N^o d'ordre : D.U. 298

EDSPIC : 487

UNIVERSITÉ BLAISE PASCAL – CLERMONT II

Ecole Doctorale Sciences Pour l'Ingénieur

HABILITATION À DIRIGER DES RECHERCHES

Préparée au LASMEA, UMR 6602 CNRS / Université Blaise Pascal
(Laboratoire des Sciences des Matériaux pour l'Electronique, et d'Automatique)

Spécialité : Vision par Ordinateur

Présentée et soutenue publiquement par

Thierry Chateau

le 28 Septembre 2010

Contributions à l'estimation d'état dans des images

— Synthèse des travaux et activités scientifiques sur 2001 – 2009 —

Devant le jury composé de

Président :	Roger Mohr	Professeur émérite, ENSIMAG, Grenoble
Rapporteurs externes :	James Crowley	Professeur, Grenoble I.N.P.
	Eric Marchand	Professeur, IRISA, Université de Rennes 1
Rapporteur interne :	Michel Dhome	Directeur de Recherche CNRS – LASMEA
Examinateurs :	Vincent Lepetit	Professeur, CVLab, EPFL, Suisse
	Patrick Perez	Directeur de Recherche, Thomson, Rennes
	Jean-Thierry Lapresté	Professeur, ENSCCF, Clermont-Ferrand

Avant-propos

Ce document dresse une synthèse de mes activités de recherche sur la période 2001-2009, dans l'équipe ComSee du groupe GRAVIR du LASMEA. Il ne revient pas sur les travaux que j'ai réalisés durant ma thèse[1].
Les travaux que j'ai menés s'inscrivent dans dans le thème scientifique de la Vision par Ordinateur.
De manière générique, ils concernent l'étude et la mise en œuvre de méthodes d'estimation d'état dans une ou plusieurs images pouvant provenir, d'une, de plusieurs caméras, et/ou d'une séquence vidéo. Dans ce cadre, mes travaux ont plus particulièrement porté sur deux points clefs :

- ▷ le premier concerne la modélisation paramétrique du lien entre les mesures (ce que l'on observe) et l'état (ce que l'on cherche), et l'estimation, à partir d'un ensemble d'apprentissage, des paramètres associés au modèle,
- ▷ le second la modélisation stochastique de l'état par un ensemble d'échantillons, approximant sa distribution de probabilités, et les techniques d'exploration efficaces associées à cette approximation.

Le document est structuré en deux chapitres principaux reprenant ces deux contributions. Pour chaque chapitre, je propose, dans une première partie, une synthèse des outils utilisés pour modéliser la problématique générale, puis je présente une sélection des travaux les plus représentatifs que j'ai effectués autour de cette problématique.
Dans la dernière partie, je conclus et propose des pistes de recherches pouvant donner lieu à différents travaux tels que des mémoires de thèses, post-doctorants, ou projets de recherche.

[1] Ma thèse s'intitule : Détection robuste d'interface par fusion d'informations incertaines : application à l'aide au guidage d'engins agricoles. Elle s'est déroulée au CEMAGREF à Varennes sur Allier, sous la direction de Pierre, BONTON

Remerciements

Les résultats présentés dans ce manuscrit sont avant tout issus d'un travail collectif intense auquel a participé une grande partie des chercheurs et enseignants chercheurs de l'équipe GRAVIR du Lasmea, comme l'atteste le nombre de co-auteurs permanents avec lesquels j'ai eu la joie de publier sur la période 2000-2009[2]. Mes remerciements vont donc à tous ceux, secrétaires, techniciens, ingénieurs, chercheurs, enseignants, dirigeants, qui par leur action individuelle maintiennent le collectif Lasmea à un niveau d'excellence particulièrement élevé et reconnu ; et qui s'engagent, manches retroussées, dans la construction du futur institut Pascal.
Au delà de ces nombreuses et fructueuses collaborations intra et inter thèmes, je remercie plus particulièrement les collègues de l'équipe ComSee pour toutes les discussions passionnées qui ont contribuées à la synthèse présentée dans ce document.
Je tiens à remercier tout particulièrement tous les étudiants (doctorants, post-doctorants et stagiaires), qui ont travaillé sous ma responsabilité, et qui reconnaîtront leurs travaux dans ce manuscrit. Plutôt que d'insister sur leur rôle essentiel dans une structure de recherche, je préfère illustrer mon sentiment par l'histoire suivante :

Deux entreprises, dont une de notre pays, decident de faire une course d'aviron dans le but de montrer leur savoir-faire dans le domaine de la galvanisation des troupes. Les deux equipes s'entrainent dur. Lors de la premiere épreuve, les étrangers gagnent avec plus d'un kilomètre d'avance.
Les nôtres sont très affectés. Le management se réunit pour chercher la cause de l'échec. Une équipe d'audit constituée de seniors managers est désignée. Après enquête, ils constatent que notre équipe, qui est constituée de dix personnes n'a qu'un rameur, alors que l'équipe étrangère comporte un barreur et neuf rameurs. La direction décide de faire appel au service de consultants internes. Leur avis, entouré de précautions oratoires, semble préconiser l'augmentation du nombre de rameurs. Apres réflexion, la direction décide de procéder à une réorganisation. Il est décidé de mettre en place un manuel qualité, des procédures d'application, des documents de suivis...
Une nouvelle stratégie est mise en place, basée sur une forte synergie. Elle doit améliorer le rendement et la productivité grâce à ces modifications structurelles. On parle même de zéro défaut dans tous les repas de brainstorming. La nouvelle équipe constituée comprend maintenant : 1 directeur général d'aviron, 1 directeur adjoint d'aviron, 1 superviseur d'aviron, 1 consultant de gestion d'aviron, 1 contrôleur de gestion d'aviron, 1 chargé de communication d'aviron, 1 coordinateur d'aviron, 1 barreur et 1 rameur.
La course a lieu et notre équipe a 2 kilomètres de retard. Humiliée, la direction prend des décisions rapides et courageuses : elle licencie le rameur n'ayant pas atteint ses objectifs, vend le bateau et annule tout investissement. Avec l'argent économisé, elle récompense les managers et superviseurs en leur donnant une prime, augmente le salaire des directeurs et s'octroie une indemnité exceptionnelle de fin de mission.

Enfin, je termine cet exercice de style avec une pensée toute particulière pour mes parents, ma soeur, ma femme Sandrine et nos deux merveilleux biquets Gabin et Léna, sans qui ma vie n'aurait aucun sens.

[2] 23 Chercheurs, enseignants chercheurs ou ingénieurs du groupe GRAVIR, qui compte 32 permanent au 1er Juin 2010

Table des matières

Table des figures

Liste des tableaux

CHAPITRE

1

ACTIVITÉS SCIENTIFIQUES ET ADMINISTRATIVES

Ce chapitre présente une synthèse de mes activités scientifiques, d'enseignement et d'administration de la recherche. La première partie aborde les activités du Lasmea, du groupe Gravir (GRoupe Automatique : VIsion et Robotique du LASMEA) dont j'ai la co-responsabilité, et de l'équipe ComSee (*Computers that See*), que j'anime.

La seconde partie dresse la liste des encadrements auxquels j'ai participé, en détaillant, pour chacun d'entre eux, le sujet de recherche associé.

La troisième partie détaille la liste des publications que j'ai réalisées depuis l'année 2000.
la quatrième partie aborde les activités d'enseignement qui m'ont été confiées depuis ma nomination à Polytech' Clermont-Ferrand.
La dernière partie présente une synthèse des activités de recherche que j'ai menées au LASMEA, au sein de l'équipe ComSee du groupe Gravir. Ces travaux concernent l'estimation d'état dans des images.

1.1 Récapitulatif

Publications, brevets et codes APP

Le tableau ci-dessous synthétise l'ensemble des publications parues sur la période 2000-2009, pour lesquelles je suis auteur principal ou co-auteur. Il ne prend pas en compte les publications parues durant ma thèse, ainsi que les séminaires invités ou les journées du Gdr Isis auxquels j'ai participé.

	Total	Premier / co-auteur
Revues	11	36 % / 73 %
Congrès et ateliers internationaux	33	18 % / 82 %
Congrès et ateliers nationaux	26	19 % / 81 %
Actes	2	100 % / 0 %
Brevets et Codes APP	4	/

Encadrements

Le tableau suivant synthétise l'activité d'encadrement réalisée depuis 2000. Le tableau est divisé en trois niveaux. Pour chaque niveau, le nombre d'étudiants encadré est regroupé par taux d'encadrement. Les projets d'étudiants d'écoles d'ingénieurs ne sont pas comptabilisés ici. D'autre part, j'ai co-encadré deux post doctorants qui ne figurent pas dans ce tableau.

	Plus ou 50%	Moins de 50%	Total
Niveau thèse	4	4	8
Niveau Master 2	10	0	10
Niveau inférieur	7	0	7

Projets de recherche

Je participe ou j'ai participé à 1 projet européen, 9 projets nationaux et 4 collaborations de recherche ou prestations avec des industriels français.

Enseignements

Je donne ci-dessous le nombre d'heures de cours et de TDe (Travaux Dirigés d'expérimentation) que j'ai dispensés par année.

	2001	2002	2003	2004	2005	2006	2007	2008
Cours	32	30	38	63	97	82	74	72
TD	131	143	127	209	158	82	108	107
TP	88	136	144	112	100	84	108	96

Responsabilités pédagogiques

- ▷ Je suis responsable des stages de 2ème année (depuis 2006) et de 3ème année (depuis 2008) dans le département Génie Electrique de Polytech Clermont-Ferrand.
- ▷ Je suis responsable des parcours IV (Imagerie et vision) et RPM (Robotique et Perception Multisensorielle) du Master Informatique MSIR de l'Université Blaise Pascal.

Autres éléments

- ▷ Je suis co-responsable de l'équipe ComSee du Lasmea depuis 2005 et du groupe Gravir depuis 2009.
- ▷ J'ai siègé dans diverses commissions au niveau du LASMEA et des Universités Clermontoises (2ème vice président de la commission de spécialistes 61ème section de l'Université Blaise Pascal de 2002 à 2006).
- ▷ J'expertise des articles pour de revues et congrès, et pour des projets ANR.
- ▷ J'ai co-organisé 2 congrès ou *workshop*.
- ▷ Je suis membre du bureau de l'école doctorale science pour l'ingénieur de l'Université Blaise Pascal.

1.2 Présentation du Lasmea, du Groupe Gravir et de l'équipe ComSee

1.2.1 Lasmea

Le Lasmea rassemble des automaticiens, des physiciens et des électroniciens. La diversité des thématiques investiguées explique sa structuration en deux groupes de recherche :

- ▷ Gravir (GRoupe Automatique : VIsion et Robotique) ;
- ▷ Matelec (MATériaux pour l'Electronique, ELECtromagnétisme).

Le groupe Matelec développe des activités théoriques adressant :

- ▷ « les nanostructures de semiconducteurs » : modélisations et analyses expérimentales optiques associées du couplage lumière-matière en microcavités et structures quantiques ;
- ▷ « la photonique » : modélisations optiques et électromagnétiques des cristaux photoniques, modélisations électromagnétiques des métamatériaux et structures plasmoniques ;
- ▷ « la nanostructuration » : modélisation atomistique des structurations de surface hors équilibre.

1.2.2 Gravir

Historiquement, dans les années 80, des enseignants-chercheurs et chercheurs clermontois ont choisi de s'investir dans les problématiques, alors naissantes de la vision par ordinateur et du traitement d'images. Au fil des années, ces orientations scientifiques ont évolué et se sont renforcées pour donner naissance en 1996 au GRoupe Automatique : VIsion et Robotique du Lasmea.
Ce groupe, dont je suis co-responsable, compte aujourd'hui 67 personnes dont 27 enseignants-chercheurs et chercheurs, 31 doctorants, 5 emplois temporaires (CDD, ATER post-doc), et 5 personnes au service technique. Gravir est au plan national un des plus important effectif universitaire-CNRS rassemblé sous la bannière de la « Vision et Robotique ». L'activité scientifique globale du groupe est structurée en trois thèmes scientifiques :

- ▷ Persyst : Système de perception
- ▷ Comsee : Vision artificielle
- ▷ Rosace : Commande des systèmes robotiques

Le groupe Gravir comporte également une équipe technique et deux projets scientifiques majeurs :

- ▷ V2I : véhicules et infrastructures intelligents,
- ▷ M2I : machines et mécanismes innovants.

1.2.3 Comsee

L'équipe Comsee que j'anime est constituée (au 1er janvier 2009) de 6 enseignants chercheurs (2 professeurs et 4 maîtres de conférences) et de 3 chercheurs CNRS (1 directeur de recherche et 2 chargés de recherche). Elle focalise ses activités autour du domaine de la vision par ordinateur et de la photogramétrie dans le cadre d'applications de localisation et de reconstruction tridimensionnelle, de calibration de capteurs de vision, de reconnaissance et de suivi d'objets. Ces travaux ont donné lieu au dépôt de brevets et de codes APP, à plusieurs actions de transfert de technologie, et à la création de deux sociétés : Poseidon et DxO. L'animation de l'équipe s'organise autour de trois thèmes principaux :

- ▷ reconstruction tridimensionnelle de scènes rigides et métrologie par vision,
- ▷ identification visuelle et suivi,
- ▷ vision en milieu déformable.

1.3 Encadrements

Cette section reprends la liste des encadrements d'étudiants auxquels j'ai participé sur la période 2000-2008

1.3.1 Doctorants

[1] Joel Falcou. *Reconstruction stéréo temps réel. Une approche adaptative et parallèle*. Thèse (encadrement 30%), durée : 2003-2006, Université Blaise-Pascal, Clermont-Ferrand, Décembre 2006.

[2] Eric Royer. *Localisation d'un Robot Mobile par Vision artificielle*. Thèse (encadrement 30%), durée : 2003-2006, Université Blaise-Pascal, Clermont-Ferrand, Décembre 2006.

[3] Yann Goyat. *Systèmes vidéos pour l'analyse de trajectoires*. Thèse (encadrement 30%), durée : 2005-2008, Université Blaise-Pascal, Clermont-Ferrand, Décembre 2008.

[4] Francois Bardet. *Suivi temps réel d'un nombre variable d'objets par vision*. Thèse (encadrement 50%), durée : 2003-2009, Université Blaise-Pascal, Clermont-Ferrand, Octobre 2009.

[5] Samuel Gidel. *détection et suivi de piétons par télémètrie laser*. Thèse (encadrement 10%), durée : 2006-2010, Université Blaise-Pascal, Clermont-Ferrand, soutenance prévue en 2010.

[6] Laetitia Gond. *Estimation de posture par vision*. Thèse (encadrement 70%), durée : 2005-2009, Université Blaise-Pascal, Clermont-Ferrand, Mai 2009.

[7] Laetitia Leyrit. *Reconnaissance d'objets temps réel*. Thèse (encadrement 50%), durée : 2006-2010, Université Blaise-Pascal, Clermont-Ferrand, soutenance prévue en 2010.

[8] Bertrand Luvison. *Détection d'évènements dans les videos de foules*. Thèse (encadrement 50%), durée : 2008-2011, Université Blaise-Pascal, Clermont-Ferrand, soutenance prévue en 2011.

1.3.2 Postdoctorants et stagiaires

[1] Romain Desgeorges. *Suivi et reconstruction 3D de points clefs sur une personne*. stage de deuxième année ingénieur informatique ISIMA (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Septembre 2002.

[2] Olivier Duval. *Analyse de mouvements humains pour la formation de stagiaires chefs de manoeuvre sur ponts roulants*. stage de deuxième année ingénieur informatique ISIMA (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Septembre 2002.

[3] Lamri Nehaoua. *Suivi de motifs basé sur l'utilisation des ondelettes de haar : application au suivi de véhicules.* stage du master recherche Composants et Systèmes pour le Traitement de l'Information option VIsion pour la RObotique de l'Université Blaise Pascal (encadrement 100%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2002.

[4] Benjamin Ninassi. *Animation 3D temps réel d'un avatar*. stage de deuxième année ingénieur informatique ISIMA (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Septembre 2002.

[5] Omar Ait-Aider. *Localisation d'un robot mobile par vision basée sur la mémoire visuelle*. 9 mois, encadrement (50%), Stage de PostDoctorat au Lasmea, UMR6602, CNRS Université Blaise Pascal, 2003.

[6] Agnes Caron. *Reconnaissance des gestes d'un chef de manoeuvre*. stage de fin d'étude ingénieur informatique INPG Ensimag (encadrement 50%), durée : 5 mois, Ecole Nationale Polytechnique de Grenoble, Clermont-Ferrand, Septembre 2003.

[7] Alfredo Gardel. *Realtime pattern matching applied to vehicle tracking*. 6 mois, encadrement (50%), Stage de PostDoctorat au Lasmea, UMR6602, CNRS Université Blaise Pascal, 2004.

[8] Nadir Karam. *Suivi d'objet par Support Vector Machine. Application au suivi de véhicule*. stage de master recherche Composants et Systèmes pour le Traitement de l'Information option VIsion pour la RObotique de l'Université Blaise Pascal (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2004.

[9] Antoine Vacavant. *Suivi de gestes temps réel par traitement d'images couleur*. Master 1 informatique (encadrement 100%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, septembre 2004.

[10] Vincent Gay-Bellile. *Détection de piétons dans des images*. stage de master recherche Composants et systèmes pour le Traitement de l'Information option VIsion pour la RObotique de l'Université Blaise Pascal (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2005.

[11] Hala Najmeddine. *Détection de défauts par traitement d'image thermique*. stage de 2eme année ingénieur du CUST, université Blaise Pascal (encadrement 100%), durée : 2 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2005.

[12] Laurence Ngako-Pangop. *Reconnaissance de pièces manufacturées par vision artificielle*. stage de master recherche Composants et Systèmes pour le Traitement de l'Information option VIsion pour la RObotique de l'Université Blaise Pascal (encadrement 100%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2005.

[13] Guillaume Vignal. *Suivi Visuel de posture 3D par filtrage particulaire*. Master Informatique et stage de fin d'étude ingénieur ISIMA (encadrement 100%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, septembre 2005.

[14] Nathalie Butot. *Reconnaissance et suivi d'objets temps réel*. Master Informatique et stage de fin d'étude ingénieur ISIMA (encadrement 100%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, septembre 2006.

[15] Hala Najmeddine. *Détection et suivi de véhicules dans les séquences vidéo : Développement d'algorithmes de suivi multi-pistes*. stage du master recherche Composants et Systèmes pour le Traitement de l'Information option VIsion pour la RObotique de l'Université Blaise Pascal (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2006.

[16] Yves Varesco. *Etude et mise en oeuvre d'une caméra pan tilt zoom virtuelle par liaison sans fil*. stage de fin d'étude ingénieur du CNAM (encadrement 100%), durée : 9 mois, CNAM, Conservatoire National des Arts et Métiers, Clermont-Ferrand, Juin 2006.

[17] Pierre Lebraly. *Estimation de trajectoires de véhicules par vision*. stage de 2eme année de l'ENSEA, Ecole Nationale Supérieure de l'Electronique et de ses Applications (encadrement 100%), durée : 1 mois, Université Blaise-Pascal, Clermont-Ferrand, Juin 2007.

[18] Datta Ramadasan. *Développement de méthodes de suivi d'objets*. stage du Master 1 informatique (encadrement 100%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Septembre 2007.

[19] Datta Ramadasan. *Développement d'une application de suivi multi-objets*. stage du Master 2 système d'information et aide à la décision (encadrement 50%), durée : 5 mois, Université Blaise-Pascal, Clermont-Ferrand, Septembre 2008.

[20] P. Bouges. *Catégorisation de visages pour le maquillage virtuel*. stage de troisième année ingénieur informatique ISIMA (encadrement 50%), durée : 6 mois, Université Blaise-Pascal, septembre 2009.

1.4 Participation à des projets de recherche

[1] PAROTO. *Projet Anticollision Radar Optronique pour l'auTOmobile*. Projet national, PREDIT : programme de recherche et d'innovation dans les transports terrestres, 2001-2004.

[2] ROADSENSE. *Industry standard evaluation framework for new Human Vehicle Interactions strategies*. Projet européen, European 5th Framework Programme project, http ://www.cvisproject.org/en/links/roadsense.htm, visité en Octobre 2008, 2001-2004.

[3] WACIF. *Indoor Navigation of a Wheeled Mobile Robot along Visual Routes*. Projet national, RNTL : Réseau National des Technologies Logicielles, 2002-2004.

[4] BODEGA. *Navigation autonome et sûre en environnement urbain*. Projet national, ROBEA : programme national robotique et entités artificielles du CNRS, 2003-2005.

[5] MOBIVIP. *Véhicules Individuels Publics pour la Mobilité en centre ville*. Projet national, PREDIT 3 : programme de recherche et d'innovation dans les transports terrestres, 2003-2006.

[6] SARI/RADARR. *Recherche des Attributs pour le Diagnostic Avancé des Ruptures de la Route*. Projet national, action concertée du PREDIT : programme de recherche et d'innovation dans les transports terrestres, 2005-2008.

[7] LOVE. *Logiciels d'Observation des Vulnérables*. Projet national, PREDIT : programme de recherche et d'innovation dans les transports terrestres, 2006-2009.

[8] DIVAS. *Dialogue Infrastructure - Véhicules pour Améliorer la Sécurité routière*. Projet national, Projet ANR-PREDIT 3 : programme de recherche et d'innovation dans les transports terrestres, 2007-2010.

[9] CITYVIP. *Déplacement sûr de véhicules individuels adaptés à l'environnement urbain*. Projet national, Projet ANR-PREDIT : programme de recherche et d'innovation dans les transports terrestres, 2008-2010.

[10] BIORAFALE. *Identification des interdits de stades*. Projet national, Projet OSEO, aide à l'innovation, 2009-2012.

1.5 Publications

Articles dans des revues internationales à comité de lecture

[1] T. Chateau, F. Collange, L. Trassoudaine, C. Debain, and J. Alizon. Automatic Guidance of Agricultural Vehicles Using a Laser Sensor. *Computers and Electronics in Agriculture, Elsevier Sciences*, 28 :243–257, 2000.

[2] C. Debain, T. Chateau, M. Berducat, P. Bonton, and P. Martinet. A help guidance system for agricultural vehicles. *Elsevier, Computers and Electronics in Agriculture, Special issue Navigating Agricultural Field Machinery*, 25(1) :29–51, Janvier 2000.

[3] J. Falcou, J. Sérot, T. Chateau, and J. T. Lapresté. Quaff : Efficient C++ Design for Parallel Skeletons, volume = 32, year = 2006. *Parallel Computing*, (7-8) :604–615.

[4] T. Chateau and J. T. Lapresté. Realtime kernel based tracking. *Electronic Letters on Computer Vision and Image Analysis*, 8(1) :27–43, 2009.

[5] S. Gidel, P. Checchin, C. Blanc, T. Chateau, and L. Trassoudaine. Pedestrian detection and tracking in urban environment using a multilayer laserscanner. *IEEE Transactions on Intelligent Transportation Systems*, page to appear, 2010.

[6] Y. Goyat, T. Chateau, L. Trassoudaine, and L. Malaterre. Trajectory measurement of vehicles : a new observation. *Advances in Transportation Studies*, 8(1) :5–17, July 2009.

[7] Y. Goyat, T. Chateau, and L. Trassoudaine. Tracking of vehicle trajectory by combining a camera and a laser rangefinder. *Springer MVA : Machine Vision and Application*, online, March 2009.

[8] L. Leyrit, T. Chateau, and J.T. Lapresté. *Machine Learning*, chapter Classifiers Association for High Dimensional Problem. IN-TECH, To appear.

Articles dans des revues nationales à comité de lecture

[9] T. Chateau, L. Trassoudaine, F. Collange, P. Bonton, and C. Debain. Fusion d'attributs incertains : application à l'aide au guidage d'engins agricoles. *Traitement du signal, numéro spécial Perception pour la localisation de véhicules intelligents*, 17(3) :249–261, 2000.

[10] T. Chateau and A. Vacavant. Suivi de gestes temps réel par traitement d'images couleur. *Traitement du signal*, 21(1), January 2005.

[11] Y. Goyat, T. Chateau, L. Malaterre, L. Trassoudaine, and F. Menant. Un observatoire de trajectoires en virages fondé sur la vision artificielle. *RTS : Recherche, Transport et Sécurité*, 98 :73–88, Mars 2008.

