pattern Recognition*, pages 601–605, Vienna, Austria, August 1996.

[129] M. Li and J. Lavest. Some aspects of zoom lens camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 18 :1105–1110, November 1996.

[130] Z. Liang and C. Thorpe. Stereo- and neural network-based pedestrian detection. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 298–303, Tokyo, Japan, Spring, 1999.

[131] C. Lin and W. Lin. Extracting facial features by an inhibitory mechanism based on gradient distributions. *Pattern Recognition*, 29 :2079–2101, 1996.

[132] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Transactions Neural Networks*, 8 :114–131, january 1997.

[133] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2) :113–122, 1992.

[134] M. Lützeler and E. Dickmanns. Road recognition with marveye. In *IEEE International Conference on Intelligent Vehicles*, volume 1, pages 341–346, Stuttgart, Allemagne, october 1998.

[135] J. Ma and S. Olsen. Depth from zooming. *Journal of the American Optical Society*, 10(7) :1883–1890, 1990.

[136] E. Malis. Vision-based control invraint to camera intrinsic parameters : stability analysis and path tracking. In *IEEE International Conference on Robotics and Automation, ICRA'02*, pages 217–222, Washington, USA, May 2002.

[137] E. Malis, J.-J. Borrelly, and P. Rives. Intrinsics-free visual servoing with respect to straight lines. In *IEEE International Conference on Intelligent Robots and Systems*, pages 384–389, Lausanne, Switzerland, October 2002.

[138] E. Malis, F. Chaumette, and S. Boudet. 2D 1/2 visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2) :238–250, april 1999.

[139] S. Mallat. A theory for multiresolution signal decomposition : the wavelet representation. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 11 :674–693, 1989.

[140] E. Marchand, F. Chaumette, and A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'96*, volume 3, pages 1083–1090, Osaka, Japan, nov 1996.

[141] E. Marchand and G. Hager. Dynamic sensor planning in visual servoing. In *IEEE International Conference on Robotics and Automation, ICRA'98*, volume 3, pages 1988–1993, Louvain, Belgique, may 1998.

[142] F. Marmoiton. *Detection et suivi par vision monoculaire d'obstacles mobiles cooperatifs a partir d'un vehicule experimental automobile*. PhD thesis, Ecole Doctorale Sciences Pour l'Ingénieur de Clermont-Ferrand, Université Blaise Pascal, Clermont-Ferrand, France, January 2000.

[143] P. Martinet. *HDR in visual servoing*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France, January 1999.

[144] N. Matthews, P. An, D. Charnley, and C. Harris. Vehicle detection and recognition in greyscale imagery. In *2nd International Workshop on Intelligent Autonomous Vehicles*, pages 1–6, Helsinki, 1995. IFAC.

[145] M. Mirmehdi, P. Palmer, and J. Kittler. Towards optimal zoom for automatic target recognition. In *Proceedings of the Tenth Scandinavian Conference in Image Analysis*, volume 1, pages 447–453, Lappeenranta, Finland, June 1997.

[146] B. Moghaddam and A. Pentland. A subspace method for maximum likelihood target detection. Technical report, M.I.T. Media Laboratory Perceptual Computing Section, 1995.

[147] Y. Mézouar. *Planification de trajectoires pour l'asservissement visuel*. PhD thesis, Université de Rennes 1, IRISA/INRIA, Rennes, France, Novembre 2001.

[148] H. Nanda and L. Davis. Probabilistic template based pedestrian detection in infrared videos. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[149] S. K. Nayar, S. A. Nene, and H. Murase. Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation on Vision-Based Control of Robot Manipulators*, 12(5) :750–758, october 1996.

[150] M. Nishigaki, M. Saka, T. Aoki, H. Yuhara, and M. Kawai. Fail Output Algorithm of Vision Sensing. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 581–584, Detroit, USA, Oct. 2000.

[151] D. Oberkampf, D. DeMenthon, and L. Davis. Iterative pose estimation using coplanar feature points. *CVGIP : Image Understanding*, 63 :495–511, 1996.

[152] N. Oliver, A. Pentland, and F. Berard. LAFTER : Lips and face real time tracker. In *Proceedings IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, pages 123–129, Puerto Rico, June 1997.

[153] C. Olson and D. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *IEEE Transactions on Image Processing*, 6(1) :103–113, january 1997.