Articles dans des conférences internationales avec actes et comité de lecture

[12] R. Chapuis, J. Laneurit, R. Aufrere, F. Chausse, and T. Chateau. Accurate vision based road tracker. In *IV IEEE International Conference on Intelligent Vehicles*, June 2002.

[13] T. Chateau, F. Jurie, M. Dhome, and X. Clady. Real-time tracking using wavelets representation. In *Symposium for Pattern Recognition, DAGM*, pages 523–530, Zurich, September 2002. Springer.

[14] O.A. Aider, T. Chateau, and J.T. Lapresté. Indoor autonomous navigation using visual memory and pattern tracking. In S. Barman and T. Ellis, editors, *BMVC2004, British Machine Vision Conference*, volume 1, pages 657–666, Kingston, England, September 2004.

[15] O. Ait-Aider, G. Blanc, Y. Mezouar, T. Chateau, and P. Martinet. Indoor navigation of mobile robot : An image based approach. In *ISR, International Symposium on Robotics*, Paris, Mars 2004.

[16] G. Blanc, O. Ait-Ader, Y. Mezouar, and T. Chateau. Autonomous image based navigation in indoor enveronment. In *IAV'2004, IFAC Symposium On intelligent Autonomous Vehicles*, Lisbone, Portugal, July 2004.

[17] T. Chateau and J. T. Lapresté. Real time tracking with occlusion and illumination variations. In *ICPR, IAPR International Conference on Pattern Recognition*, volume 4, pages 763–767, Cambridge, England, August 2004.

[18] T. Chateau and J.T. Lapresté. Robust real time tracking of a vehicle by image processing. In *IEEE Intelligent Vehicles Symposium,*, Parma, Italy, June 2004.

[19] E. Royer, M. Lhuillier, T. Chateau, and M. Dhome. Towards an alternative gps sensor in dense urban environment from visual memory. In S. Barman and T. Ellis, editors, *BMVC, British Machine Vision Conference,*, volume 1, pages 197–206, September 2004.

[20] T. Chateau, A. Vacavant, and J.M. Lavest. Skin detection and tracking by monocular vision. In *ISSCS'05 IEEE international Symposium on Signal Circuits and Systems*, Iasi, Roumania, 2005.

[21] J. Falcou, J. Sérot, T. Chateau, and F. Jurie. A parallel implementation of a 3d reconstruction algorithm for real-time vision. In *PARCO, Parallel Computing*, Malaga, Spain, 2005.

[22] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau. Localization in urban environments : monocular vision compared to a differential gps sensor. In *IEEE CVPR, Computer Vision and Pattern Recognition*, San Diego, USA, June 2005.

[23] A. Vacavant and T. Chateau. Real time head anb hand tracking by monocular vision. In *ICIP - International Conference on Image Processing*, Genova, Italy, September 2005.

[24] T. Chateau, V. Gay-Belille, F. Chausse, and J. T. Lapresté. Real-time tracking with classifiers. In *WDV - WDV Workshop on Dynamical Vision at ECCV2006*, Grazz, Austria, May 2006.

[25] J. Falcou, T. Chateau, J. Sérot, and J.T. Lapresté. Real time parallel implementation of a particle filter based visual tracking. In *CIMCV 2006 - Workshop on Computation Intensive Methods for Computer Vision at ECCV*, Grazz, Austria, May 2006.

[26] Y. Goyat, T. Chateau, L. Malaterre, and L. Trassoudaine. Vehicle trajectories evaluation by static video sensors. In *ITSC06 2006 - 9th International IEEE Conference on Intelligent Transportation Systems*, Toronto, Canada, September 2006.

[27] A. Gardel, J. L. Lazaro, I. Bravo, J.-P. Derutin, and T. Chateau. Parallel implementation of modified 2d pattern matching. In *ISIE IEEE International Symposium on Industrial Electronics*, Vigo, Spain, June 2007.

[28] S. Treuillet, E. Royer, T. Chateau, M. Dhome, and J-M. Lavest. Body mounted vision system for visually impaired outdoor and indoor wayfindind assistance. In *CVHI 2007 - Conference and Workshop on Assistive Technologies for People with Vision and Hearing Impairments*, Granada, Espagne, August 2007.

[29] F. Bardet and T. Chateau. Mcmc particle filter for real-time visual tracking of vehicles. In *11th International IEEE Conference on Intelligent Transportation Systems*, Beijing, China, october 2008.

[30] S. Gidel, P. Checchin, C Blanc, T. Chateau, and L. Trassoudaine. Decentralized fusion of a 4-layer sensor based on parzen method : Application to pedestrian detection. In *ICRA 2008 Workshop : Human Detection from Mobile Robot Platforms : Different Perspetive, Different Modalities*, Pasadena, USA, May 2008.

[31] S. Gidel, P. Checchin, C. Blanc, T. Chateau, and L. Trassoudaine. Pedestrian detection method using a multilayer laserscanner : Application in urban environment. In *IROS IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, September 2008.

[32] S. Gidel, C. Blanc, T. Chateau, P. Checchin, and L. Trassoudaine. Non parametric data association for particle filter based multi-object tracking : Application to multi-pedestrian tracking. In *IEEE Intelligent Vehicles Symposium*, Eindhoven, The Netherlands, June 2008.

[33] S. Gidel, C. Blanc, T. Chateau, P. Checchin, and L. Trassoudaine. Parzen method for fusion of laserscanner data : Application to pedestrian detection. In *IEEE Intelligent Vehicles Symposium*, Eindhoven, The Netherlands, June 2008.

[34] L. Gond, P. Sayd, T. Chateau, and M. Dhome. A 3d shape descriptor for human pose recovery. In *ADMO, V Conference on Articulated Motion and Deformable Objects*, Andratx, Spain, July 2008.

[35] L. Leyrit, T. Chateau, and J.T. Lapresté. Visual pedestrian recognition in weak classifier space using nonlinear parametric models. In *ICIP IEEE International Conference on Image Processing*, San Diego, USA, october 2008.

[36] L. Leyrit, T. Chateau, C. Tournayre, and J.T. Lapresté. Association of adaboost and kernel based machine learning methods for visual pedestrian recognition. In *IEEE Intelligent Vehicles Symposium*, Eindhoven, The Netherlands, June 2008.

[37] F. Bardet, T. Chateau, and J.T. Lapresté. Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous tracking and classification. In *ICCV 2009, International Conference on Computer Vision*, Tokyo, Japan, 09 2009.

[38] François Bardet, Thierry Chateau, and Datta Ramadasan. Unifying real-time multi-vehicle tracking and categorization. In *Intelligent Vehicle Symposium*, volume 1, 2009.

[39] F. Bardet and T. Chateau. Real time multi-object tracking with few particles. In *Visapp, International Conference on Vision Theory and Applications*, Lisboa, Portugal, Fevrier 2009.

[40] T. Chateau, Y. Goyat, and L. Trassoudaine. M2sir, a multi modal sequential importance resampling algorithm for particle filters. In *ICIP IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[41] S. Gidel, C. Blanc, P. Checchin, T. Chateau, and L. Trassoudaine. Non-parametric laser and video data fusion : Application to pedestrian detection in urban. In *in 12th IEEE International Conference on Information Fusion*, Seattle, USA, July 2009.

[42] S. Gidel, C. Blanc, T. Chateau, P. Checchin, and L. Trassoudaine. A method based on multilayer laserscanner to detect and track people in urban environment. In *in IEEE Intelligent Vehicle Conference*, Xi'an, China, June 2009.

[43] L. Gond, P. Sayd, T. Chateau, and M. Dhome. A regression-based approach to recover human pose from voxel data. In *Second IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS2009) at ICCV 2009,*, Kyoto Japon, Septembre 2009.

[44] B. Luvison, T. Chateau, P. Sayd, Q.C. Pham, and J.T. Lapresté. An unsupervised learning based aprroach for unexpected event detection. In *Visapp, International Conference on Vision Theory and Applications*, Lisboa, Portugal, Fevrier 2009.

Articles dans des conférences nationales avec actes et comité de lecture

[45] B. Luvison, T. Chateau, P. Sayd, Q.C. Pham, and J.T. Lapresté. Estimation parcimonieuse de densité par des fonctions noyaux : application à la détection temps réel d'événements rares. In *RFIA : 17e congrés francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Caen, France, Janvier 2010.

[46] T. Chateau, .F Collange, L. Trassoudaine, C. Debain, and J. Alizon. Fusion d'attributs : application au guidage d'engins agricoles. In *LFA 99, Rencontres francophones sur la logique floue et ses applications*, page 8pp, Valenciennes, October 1999.

[47] T. Chateau, F. Jurie, N. Allezard, and R. Marc. Suivi tridimensionnel et reconnaissance de gestes temps réel par vision monoculaire. In *Orasis 2003*, Gerardmer, May 2003.

[48] F. Bardet, T. Chateau, F. Jurie, and M. Naranjo. Interactions geste-musique par vision artificielle. In *Workshop Acquisition du geste humain par vision artificielle, dans RFIA04, Congrès sur la Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, January 2004. Actes sur CDROM.

[49] T. Chateau, F. Jurie, R. Marc, and M. Dhome. Suivi et reconnaissance de gestes par vision monoculaire en temps reél : application la formation des chargés de manoeuvres pour la conduite des ponts polaires. In *RFIA 04, Congrès sur la Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, January 2004. Actes sur CDROM.

[50] T. Chateau, F. Jurie, and R. Marc. Reconnaissance de gestes par vision monoculaire en temps reél : application la formation des chargés de manoeuvres pour la conduite des ponts polaires. In *Workshop Acquisition du geste humain par vision artificielle, dans RFIA 04, Congrès sur la Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, January 2004. Actes sur CDROM.

[51] J. Falcou, J. Sérot, T. Chateau, and F. Jurie. Un cluster de calcul hybride pour les applications de vision temps réel. In *GRETSI 2005 - 20e colloque GRETSI sur le traitement du signal et des images*, Louvain, Belgium, September 2005.

[52] A. Vacavant and T. Chateau. Suivi de la tête et des mains en vision monoculaire temps réel. In *ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur*, Fournol, France, May 2005.

[53] J. Falcou, J. Sérot, T. Chateau, and J. T. Lapresté. Nt2 : Une bibliothèque haute-performance pour la vision artificielle. In *Orasis 2007, Congrès Jeunes Chercheurs en Vision par Ordinateur*, volume 32, pages 604–615, Obernai, Mai 2007.

[54] Y. Goyat, T. Chateau, L. Malaterre, and L Trassoudaine. Trajectographie des véhicules en vision monoculaire. In *GRETSI - 11eme Colloque de traitement du signal et des images*, Troyes, Septembre 2007.

[55] Y. Goyat, T. Chateau, L. Malaterre, and L. Trassoudaine. Estimation précise de la trajectoire d'un véhicule par vision monoculaire. In *Congrès Jeunes Chercheurs en Vision par Ordinateur*, Obernai, Juin 2007.

[56] F. Bardet and T. Chateau. Performances comparées de rééchantillonnage pour filtres de monte-carlo. In *RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Amiens, France, January 2008.

[57] T. Chateau, J. T. Lapresté, D. Ramadasan, and S. Treuillet. Suivi de motifs planaires temps réel par combinaison de traqueurs. In *RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Amiens, France, January 2008.

[58] S. Gidel, P. Checchin, C. Blanc, L. Trassoudaine, and T. Chateau. Détection de piétons à l'aide d'un capteur laser quatre nappes embarqué. In *RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Amiens, France, January 2008.

[59] Y. Goyat, T. Chateau, and L. Trassoudaine. Métrologie des trajectoires de véhicules. In *Conférence Internationale Francophone d'Automatique*, Bucarest, Roumania, September 2008.

[60] Y. Goyat, T. Chateau, L. Malaterre, and L Trassoudaine. Estimation précise de la trajectoire d'un véhicule par fusion vision télémètre laser. In *RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Amiens, France, January 2008.

[61] F. Bardet, D. Ramadasan, and T. Chateau. Suivi et classification visuels temps réel d'un nombre variable d'objets : application au suivi de véhicules. In *ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur*, Tregastel, June 2009.

[62] L. Leyrit, T. Chateau, and J.T. Lapresté. Descripteurs pour la reconnaissance de piétons. In *ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur*, Tregastel, June 2009.

[63] L. Leyrit, C. Tournayre, and T. Chateau. Association de classifiers pour la reconnaissance de piétons dans les images. In *13ème colloque national Compression et Representation des SIgnaux Audiovisuels (CORESA'2009)*, Toulouse, mars 2009.

[64] B. Luvison, T. Chateau, P. Sayd, and Q.C. Pham. Méthode d'apprentissage non supervisée pour la détection d'évènements inattendus. In *CORESA (COmpression et REprésentation des Signaux Audiovisuels)*, Toulouse, Mars 2009.

[65] B. Luvison, T. Chateau, Q.C. Pham, P. Sayd, and J.T. Lapresté. A single-class learning method for classification. In *ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur*, Tregastel, June 2009.

[66] T. Penne, V. Barra, C. Tilmant, and T. Chateau. Une version modifiée de l'ensemble tracking. In *ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur*, Tregastel, June 2009.

[67] D. Ramadasan, F. Bardet, and T. Chateau. Suivi visuel temps réel d'un nombre variable d'objets avec peu de particules. In *CORESA (COmpression et REprésentation des Signaux Audiovisuels)*, Toulouse, Mars 2009.

[68] Y. Goyat, T. Chateau, and F. Bardet. Méthode spatio-temporelle mcmc pour l'estimation des trajectoires de véhicule. In *RFIA : 17e congrés francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Caen, France, Janvier 2010.

[69] T. Penne, V. Barra, T. Chateau, and C. Tilmant. Ensemble tracking modulaire. In *RFIA : 17e congrés francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Caen, France, Janvier 2010.

[70] F. Bardet, T. Chateau, and D. Ramadasan. Suivi et classification conjoints de multiples objets et de la source lumineuse par filtre particulaire mcmc. In *RFIA : 17e congrés francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Caen, France, Janvier 2010.

Actes de Conférences

[71] T. Chateau and A. Bartoli, editors. *Congrès des jeunes chercheurs en vision par ordinateur*, volume 1, Fournol, France, May 2005.

[72] T. Chateau, A. Vacavant, and P. Sayd, editors. *Atelier VISAGE : Vidéo-surveillance Intelligente : Systèmes et AlGorithmES*, volume 1, Caen, France, January 2010.

Codes APP et Brevets

[73] F. Bardet, T. Chateau, and D. Ramadasan. suivi multi-objet en temps réel par filtre particulaire mcmc. Code APP : IDDN.FR (en cours), Octobre 2009.

[74] T. Chateau, Y. Goyat, L. Trassoudaine, and L. Malaterre. Logiciel de mesure de la trajectoire d'objets mobiles passifs. Code APP : IDDN.FR (En cours), Octobre 2009.

[75] T. Chateau, Y. Goyat, L. Trassoudaine, and L. Malaterre. Procédé et dispositif pour la mesure de la trajectoire d'objets mobiles passifs. Demande de brevet francais 09 05 153, Octobre 2009.

[76] E. Royer, M. Lhuillier, M. Dhome, J.M. Lavest, and T. Chateau. Algorithme de création d'une mémoire visuelle tridimentionnelle d'images visuelles à partir d'un flot vidéo. Code APP : IDDN.FR.001.120007.000.S.P.2009.21000, Mars 2009.

1.6 Enseignements

Je suis rattaché aux départements Génie Electrique et Génie Physique de Polytech Clermont-Ferrand, où j'effectue la majorité de mon enseignement. J'interviens également, à l'ISIMA (Ecole d'ingénieur en Informatique) et au Master M2 Recherche dans l'option IV (imagerie-vision). Le tableau 1.1 récapitule les volumes horaires dispensés dans les principales disciplines dans lesquelles j'interviens. La rubrique divers est composée d'heures d'encadrement de projet et de stages. La figure 1.1 montre la répartition globale des disciplines enseignées.
La figure 1.2 montre la répartition globale de mes enseignements en fonction de leur nature (cours, td, tp).

	2001	2002	2003	2004	2005	2006	2007	2008
Reconnaissance des formes	/	/	/	24	24	24	12	9
Imagerie	/	/	21	38	36	51	47	43
Logique	20	16	/	/	/	/	/	/
Automatique	139	132	147	161	174	114	178	158
Robotique	36	68	63	62	49	/	/	/
Divers	39	63	92	94	88	72	54	69

TABLE 1.1 – Tableau récapitulatif des heures (équivalent TD) effectuées dans les principales disciplines d'enseignement

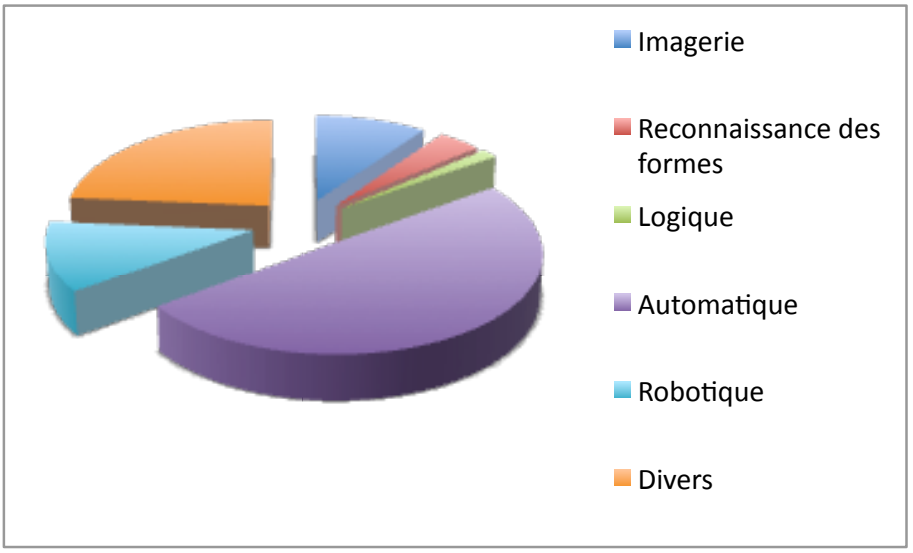


FIGURE 1.1 – Synoptique illustrant le répartition globale des disciplines enseignées en volume horaire, illustration en couleur.

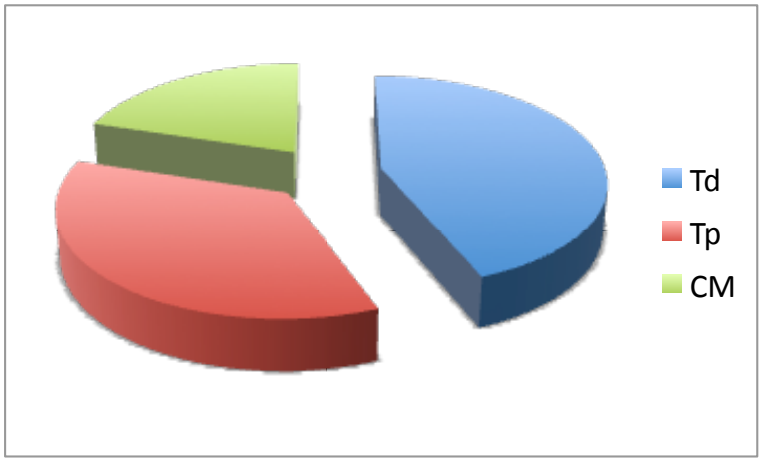


FIGURE 1.2 – Synoptique illustrant le répartition globale des enseignements en fonction de leur nature (cours, td, tp), illustration en couleur.

1.7 Thématiques scientifiques abordées

Ce chapitre synthétise les activités de recherche que j'ai menées au LASMEA, au sein de l'équipe ComSee du groupe Gravir. Elles s'inscrivent dans le domaine de la vision par ordinateur et concernent plus précisément

l'estimation d'état dans les images. L'état est vu comme une variable cachée définie par «ce que l'on cherche» et qui contribue à générer une ou plusieurs images constituant les observations d'entrée du système. La figure 1.3 illustre cette modélisation.

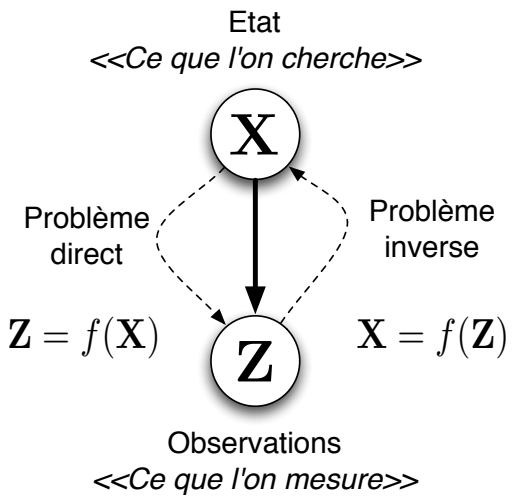


FIGURE 1.3 – Synoptique illustrant le principe de l'estimation d'état par vision. L'état $\mathbf{X}$ est une variable cachée qui génère une observation $\mathbf{Z}$. Le processus consistant à construire une observation à partir de l'état est un processus direct qui peut s'apparenter à de la synthèse. Par contre, l'estimation de l'état à partir d'une observation est un problème inverse, souvent mal posé, et plus complexe à résoudre.

Soit $\mathbf{X}$ un vecteur composé de variables définissant l'état de «ce que l'on recherche». Dans le cas d'un problème de détection d'objet, $\mathbf{X}$ est une variable discrète de dimension un pouvant prendre deux valeurs, définissant la présence ou l'absence de l'objet recherché. Dans le cas de l'estimation de la pose d'un humain, l'état est un vecteur de variables continues (paramètres d'un modèle de corps humain) codant la configuration articulaire de la pose. Dans le cas du suivi d'un objet, l'état $\mathbf{X}$ devient une séquence temporelle d'états indexés sur le temps. Dans le cas général, l'état n'est pas forcément de taille connue *a priori* et il peut être composé d'une partie discrète et d'une partie continue.
Soit $\mathbf{Z}$ le vecteur constitué des observations disponibles, généré par l'état. Le processus consistant à construire une observation à partir de l'état est un processus direct qui peut s'apparenter à de la synthèse. Par contre, l'estimation de l'état à partir d'une observation est un problème inverse, souvent mal posé, et plus complexe à résoudre. L'estimation visuelle peut être vue comme l'ensemble des techniques consistant à modéliser la relation entre les observations et l'état. Dans le cas d'applications de suivi visuel, l'état est indexé par le temps et la notion de séquence temporelle d'état et d'observation est introduite.

1.7.1 Méthodes d'apprentissage pour l'estimation d'état

Il existe principalement deux grandes familles d'approches pour estimer un état à partir d'images : les méthodes basées sur un modèle et les méthodes basées sur des exemples (basées apprentissage) .

- ▷ Les méthodes basées modèle reposent sur l'utilisation d'un modèle explicite de ce l'on cherche. Dans le cas d'estimation de pose par exemple, un modèle du corps humain est défini a priori pour représenter la personne observée dans les images. La pose du corps est alors estimée par une approche de type «analyse-synthèse» : des prédictions sont effectuées sur la configuration du modèle, et sont ensuite mises à jour grâce aux informations contenues dans l'image. Plusieurs étapes de modélisation sont nécessaires dans la mise en œuvre de ces techniques : il faut d'abord définir le modèle du corps humain et la façon dont il se projette dans l'image pour prédire une apparence. Une fonction de vraisemblance, ($\mathbf{Z} = f(\mathbf{X})$) s'appuyant sur différentes primitives extraites de l'image doit également être construite pour mesurer la concordance entre les données visuelles et l'apparence générée du modèle dans l'image.

La configuration optimale du modèle, c'est-à-dire qui maximise cette fonction de vraisemblance est ensuite estimée. Les méthodes d'estimation basées sur un modèle sont souvent assez précises mais aussi coûteuses en temps de calcul : elles requièrent l'optimisation ou l'exploration d'une fonction de coût très complexe (le modèle doit être rendu dans les images et la fonction de coût évaluée pour chaque hypothèse sur l'état).