[154] E. Osuna, R. Freund, and F. Girosi. Support vector machines : Training and applications. Technical Report AIM-1602, MIT A. I. Lab., 1997.

[155] C. Papageorgiou and T. Poggio. A pattern classification approach to dynamical object detection. In *International Conference on Computer Vision*, pages 1223–1228, Corfu , Greece, September 1999.

[156] C. P. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *Proceedings of Intelligent Vehicles*, pages 241–246, Stuttgart, Germany, October 1998.

[157] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of International Conference on Computer Vision*, pages 555–562, Bombay, India, January 1998.

[158] C. Papageorgious and T. Poggio. A trainable object detection system : Car detection in static images. Technical report, MIT AI Memo 1673 (CBCL Memo 180), 1999.

[159] D. Paulus and G. Schmidt. Approaches to Depth Estimation from Active Camera Control. In *Speech and Image Understanding*, pages 281–290, Ljubljana, Slovenien, 1996.

[160] M. Pellkofer and E. Dickmanns. Ems-vision : Gaze control in autonomous vehicles. In *IEEE International Conference on Intelligent Vehicles*, pages 296–301, Dearborm, USA, october 2000.

[161] H. Pfannschmidt. Safety measures in transportation systems. In *Proceeding of the World Engineers'Convention : Professional Congress Mobility*, pages 153–176, Hannover, Germany, June 2000.

[162] D. Pomerleau. RALPH : Rapidly adaptating lateral position handler. In *Proceedings of the Intelligent Vehicles'95 Symposium*, pages 506–511, Detroit, U.S.A., october 1995.

[163] V. Rehrmann. Object oriented motion estimation in color image sequences. In *Proceedings European Conference on Computer Vision*, volume 1, Freiburg, Germany, 1998.

[164] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators : The generalized symmetry transform. *International Journal Computer Vision, Special Issue on Purposive Vision*, 14 :119–130, 1995.

[165] D. Reisfeld and Y. Yeshurun. Preprocessing of face images - detection of features and pose normalization. *Computer Vision and Image Understanding*, 71(3) :413–430, 1998.

[166] RIEGL. Laser mirror scanner lms-z210-60 complement to the user's manual lms-z210(-ht). Technical report, RIEGL Laser Measurement Systems GmbH, 2002.

[167] S. Rémy. *Etalonnage d'un système de vision embarqué*. PhD thesis, Ecole Doctorale SPI, LASMEA, Université Blaise Pascal, Clermont-Ferrand, France, july 1998.

[168] D. Roth, M. Yang, and N. Ahuja. A snow-based face detector. In *Advances in Neural Information Processing Systems 12 (NIPS-12)*, december 1999.

[169] C. Rother and H.-H. Nagel. Analyse initialer positionsschätzungen bei der bildfolgenauswertung. In *Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM) -Symposium*, pages 189–196, Bonn, Germany, September 1999.

[170] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1) :23–38, January 1998.

[171] S. Rüping. mysvm-manual. Technical report, University of Dortmund, Lehrstuhl Infomatik 8, 2000.

[172] Y. Ruichek, H. Issa, and J.-G. Postaire. Genetic Approach for Obstacle Detection using Linear Stereo Vision. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 261–266, Detroit, USA, Oct. 2000.

[173] P. Sahoo, S. Soltani, and A. Wong. A survey of thresholding techniques. *Computer Vision Graphics Image*, 41 :233–260, 1988.

[174] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control. The task function approach*. ISBN 0-19-8538057. Clarendon Press, Oxford, 1991.

[175] A. Sanderson and L. Weiss. Image-based visual servo control using relational graph error signals. In *Proceddings IEEE International Conference on Cybernetics and Society*, pages 1074–1077, Cambridge, MA, USA, october 1980.

[176] P. Sayd, R. Chapuis, R. Aufrère, and F. Chausse. A dynamic vision algorithm to recover the 3d shape of a non-structured road. In *IEEE International Conference on Intelligent Vehicles*, volume 1, pages 80–86, Stuttgart,Allemagne, october 1998.

[177] B. Scassellati. Eye finding via face detection for a foveated active vision system. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference*, pages 969–976, Madison, Wisconsin, USA, june 1998.