▷ Contrairement aux approches basées sur des modèles, les méthodes basées apprentissage fournissent directement une estimation de l'état à partir des observations, sans passer par des prédictions sur un modèle. Elles reposent sur l'utilisation d'une base d'exemples créée à l'avance et qui contient un ensemble de paires observation-état. Les données d'entraînement sont généralement obtenues à partir de bases de données ou de logiciels de synthèse d'images. Les principaux avantages de ces techniques sont d'une part qu'elles évitent lors de l'estimation de passer par l'optimisation d'une fonction de vraisemblance complexe et d'effectuer les rendus d'un modèle. L'inconvénient est que la base d'apprentissage doit être capable de traduire un sous-ensemble représentatif des états à estimer ; ce qui peut conduire à des bases de très grande taille dans le cas d'états de dimensions importantes. D'autre part, ces méthodes doivent posséder de bonnes propriétés de généralisation. Au sein des techniques utilisant un ensemble d'exemples dans l'apprentissage, on distingue deux grandes classes de méthodes : les méthodes non paramétriques basées sur une comparaison aux exemples de la base, dans lesquelles la base d'exemples est explicitement stockée en mémoire et sert ensuite de référence pour comparer les nouveaux exemples ; et les méthodes basées sur un apprentissage, dans lesquelles un entraînement, effectué hors ligne, produit un modèle paramétrique qui généralise les propriétés de la base d'exemples.

Dans le cadre de mes travaux de recherche je me suis particulièrement intéressé à l'étude de ce dernier type de modèle pour la mise en relation entre l'observation et l'état. Je présente, dans le chapitre 2, une synthèse des principales recherches que j'ai menées autour de cette problématique. Ces dernières ont en commun l'utilisation d'une machine de régression qui repose sur un modèle paramétrique décrit par une combinaison linéaire d'un ensemble de fonctions de base prédéfinies :

$$\mathbf{X} = \sum_{k=1}^{K} w_k \phi_k(\mathbf{Z}) + \boldsymbol{\epsilon} \tag{1.1}$$

Ici, $\{\phi_k(\mathbf{Z}) | k = 1, .., K\}$ sont les fonctions de base, $\mathbf{w}_k \in I\!R^m$ sont les vecteurs de poids et $\boldsymbol{\epsilon}_k$ est un terme contenant les erreurs résiduelles additives.
Ce modèle peut être utilisé pour apprendre le lien entre l'état et l'observation, aussi bien dans un contexte de régression (la sortie est un réel ou un vecteur de réels), que dans un contexte de classification (la sortie est une variable binaire). La principale problématique associée à ce modèle concerne la stratégie mise en place pour l'estimation des paramètres w_k, et plus particulièrement, la définition du critère associé. Ainsi, dans les méthodes de type *Support Vector Machine* (SVM), il s'agit de maximiser une marge (critère géométrique définit dans l'espace des paramètres). Dans Les méthodes de type *Relevant Vector Machine* (RVM), les paramètres sont définis dans un cadre probabiliste et l'estimation s'effectue par une méthode de maximum *a posteriori*. Il est aussi possible d'utiliser un critère de type moindre carré pénalisé pour estimer le jeu de paramètres. Dans le chapitre 2, je reprends les différentes contributions que nous avons réalisées autour de l'utilisation de ces modèles.
Dans un premier temps, nous avons abordé le problème de la détection d'une catégorie d'objet dans une image. Nous avons proposé, dans le cadre d'une application de détection de piétons à partir d'une caméra mobile, de combiner deux types de classifieurs populaires : l'algorithme Adaboost et les méthodes à noyau. L'originalité porte sur l'utilisation de la réponse des classifieurs faibles obtenus par Adaboost comme vecteur de primitives à l'entrée d'une machine à noyau. Nous avons montré que cette stratégie est généralement plus performante qu'un Adaboost seul.
Nous nous sommes ensuite intéressés à l'utilisation de méthodes à noyaux dans un problème de régression multi-variables. Le contexte applicatif associé est l'estimation de la pose d'une personne à partir d'une ensemble d'images provenant d'un réseau de caméras calibrées. Les machines à noyaux ont été apprises à

l'aide d'un ensemble d'images générées de manière synthétique. Nous avons montré que les performances des machines ainsi apprises sont au moins comparables à l'état de l'art. De plus, nous avons proposé un descripteur permettant un codage original de l'enveloppe 3D de la personne, utilisé ensuite comme vecteur de paramètres à l'entrée de la machine d'apprentissage.
Le Lasmea travaillant depuis plus de dix ans sur des problématiques de suivi d'objets dans des images, nous avons proposé d'utiliser des modèles à noyaux pour apprendre le lien entre la variation de primitives images et la variation du mouvement d'un objet planaire texturé associé. Nous avons montré que l'algorithme de suivi d'objet obtenu est plus performant en terme de convergence et de robustesse au bruit que les modèles classiques d'ordre un.

1.7.2 Méthodes de Monte-Carlo pour l'estimation d'état

La formalisation du problème de l'estimation d'état par vision se divise en deux grandes catégories, en fonction de la nature du vecteur d'état. Lorsque celui-ci est composé de variables déterministes, on cherche à estimer la valeur du vecteur d'état qui explique au mieux les observations (minimisation d'un critère). Lorsque le vecteur d'état est composé de variables aléatoires, il s'agit d'estimer la densité de probabilité de l'état, étant donné l'observation $p(\mathbf{X}|\mathbf{Z})$:

- ▷ Les méthodes déterministes considèrent que le vecteur d'état est de nature déterministe est qu'il faut estimer sa valeur en fonction des observations disponibles. Dans le cas où l'on a une fonction analytique décrivant le problème inverse ($\mathbf{X} = f(\mathbf{Z})$), l'estimation de l'état utilise directement cette fonction. Dans le cas où l'on dispose uniquement d'une fonction décrivant le problème direct ($\mathbf{Z} = f(\mathbf{X})$), l'estimation de l'état peut être vu comme un problème d'optimisation où l'on recherche l'état qui minimise une distance entre les observations et la fonction de vraisemblance : $\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} d(\mathbf{Z} - f(\mathbf{X}))$. L'état estimé est celui qui explique au mieux les mesures disponibles. Le principal inconvénient des méthodes déterministes est lié au fait qu'elles ne sont pas capables de gérer les problèmes de multi-modalité dus à la nature mal posée du problème inverse : une observation peut souvent expliquer plusieurs états. Dans le cas des méthodes basées sur des techniques d'optimisation, cela se traduit par la présence de minima locaux dans la fonction de vraisemblance.

- ▷ Contrairement aux méthodes déterministes, les méthodes probabilistes considèrent l'état comme un vecteur aléatoire, dont il faut estimer la densité de probabilité, connaissant les mesures. La vraisemblance des mesures par rapport à l'état étant connue ($p(\mathbf{Z}|\mathbf{X})$), la résolution du problème est obtenue en utilisant la règle de Bayes :

$$p(\mathbf{X}|\mathbf{Z}) = \frac{p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z})} \tag{1.2}$$

Lorsque les états contiennent des variables aléatoires continues, une des difficultés consiste à représenter leur densité de probabilité. Il existe alors deux principaux modèles de représentation. Le premier propose de définir les densités de probabilité par une fonction paramétrique, le but étant d'obtenir une forme analytique de la formule de Bayes. Les modèles Gaussiens sont les plus utilisés lorsque l'on applique la règle de Bayes à de tels modèles, la densité obtenue suit également une loi Gaussienne. Le principal inconvénient de ces techniques vient du fait que l'on fait une hypothèse a priori sur la forme des densités de probabilité, hypothèse qui n'est pas toujours vraie. Le deuxième modèle de représentation consiste à approcher les densités de probabilité par une somme d'échantillons, générés selon des techniques de Monte-Carlo. Ces méthodes permettent de gérer des densités de probabilité de forme quelconque et de dimensions variables ; ce qui est essentiel dans le cas d'applications comme le suivi d'un nombre variable d'objets.
Dans le cadre de mes travaux j'ai principalement utilisé une formalisation de type probabiliste, avec une représentation des densités de probabilité sous la forme d'échantillons (méthodes de Monte-Carlo). J'expose, dans le chapitre 3 une synthèse des contributions réalisées dans ce domaine, essentiellement attachées à des applications de suivi d'un ou plusieurs objets.

Dans un premier temps , nous avons étudié les performances des modèles d'observation basés apprentissage dans des filtres à particules. Ces travaux se positionnent à l'intersection des deux problématiques abordées dans ce manuscrit, de par l'utilisation d'algorithmes de type Adaboost ou SVM pour construire la fonction de vraisemblance, de nature probabiliste. Nous avons alors proposé une méthode basée apprentissage pour construire des probabilités calibrées à partir de la sortie réelle des classifieurs (marge, score, ..). L'algorithme résultant est particulièrement performant tant dans des applications de suivi de catégories d'objets, que dans le cas du suivi d'un objet en particulier ; l'apparence de ce qui doit être suivi étant appris par une base d'images constituée d'instances positives et négatives.
Nous avons ensuite abordé la problématique du suivi d'un nombre variable d'objets par filtrage particulaire. Il s'agit d'un problème difficile où le vecteur d'état peut avoir une dimension variable, ce qui entraîne la mise en œuvre d'un formalisme capable de gérer les changements de dimension dans le processus d'exploration. Le filtre utilisé, appelé RJMCMC-PF (*Reversible Jump Markov Chain Monte Carlo Particle Filter*) s'avère très performant pour ce type d'application. Dans le cas où les objets à suivre sont de catégories différentes (taille différentes), nous avons proposé d'explorer la catégorie dans le filtre. De plus, la plupart des applications abordées étant en milieu extérieur, nous proposons d'estimer la présence et la position du soleil dans le filtre dans le but de prendre en compte ces dernières dans les fonctions de vraisemblance. Au final, le processus explore un état de dimension variable comprenant à la fois les configurations dynamiques des objets (position, vitesse, ...), leur catégorie, la présence de soleil et sa position. Le système obtenu est capable de suivre, en temps réel, un nombre important d'objets de catégories différentes dans un environnement dont les conditions d'éclairement varient.
Bien que les méthodes de Monte-Carlo connaissent une grande popularité dans le domaine du suivi d'objet, elles peuvent aussi être utilisées pour estimer des états dans un cadre différent du contexte de suivi séquentiel. Nous avons abordé la problématique de l'estimation d'une trajectoire, à partir de données multi-capteurs (laser, vidéo) dans le contexte d'applications routières. La dynamique d'un véhicule pouvant être approximée par des modèles assez simples, nous avons proposé un modèle de trajectoire paramétrique prenant en compte des *a priori* sur le comportement du conducteur, ainsi que de la géométrie de l'infrastructure observée (rayon de courbure dans le cas d'un virage par exemple). Le vecteur d'état est ainsi formé des paramètres composant le modèle paramétrique de la trajectoire, dont nous estimons la densité de probabilité par MCMC (*Markov Chain Monte Carlo*). Nous avons montré que les trajectoires estimées sont plus précises avec une exploration globale qu'avec un filtrage temporel classique.
Dans ce document, j'ai volontairement omis de développer certains travaux réalisés, soit parce qu'ils s'inséraient mal dans l'un des deux chapitres, soit parce que je considère que ma contribution à ces travaux est n'est pas suffisante pour qu'ils fassent l'objet d'une section dans ce document. Il s'agit, entre autres :

- ▷ des travaux fondateurs menés sur la localisation par mémoire visuelle dans le cadre de la thèse d'Eric Royer et de celle de Guillaume Blanc. Ces derniers ont permis de définir la notion de mémoire visuelle pour une utilisation dans le domaine de la localisation de robots mobiles,
- ▷ des travaux sur la détection et le suivi multi-piétons multi-capteurs embarqués sur un véhicule routier menés dans le cadre de la thèse à Samuel Gidel,
- ▷ des travaux sur la parallélisation des filtres à particules menés dans le cadre de la thèse de Joel Falcou et de celle de Francois Bardet,

CHAPITRE

2

MODÈLES BASÉS APPRENTISSAGE POUR L'ESTIMATION D'ÉTAT

Cette partie dresse une synthèse des contributions que j'ai réalisées autour de l'utilisation de méthodes basées apprentissage pour la mise en relation entre les observations et l'état. Les applications associées concernent essentiellement le suivi d'objets planaires, l'estimation de posture et la détection de piétons. Pour chaque application présentée, une synthèse est effectuée, reprenant son positionnement bibliographique, le principe de la méthode , les résultats, ainsi que les publications associées que le lecteur est invité à consulter pour plus de détails.

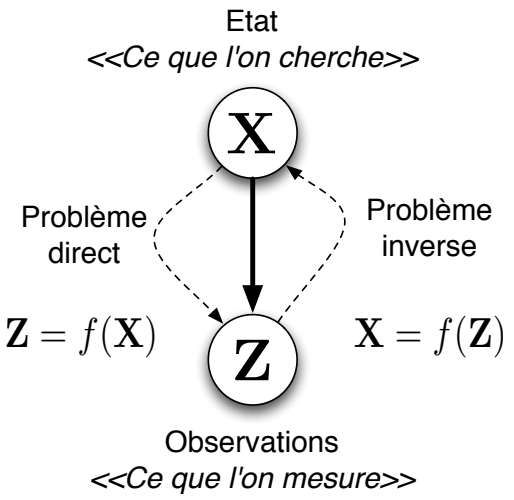


FIGURE 2.1 – Synoptique illustrant le principe de l'estimation d'état par vision. L'état $\mathbf{X}$ est une variable cachée qui génère une observation $\mathbf{Z}$. Dans ce chapitre, nous nous intéresserons plus particulièrement à l'étude du lien entre l'état et l'observation, par des méthodes basées apprentissages.

2.1 Introduction

Je présente une synthèse de mes activités de recherche autour des méthodes basées apprentissage pour l'estimation d'état dans des images. Le principe général de l'estimation d'état par vision est illustré sur la figure 2.1. Ce chapitre se focalise sur la partie modélisation du lien entre l'état et les observations, par des techniques d'apprentissage. Il s'agit, à partir d'un ensemble d'exemples de couples {observation, état }, d'apprendre un modèle permettant d'inférer un état à partir d'une nouvelle observation.

Je présente, dans une première partie un modèle paramétrique générique et certains algorithmes récents utilisés pour l'estimation des paramètres de ce modèle. Ces algorithmes sont alors comparés, en testant leurs performances dans le cas d'un exemple de classification simple. Ce modèle générique est alors utilisé dans le cadre de trois applications de vision : la détection de piétons, l'estimation de posture et le suivi d'objets planaires.

La deuxième partie est focalisée sur une application de détection de piétons. Nous étudions, d'une part, la combinaison de techniques de sélection de primitives basées sur l'Adaboost avec une machine d'apprentissage à noyau, et d'autre part, la comparaison de primitives images.

La troisième partie synthétise des travaux adressant une application d'estimation de la pose d'une personne à partir d'un ensemble d'images statiques, issues de caméras calibrées. Des machines de régression sont apprises à partir d'un ensemble d'images de synthèse, pour lesquelles les poses sont connues. Nous proposons un descripteur original permettant de coder la silhouette binaire 3D de la personne.

La quatrième partie aborde le problème du suivi d'un motif plan texturé en apprenant la relation entre le déplacement du motif et la variation de son apparence.

Dans la dernière partie, je conclus sur ces travaux et je propose des perspectives de recherche associées.

2.2 Techniques d'apprentissage pour l'estimation d'état

On dispose d'un ensemble de données d'apprentissage formées par des couples {état, mesure} : $\mathcal{L} \doteq \{\mathbf{X}_n, \mathbf{Z}_n\}_{n=1}^N$. Il s'agit d'apprendre la relation entre les mesures et l'état, à partir de $\mathcal{L}$, afin d'estimer un état $\mathbf{X} \doteq (x^1, x^2, ...x^M) \in I\!R^M$, connaissant les mesures $\mathbf{Z} \in I\!R^D$ que l'observation de ce dernier à généré. Bien que cela ne soit généralement pas vrai, on fait l'hypothèse que cette relation est approximée par une

fonctionnelle, dont le modèle est une combinaison linéaire d'un ensemble de fonctions de base prédéfinies :

$$\mathbf{X} = \sum_{k=1}^{K} \mathbf{w}_k \phi_k(\mathbf{Z}) + \boldsymbol{\epsilon} \tag{2.1}$$

$$\text{ou encore : } \mathbf{X} = \mathtt{W}\boldsymbol{\phi}(\mathbf{Z}) + \boldsymbol{\epsilon} \tag{2.2}$$

Ici, $\{\phi_k(\mathbf{Z})|k = 1, .., K\}$ sont les fonctions de base, $\mathbf{w}_k \in I\!R^k$ sont les vecteurs de poids et $\boldsymbol{\epsilon}_k$ est une matrice contenant les erreurs résiduelles additives. La deuxième forme de l'équation est une formalisation plus compacte du modèle avec $\mathtt{W} \doteq (\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_K)$, une matrice de poids de taille $M \times K$ et $\boldsymbol{\phi}(\mathbf{Z}) \doteq (\phi_1(\mathbf{Z}), \phi_2(\mathbf{Z}), ...\phi_K(\mathbf{Z}))^T$ une fonction qui retourne un vecteur de dimension K. Il peut être utile d'introduire un offset constant dans le modèle : $\mathtt{W}\boldsymbol{\phi}(\mathbf{Z}) + b$ que l'on obtient aussi en choisissant pour dernière fonction de base $\phi_K(\mathbf{Z}) \doteq 1$ dans $\boldsymbol{\phi}(\mathbf{Z})$.
L'apprentissage consiste à calculer, à partir d'un ensemble $\mathcal{L} \doteq \{\mathbf{X}_n, \mathbf{Z}_n\}_{n=1}^N$, les valeurs à affecter aux $M \times K$ paramètres de la matrice de poids $\mathtt{W}$. Lorsque une proportion importante des paramètres de $\mathtt{W}$ sont nuls, on parle de modèles parcimonieux ou épars.
L'apprentissage du lien entre les mesures et l'état revient à l'étude et la mise en place de stratégies d'estimation des paramètres $\mathtt{W}$. Je présente quelques stratégies utilisées en *machine learning*.

2.2.1 Apprentissage par optimisation sous contrainte (SVM)

Les techniques SVM (*Support Vector Machine*), (111) très populaires dans des applications de classification, estiment $\mathtt{W}$ comme solution d'un problème de minimisation de fonctionnelle sous contrainte. Pour chaque paramètre x^m du vecteur d'état $\mathbf{X}$, le modèle général (2.2) peut se ré-écrire :

$$x^m = \mathbf{w}^m.\phi(\mathbf{Z}) \doteq f^m(\mathbf{Z}; \mathbf{w}^m) \tag{2.3}$$

où $\mathbf{w}^m$ représente la transposée de la ligne m de la matrice $\mathtt{W}$. Vapnik propose de mesurer la qualité d'un modèle SVM par une fonction de coût $L(x^m, f^m(\mathbf{Z}; \mathbf{w}^m))$, appelée ϵ-*insensitive* :

$$L(x^m, f^m(\mathbf{Z}; \mathbf{w}^m)) = \begin{cases} 0 & \text{si } |x^m - f^m(\mathbf{Z}; \mathbf{w}^m)| \leq \epsilon \\ |x^m - f^m(\mathbf{Z}; \mathbf{w}^m)| - \epsilon & \text{sinon} \end{cases} \tag{2.4}$$

En appliquant cette fonction de coût à un ensemble d'apprentissage, le risque empirique associé se définit simplement comme la somme des coûts sur la base d'apprentissage :

$$R_{emp}(\mathbf{w}^m) \doteq N^{-1} \sum_{n=1}^{N} L(x_n^m, f^m(\mathbf{Z}_n; \mathbf{w}^m)) \tag{2.5}$$

Les méthodes SVM proposent de construire l'estimation de $\mathbf{w}^m$ sur une double contrainte. D'une part, il faut minimiser le risque empirique afin d'apprendre au mieux la relation entre l'état et les mesures. D'autre part, on cherche à réduire la complexité du modèle en minimisant la norme du vecteur de paramètres $||\mathbf{w}^m||$. La combinaison de ces deux termes conduit à la minimisation sous contraintes suivante :

$$\begin{array}{l} \mathbf{w}_{SVM}^m = \arg\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{n=1}^{N} (\xi_n + \xi'_n) \\ \begin{cases} x_n^m - f^m(\mathbf{Z}_n; \mathbf{w}^m) \leq \epsilon - \xi'_n \\ x_n^m - f^m(\mathbf{Z}_n; \mathbf{w}^m) \geq \epsilon - \xi_n \\ \xi_n, \xi'_n \geq 0 \end{cases} \end{array} \tag{2.6}$$

L'utilisation d'une formalisation Lagrangienne permet d'écrire ce problème sous sa forme duale, pour le résoudre. Plus de détails sont disponibles dans (111). Les propriétés de généralisation du modèle dépendent, en grande partie, d'un choix optimal des fonctions de base, des paramètres associés, et des deux paramètres de minimisation C et ϵ. De manière pratique, ces réglages s'obtiennent, soit par des méthodes ad-hoc, soit par de techniques de validation croisée.

2.2.2 Apprentissage par minimisation au sens des moindres carrés

Une méthode assez classique, pour estimer le jeu de paramètres, consiste à utiliser un critère au sens des moindres carrés. Ce type de méthode a été utilisé par Agarwal et Triggs (2) pour une application d'estimation de posture.
L'apprentissage des paramètres de $\mathtt{W}$ est issu d'une minimisation de l'erreur euclidienne entre la pose estimée et la pose réelle ; ce qui peut se définir de la manière suivante :

$$\mathtt{W}_{lsq} = \arg\min_{\mathtt{W}} \left\{ \sum_{n=1}^{N} ||\mathtt{W}\boldsymbol{\phi}(\mathbf{Z}_n) - \mathbf{X}_k||^2 + R(\mathtt{W}) \right\} \tag{2.7}$$

$R(\mathtt{W})$ est un terme de régularisation dont but principal est d'éviter les problèmes de sur-apprentissage[1]. En regroupant l'ensemble des états dans la matrice $\mathtt{X} \doteq (\mathbf{X}_1, \mathbf{X}_2, ...\mathbf{X}_N)$, ainsi que les sorties de fonction de base dans une matrice de taille $K \times N$ définie par $\mathtt{\Phi} \doteq (\boldsymbol{\phi}(\mathbf{Z}_1), \boldsymbol{\phi}(\mathbf{Z}_2), ..., \boldsymbol{\phi}(\mathbf{Z}_N)$, la minimisation de $\mathtt{W}$ se ré-écrit :

$$\mathtt{W} = \arg\min_{\mathtt{W}} \left\{ ||\mathtt{W}\mathtt{\Phi} - \mathtt{X}||^2 + R(\mathtt{W}) \right\} \tag{2.8}$$

avec $||.||$, opérateur définissant la norme de Frobenius.
Dans le cas de problèmes de dimension élevé, il est fréquent que le système linéaire soit mal conditionné. La résolution de $\mathtt{W}$ sans terme de régularisation peut alors conduire à l'obtention d'un modèle qui modélisera le bruit d'apprentissage, et dont les propriétés de généralisation risquent d'être très pauvres. Pour éliminer ce phénomène, on ajoute classiquement une contrainte de lissage, en introduisant, par exemple, un terme de régularisation dont le but est de pénaliser les paramètres de $\mathtt{W}$ dont la valeur est élevée. Le choix le plus simple consiste à à poser $R(\mathtt{W}) \doteq \lambda||\mathtt{W}||^2$ où λ est un paramètre d'équilibrage :

$$\mathtt{W} = \arg\min_{\mathtt{W}} \left\{ ||\mathtt{W}\tilde{\mathtt{\Phi}} - \tilde{\mathtt{X}}||^2 \right\} \tag{2.9}$$

$$||\mathtt{W}\tilde{\mathtt{\Phi}} - \tilde{\mathtt{X}}||^2 = ||\mathtt{W}\mathtt{\Phi} - \mathtt{X}||^2 + \lambda||\mathtt{W}||^2 \tag{2.10}$$

avec $\tilde{\mathtt{\Phi}} \doteq (\mathtt{\Phi} \;\; \lambda\mathbf{I})$ et $\tilde{\mathtt{X}} \doteq (\mathtt{X} \;\; \mathbf{0})$
Cette méthode de pénalisation n'est pas homogène lorsque les données d'entrée ne sont pas toutes à la même échelle. Il faut donc veiller à normaliser les vecteur de mesure et d'état pour qu'ils soient à variance unitaire. Le paramètre λ doit être suffisamment important pour éviter les problèmes de sur-apprentissage, et suffisamment faible pour ne pas forcer tous les paramètres de $\mathtt{W}$ à être nuls. L'ajustement de ce paramètre peut être effectué par validation croisée.
Lorsque le modèle est défini avec un biais, la régularisation ne doit pas concerner ce dernier et on pose :
$\tilde{\mathtt{\Phi}} \doteq \begin{pmatrix} \mathtt{\Phi} & \lambda\mathbf{I} \\ \mathbf{1} & \mathbf{0} \end{pmatrix}$ et $\tilde{\mathtt{X}} \doteq (\mathtt{X} \;\; \mathbf{0})$

2.2.3 Apprentissage par maximisation de la probabilité a posteriori (RVM)

En 2000, Tipping (107) a proposé une approche probabiliste pour l'estimation de la matrice de paramètres $\mathtt{W}$. Pour cela, le problème d'optimisation est ré-écrit dans un contexte Bayesien, utilisant des mécanismes similaires aux processus Gaussiens.(65). On considère que la séquence d'état est issue d'un processus aléatoire, réalisation du modèle 2.2 ($\mathbf{X} = \sum_{k=1}^{K} \mathbf{w}_k \phi_k(\mathbf{Z}) + \boldsymbol{\epsilon} \doteq \mathtt{W}\boldsymbol{\phi}(\mathbf{Z}) + \boldsymbol{\epsilon}$) dans lequel on cherche le jeu de paramètres qui minimise un critère lié à l'erreur $\boldsymbol{\epsilon}$. Classiquement, un critère au sens des moindres carrés consiste à minimiser la somme quadratique des termes de la matrice $\boldsymbol{\epsilon}$, soit $\sum_{i,j} \epsilon_{i,j}^2$. De plus, on considère que le modèle de bruit est aléatoire et suit une loi normale de moyenne nulle et de variance σ^2 :

[1] le sur-apprentissage (*overfitting*) apparaît généralement lorsque le phénomène à modéliser comporte un bruit important et que l'on impose un modèle à apprendre qui possède trop de degrés de liberté. Le résultat est un modèle dont les capacités de généralisation sont mauvaises

$p(\epsilon_{i,j}|\sigma^2) = \mathcal{N}(0, \sigma^2)$. l'état suit alors un processus aléatoire dont la densité de probabilité est une loi normale centrée sur la prédiction et de variance σ^2 :

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}, \mathtt{W}, \sigma^2) &= \prod_{m=1}^{M} \prod_{n=1}^{N} p(X_n^m|\mathbf{Z}_n, \mathtt{W}, \sigma^2) \\ &= \prod_{m=1}^{M} \prod_{n=1}^{N} (2\pi\sigma^2)^{-1/2} \exp\left[-\frac{\{X_n^m - \mathbf{w}^m.\boldsymbol{\phi}(\mathbf{Z}_n)\}^2}{2\sigma^2}\right] \end{aligned} \tag{2.11}$$

L'estimation du jeu de paramètres $\mathtt{W}$ peut s'effectuer via la méthode du maximum de vraisemblance, en maximisant le log de la vraisemblance, soit :

$$\log p(\mathbf{X}|\mathbf{Z}, \mathtt{W}, \sigma^2) = -\frac{Nm}{2}\log(2\pi\sigma^2) - \frac{1}{2\pi\sigma^2}\sum_{m=1}^{M}\sum_{n=1}^{N}\{X_n^m - \mathbf{w}^m.\boldsymbol{\phi}(\mathbf{Z}_n)\}^2 \tag{2.12}$$

Le premier terme ne dépendant pas des données, cette estimation du jeu de paramètres conduit aux mêmes résultats qu'un critère de type moindres carrés, avec le même risque de sur-apprentissage.
Il est donc important, tout comme pour l'approche de type moindres carrés, d'introduire des *a priori* afin 1) de régulariser le modèle obtenu et 2) de le rendre parcimonieux. Ces derniers sont spécifiés dans un cadre Bayesien sous la forme d'une distribution *a priori* du jeu de paramètres $\mathtt{W}$:

$$p(\mathtt{W}|\alpha) = \prod_{m=1}^{M}\prod_{n=1}^{N}\left(\frac{\alpha}{2\pi}\right)^{1/2} \exp\left[-\frac{\alpha}{2}(w_n^m)^2\right] \tag{2.13}$$

La répartition du jeu de paramètres suit donc une loi normale de moyenne nulle, ce qui favorise les faibles valeurs, et ainsi régularise le modèle, de la même façon que ce dernier est régularisé par le paramètre λ dans le cas des moindres carrés pénalisés (PLS). La valeur de α joue un rôle similaire à celle de λ en ajustant la part de régularisation souhaitée. Connaissant la vraisemblance et les *a priori*, le calcul de la densité *a posteriori* s'effectue alors en appliquant la règle de Bayes :

$$p(\mathtt{W}|\mathbf{X}_1, ...\mathbf{X}_N, \alpha, \sigma^2) = \frac{\text{vraisemblance } \times \ a\ priori}{\text{probabilité marginale}} = \frac{p(\mathbf{X}_1, ...\mathbf{X}_N|\mathtt{W}, \sigma^2)p(\mathtt{W}|\alpha)}{p(\mathbf{X}_1, ...\mathbf{X}_N|\alpha, \sigma^2)} \tag{2.14}$$

Une méthode d'estimation des paramètres de $\mathtt{W}$ est proposée dans (106), en traitant spécifiquement du cas où $\mathbf{X}$ est de dimension supérieure à un. La méthode la plus populaire consiste à décomposer l'état selon chacune de ses composantes et à calculer un modèle pour chaque composante :

$$p(\mathbf{w}^m|x_1^m, ...x_N^m, \alpha, \sigma^2) = \frac{p(x_1^m, ...x_N^m|\mathbf{w}^m, \sigma^2)p(\mathbf{w}^m|\alpha)}{p(x_1^m, ...x_N^m|\alpha, \sigma^2)} \tag{2.15}$$

Pour alléger les notations, l'indice m sera omis dans la suite de ce paragraphe. Comme la fonction de vraisemblance (*likelihood*) et les *a priori* (*prior*) suivent une loi Gaussienne, la distribution *a posteriori* est de même type :

$$p(\mathbf{w}|x_1, ...x_N, \alpha, \sigma^2) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{2.16}$$

$$\boldsymbol{\mu} = (\Phi^T\Phi + \sigma^2\alpha\mathtt{I})^{-1}\Phi^T(x_1, ...x_N)^T \tag{2.17}$$

$$\Sigma = \sigma^2(\Phi^T\Phi + \sigma^2\alpha\mathtt{I})^{-1} \tag{2.18}$$

L'estimation de $\mathbf{w}$ par une règle de maximum *a posteriori* (MAP) consiste à rechercher le jeu de paramètres le plus probable, pour la distribution $p(\mathbf{w}|x_1, ...x_N, \alpha, \sigma^2)$. Comme le dénominateur de la règle de Bayes est indépendant du jeu de paramètres, c'est équivalent à maximiser le numérateur, ou encore, minimiser : $E_{MAP}(\mathbf{w}) = -\log p(x_1, ...x_N|\mathbf{w}, \sigma^2) - \log p(\mathbf{w}|\alpha)$. En conservant uniquement les termes non constants, on obtient l'expression suivante :

$$E_{MAP}(\mathbf{w}) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}\left[\{x_n - \mathbf{w}\boldsymbol{\phi}(\mathbf{Z_n})\}^2 + \frac{\alpha}{2}w_n^2\right] \tag{2.19}$$

Le critère obtenu est équivalent à celui de la méthode des moindres carrés pénalisés (équation 2.10) avec $\lambda = \sigma^2\alpha$.
Dans le cas des techniques RVM (*Relevance Vector Machine*), l'*a priori* sur les paramètres est modifié de la manière suivante :

$$p(\mathbf{w}|\alpha_1, ...\alpha_N) = \prod_{n=1}^{N} \left[(2\pi)^{-1/2} \alpha_n^{1/2} \exp\left\{ \frac{-1}{2} \alpha_n w_n^2 \right\} \right] \tag{2.20}$$

Un hyper-paramètre est associé à chaque paramètre du vecteur de poids $\mathbf{w}$, contrôlant l'inverse de leur variance. De cette manière, et en jouant sur les *a priori* des hyper-paramètres α_n, on définit un comportement plus ou moins épars du modèle obtenu. Plus de détails sur cette méthode sont disponibles dans (108).

2.2.4 Cas de la classification.

On s'intéresse ici au problème de la classification d'objets. Ce dernier peut être vu comme un cas particulier de la régression dans lequel l'état est discret et définit les classes possibles d'appartenance de l'objet que l'on cherche à classer. On dispose donc d'un ensemble d'apprentissage composé par des couples {étiquette, mesure} : $\mathcal{L} \doteq \{\mathcal{X}, \mathcal{Z}\} \doteq \{X_n, \mathbf{Z}_n\}_{n=1}^{N}$. Dans un problème d'apprentissage classique, on souhaite lier une mesure à une classe parmi L. Lorsqu'il s'agit de détecter la présence d'une catégorie d'objets, le problème est restreint à deux classes. Soit X, une variable aléatoire discrète qui peut prendre deux états possible : $X = X_{in}$ si X est un objet de la classe recherchée et $X = X_{out}$ si X n'est pas un objet de la classe recherchée. Généralement, on associe la valeur numérique 1 à la classe X_{in} et la valeur numérique -1 à la classe X_{out} ; ce qui fait que l'état $X \in \{-1; 1\}$. Dans ce cadre, le modèle 2.2 est utilisé pour définir une règle de décision :

$$X = \text{sign}\left(\sum_{k=1}^{K} w_k \phi_k(\mathbf{Z}) \right) \tag{2.21}$$

$$X = \text{sign}\big(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{Z})\big) \tag{2.22}$$

2.2.5 Illustration des méthodes d'apprentissage sur un exemple simple

Les performances des méthodes à noyaux présentées dans la section précédente dépendent d'un certain nombre de facteurs comme le choix des fonctions de base et les paramètres associés, ou encore le terme de régularisation. Une illustration de l'effet de ces facteurs sur la classification est proposée à partir d'une base de synthèse simple fournie par Ripley (88). Cette dernière est composée de deux classes dont les observations sont fournies par des vecteurs de dimension deux. L'ensemble d'apprentissage est composé de cinquante exemples de chaque classe. D'autre part, on dispose d'un ensemble de test composé de 1000 éléments.
Pour chaque test, nous donnons une représentation graphique de la frontière entre les deux classes, ainsi que l'erreur de classification, exprimée en pourcentage. Les différents algorithmes d'apprentissages testés sont les suivants :

- ▷ Apprentissage du jeu de paramètres par un critère au sens des moindres carrés (LS),
- ▷ apprentissage du jeu de paramètres par un critères au sens des moindres carrés pénalisés (PLS),
- ▷ apprentissage du jeu de paramètres par la méthode *Relevant Vector Machine* (RVM),
- ▷ apprentissage du jeu de paramètres par la méthode *Support Vector Machine* (SVM),
- ▷ approche non paramétrique par la méthode des fenêtres de Parzen (KDE) (*Kernel Density Estimation*)

Tous ces algorithmes sont basé sur le modèle 2.2. La méthode non paramétrique utilisant les fenêtres de Parzen peut être considérée comme méthode de référence. Il s'agit d'une régle de décision basée sur un log.

ratio de vraisemblance ; cette dernière étant estimée par la méthode des fenêtres de Parzen[2]
D'autre part, des tests ont également été réalisés pour illustrer l'effet de la modification de la largeur du noyau ou le choix d'un noyau linéaire.

Méthode	Type de fonction de base	Largeur du noyau	Erreur	Nb de vecteurs de base
KDE	Gaussien	0.3	9.6%	-
LS	Gaussien	0.5	40.7%	-
LS	linéaire	-	27%	-
PLS	Gaussien	0.5	8.4%	14
SVM	Gaussien	1	8.8%	29
RVM	Gaussien	0.5	8.2%	4
SVM	Linéaire	–	10.5%	63
LS	Gaussien	7	11.3%	-

TABLE 2.1 – Comparaison des performances des différentes méthodes testées sur la base *Ripley*, en terme d'erreur de classification et de nombre de fonctions de bases utilisées dans le modèle (pour les méthodes dites parcimonieuses).

La figure 2.2 montre les frontières de décision obtenues dans le cas des cinq algorithmes testés. Le tableau 2.1 synthétise les erreurs de classification issues des cinq algorithmes, ainsi que le nombre de vecteurs de base utilisés dans le cas de modèles parcimonieux.
L'algorithme le plus simple consiste à considérer que tous les points d'apprentissage ont le même poids et contribuent à la décision finale. La densité de probabilité de chaque classe est estimée par un modèle de type KDE (*Kernel Density Estimation*), aussi appelé fenêtres de Parzen. La frontière de décision est obtenue par le passage à zéro d'une fonction discriminante de type *log. ratio.* de vraisemblance.
La deuxième méthode testée utilise un critère au sens des moindres carrés. Dans le cas de fonctions de bases non linéaires (algorithme LS Gaussien par exemple), l'erreur de classification est très importante. L'utilisation de fonctions noyaux de type gaussiennes rend le problème séparable dans l'espace de redescription (le nombre de degrés de liberté du modèle devient très important, causant des problèmes de sur échantillonnage), et le système linéaire ainsi formé possède une solution, qui explique exactement les couples observation, état de l'ensemble d'apprentissage. Néanmoins, cette solution ne possède pas de bonnes propriétés de généralisation, ce qui conduit à une erreur de classification importante (40.7%). Ce phénomène se constate également en observant la forme de la frontière de décision obtenue. Elle englobe les données d'apprentissage mais est très irrégulière. Il est donc nécessaire d'injecter des contraintes de régularisation au niveau du critère.
En pénalisant le critère d'attache aux données (moindre carré) par une contrainte de régularisation basée sur la norme du jeux de paramètres, on obtient l'algorithme des moindres carrés régularisés (PLS Gaussien). La difficulté consiste alors à ajuster l'influence du terme de régularisation par rapport au terme d'attache aux données. Pour un bon compromis entre les deux termes (obtenu par validation croisée), on constate qu'un certain nombre de paramètres sont très proches de zéro. et ont une influence négligeable sur le calcul de l'estimation. Dans l'exemple donné, seuls 14 paramètres sont conservés. Les propriétés de généralisation du modèle obtenu sont meilleures et l'erreur de classification, sur la base de test est ramenée à 8.4%.
Une autre forme de pénalisation des paramètres consiste à faire l'hypothèse qu'ils suivent une loi normale de moyenne nulle. C'est une manière de les contraindre à être proches de zéros. L'algorithme RVM (Gaussien)

[2]Estimation de vraisemblance par la méthode des fenêtres de Parzen (KDE, *Kernel Density Estimation*)

$$p(\mathbf{Z}|X = X_{in}) \propto \sum_{X \in \mathcal{X}|X=X_{in}} \phi_n(\mathbf{Z})$$

$$X = \text{sign}\left(\sum_{n=1}^{N} X_n \phi_n(\mathbf{Z})\right)$$

où $\phi_n(\mathbf{Z})$ est une fonction de base

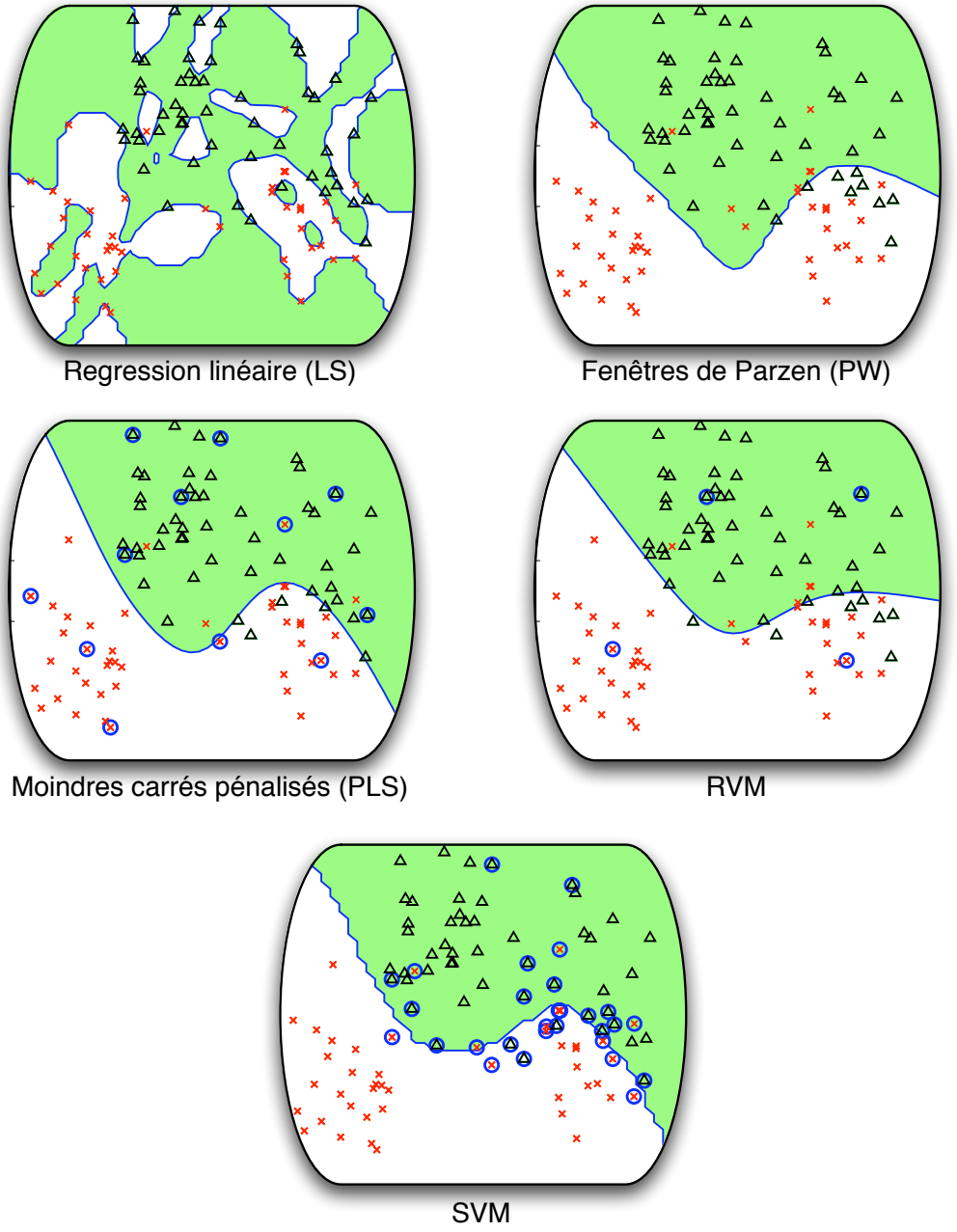


FIGURE 2.2 – Comparaison des frontières de décision obtenues pour plusieurs types d'algorithmes d'estimation du jeux de paramètres `W` du modèle 2.2. Les croix rouges et les triangles noirs représentent les points d'apprentissage. Les rond bleus représentent les vecteurs de base utilisés dans le cas d'algorithmes parcimonieux. La surface verte représente la région de l'espace des paramètres dans laquelle les points appartiennent à la classe des triangles noirs et la surface blanche représente la région dans laquelle les points appartiennent à la classe de points rouges. Illustration en couleur

fonctionne sur ce principe. Dans l'exemple donné, on conserve uniquement 4 paramètres non nuls, tout en conservant un taux d'erreur faible sur la base de test, 8.2%.
Dans le cas de l'algorithme SVM (Gaussien), on constate que le taux d'erreur obtenu est également faible (8.4%) mais le nombre de vecteurs supports nécessaires (paramètres non nuls) est un plus important que dans le cas des algorithmes RVM ou PLS.
Cette section a décrit, de manière synthétique, les concepts utilisés dans le cadre de mes travaux sur l'estimation d'état basé apprentissage. Dans les exemples qui suivent, ces derniers sont utilisés dans des contextes applicatifs tels que la détection de piétons, l'estimation de posture, ou encore le suivi d'objets planaires dans une séquence d'images.

2.3 Application à la détection de piétons

La détection de piétons dans les images ou des séquences d'images a pris un essor important avec l'apparition des caméras dans la vie de tous les jours. On les utilise aujourd'hui pour faire de la vidéo surveillance dans les magasins ou dans les lieux sensibles (aéroport, banques,...), pour développer des systèmes d'aide à la conduite dans les véhicules (évitement de collisions, activation d'airbags,...),... . Toutefois, un piéton reste un objet difficile à détecter de par sa grande variabilité intrinsèque (taille des personnes, habillement,...) et extrinsèque (changement de pose et d'illumination). Il semble donc naturel de chercher à modéliser cette complexité à l'aide de méthodes basées apprentissage. Les travaux présentés dans cette partie sont principalement issus de la thèse de Laetitia Leyrit, qui s'est déroulée dans le cadre du projet ANR-LOVE[3]

2.3.1 Positionnement bibliographique

La conception d'une méthode de détection de piétons basée apprentissage s'articule autour de deux éléments clefs :

1. **La description de l'image**. Il s'agit de construire un vecteur de primitives codant le contenu d'une image. Ce contenu étant à la fois géométrique et photogrammétrique, certaines descriptions encodent plutôt la répartition spatiale de l'image tandis que d'autres se focalisent sur la répartition globale des caractéristiques photogrammétriques des pixels dans l'image ; un bon descripteur d'image devant combiner ces deux informations complémentaires. Parmi les descriptions spatiales, les plus populaires sont les descriptions sous la forme d'ondelettes de Haar, initialement proposées dans (81), puis reprises dans (115). Dalal et Triggs proposent d'utiliser des histogrammes d'orientation des Gradients (HOG)(27), qui encodent de manière efficace la silhouette d'un piéton. Les *local binary patterns* (LBP) (76) (61)(71), ont également été utilisées pour coder les relations spatiales entre les pixels, avec de bons résultats, notamment dans des applications de détection de visages (5). Récemment, des codages basés sur la covariance de primitives regroupant à la fois des informations spatiales et photogrammétriques ont été utilisées (110).

   Le descripteur, codé sous la forme d'un vecteur (aussi appelé vecteur de primitives), doit permettre de séparer les objets de la classe recherchée avec des objets qui n'appartiennent pas à cette classe. On parle alors de distance inter-classe. D'autre part, il doit également être capable de regrouper tous les objets d'une même classe, mais dont l'apparence est différente. On parle alors de distance intra-classe. La performance d'un descripteur, traduit par une distance intra-classe faible et une distance inter-classe importante, nécessite donc la définition d'une mesure de distance entre deux descripteurs. Dans le cas de descripteurs issus d'une matrice de covariance (110), où l'ensemble des descripteurs possibles est positionné sur une variété de l'espace, le calcul de la distance doit prendre en compte cette particularité.

[3]LOVe propose de contribuer à la sécurité routière en mettant principalement l ?accent sur la sécurité des piétons. L'objectif est d ?aboutir à des logiciels d'observation des vulnérables fiables et sûrs implantés sur des matériels compatibles avec une exploitation industrielle rapide. Plusieurs solutions algorithmiques de perception sont mises en ?uvre dans un double objectif : celui de la concurrence afin d'identifier les solutions à même de répondre au mieux aux exigences et celui de la complémentarité afin d'aboutir à des systèmes combinés lorsque les contraintes l'exigent.

Deux objets visuellement proches doivent se traduire par une faible distance entre leurs descripteurs. Enfin, la dimension du vecteur de description est un point important dans son choix. Ce dernier doit être en relation avec le nombre d'exemples disponibles pour réaliser un apprentissage. En effet, le but de toute machine d'apprentissage étant de rechercher un séparateur entre les exemples positifs et les exemples négatifs, dans l'espace de description, la taille de ce dernier doit être bien inférieure au nombre d'exemples utilisés. Il est alors possible de passer par une sélection de primitives (54) ou une méthode de réduction de la dimension de type ACP (Analyse en Composantes Principales) pour réduire la taille du vecteur de primitives. Parfois, la stratégie de sélection est intégrée à l'intérieur de la machine d'apprentissage (55) (29) .