[178] H. Schneiderman and T. Kanade. Probalistic madeling of local appearance and spatial relationships for object recognition. In *Computer Vision and Pattern Recognition*, pages 45–51, Santa-Barbara, CA, 1998. Proceedings IEEE Conference on Computer Vision and Pattern Recognition.

[179] H. Schneiderman and T. Kanade. A statistical model for 3d object detection applied to faces and cars. In *Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, USA, June 2000. Proceedings IEEE Conference on Computer Vision and Pattern Recognition.

[180] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 17(6) :545–561, June 1995.

[181] W. B. Seales. Measuring time-to-contact using active camera control. In *Proceedings 6th International Conference on Computer Analysis of Images and Patterns*, pages 944–949, Prague, September 1995. CAIP'95.

[182] Y. Seo and K. Hong. About the self-calibration of a rotating and zooming camera : Theory and practice. In *Proceedings IEEE International Conference on Computer vision*, pages 183–188, Corfu, Greece, 1999.

[183] R. Sharma and S. Hutchinson. Optimizing hand/eye configuration for visual-servo systems. In *IEEE International Conference on Robotics and Automation*, pages 172–177, Nagoya, Japan, may 1995.

[184] N. Shimomura and K. Fujimoto. An Algorithm for Distinguishing the Types of Objects on the Road using laser Radar and Vision. In *Proceedings IEEE Intelligent Vehicles Symposium 2001*, Tokyo, Japan, june 2001.

[185] J. W. Sinko and R. C. Galijan. An evolutionary automated highway system concept based on GPS. In *International Technical Meeting of the Satellite Division of the Institute of Navigation*, pages 535–544, Kansas City, Missouri, USA, September 1996.

[186] C. Smith, C. Richards, S. Brandt, and N. Papanikolopoulos. Visual tracking for intelligent vehicle-highway systems. *IEEE Transactions on Vehicular Technology*, 46 :732–743, 1996.

[187] S. M. Smith and J. M. Brady. Asset-2 : Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8) :814–820, 1995.

[188] A. Smolic, J.-R. Ohm, and T. Sikora. Object-based global motion estimation using a combined feature matching and optical flow approach. In *Proceedings of International Workshop onVery Low Bitrate Video Coding*, pages 254–260, Urbana, USA, October 8-9 1998.

[189] B. Steux. *^{RT}Maps, un environnement logiciel dédié à la conception d'applications embarqués temps-réel. Utilisation pour la détection automatique de véhicules par fusion radar/vision.* PhD thesis, Ecole des Mines de Paris, Paris, France, December 2001.

[190] C. Stiller and J. Konrad. Estimating motion in image sequences : A tutorial on modeling and computation of 2d motion. *IEEE Signal Processing Magazine*, 16 :70–91, July 1999.

[191] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1) :39–51, 1998.

[192] I. Takahashi and K. Deguchi. Image-based control of robot and target object motions by eigen space method. In *IAPR Workshop on Machine Vision Applications*, pages 95–101, Chiba, Japan, november 1998.

[193] S. Terakubo, M. Morii, and T. Kashiara. Development of a road obstacle sensor combining image processing and a laser radar. In *Proceedings of the IEEE Inter-*

national Conference on Intelligent Vehicles, volume 2, pages 521–526, Stuttgart, Germany, October 1998.

[194] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice. Position based visual servoing : keeping the object in the field of vision. In *IEEE International Conference on Robotics and Automation, ICRA'02*, pages 1624–1629, Washington, USA, May 2002.

[195] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91132, Pittsburgh : Carnegie Mellon University, School of Computer Science, 1991.

[196] L. Trassoudaine. *Solution multisensorielle temps réel pour la détection d'obstacles routiers*. PhD thesis, Université Blaise Pascal, Clermont-Ferrand, France, February 1993.

[197] S. Tsugawa, S. Kato, K. Tokuda, T. Matsui, and H. Fujii. An overview on demo 2000 cooperative driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2001, IV 2001*, pages 327–332, Tokyo, Japan, june 2001.

[198] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1) :71–86, 1991.

[199] C. Tzomakas and W. von Seelen. Vehicle detection in traffic scenes using shadows. Technical report, IRINI 98-06, Institut fur Neuroinformatik, Ruhr-Universitat Bochum, D-44780 Bochum, Germany, August 1998.