2. **La machine d'apprentissage**. Le but d'une machine d'apprentissage est de produire une fonction capable de décider si une image appartient à une classe recherchée ou pas. La construction de cette fonction s'effectue selon un apprentissage, à partir d'une collection d'images de la classe recherchée (exemples positifs) et une collection d'objets autres (exemples négatifs). Pour la détection de piétons, les méthodes se divisent en deux catégories. La première utilise des modèles à noyau comme ceux présentés au début de ce chapitre. Les algorithmes de type SVM (81), ou RVM (27) sont alors utilisés pour calculer le séparateur non-linéaire entre les deux classes. La deuxième méthode est basée sur des techniques de Boosting. Il s'agit de combiner un ensemble, souvent important, de classifieurs faibles. Les paramètres de la combinaison linéaire formant le classifieur fort obtenu sont par exemple calculés par l'algorithme Adaboost (115). Une des difficultés associées aux machines d'apprentissage dans le cadre des applications de détection de piétons concerne le grand nombre d'exemples nécessaires pour représenter correctement la variabilité intra-classe. Il en résulte des problématiques d'apprentissage dans des grandes bases et il faut alors réfléchir aux stratégies d'apprentissage à mettre en oeuvre pour que les algorithmes proposés puissent se déployer sur les architectures actuelles.

De nouvelles approches ont été développées afin de combiner les avantages des descripteurs les plus utilisés. Dans (37), deux types de descripteurs sont utilisés : des ondelettes de Haar et une variante des histogrammes de gradients. Ces deux descripteurs sont représentés par des valeurs réelles et des classifieurs faibles sont simplement créés en établissant une règle de décision à partir d'un seuil ; un algorithme AdaBoost est utilisé en cascade afin de créer le classifieur fort final. Dans (73), une association de deux descripteurs est présentée pour la reconnaissance de voitures. Il s'agit de concaténer un descripteur rectangulaire (de type ondelettes de Haar) à un descripteur d'histogrammes de gradients orientés dans le cadre d'une cascade de classifieurs AdaBoost. Dans les deux cas, la fusion de deux descripteurs permet d'augmenter les scores de reconnaissance. Le travail que nous avons réalisé tente de combiner, d'une part, différents types de descripteurs, et d'autre part, les deux classifieurs les plus populaires qui sont les méthodes SVM avec l'algorithme Adaboost.

2.3.2 Méthode

2.3.2.1 Les descripteurs utilisés

Notre première contribution concerne l'étude des performances d'un système de détection de piéton combinant plusieurs types de descripteurs. Dans cette étude, trois types de descripteurs ont été sélectionnés : les ondelettes de Haar, les histogrammes d'orientation des gradients et une variante des *Local Binary Pattern*.

- ▷ **Les ondelettes de Haar** : il s'agit du descripteur le plus utilisé pour la reconnaissance de piétons (81; 113). Dans (81), les auteurs définissent un dictionnaire complet d'ondelettes pour la classification. Il s'agit du calcul des composantes des ondelettes dans trois directions principales : horizontale, verticale et diagonale (voir figure 2.3). La taille de ces ondelettes est ensuite adaptée à la résolution de l'objet à décrire.

- ▷ **Les histogrammes de gradients orientés** : l'utilisation de ce type de descripteurs a été proposé dans (27). Le principe est d'utiliser une version simplifiée du descripteur SIFT (63) pour de la reconnaissance temps réel d'objet. L'image de l'objet à caractériser est découpée en plusieurs cellules pour lesquelles on comptabilise les occurrences de l'orientation du gradient dans un histogramme. Plusieurs versions de

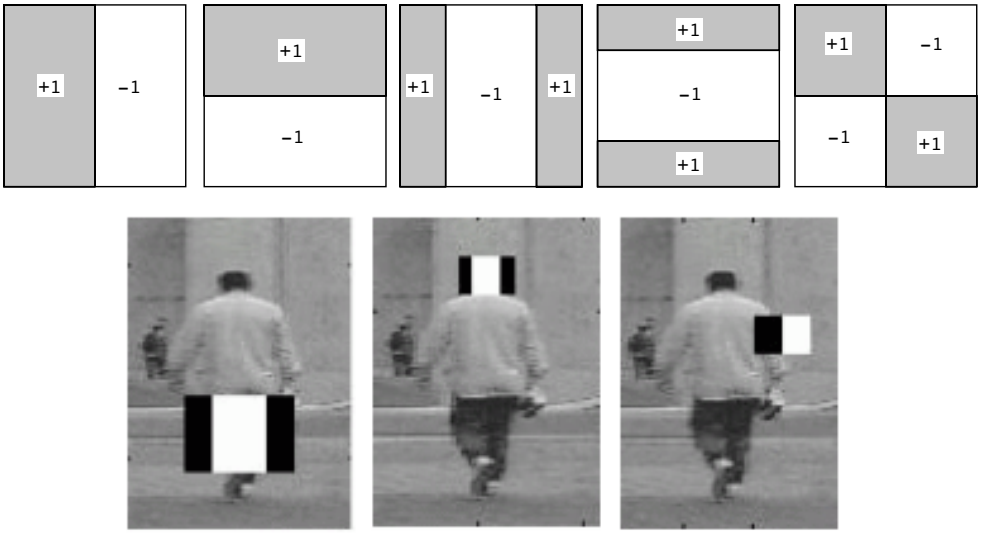


FIGURE 2.3 – Ondelettes de Haar : elles se déclinent sous la forme de masques binaires (partie haute de la figure). La partie basse de la figure illustre quelques ondelettes pertinentes pour la détection de piétons

ce type de descripteur ont été utilisées : certaines agissent sur la normalisation des histogrammes (101), d'autres comptabilisent la magnitude du gradient au lieu des occurrences seules (13). De même que pour les ondelettes de Haar, ce type de descripteur doit également être adapté à la taille et la résolution de l'objet à reconnaître, notamment pour la taille des cellules.

▷ **Le descripteur binaire de comparaison de pixels** : ces types de descripteurs simples se développent de plus en plus car ils ont l'avantage d'être rapides et d'assez bonne qualité pour des images de faible résolution (61; 71). Dans le cas de la reconnaissance de piétons, ils sont particulièrement bien adaptés au vue des applications : les piétons sont extraits d'images provenant de caméras avec des résolutions standard (640x480 pixels) et ne sont en général représentés que par quelques dizaines de pixels. Nous utiliserons ici le descripteur présenté dans (61). Il effectue une comparaison de l'intensité de certains pixels de l'image et renvoie une information binaire (voir figure 2.4). Les meilleurs couples de points sont choisis au préalable avec une sélection de variable par l'algorithme AdaBoost (29).

Dans (37) et (73), deux types de descripteurs sont utilisés et concaténés afin d'améliorer les résultats de la classification. Au lieu de concaténer les différents descripteurs pour chaque image, ce qui donnerait un vecteur de caractéristiques de très grande dimension pour chaque objet, nous proposons dans un premier temps de fusionner les résultats des différents classifieurs (chacun utilisant un descripteur différent) par des règles de combinaison basiques :

▷ **la moyenne des descripteurs**. La somme des valeurs réelles en sortie des classifieurs associés à chaque type de descripteur est calculée ; la règle de décision est ensuite donnée par le signe de cette somme.

▷ **le max des descripteurs**. On conserve le classifieur dont l'amplitude de la sortie est maximale. la règle de décision est ensuite donnée par le signe de ce dernier.

2.3.2.2 Combinaison de classifieurs (Adaboost, machine à noyau)

La deuxième contribution de ce travail concerne l'étude des performances d'un classifieur combinant les deux classifieurs les plus utilisés en détection de piétons (l'algorithme Adaboost et les machines à noyau). Le principe de cette méthode est illustré sur la figure 2.5. L'algorithme Adaboost permet de construire un ensemble de classifieurs faibles et le classifeur fort associé qui peut être vu comme un séparateur linéaire dans

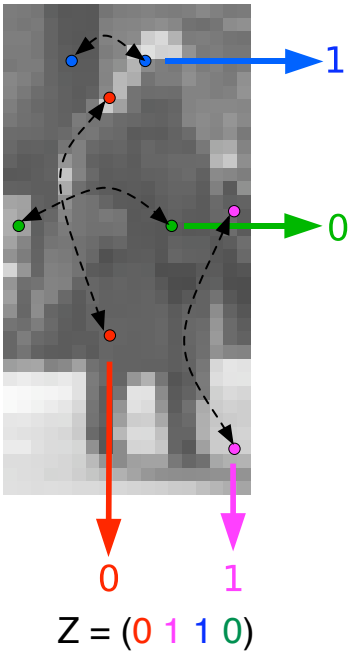


FIGURE 2.4 – Descripteur basé sur la comparaison de niveaux de gris

Pour un couple de points données, le descripteur retourne une valeur logique $d \in \{0, 1\}$ *correspondant au résultat du test* $\{Intensité(point_1) \geq Intensité(point_2)\}$

l'espace des classifieurs faibles. Les classifieurs faibles sont alors utilisés comme des descripteurs binaires en entrée d'une machine d'apprentissage basée sur un modèle à noyau. Le classifieur généré construit alors un séparateur non linéaire dans l'espace des classifieurs faibles.

De plus, le surcoût algorithmique engendré par la combinaison de l'adaboost et du modèle à noyau demeure faible car les opérations effectuées par le modèle à noyau concernent toutes des paramètres binaires.

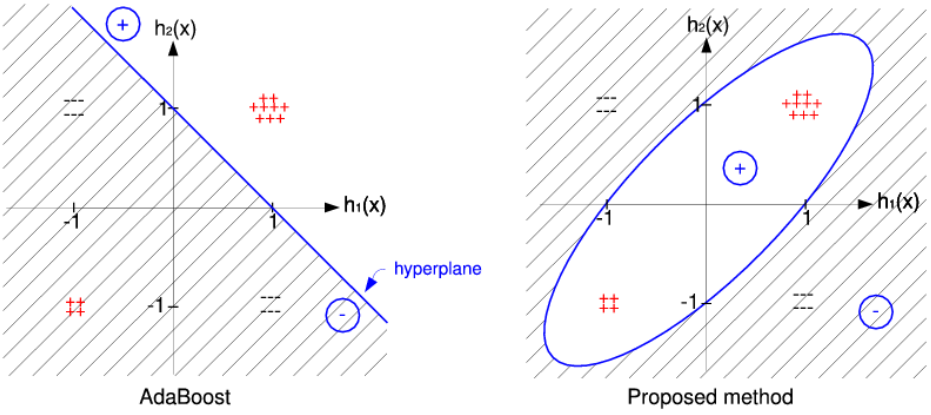


FIGURE 2.5 – Synoptique illustrant le principe de la méthode d'apprentissage combinant un algorithme adaboost, suivi d'un modèle à noyau. La frontière de décision obtenue est non linéaire dans l'espace des classifieurs faibles. Illustration en couleur

2.3.3 Résultats

Les expérimentations ont porté sur trois points : l'apport de la méthode combinant l'adaboost suivi d'un modèle à noyau, la comparaison des trois types de descripteurs utilisés et l'association des descripteurs.
La base de piétons utilisée est une base publique, disponible dans (72) . Elle est subdivisée en cinq parties ; chacune contenant 4500 images positives et 5000 images négatives. Chaque image est codée sur 256 niveaux de gris, pour une taille de 36x18 pixels. Dans les images d'exemples positifs, les piétons se tiennent debout et sont entièrement visibles ; ils ont été pris dans différentes postures, et dans conditions d'illumination de fond variables. Chaque image de piéton a été aléatoirement décalée de quelques pixels dans les directions horizontale et verticale. Les images d'exemples négatifs représentent l'environnement urbain : bâtiments, arbres, voitures, panneaux de signalisations,... .
Les ondelettes de Haar, ont été calculées sur la base des paramètres utilisés dans (72). Deux tailles d'ondelettes sont conservées (4x4 et 8x8) et calculées pour les trois orientations, et avec un recouvrement d'$\frac{1}{4}$ la taille de l'ondelette. Le vecteur de primitive calculé est donc de taille 1755.
Pour les histogrammes de gradients, des cellules de taille 3x3 ont été utilisées, sans recouvrement, et les histogrammes sont calculés sur huit classes. Un vecteur de caractéristiques résultant est de taille 576.
En ce qui concerne le descripteur binaire, une phase de sélection de variables à partir de l'algorithme AdaBoost a permis d'obtenir un vecteur de caractéristiques de taille 2468 pour chaque image.
Le critère utilisé pour les comparaisons est une courbe ROC, affichant le taux de faux positifs en abscisse et le taux de bonnes détection en ordonnée.

- ▷ **Combinaison Adaboost, méthode à noyau** :
  La figure 2.6 présente une comparaison entre trois stratégies d'apprentissage. La première est la stratégie que nous proposons avec sélection d'attributs par AdaBoost et utilisation d'une machine à noyaux de type SVM. Elle est évaluée par rapport à la sortie du classifieur AdaBoost seul qui a gardé 2000 classifieurs faibles et par un apprentissage par SVM dont la sélection d'attributs a été faite de façon aléatoire. Les deux classifieurs SVM ont été entraînés sur un ensemble d'apprentissage contenant 1600 images de positifs et 3200 de négatifs. L'AdaBoost a sélectionné 2000 classifieurs faibles sur l'ensemble initial. Le taux de reconnaissance de la mise en cascade de l'Adaboost avec des SVM est supérieure à celle des SVMs seuls pour des scores de fausses détections identiques. Les performances d'un classifeur AdaBoost seul sont un peu en deça des performances des deux autres méthodes. Par exemple, pour un taux de fausses détections de 10%, La méthode proposée reconnaît 99,1% de piétons, les SVMs avec sélection aléatoire de variables en identifie 96,3% et l'AdaBoost seulement 81,4%.

- ▷ **Combinaison de descripteurs** : Pour comparer les différents descripteurs, ainsi que leurs associations possibles, nous travaillons sur les trois bases d'apprentissage puis testons ces apprentissages sur les bases de test 4 et 5. Les images sont décrites par les trois descripteurs présentés dans cette section. Des classifieurs sont entraînés à partir de chacune de ces descriptions sur les trois bases d'apprentissages, ce qui nous donne 9 apprentissages. Ces apprentissages sont ensuite testés en classification sur les bases de test 4 et 5. Les performances des apprentissages sont évaluées au moyen de courbes ROC (*Receiver Operating Curves*) qui représentent le taux de faux négatifs (des non-piétons pris pour des piétons) en fonction du taux de vrais positifs (des piétons bien reconnus en tant que tels) pour différents seuils de discrimination.

  La figure 2.7 présente les performance des classifieurs, en fonction de type de descripteur utilisé, ainsi que pour les deux stratégies de combinaison proposées. On constate que le descripteur le plus performant est l'histogramme d'orientation des gradients. D'autre part, les deux types d'association proposés permettent d'accroître les performances du classifieur, l'association la plus performante étant la moyenne des trois classifieurs. Pour 10% de fausses alarmes, l'association moyennée des trois classifieurs (chacun basé sur un descripteur différent) atteint 90,20% de bonnes détections, le choix de la valeur maximale donne 84,04% de bonnes détections, tandis que l'apprentissage à partir des histogrammes de gradients n'obtient que 70,45% de bonnes détections.

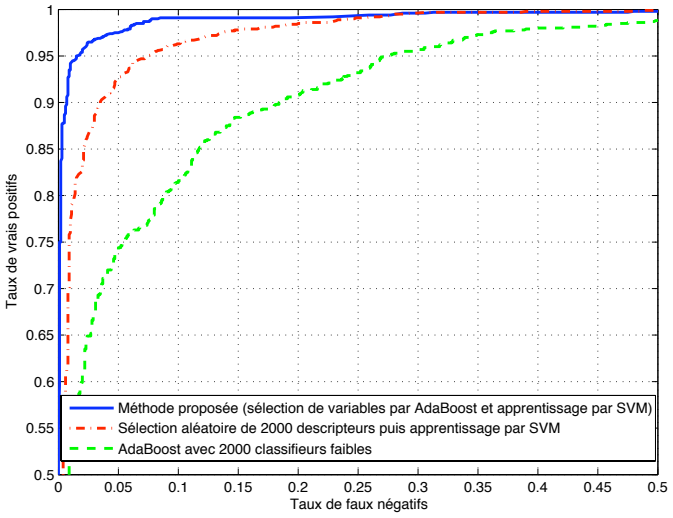


FIGURE 2.6 – Comparaison de la méthode proposée (combinaison Adaboost et SVM avec un classifieur SVM ayant une sélection aléatoire de variables, et une classification par AdaBoost. Illustration en couleur

Nous avons également comparé cette méthode d'association par la moyenne algébrique aux résultats présentés dans (72). Toutes les méthodes développées ont été apprises et testées sur la même base que nous avons utilisée. Nous avons retenu deux systèmes qui atteignent de bons taux de reconnaissance ; il s'agit d'une part d'un descripteur LRF (*Local Receptive Features*) avec un classifieur SVM (*Séparateur à Vaste Marge*) et d'autre part d'un descripteur avec ondelettes de Haar avec un classifieur SVM. Nous avons appris et testé la méthode de combinaison dans les mêmes conditions. Les résultats sont présentés en figure 2.8. Là encore, la méthode d'association par la moyenne obtient des scores de reconnaissance supérieurs à ceux présentés dans (72).

2.3.4 Publications associées

1 **Machine Learning, chapter Classifiers Association for High Dimensional Problem.**
L. Leyrit, T. Chateau, and J.T. Lapresté.
IN-TECH, To appear

2 **Visual pedestrian recognition in weak classifier space using nonlinear parametric models.**
L. Leyrit, T. Chateau, et J. Lapresté.
IEEE International Conference on Image Processing (ICIP),San Diego, USA, Octobre 2008

3 **Association of adaboost and kernel based machine learning methods for visual pedestrian recognition**
L. Leyrit, T. Chateau, C. Tournayre, et J.T. Lapresté.
IEEE Intelligent Vehicles Symposium (IV), Eindhoven, Pays Bas, Juin 2008

4 **Descripteurs pour la reconnaissance de piétons**
L. Leyrit, T. Chateau, and J. Lapresté
ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur, Tregastel, Juin 2009.

5 **Association de classifiers pour la reconnaissance de piétons dans les images**
L. Leyrit, C. Tournayre, and T. Chateau
13ème colloque national Compression et Representation des SIgnaux Audiovisuels (CORESA ?2009), Toulouse, Mars 2009.

2.4 Application à l'estimation de postures

L'estimation du mouvement humain à partir d'images est un problème complexe qui a fait l'objet de très nombreux travaux ces dernières années. Les applications associées sont multiples et vont de la capture de

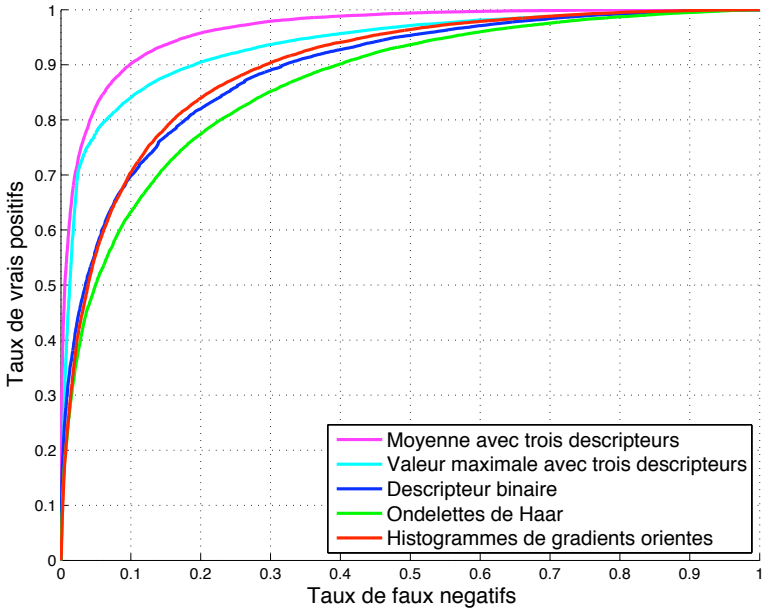


FIGURE 2.7 – Association de trois classifieurs par rapport à un apprentissage à partir d'un seul descripteur. Illustration en couleur

L'association des trois descripteurs par la moyenne augmente très nettement les scores de reconnaissance de piétons.

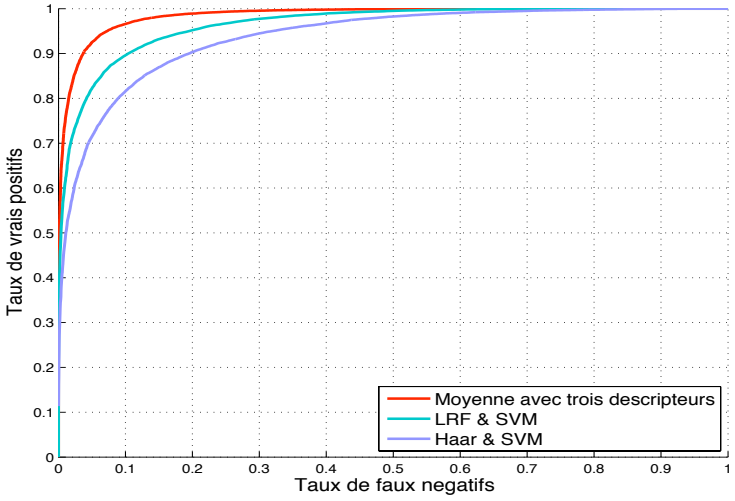


FIGURE 2.8 – Comparaison de la méthode d'association par la moyenne aux méthodes présentées dans (72). Illustration en couleur

C'est la méthode de combinaison par la moyenne qui obtient les meilleurs taux de reconnaissance.

mouvement pour l'animation d'avatars à la détection de mouvements anormaux dans un contexte de vidéo-surveillance. Le dispositif d'acquisition rend ce problème plus ou moins bien posé. Dans le cas où l'on dispose de séquences vidéos du mouvement d'une personne, acquises à partir de plusieurs caméras calibrées, le problème est plutôt bien posé et les techniques proposées cherchent à maximiser la précision de la pose estimée. Dans le cas où l'on observe un mouvement à partir d'une seule caméra, le problème devient moins bien posé et fait apparaître des configurations singulières où une même observation peut générer deux poses différentes. Il sera alors nécessaire d'injecter une hypothèse sur la continuité temporelle du mouvement pour lever l'ambiguïté. Le problème traité ici concerne l'estimation d'une pose statique acquise à partir d'un ensemble de caméras calibrées. Une seule image par caméra est disponible. Les travaux présentés dans cette partie sont issus de la thèse de Laetitia Gond, en co-tutelle avec l'équipe LIST du CEA Saclay.

2.4.1 Positionnement bibliographique

Plusieurs états de l'art sur ce sujet ont été proposés dans la littérature (35; 67; 83; 117). Il existe principalement deux grandes familles d'approches pour estimer la posture à partir d'images : les méthodes basées sur un modèle et les méthodes basées sur des exemples. Les méthodes basées modèle reposent sur l'utilisation d'un modèle explicite du corps humain, défini *a priori*, pour représenter la personne observée dans les images. La pose du corps est estimée par une approche de type "analyse-synthèse" : des prédictions sont effectuées sur la configuration du modèle, et sont ensuite mises à jour grâce aux informations contenues dans l'image. Dans ce type de méthodes, le choix du modèle de corps humain est un élément clef. Certaines approches utilisent un modèle 2D (21; 48; 70), tandis que d'autres sont plutôt basées sur un modèle 3D (9; 12; 28; 36; 50; 98; 116).
Contrairement aux approches basées modèle, les méthodes basées exemples (ou basées apprentissage) fournissent directement une estimation de la pose 3D à partir des données bas niveau de l'image, sans passer par des prédictions sur un modèle. Elles reposent sur l'utilisation d'une base d'exemples, créée à l'avance, qui contient un ensemble de paires image-pose. Les données d'entraînement sont généralement obtenues à partir de logiciels d'animation d'avatars et de rendu 3D. Aucune modélisation explicite du corps humain (sur l'apparence des membres ou les contraintes de la chaîne cinématique) n'est utilisée, toutes ces données sont implicitement modélisées dans la construction de la base d'exemples. On distingue deux grandes classes de méthodes : les méthodes non paramétriques basées sur une comparaison aux exemples de la base, dans lesquelles la base d'exemples est explicitement stockée en mémoire et sert ensuite de référence pour comparer les nouveaux exemples (68; 78; 83; 84; 94; 104), et les méthodes basées sur un apprentissage, dans lesquelles un entraînement, effectué hors ligne, permet de générer un modèle paramétrique qui généralise les propriétés de la base d'exemples (4; 30; 90; 91; 97; 103).
Il existe également des méthodes qui sont basées sur la détection et l'assemblage des différentes parties du corps (31; 32; 69; 89)
Lorsque l'estimation est réalisée à partir des images d'une séquence vidéo, il est possible d'utiliser l'information de cohérence temporelle entre les images consécutives de la séquence. Il existe plusieurs manières de la prendre en compte. Dans certain cas (28; 66; 116; 120), l'estimation est entièrement basée sur l'exploitation d'une continuité du mouvement du corps dans le temps, alors que dans d'autres (4; 96; 97), la cohérence temporelle n'est pas indispensable mais vient simplement compléter et améliorer une estimation effectuée à partir d'une seule image.
L'étude que nous menons s'inscrit dans les méthodes basées modèle, et plus précisément les méthodes basées sur un apprentissage. Nous proposons d'étudier les performances des machines d'apprentissage, dans le cadre de l'estimation de pose. Le but de l'algorithme d'apprentissage est de modéliser le lien entre les observations effectuées (codées sous la forme d'un descripteur) et la pose (liée à un modèle géométrique d'avatar).
Toutes les méthodes d'estimation de la pose ont en commun le fait de devoir tirer profit au mieux des informations bas niveau présentes dans l'image pour déduire une estimation haut niveau sur la configuration du corps humain qui a généré ces observations. Elles peuvent pour cela s'appuyer sur différents types de primitives extraites de l'image.
Parmi les primitives les plus couramment utilisées, on trouve :

- ▷ **la silhouette et ses contours externes de la silhouette :** la silhouette et ses contours externes peuvent

généralement être extraits de manière robuste à partir du moment où la caméra est statique et le fond stable. Comme souligné dans (4), la silhouette contient déjà une grande quantité d'informations intéressantes pour estimer la pose, tout en étant invariante à la plupart des caractéristiques inutiles comme la couleur des vêtements ou leur texture. Son principal défaut est qu'elle rend invisible certains degrés de liberté du corps, et que des poses assez différentes peuvent avoir des silhouettes similaires. Si par exemple les bras sont le long du corps, il est très difficile d'analyser leur mouvement simplement à partir des contours de la silhouette. Elle introduit aussi des ambiguïtés de symétrie. Par exemple, si une personne se déplace en marchant parallèlement au plan de la caméra, la silhouette ne donne aucun moyen d'identifier la jambe gauche ou la jambe droite. Avec une seule image, elle ne permet pas non plus de savoir si une personne est de face ou de dos. Toutes ambiguïtés sont évidemment réduites si plusieurs silhouettes extraites de différents points de vue sont utilisées. Les performances de l'estimation peuvent aussi être limitées par la présence sur la silhouette de bruits ou d'artéfacts (trous, ombres...).
Il est également possible de segmenter la silhouette d'une personne en mouvement grâce à des méthodes basées sur les contours actifs (28).
Dans le cas où les caméras sont calibrées, une reconstruction 3D peut être construite à partir des silhouettes extraites des différentes images : l'enveloppe visuelle peut donc être vue comme une utilisation particulière des silhouettes. Plusieurs approches s'appuyant sur une reconstruction 3D de l'enveloppe ont été proposées (17; 66; 102).

- ▷ **les contours de l'objet :** les contours de l'objet peuvent être extraits de manière robuste à un faible coût. Les contours internes permettent d'accéder à des informations sur des parties du corps situées à l'intérieur de la silhouette, par exemple en cas d'auto-occultation. Pour ne considérer que les contours utiles de l'objet, l'analyse ne porte souvent que sur les contours situés à l'intérieur de la silhouette (94) ou dans la boite englobante de la silhouette (83). Dans les méthodes basées sur des exemples ou un apprentissage, ces contours sont souvent décrits par des histogrammes d'orientation de gradient (3; 83; 94), ou par le Shape Context (68). L'inconvénient principal est qu'il est difficile de différencier les contours utiles (ceux qui délimitent une partie du corps) des autres contours (par exemple un pli ou un motif sur les vêtements), qui constituent alors un bruit dans le descripteur.

- ▷ **la couleur ou la texture :** leur utilisation se base sur le fait que la couleur ou la texture des membres du corps reste inchangée le long d'une séquence, même lorsque leur pose varie. Elle nécessite en général de connaître *a priori* la couleur ou la texture de l'objet d'intérêt, par exemple en réalisant un apprentissage hors-ligne. Dans (60), une phase préliminaire d'apprentissage permet de modéliser la texture des vêtements (volontairement très texturés) pour recaler un modèle de la jambe sur les images. Dans (86), l'apparence des membres du corps est apprise à partir de la vidéo, puis utilisée pour améliorer la détection dans les images suivantes. Dans (69), la reconnaissance des parties du corps est guidée en faisant l'hypothèse que des parties symétriques (par exemple le bras gauche et le bras droit) doivent avoir la même couleur dans l'image. Ce type de primitives est toutefois assez sensible, et les performances de la méthode risquent d'être dégradées par des variations d'éclairage, la déformation des vêtements, ou le manque de texture.

- ▷ **le mouvement :** le flot optique mesuré dans l'image peut fournir une indication intéressante sur le mouvement du modèle qui l'a généré. Dans (48), le mouvement de patchs modélisant en 2D les différentes parties du corps est estimé en appliquant des contraintes sur le flot optique des pixels de l'image situés dans la zone délimitée par un patch. Dans (19), Bregler et Malik introduisent des outils (*Twist Motion Model*) permettant de simplifier la relation entre le mouvement d'un modèle 3D et le mouvement mesuré dans l'image.

- ▷ **détection de la peau ou de parties du corps :** Certaines méthodes sont entièrement basées sur une détection préalable des différentes parties du corps dans l'image. Dans d'autres approches, la détection de parties du corps peut être utilisée en complément d'autres observations pour guider l'estimation de la pose. Certains travaux tirent notamment profit du fait que la peau possède une couleur bien spécifique et peut se distinguer facilement des autres éléments de l'image. Dans (18; 57), des *blobs* correspondant au

visage ou aux mains sont ainsi extraits de l'image grâce à une analyse de la couleur. Dans (57), le visage également est détecté par AdaBoost.

▷ **une combinaison de plusieurs primitives** : pour tirer un maximum de bénéfice des informations contenues dans l'image, il est bien sûr possible de combiner plusieurs de ces primitives. Dans (57), les auteurs combinent par exemple une détection du visage et de la peau et une analyse des contours. Dans (98), la fonction de vraisemblance est construite à partir d'informations sur l'intensité, les contours et le flot optique.

Le choix du descripteur est un point très important dans toute méthode d'estimation de pose. Dans le cas où une silhouette binaire de la personne (qui peut être soit 2D, soit 3D) est disponible, il est possible d'utiliser une description statistique de la silhouette par des moments 2D ou 3D. Dans le cas d'une silhouette 2D, Les moments de Hu sont assez populaire et ont été utilisés dans (90; 91) pour de l'estimation de pose. De nombreux autres moments peuvent être définis comme par exemple les moments de Zernike (105), les moments basés sur des ondelettes (95), la transformée en cosinus discrète (109). De la même façon qu'en 2D, des moments invariants par translation, rotation et changement d'échelle peuvent être dérivés (voir (62)). De nombreuses variantes de ces moments ont été définis ; on trouve par exemple une extension en 3D des moments de Zernike (20). Des moments 3D basées sur des ondelettes sont aussi proposés dans (118). Un autre type de descripteur utilisé pour le codage des silhouette est le *Shape Context*. Introduit dans (14), puis repris dans (4), c'est une description locale non-paramétrique de la forme d'un objet, s'appuyant sur des points échantillonnées le long de ses contours (internes et externes). En 3D, une extension du *Shape Context* a été introduit dans (56), et utilisé par les auteurs de (102). D'autres descriptions basées sur les contours externes de la silhouette sont également possibles, notamment celles qui traitent le contour comme une courbe paramétrée continue, comme par exemple les représentations basées sur les coefficients de Fourier ((84)) ou d'ondelettes. Il est également possible de représenter la silhouette par une mixture de gaussiennes 2D (42). Enfin, les auteurs de (23) présentent également une façon de décrire la forme 3D reconstruite à partir de silhouettes extraites avec plusieurs caméras. Pour une silhouette 2D, les auteurs définissent un cercle de référence, sur lequel sont régulièrement répartis des points de contrôle. Pour chaque point de contrôle P_i, un histogramme décrit la répartition des points de la silhouette en coordonnées polaires dans un repère ayant pour origine le point P_i. L'espace est partitionné suivant un ensemble de subdivisions radiales et angulaires. L'histogramme dénombre ensuite le nombre de points de la silhouette contenus dans ces différentes subdivisions. L'invariance par rotation est obtenue en sommant l'ensemble des histogrammes construits pour tous les points de contrôle. Pour généraliser cette description au cas d'une silhouette 3D, le cercle de référence est remplacé soit par un cylindre dont l'axe principal passe par le centre de gravité de la reconstruction , soit par une sphère. Une des contributions du travail présenté ici concerne l'étude d'un nouveau descripteur 3D, basé sur la silhouette d'une personne. La deuxième contribution se positionne par rapport au sujet principal de de chapitre, à savoir l'utilisation de machines d'apprentissage pour l'estimation d'état ; ce dernier étant défini par la pose à rechercher. Dans le domaine de l'estimation de pose, les travaux les plus proches sont ceux d' Agarwal et Triggs (4), et de Sun (102).

2.4.2 La méthode

Nous proposons un système d'estimation de la pose d'un modèle articulé du corps à partir des images acquises par un système de plusieurs caméras (4 ou 5) statiques et calibrées. Comme beaucoup d'autres travaux sur cette application, nous avons choisi de baser les observations sur les silhouettes extraites dans les différentes images par un algorithme de soustraction de fond. Cette approche est envisageable dans le cas où les caméras sont fixes et où l'environnement dans lequel évolue la personne est relativement stable au cours du temps. La silhouette peut alors être calculée de manière robuste, et représente une primitive intéressante car elle est invariante aux changements d'éclairage et d'habillement.
Les silhouettes 2D sont ensuite combinées par une reconstruction en 3D de l'enveloppe visuelle : la silhouette 3D fusionne toutes les informations du système (données images et calibrage du système) en un seul élément, et rend l'estimation plus indépendante de la configuration des caméras. Il faut préciser que notre système

d'acquisition n'est pas synchronisé électroniquement : il s'agit d'un réseau IP représentatif des réseaux de vidéo surveillance. Les images venant des différents capteurs sont associées en fonction de leur moment d'arrivée au PC de calcul. Il en résulte qu'un décalage de quelques ms peut exister entre les acquisitions, ce qui peut induire des imprécisions dans la silhouette 3D reconstruite.
Pour évaluer la pose du corps à partir de l'enveloppe visuelle reconstruite, nous proposons d'apprendre les paramètres d'une fonction de régression : cette méthode évite d'une part les calculs liés à des prédictions sur la configuration d'un modèle du corps et à l'optimisation d'une fonction de vraisemblance complexe, et permet d'autre part de déterminer la pose à partir d'une seule image, et de se dispenser des problèmes d'initialisation ou de perte de suivi.
Une vue d'ensemble de la méthode proposée est donnée sur la figure 2.9. Pour chaque caméra, on suppose qu'un apprentissage préalable a permis de modéliser une image du fond. La silhouette de la personne est extraite dans chaque image. La connaissance des paramètres de calibrage intrinsèques et extrinsèques nous permet ensuite de synthétiser les données image en une reconstruction 3D de l'enveloppe visuelle. La suite de notre méthode est basée sur un algorithme d'apprentissage. On suppose qu'on dispose d'une base d'exemples contenant d'un côté des données image de silhouettes et de l'autre la vérité terrain sur la posture de la personne. Cette base nous permet de modéliser par apprentissage l'application permettant de passer d'une silhouette 3D (observations) à la posture (état). Les caractéristiques de la forme 3D sont encodées par un descripteur 3D, qui synthétise en un vecteur compact l'information de la reconstruction. A l'estimation, la reconstruction 3D est calculée à partir des silhouettes extraites et son descripteur est donné directement en entrée du modèle qui a été appris, pour fournir une estimation de la pose.
La figure 2.10 illustre le fonctionnement de l'apprentissage de la machine de régression. Un logiciel de synthèse d'avatar (POSER) est utilisé comme modèle direct pour générer des couples d'apprentissage $\{$ Etat, Observation $\}$. L'ensemble de ces derniers est alors pris en compte dans l'apprentissage de la machine de régression, dont le modèle est identique à celui présenté dans la section 2.2 :
$\mathbf{X} = \sum_{k=1}^{K} \mathbf{w}_k \phi_k(\mathbf{Z}) + \boldsymbol{\epsilon} \doteq \mathtt{W}\boldsymbol{\phi}(\mathbf{Z}) + \boldsymbol{\epsilon}$

2.4.2.1 L'observation (descripteur)

La figure 2.11 illustre le dispositif d'acquisition, constitué de quatre caméras calibrées. Les images qui en sont issues sont alors binarisées par une technique d'extraction de fond. Ces dernières sont alors utilisées pour construire une représentation sous la forme de voxels de l'enveloppe visuelle 3D reconstruite à partir des quatre silhouettes. La figure 2.12 présente deux exemples de reconstruction de silhouettes. Elle permet d'illustrer la présence d'artefacts dans la reconstruction, diminuant lorsque le nombre de caméras utilisées pour la reconstruction augmente.
Nous avons proposé un nouveau descripteur 3D, basé sur la répartition spatiale des voxels à l'intérieur d'un cylindre de référence englobant la forme 3D reconstruite. Ce descripteur se rapproche de celui proposé par Cohen et Li dans (23), mais dans leur cas, le fait de sommer les composantes pour les différents points de contrôle efface une partie de l'information utile sur la pose (dans leur travaux, les auteurs utilisent cette description simplement pour de la classification). La formulation du descripteur proposé est assez intuitive et ses composantes rendent directement compte de la répartition de la matière dans l'enveloppe reconstruite.
La figure 2.13 illustre le principe du descripteur 3D proposé. Etant donnée une silhouette 3D reconstruite en voxels, on définit un cylindre de référence dont l'axe principal est l'axe vertical passant par le centre de gravité de la reconstruction et dont le rayon est proportionnel à la hauteur de la reconstruction (on suppose que les points les plus hauts de la reconstruction correspondent à des points de la tête, et donc que la hauteur de la reconstruction est égale à la taille de la personne). Pour chaque section horizontale des voxels, le disque défini par le cylindre est divisé en une grille composée de graduations suivant le rayon et l'angle. Un histogramme 2D est calculé en comptant le nombre de voxels de la forme 3D contenus dans chaque secteur (la position du centre d'un voxel détermine son appartenance à un secteur).
La hauteur de la forme 3D étant divisée en tranches verticales, pour chaque tranche, un nouvel histogramme 2D est calculé en moyennant l'ensemble des histogrammes des couches de voxels contenues dans la tranche. Le descripteur 3D est constitué de la concaténation de ces descripteurs moyens. De plus, afin d'éviter les

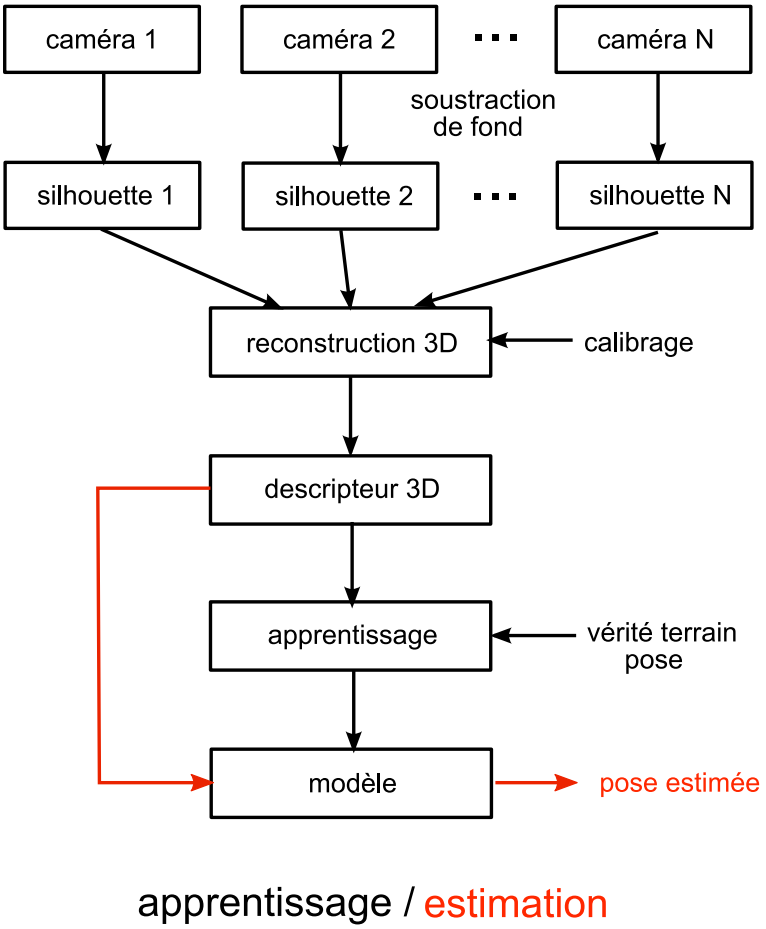


FIGURE 2.9 – Vue d'ensemble de l'application d'estimation de pose

phénomènes de frontière lors de la discrétisation, un noyau gaussien est utilisé lors du vote, permettant à un voxel de voter également pour des secteurs voisins du sien.

2.4.2.2 Estimation de la pose par régression

La méthode proposée est basée sur des techniques d'apprentissage statistique : un apprentissage, réalisé hors ligne, permet de modéliser par une unique application la relation entre le descripteur 3D de l'enveloppe visuelle et le vecteur décrivant la pose. A l'issue de l'apprentissage, cette application génère directement une estimation de la pose à partir des données image encodées par le descripteur, sans passer par des prédictions sur la configuration d'un modèle du corps. Cette application résume en un modèle compact les propriétés des exemples de la base d'apprentissage pour effectuer des prédictions sur des données non-apprises. Cette technique permet donc d'intégrer des connaissances *a priori* sur la nature des mouvements réalisés grâce aux exemples de la base d'apprentissage : au lieu d'imposer des contraintes par l'intermédiaire d'un modèle détaillé du corps, celles-ci sont implicitement modélisées dans l'apprentissage. Ce type d'approche présente enfin l'avantage de réduire les temps de calcul associés à l'estimation de la pose, car la plus grosse partie des calculs nécessaires est effectuée hors-ligne pendant la phase d'apprentissage.
Le principal problème des méthodes d'estimation basées sur un apprentissage est lié à la taille de l'espace d'apprentissage et la quantité d'exemples nécessaires dans la base pour bien représenter sa structure. Même en utilisant une machine de régression possédant de bonnes capacités de généralisation, il est nécessaire de bien couvrir dans la base d'apprentissage toutes les possibilités de poses que l'on souhaite reconnaître. Dans le cas d'une personne qui marche, l'estimation de la configuration des membres du corps pourrait être facilitée si l'orientation globale du corps dans l'espace (l'azimut) était connu : il serait alors possible de recaler le descripteur sur cette orientation et d'estimer les autres angles avec un descripteur aligné avec l'orientation du

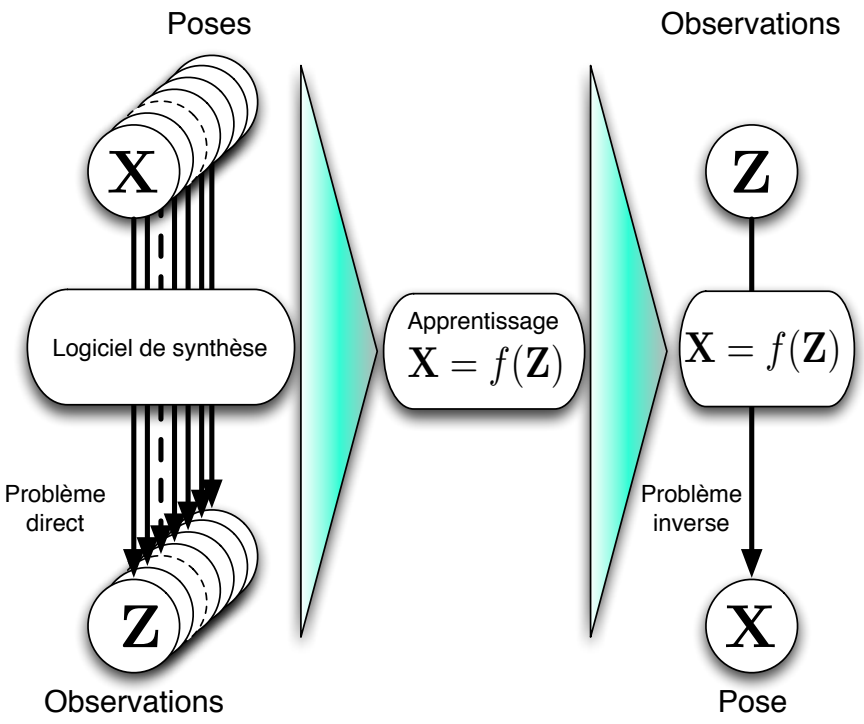


FIGURE 2.10 – Synoptique illustrant l'apprentissage de la machine de régression. L'ensemble des couples d'apprentissage { Etat, Observation } est généré selon un modèle direct issu d'un logiciel de synthèse d'images. Ces derniers sont alors utilisés pour apprendre une fonction de régression que l'on utilise comme modèle inverse pour estimer une pose à partir d'observations.

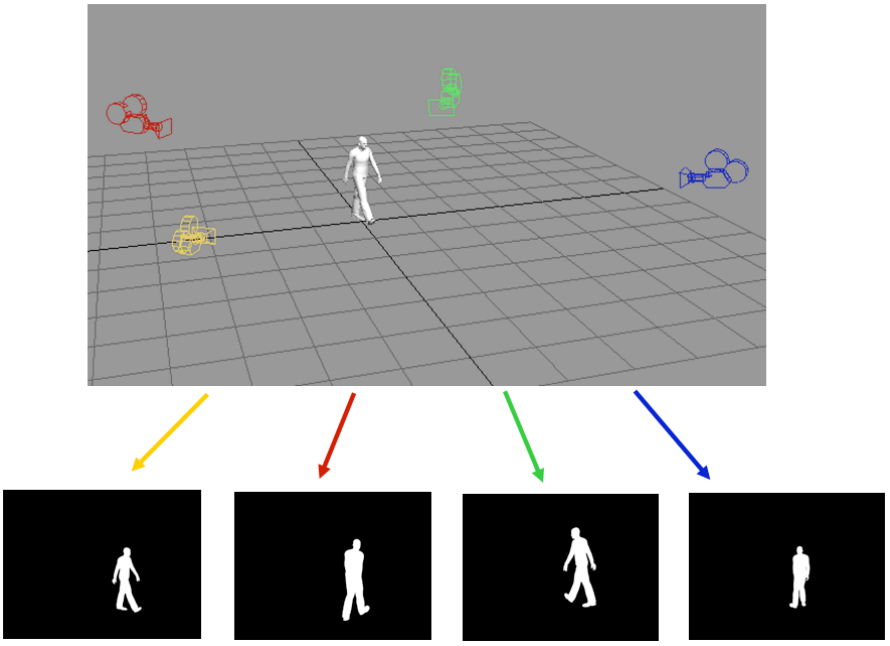


FIGURE 2.11 – Illustration du dispositif d'acquisition, simulé sous un logiciel de synthèse pour construire la base d'apprentissage. Dans cette exemple, les quatre caméras utilisées sont supposées calibrées.

corps. L'orientation du descripteur permettrait de réduire la complexité de l'espace des poses. En effet, si le descripteur n'est pas orienté, pour estimer correctement la pose quelque soit l'orientation, il faudrait en théorie

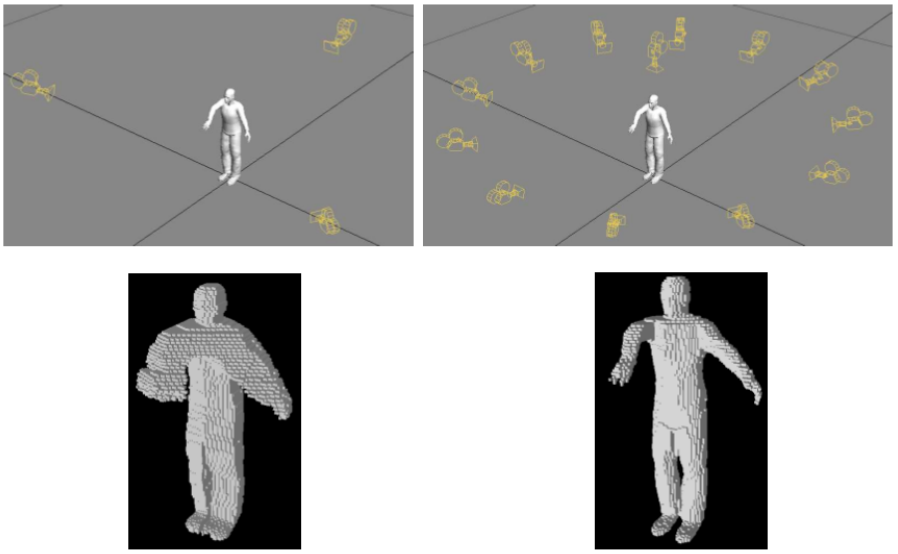


FIGURE 2.12 – Exemples de silhouettes 3D reconstruites à partir de données de synthèse. L'exemple de gauche utilise trois caméras. De nombreux artefacts apparaissent au niveau de la reconstruction. L'exemple de droite utilise treize caméras, dont une au plafond. La plupart des artefacts a disparu.