[200] T. Uebo, T. Kitagawa, and T. Iritani. Short range radar utilizing standing wave of microwave or millimeter wave. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2001, IV 2001*, Tokyo, Japan, june 2001.

[201] M. Unser. Texture classification and segmentation using waveletes frames. *IEEE Transactions on Image Processing*, 4(11) :1549–1560, november 1995.

[202] M. B. van Leeuwen and F. C. Groen. Vehicle detection with a mobile camera. Technical report, Computer Science Institut, University of Amsterdam, The Netherlands, 2001.

[203] V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, USA, 1998.

[204] Verein Deutscher Ingenieure, Hannover, Germany. *Proceeding of the World Engineers'Convention : Professional Congress Mobility*, June 2000.

[205] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 511–518, 2001.

[206] C. von Seelen and R. Bajcsy. Adaptive correlation tracking of targets with changing scale. *Oscar Firschein, editor, Reconnaisance, Surveillance, and Target Acquisition (RSTA) for the Unmanned Ground Vehicle. Morgan Kaufmann Publishers*, 1997.

[207] G. Wells, C. Venaille, and C. Torras. Promising research : vision-based robot positioning using neural networks. *Image and Vision Computing*, 14(10) :715–732, 1996.

[208] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. *Feature Selection for SVMs, in Advances in Neural Information Processing Systems*, volume 13. MIT Press, Cambridge, MA, USA, 2001.

[209] D. Willersinn and W. Enkelmann. Robust obstacle detection and tracking by motion analysis. In *IEEE Conference on Intelligent Transportation Systems*, pages 717–722, Boston, USA, november 1997.

[210] T. Williamson and C. Thorpe. Detection of small obstacles at long range using multibaseline stereo. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, volume 1, pages 311–316, Stuttgart, Germany, October 1998.

[211] R. G. Wilson. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Carnegie Mellon University, 1999. CMU-RI-TR-94-03.

[212] W. J. Wilson, C. C. W. Hulls, and G. S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5) :684–696, October 1996.

[213] M. Xie, L. Trassoudaine, J. Alizon, and J. Gallice. Road obstacle detection and tracking by an active and intelligent strategy. *Machine Vision and Applications*, 7 :165–177, 1994.

[214] F. Xu and K. Fujimura. Pedestrian detection and tracking with night vision. In *Proceedings of the IEEE Intelligent Vehicles Symposium 2002, IV 2002*, Versailles, France, june 2002.

[215] G. Yang and T. Huang. Human face detection in a complex background. *Pattern Recognition*, 27 :53–63, 1994.

[216] M. Yang, N. Ahuja, and D. Kriegman. Face detection using mixtures of linear subspaces. In *IEEE Conference on Automatic Face and Gesture Recognition*, pages 70–76, Los Alamitos, CA, USA, 2000.

[217] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :34–58, january 2002.

[218] Z. Yi, H. Y. Khing, C. C. Seng, and Z. X. Wei. Multi-ultrasonic Sensor Fusion for Mobile Robots. In *Proceedings IEEE Intelligent Vehicles Symposium 2000*, pages 387–391, Detroit, USA, Oct. 2000.

[219] K. Yow and R. Cipolla. Feature-based human face detection. *Image and Vision Computing (UK)*, 15(9) :713–735, 1998.

[220] X. Yu, S. Beucher, and M. Bilodeau. Lane segmentation and obstacle recognition by mathematical morphology. In *Proceedings Intelligent Vehicles'92 Symposium*, pages 166–170. 20, Detroit, USA, June/July 1992.

[221] A. Yuille, P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2) :99–111, 1992.

[222] B. Zana and C. LeMoine. Autant d'espèces, autant de monde. *La perception visuelle : un système de haute technologie. tdc - textes et documents pour la classe*, 817, 1er au 15 june 2001.

[223] W. B. Zhang and R. E. Parsons. An intelligent roadway reference system for vehicle lateral guidance/control. In *Proceeding of the 1990 American Control Conference*, San Diego, USA, May 1990.

[224] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. Face recognition : a literature survey. Technical Report UMD–TR4167, University of Maryland, USA, october 2000.

[225] T. Zielke, M. Brauckmann, and W. von Seelen. Intensity and edgebased symmetry detection applied to car-following. In *G. Sandini, editor, Proceedings 2nd European Conference on Computer Vision*, pages 865–873, Santa Margherita Ligure (Italy), 1992.