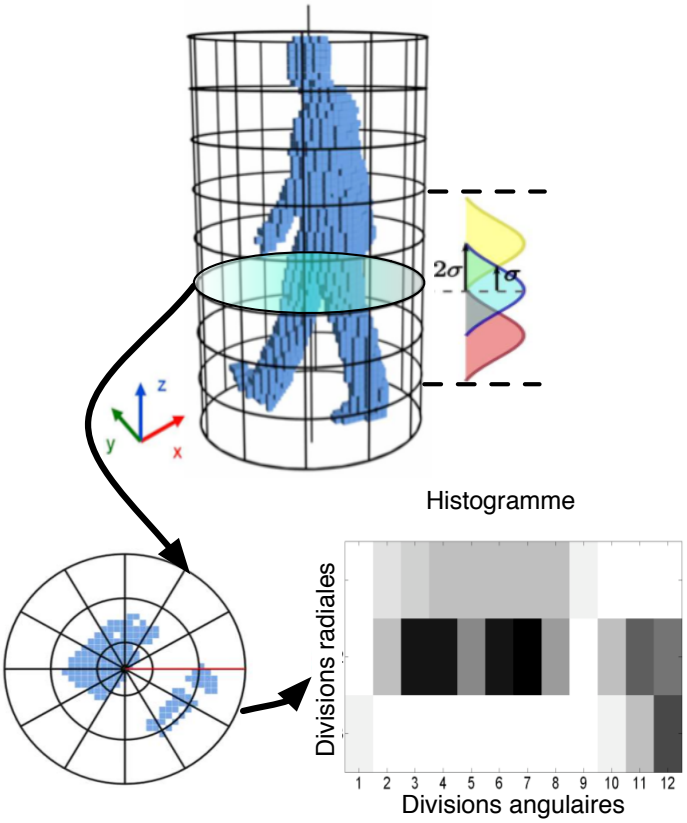


FIGURE 2.13 – Synoptique illustrant le principe du descripteur 3D. La forme 3D est divisée en tranches verticales. Pour chaque tranche, une grille composée de graduations suivant le rayon et l'angle est définie. Un histogramme 2D est alors calculé en comptant le nombre de voxels de la forme 3D contenus dans chaque secteur de la grille. Le descripteur 3D final est la concaténation des histogrammes 2D.

	corps entier	orientation	épaule gauche	jambe droite
(4)	6.0	17	7.5	4.2
(102)	5.2	8.8	6.3	3.2
notre approche	3.0	4.6	3.7	2.8
avec recalage	2.5	-	3.4	2.1

TABLE 2.2 – Comparaison des RMS des erreurs (angles en degrés) entre différentes approches sur une séquence synthétique de 418 exemples de marche en spirale. **1re ligne :** résultats de l'approche monoculaire présentée par Agarwal et Triggs dans (4). **2e ligne :** résultats présentés dans (102) avec 6 caméras circulaires. **3e ligne :** résultats obtenus avec notre méthode avec 6 caméras, sans recaler le descripteur sur l'orientation. **4e ligne :** même chose avec l'estimation en deux temps (recalage du descripteur sur l'orientation estimée pour les angles internes).

que la base d'apprentissage contienne des exemples de chaque pose avec toutes les orientations possibles. Nous avons donc choisi de décomposer l'estimation de la pose en deux étapes. Dans un premier temps, seule l'orientation du corps est estimée, puis les autres degrés de liberté du corps sont évalués grâce à un second descripteur, recalé sur l'orientation globale du corps. Dans nos travaux, l'orientation du torse sert de référence pour l'orientation globale du corps (elle est estimée à partir de l'orientation par rapport à la verticale du segment reliant les deux clavicules.
Deux machines de régression ont donc été apprises : la première estime uniquement l'orientation du corps, à partir d'un descripteur calé en rotation sur une référence absolue. La deuxième estime les autres paramètres de pose, à partir d'un descripteur recalé en rotation par rapport à l'orientation estimée par la première machine de régression.

2.4.3 Résultats

La méthode proposée a été comparée avec les méthodes présentées dans (4) et (102). Ces deux références sont intéressantes car elles adoptent des approches similaires à la nôtre : elles sont toutes les deux basées sur l'utilisation des silhouettes (en monoculaire dans (4) et via une reconstruction 3D en voxels pour (102)) et une régression. La comparaison avec (4) permet de mettre en avant les bénéfices d'une méthode d'estimation multi-vues par rapport au cas monoculaire, et avec (102) de mesurer les performances de notre descripteur par rapport aux histogrammes de Shape Context 3D. Les résultats de ces deux publications sont résumées dans le tableau 1 de (102). Pour mieux nous comparer à (102), un système circulaire de 6 caméras, similaire à celui qui est utilisé pour l'apprentissage et les tests de l'article, a été employé.
D'autre part, nous avons utilisé des données produites à partir de la capture des mouvements de 3 personnes qui marchent en spirale, et sont disponibles en ligne [4]. Pour nos tests, l'avatar par défaut du logiciel de simulation (POSER 6) a été animé avec les données provenant d'un système de capture de mouvements. Comme pour les deux autres méthodes, l'une des séquences, comprenant 418 exemples, est utilisée comme séquence de test. La base d'apprentissage contient 2537 exemples. Dans notre cas, des SVM sont utilisés pour la régression.
Le tableau 2.2 présente les résultats obtenus avec les différentes méthodes. D'aune part, les méthodes utilisant le système de 6 caméras atteignent une plus grande précision, ce qui était prévisible. D'autre part, la méthode proposée semble aussi être plus précise que celle présentée dans (102). Cette comparaison doit cependant être considérée avec prudence, car les résultats peuvent dépendre de certains facteurs qui ne sont pas toujours précisés dans les différents travaux, comme le niveau de détail de la paramétrisation du corps, le placement des caméras et le déplacement du sujet dans leur champ de vision, l'avatar qui est animé (nous avons utilisé l'avatar par défaut de Poser 6, une version antérieure du logiciel semble être utilisée dans (4), et les auteurs de (102) animent un modèle du corps composé de sphères et d'ellipsoïdes)
La figure 2.14 présente des courbes représentant les valeurs des angles estimés et de la vérité terrain le long de la séquence test, pour l'un des angles des jambes et pour l'orientation du torse. Nos résultats sont comparés

[4] www.ict.usc.edu/graphics/animWeb/humanoid

aux courbes reportées dans (4), dans le cas de l'estimation monoculaire. On peut observer que dans notre cas, l'utilisation de plusieurs caméras a permis de lever certaines ambiguïtés qui apparaissent en monoculaire dans l'orientation du corps et des jambes. Une silhouette unique ne permet pas toujours de distinguer dans une posture de marche quelle jambe est placée devant l'autre, et le régresseur peut dans ce cas choisir la mauvaise solution ou retourner une solution moyenne représentant une compromis entre deux les possibilités. Ce phénomène se manifeste par exemple dans les courbes de la figure 2.14 par une inversion des pics dans l'estimation de l'angle de la jambe (aux alentours de la vue 80).

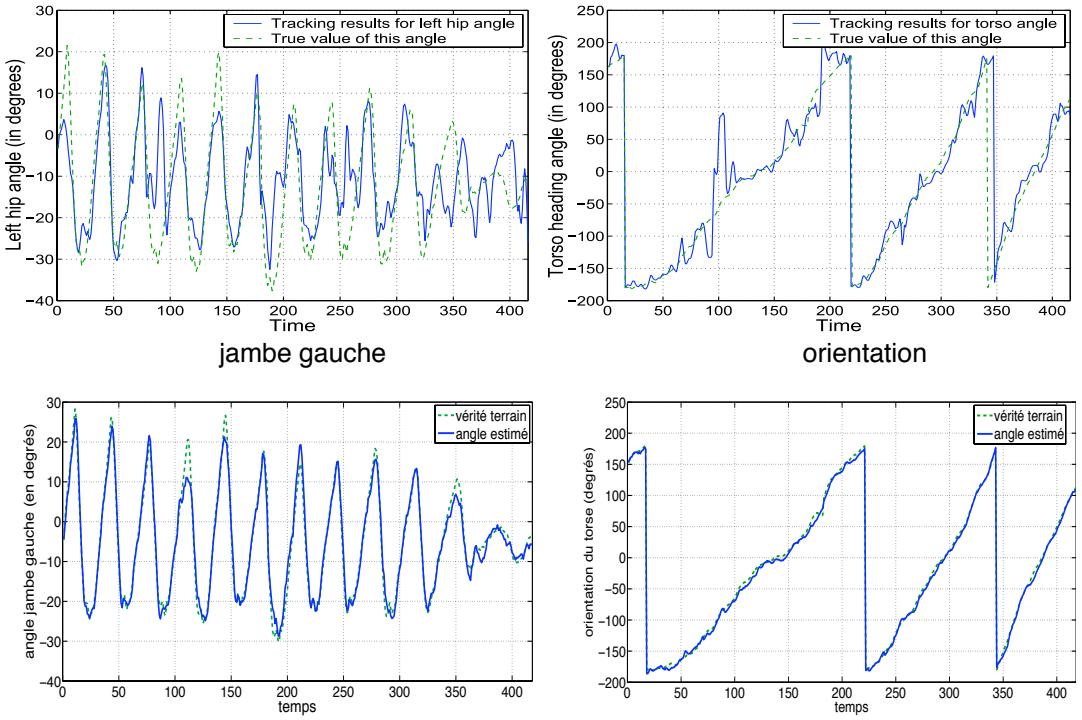


FIGURE 2.14 – Angles estimés pour l'orientation du torse et la jambe gauche le long d'une séquence de marche synthétique, comparée à la vérité terrain. **Haut :** résultats présentés dans (4). **Bas :** résultats obtenus avec notre méthode.

2.4.4 Conclusion

Cette étude a permis de montrer la faisabilité d'une méthode basée apprentissage pour l'estimation de pose, à partir d'images statiques. D'autre part, la base d'apprentissage a été totalement générée à partir d'un logiciel de synthèse d'images. La démarche utilisée a donc été de partir du modèle direct de génération d'images d'avatars à partir de poses, pour construire l'ensemble d'apprentissage, puis d'apprendre des fonctions de régressions capables d'estimer le modèle inverse. L'utilisation de logiciels de synthèse d'images comme modèles directs pour générer des ensemble d'apprentissage pourrait être étendu à d'autres applications. On pourrait par exemple générer des bases d'apprentissage spécialisées pour la détection de visages ou de personnes, en reproduisant, en synthèse, des conditions d'acquisition spécifiques (par exemple des conditions d'illumination particulières ou la présence de plusieurs caméras calibrées). Ces techniques présentent l'avantage de pouvoir prendre en compte de manière explicite le contexte dans l'apprentissage.

Ces travaux ont également permis de proposer un nouveau descripteur 3D particulièrement adapté au codage de la pose d'une personne à partir de la représentation de sa silhouette par un ensemble de voxels.

2.4.5 Publications associées

1	**A 3D Shape Descriptor for Human Pose Recovery.** L. Gond, P. Sayd, T. Chateau and M. Dhome. *In Articulated Motion and Deformable Objects (AMDO)*,Portugal, 2008
2	**A regression-based approach to recover human pose from voxel data** L. Gond, P. Sayd, T. Chateau and M. Dhome. *Second IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS2009)*, Septembre 2009, Kyoto, Japon

2.5 Application au suivi d'objets planaires

Le suivi précis et temps réel de régions planaires dans des séquences d'images est utilisé dans des applications de réalité augmentée ou d'interfaces homme machine. Le principe de la méthode consiste à lier la variation d'apparence du plan (observations) au mouvement du motif (état). la plupart des méthodes existantes considèrent un lien linéaire, ce qui n'est pas le cas en réalité. Nous proposons d'apprendre ce lien par un modèle non linéaire, tel que ceux décrits au début de ce chapitre.

2.5.1 Positionnement bibliographique

Nous abordons le problème du suivi d'un plan texturé par vision. Il s'agit d'estimer en temps réel, le déplacement d'un motif planaire dans une séquence d'images. Certaines méthodes sont basées sur une approche analytique (43) (15) ; d'autres sur une approche basée apprentissage (49) (24). La plupart du temps, il s'agit de minimiser un critère d'erreur de luminance entre un modèle et l'image courante (8), mais nous avons aussi proposé d'utiliser des fonctions d'observation plus complexes comme les ondelettes de Haar par exemple (22). Nous proposons une extension de la méthode basée apprentissage proposée de manière quasi-simultanée par Cootes et al. dans (24) et par Jurie et Dhome dans (49). Cette extension utilise des fonctions de régression paramétriques non linéaires. Cela permet d'étendre le domaine de convergence du système de manière significative.

2.5.2 La méthode

Dans l'application précédente, un logiciel de synthèse a été utilisé pour générer l'ensemble d'apprentissage. Un principe similaire, illustré figure 2.15 est également utilisé dans cette application. La position du motif plan est définie de manière supervisée dans la première image de la séquence. Sur cette même image, des mouvements virtuels du motif sont alors appliqués, auxquels on associe la variation de l'apparence qu'ils génère. La base d'apprentissage ainsi formée est ensuite utilisée pour apprendre des machines de régression. La principe du suivi est présenté sur la figure 2.16. Il s'effectue par composition des déplacements relatifs au cours du temps : le motif de référence, à l'instant t est ramené dans un repère canonique par l'homographie $\mathtt{H}_{t-1}$. une fonction de régression estime l'état (déplacement du motif $\mathbf{X}_t$ entre $t-1$ et t) en fonction de l'observation (variation de l'apparence). Le déplacement du motif dans le plan image est alors calculé par composition en appliquant $\boldsymbol{\delta}_{\mathbf{p}_t} = \mathtt{H}_{t-1}\mathbf{X}_t$.

2.5.2.1 Apprentissage

La méthode proposée est une approximation à l'ordre un de la relation entre le déplacement du motif et la variation de son apparence (modèle inverse). Elle est représentée par une relation linéaire sous la forme d'une matrice d'interaction $\mathtt{A}$ telle que $\mathbf{X}_t = \mathtt{A}\mathbf{Z}_t$ (l'indice t représente la discrétisation temporelle). La matrice d'interaction est alors estimée par un critère au sens des moindres carrés en utilisant un ensemble d'apprentissage généré sur la première image.
Ce modèle peut être vu comme un cas particulier d'un modèle plus général correspondant à celui présenté dans la section 2.2 : $\mathbf{X} = \sum_{k=1}^{K} \mathbf{w}_k \phi_k(\mathbf{Z}) + \boldsymbol{\epsilon} \doteq \mathtt{W}\boldsymbol{\phi}(\mathbf{Z}) + \boldsymbol{\epsilon}$ dans lequel $\boldsymbol{\phi}(\mathbf{Z}) = \mathbf{Z}$ (les indices temporels ont volontairement été omis pour faciliter les notations). Il semble alors naturel d'utiliser d'autres types de

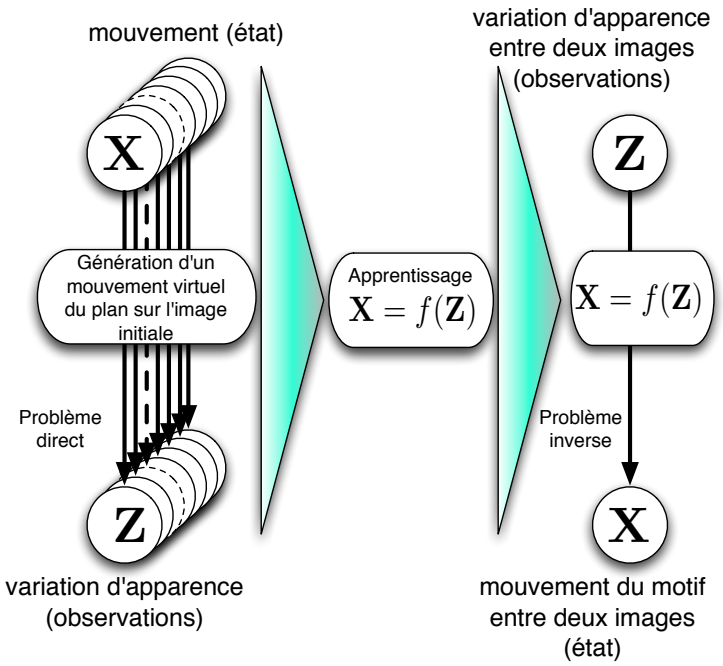


FIGURE 2.15 – Synoptique illustrant l'apprentissage de la machine de régression. L'ensemble des couples d'apprentissage { Etat, Observation } est généré selon un modèle direct issu de la génération de déplacements virtuels du plan à suivre dans la première image de la séquence. Ces derniers sont alors utilisés pour apprendre une fonction de régression que l'on utilise comme modèle inverse pour estimer un déplacement du plan à partir de la variation de son apparence entre deux images.

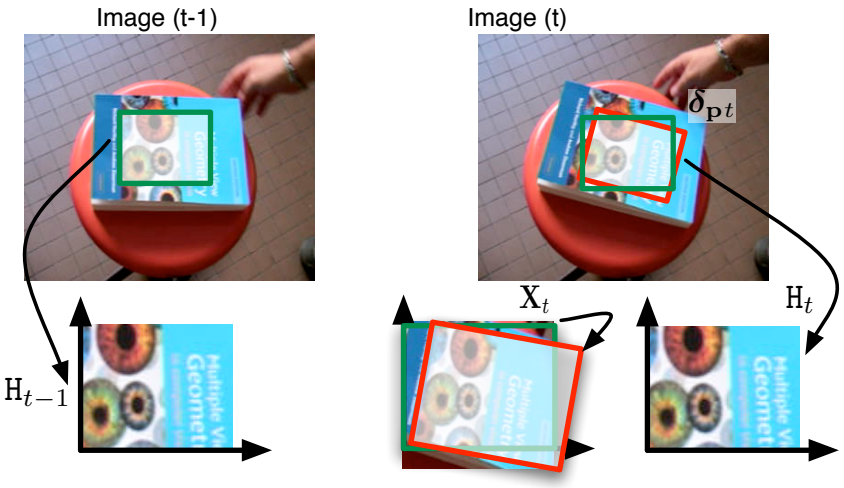


FIGURE 2.16 – Principe de l'algorithme de suivi planaire. Le motif de référence, à l'instant k est ramené dans un repère canonique par l'homographie $\mathtt{H}_{t-1}$. une fonction de régression estime l'état (déplacement du motif $\mathbf{X}_t$ entre $t-1$ et t) en fonction de l'observation (variation de l'apparence). Le déplacement du motif dans le plan image est alors calculé par composition en appliquant $\boldsymbol{\delta}_{\mathbf{p}t} = \mathtt{H}_{t-1}\mathbf{X}_t$.

fonctions de base afin de rendre le modèle non linéaire, donc de lui permettre de modéliser des liens plus complexes entre l'observation et l'état. Nous avons proposé un modèle utilisant des fonctions de base radiales sur lequel les poids associés ont été estimés, soit par un critère au sens des moindre carrés, soit par un algorithme épars de type Relevance Vector Machine.

2.5.2.2 Suivi

Une fois l'apprentissage effectué, la machine de régression est utilisée en ligne pour estimer le déplacement du motif à chaque image. Une stratégie de recalage de type "grossier vers fin" a été appliquée pour accroître à la fois le bassin de convergence et la précision du suivi. Ainsi, plusieurs machines de régression ont été apprises, pour des amplitudes de déplacement décroissantes. La machine apprise pour l'amplitude maximale pourra estimer de manière grossière des déplacements importants, mais avec peu de précision, tandis que celle apprise avec de faibles amplitudes sera très précise, mais pour de petits déplacements. Les machines apprises sont alors appliquées de manière séquentielle, en partant de la plus grossière vers la plus fine. D'autre part, chaque machine est appliquée plusieurs fois, ce qui a également pour effet d'accroître le bassin de convergence et la précision de la méthode.

2.5.3 Résultats

Nous comparons le modèle proposé, utilisant des fonctions de base non-linéaires, avec le modèle linéaire initialement proposé par Cootes et al. (24) et par Jurie et Dhome (49). Deux tests sont proposés :

- ▷ **La précision** : des mouvements de synthèse simples et connus (translations) sont générés sur une image de référence. On relève alors l'erreur d'estimation de chacune des deux méthodes, en fonction de l'amplitude des mouvements, que l'on représente sur une courbe.

- ▷ **Le taux de convergence** : des mouvements de synthèse connus sont générés sur une image de référence. On relève ensuite les estimations dont l'erreur est inférieur à un seuil pour lequel on considère que la méthode a convergé. On en déduit une courbe représentant le taux de convergence en fonction de l'amplitude du mouvement.

La fonction d'observation choisie est basée sur la luminance des pixels situés sur une grille qui échantillonne de manière régulière la zone à suivre. Il existe d'autres stratégies d'échantillonnage, basées, par exemple sur une sélection de points d'intérêt par baquets. Dans les tests qui suivent, la grille choisie est de taille 15×15 pixels, soit 225 points. Le vecteur de primitives ainsi obtenu est centré normé, pour obtenir une indépendance aux transformations affines de la luminance. Le premier comparatif concerne l'aptitude des méthodes à estimer un mouvement connu. Un déplacement horizontal a été généré à partir d'une image fixe. L'apprentissage a été effectué pour un seul niveau de bruit ($L = 1$ et $b = 0.05$ pour la courbe de gauche et $b = 0.1$ pour la courbe de droite, niveau de bruit ramené dans le repère canonique), avec $N = 300$ bruits générés et $M = N$ fonctions de base. De plus, une seule itération est appliquée pour le suivi ($I = 1$). La figure 2.17 montre l'erreur sur l'estimation de la translation en fonction de la translation réelle, dans le cas des traqueurs linéaires et non linéaires, pour quatre valeurs du paramètre d'apprentissage b (amplitude d'apprentissage associée au premier niveau de suivi). Pour de petites amplitudes d'apprentissage $b = 0.1$ and $b = 0.2$, l'estimation du déplacement est correcte pour des translations situées dans la zone d'apprentissage (inférieures à l'amplitude b). Pour de forts déplacements, l'erreur devient importante. Pour une amplitude d'apprentissage $b = 0.3$, la méthode linéaire est mise en défaut tandis que la méthode non linéaire fournit une estimation de la translation qui reste bonne. L'approximation d'ordre un effectuée par la méthode linéaire n'est plus valide. Pour l'approche non linéaire, il est possible d'apprendre une fonction de régression valide sur un domaine plus important (jusqu'à $b = 0.4$). D'autre part, lorsque le déplacement est important, la méthode à noyau retourne une estimation quasi-nulle. Ceci est dû au fait que lorsque le vecteur de primitive est éloigné de tous les vecteurs de la base, la fonction noyau retourne des valeurs très faibles. Ce n'est pas le cas de la méthode linéaire.
L'essai précédent met en évidence l'importance de notion de convergence. En fixant un seuil sur l'erreur d'estimation (erreur quadratique entre la position estimée des quatre coins du motif à suivre et la position réelle, un état de convergence peut être défini. Les quatre coins définissant le motif à suivre sont perturbés de manière aléatoire (loi uniforme). La convergence de chaque méthode est alors étudiée statistiquement pour un grand nombre de répétitions du test de perturbations. La figure 2.18 montre le pourcentage de convergence en fonction du déplacement quadratique généré. On constate que la méthode proposée possède un domaine de convergence plus élevé que la méthode linéaire classique. Pour les courbes de gauche, un bruit d'apprentissage

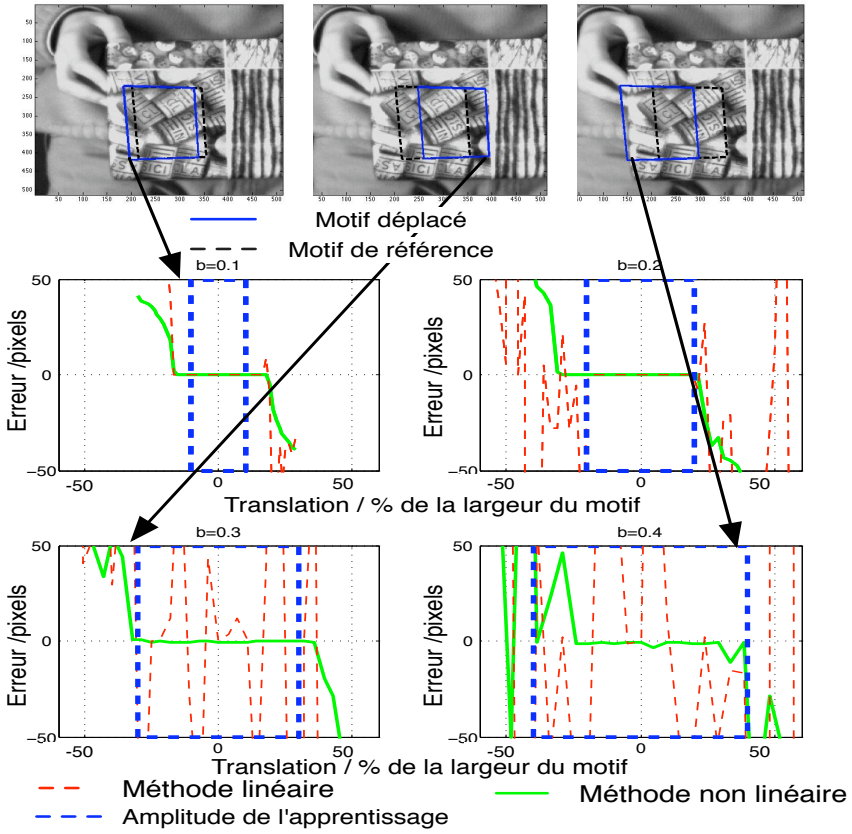


FIGURE 2.17 – Précision d'estimation : comparaison de la méthode linéaire et de la méthode non linéaire en fonction du déplacement horizontal, pour plusieurs valeurs du paramètre d'apprentissage b. Illustration en couleur

de 0.1 à été utilisé. Pour les courbes de droite, le bruit d'apprentissage a été réglé à 0.2. On constate que pour des valeurs élevées du bruit d'apprentissage, la convergence de la méthode linéaire diminue. En effet, plus les variations sont approximées sur un intervalle important, moins l'hypothèse de linéarité entre le mouvement et les observations est vraie. Pour ces essais, le nombre de niveaux d'apprentissage et le nombre de boucles par niveau sont fixés à trois.

2.5.4 Conclusion

Nous avons proposé une extension de la méthode de suivi d'objets planaires texturés introduite simultanément par Cootes et al. et par Jurie et Dhome, qui utilise une fonction de régression non linéaire (coefficients linéaires mais fonctions de base non linéaires). La méthode est basée sur l'apprentissage du lien entre la variation de l'apparence du motif à suivre et son déplacement. La méthode proposée permet d'accroître le bassin de convergence, donc d'estimer des déplacements plus importants, en modélisant des liens non linéaires, entre état et observations. D'autre part, elle est naturellement non divergente dans le sens où pour une observation très éloignée de celles apprises, le déplacement généré est nul.

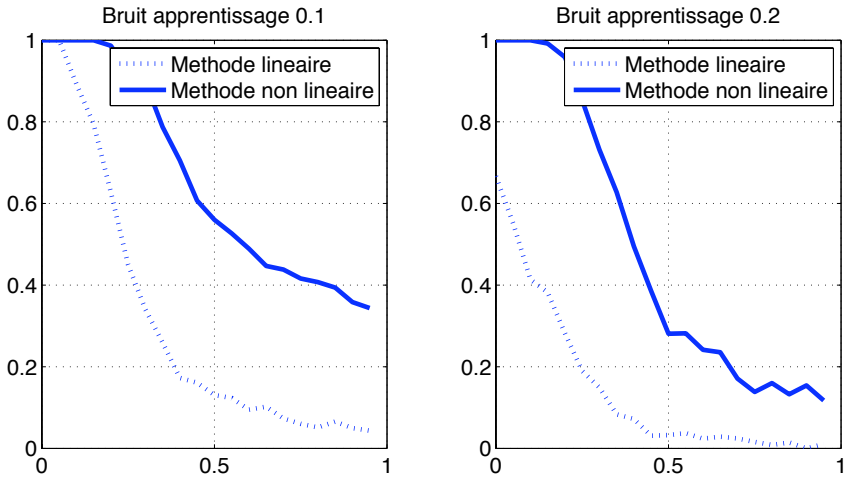


FIGURE 2.18 – Comparaison de la convergence de la méthode linéaire avec la méthode non linéaire, en fonction de la distance quadratique entre le motif et la position initiale. La courbe de gauche montre les performances pour un bruit d'apprentissage de 0.1. La courbe de droite montre les performances pour un bruit d'apprentissage de 0.2

2.5.5 Publications associées

1 **Realtime Kernel based Tracking**
T. Chateau et J.T. Lapresté
Electronic Letters on Computer Vision and Image Analysis, Vol. 8 (1), pp 27-43, 2009

2 **Suivi de Motifs Planaires Temps Réel par Combinaison de Traqueurs**
T. Chateau, J.T. Lapresté, D. Ramadasan et S. Treuillet
RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, Amiens, Janvier 2008

3 **Real-time tracking using Wavelets Representation**
T. Chateau, F. Jurie, M. Dhome and et X. Clady
Symposium for Pattern Recognition, DAGM'02, Zurich, Suisse, Septembre 2002

CHAPITRE

3

MÉTHODES DE MONTE-CARLO POUR L'ESTIMATION PAR VISION

Ce chapitre résume mes contributions autour de l'utilisation des méthodes de Monte-Carlo pour l'estimation par vision. Dans une première partie, je décris de manière synthétique les concepts utilisés. La deuxième partie étudie les performances des machines d'apprentissage, utilisées comme fonction d'observation d'un filtre à particule. La troisième partie adresse le problème difficile du suivi temps réel d'un nombre variable d'objets en environnement extérieur. La quatrième partie traite de l'estimation précise de la trajectoire d'un véhicule dans un contexte d'observation multi-capteurs.
Pour chaque application présentée, une synthèse est effectuée, reprenant le principe de la méthode, son positionnement par rapport à l'axe scientifique abordé dans le chapitre courant. Pour plus de détails, le lecteur est invité à consulter les articles associés.

3.1 Introduction

Je présente une synthèse de mes activités de recherche autour de l'utilisation des méthodes de Monte-Carlo pour l'estimation d'état par vision. Le cadre général de cette étude est le même que dans le chapitre précédent, comme illustré sur la figure 3.1. Dans ce chapitre, nous nous intéressons plus particulièrement à l'approximation de la densité de probabilité de l'état, conditionné par les observations, à l'aide de techniques de Monte-Carlo.

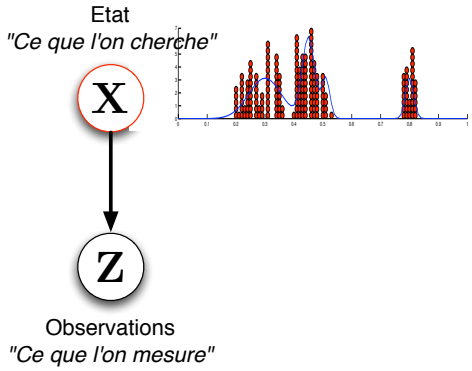


FIGURE 3.1 – Synoptique illustrant le principe de l'estimation d'état par vision. L'état $\mathbf{X}$ est une variable cachée qui génère une observation $\mathbf{Z}$.

Le principe consiste à considérer que l'état ($\mathbf{X}$) à rechercher se présente sous la forme d'une variable aléatoire, et qu'il faut déterminer sa distribution de probabilité connaissant les mesures ($\mathbf{Z}$) : $p(\mathbf{X}|\mathbf{Z})$. La vraisemblance des mesures par rapport à l'état étant connue ($p(\mathbf{Z}|\mathbf{X})$), la résolution du problème est obtenue en utilisant la règle de Bayes :

$$p(\mathbf{X}|\mathbf{Z}) = \frac{p(\mathbf{Z}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Z})} \tag{3.1}$$

Lorsque les états contiennent des variables aléatoires continues, il s'agit de représenter et de manipuler leur densité de probabilité. Il existe alors deux principaux modèles de représentation. Le premier consiste à définir les densités de probabilité par des fonctions paramétriques, le but étant d'obtenir une forme analytique de la formule de Bayes (Les modèles Gaussiens sont les plus utilisés : lorsque l'on applique la règle de Bayes à de tels modèles, la densité obtenue suit également une loi Gaussienne). Le principal inconvénient de ces techniques vient du fait qu'une hypothèse a priori est effectuée sur la forme des densités de probabilité, hypothèse qui n'est jamais vérifiée dans notre problématique. On préférera donc le deuxième modèle de représentation qui consiste à approcher les densités de probabilité par une somme d'échantillons, générés selon des techniques de Monte-Carlo. Ces méthodes permettent de gérer des densités de probabilité de formes quelconques et de dimensions variable ; ce qui est essentiel dans le cas d'applications comme du suivi multi-objets par exemple.
Les méthodes de Monte-Carlo sont très populaires en vision depuis maintenant plus de dix ans par leurs utilisations dans des applications de suivi d'objets. Néanmoins, elles sont également employées dans le cadre de l'estimation d'états statiques.
La première partie de ce chapitre expose, de manière synthétique, des méthodes de Monte-Carlo pour l'approximation de densités de probabilités stationnaires ou dynamiques. Une part importante est consacrée aux techniques dites de filtrage particulaire, très populaires en vision par ordinateur depuis une quinzaine d'années. La deuxième partie aborde des travaux qui font le lien avec le chapitre précédent. Il s'agit d'étudier comment des modèles basés apprentissage peuvent être utilisés comme fonctions d'observation dans des filtres à particules. La troisième partie traite du problème complexe du suivi d'un nombre variable d'objets de classes différentes, en temps réel, dans un contexte de vision en milieu extérieur. Pour répondre à cette problématique de manière efficace, plusieurs contributions ont été proposées. La quatrième partie compare une stratégie de suivi séquentiel par filtrage particulaire avec une stratégie d'approximation globale de la densité de probabilité

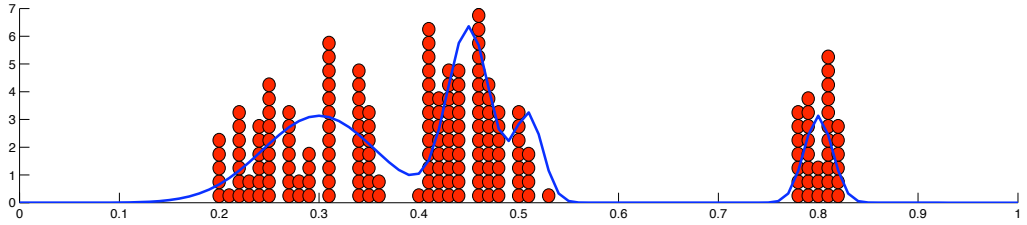


FIGURE 3.2 – Approximation d'une loi de probabilité (courbe continue bleue) par une population de N échantillons non pondérés (boules rouges).

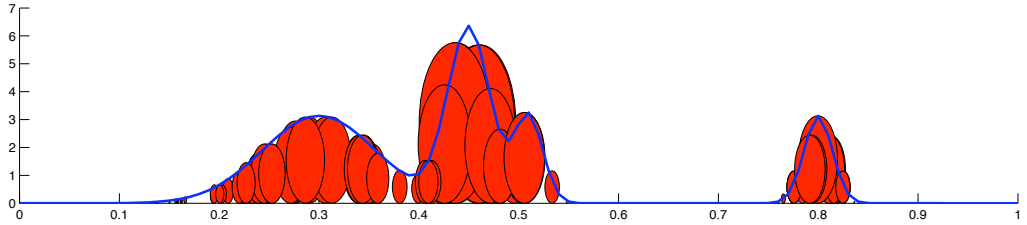


FIGURE 3.3 – Approximation d'une loi de probabilité (courbe continue bleue) par une population de N échantillons pondérés (ellipses rouges, dont la surface est proportionnelle au poids).

d'une trajectoire, dans un contexte d'estimation précise de trajectoires de véhicules à partir de données multi-capteurs. La dernière partie conclut et avance quelques perspectives pour des travaux futurs.

3.2 Méthodes de Monte-Carlo pour l'estimation de densités de probabilités.

Cette section présente, de manière synthétique, un certain nombre de méthodes utilisées en vision par ordinateur pour approximer des densités de probabilités stationnaires et dynamiques par des méthodes de Monte-Carlo. Les applications concernées portent essentiellement sur l'estimation de trajectoires d'objets dans des séquences d'images.

3.2.1 Échantillonnage de Lois de Probabilité Stationnaires

Deux grandes familles de modèles peuvent être utilisées pour représenter une loi de probabilité. La première consiste à représenter cette loi par une fonction paramétrique. Il s'agit alors d'estimer au mieux les paramètres du modèle qui explique la loi. Néanmoins, cette technique nécessite de choisir a priori une classe de fonctions paramétriques, en supposant qu'elle puisse représenter correctement la loi. La deuxième famille de modèles consiste à approximer la loi par un ensemble d'échantillons de cette dernière. Les méthodes de Monte-Carlo sont des techniques utilisées dans cette classe de modèles.
Comme illustré par la figure 3.2, une loi de probabilité de l'état $\mathbf{X}$ d'un système quelconque, peut être approximée par une moyenne de N échantillons non pondérés :

$$p(\mathbf{X}) \approx \frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{X}-\mathbf{X}^n), \text{que l'on note également : } p(\mathbf{X}) \approx \{\mathbf{X}^n\}_{n=1}^{N} \tag{3.2}$$

où δ est l'impulsion de Dirac. Comme illustré par la figure 3.3, la même distribution peut également être approximée par une somme de N échantillons pondérés de leurs poids π^n, $n \in 1...N$, tels que $\sum_{n=1}^{N}\pi^n = 1$, selon l'équation 3.3. Ces deux représentations sont deux approximations de la même loi de probabilité, où les nombres de copies de chaque particule dans la représentation non pondérée correspondent aux poids des

particules dans la représentation pondérée.

$$p(\mathbf{X}) \approx \sum_{n=1}^{N} \pi^n \delta(\mathbf{X} - \mathbf{X}^n), \text{ que l'on note également : } p(\mathbf{X}) \approx \{\mathbf{X}^n, \pi^n\}_{n=1}^{N} \tag{3.3}$$

3.2.2 Échantillonnage de lois dynamiques : filtres particulaires

Le suivi visuel d'objets peut être vu comme l'ensemble des systèmes permettant de déterminer la configuration dynamique d'un ou plusieurs objets, rigides ou déformables, à partir d'observations issues d'une ou de plusieurs caméras. La configuration dynamique est un vecteur indexé par le temps, associé à chaque objet, et dont la taille peut varier en fonction du type d'objet, du type d'observation et de l'application cible. Dans une application de suivi de piétons, il est possible de se limiter à une configuration dynamique donnée par la position 2D du centre de gravité de l'empreinte du piéton dans l'image. Par contre, dans certaines applications d'estimation de trajectoires de véhicules, la configuration dynamique peut coder l'angle au volant et l'accélération du véhicule.
On note $\mathbf{X}_t$, la configuration dynamique (état) du système suivi à l'instant t. On définit alors $\mathcal{X} \doteq \{\mathbf{X}_t\}_{t=1,\ldots,T}$ la séquence des états du système sur la fenêtre d'analyse. De manière similaire, notons $\mathbf{Z}_t$ la mesure (observation) à l'instant t et $\mathcal{Z} \doteq \{\mathbf{z}_t\}_{t=1,\ldots,T}$ la séquence des observations sur la fenêtre d'analyse. L'état $\mathbf{X}_t$ du système à l'instant t génère une observation $\mathbf{Z}_t$, dans laquelle est également présent un bruit de mesure. La séquence des états à estimer étant constituée par définition de variable cachées, c'est ce lien entre l'état et l'observation qui sera utilisé pour déterminer une estimation de la loi *a posteriori* $P(\mathbf{X}_t|\mathbf{Z}_t)$.

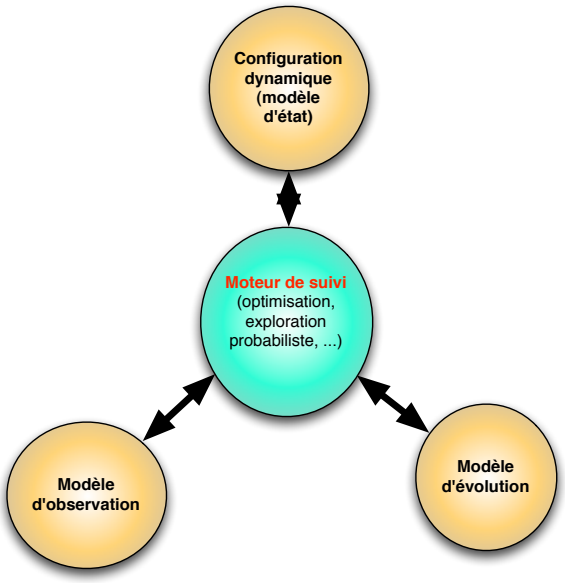


FIGURE 3.4 – Synoptique de la constitution d'une méthode de suivi d'objets : trois modèles gravitent autour du moteur de suivi.

Une méthode de suivi d'objet nécessite la définition de trois modèles (cf. figure 3.4) :

1. Le modèle d'état, aussi appelé configuration dynamique, code l'ensemble des paramètres cachés que l'on cherche à estimer.

2. Le modèle d'évolution, aussi appelé modèle de prédiction, code les contraintes dynamiques qui permettent de prédire dans quel sous-ensemble de l'espace d'état la configuration courante doit se situer, en fonction de l'ensemble des connaissances dont on dispose sur l'historique des états et des mesures.

3. Le modèle d'observation permet de faire le lien entre l'état et l'observation.

Il est souvent constitué d'un modèle de l'objet à suivre, pour lequel on recherche l'état qui explique au mieux les observations. Dans les exemples développés dans ce manuscrit, l'observation est constituée d'images et le modèle d'obversation s'appuie principalement sur des algorithmes de traitement d'images.

La méthode de suivi s'appuie sur les trois modèles définis ci-dessus pour estimer au mieux la configuration dynamique du système en fonction de la séquence d'observations dont il dispose.

3.2.2.1 Suivi Probabiliste Séquentiel

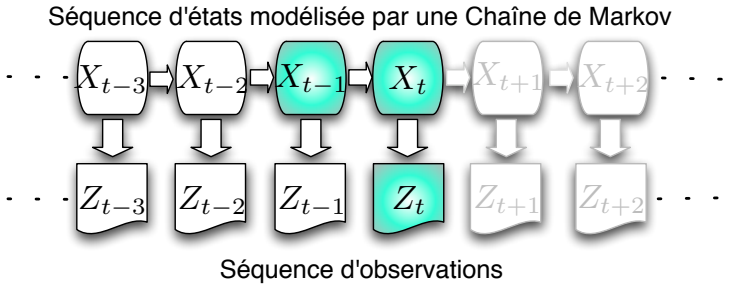


FIGURE 3.5 – Suivi d'état causal séquentiel. Seule l'observation courante et l'état précédent sont pris en compte pour inférer l'état courant.

Le suivi d'une séquence d'état peut être modélisé par un processus Markovien d'ordre 1 (l'état courant ne dépend que de l'état précédent comme l'illustre la figure 3.5). Le suivi probabiliste séquentiel, qui rentre dans ce cadre, permet d'estimer récursivement la loi de probabilité *a posteriori* $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$ de l'état $\mathbf{X}_t$ à l'instant t, conditionné à l'historique des mesures $\mathbf{Z}_{1:t}$. Dans ces conditions, la rêgle de Bayes devient :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) = \frac{p(\mathbf{Z}_t|\mathbf{X}_t)p(\mathbf{X}_t|\mathbf{Z}_{1:t-1})}{p(\mathbf{Z}_t|\mathbf{Z}_{1:t-1})}, \tag{3.4}$$

où la loi de probabilité *a priori* $p(\mathbf{X}_t|\mathbf{Z}_{1:t-1})$ est explicitée par l'équation de Chapman-Kolmogorov (3.5) :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) = \int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})d\mathbf{X}_{t-1}. \tag{3.5}$$

Dans l'équation (3.5), $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ est la loi de probabilité d'évolution temporelle de l'état, qui génère les prédictions pour l'instant t, et $p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})$ est la loi de probabilité *a posteriori* à l'instant $t-1$. Le remplacement de l'équation (3.5) dans l'équation (3.4) conduit à l'expression du filtre Bayesien récursif :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) = C^{-1}p(\mathbf{Z}_t|\mathbf{X}_t)\int_{\mathbf{X}_{t-1}} p(\mathbf{X}_t|\mathbf{X}_{t-1})p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})d\mathbf{X}_{t-1}, \tag{3.6}$$

où $C = p(\mathbf{Z}_t|\mathbf{Z}_{1:t-1})$ est une constante indépendante de l'état.
L'équation du filtre Bayesien récusif (3.6) définit la loi de probabilité de l'état d'un système dynamique à l'instant courant, à partir de la loi de probabilité de son état à l'instant précédent, et de l'observation courante. Malheureusement, l'intégrale qu'elle contient ne peut en général être calculée analytiquement. Une solution consiste alors à mettre en oeuvre des techniques d'échantillonnage pour approximer ces lois de probabilités. Les filtres particulaires produisent une approximation échantillonnée de cette intégrale. L'efficacité d'un Filtre Particulaire dépend de sa capacité à exploiter les données antérieures, pour choisir au mieux les échantillons.

3.2.2.2 Filtres particulaires : typologies

Une première famille de stratégies, les filtres particulaires parallèles, est utilisée avec succès en vision depuis plusieurs années. La méthode est basée sur l'algorithme *SIR* (Sequential Importance Resampling) ou

CONDENSATION proposé par Isard et Blake (45) : des hypothèses indépendantes sont générées, puis évaluées de manière parallèle, avant d'être propagées à l'observation suivante. Cette stratégie est intéressante car elle permet la parallélisation des calculs les plus lourds en suivi visuel : l'évaluation de la vraisemblance des hypothèses connaissant une observation lourde à traiter (une image).
Les stratégies de la deuxième famille, les filtres particulaires *MCMC* (ou *MCMC PF* : *Markov Chain Monte-Carlo Particle Filters*) (65), génèrent une chaîne de Markov d'hypothèses, la transition d'une hypothèse à la suivante étant conditionnée par l'observation. Le processus d'exploration de l'espace à un instant t est donc chaîné, ou itératif : il est plus « informé » que dans les filtres particulaires parallèles, puisque l'observation intervient dans la transition entre deux hypothèses. C'est son point fort, et l'utilisation des filtres particulaires *MCMC* en suivi visuel, bien que plus récente, a montré un potentiel certain (52).

3.2.3 Filtres particulaires *SIR*

On a vu l'approximation échantillonnée d'une loi stationnaire en section 3.2.1. Dans ce cas, on n'a en général pas d'*a priori* sur cette loi. Dans le cas d'une loi dynamique, on peut s'appuyer sur une prédiction générée à partir du passé et d'un modèle dynamique du système. C'est ce que proposent les filtres particulaires : ils propagent temporellement une approximation échantillonnée de la loi dynamique. Les filtres particulaires reposent sur l'équation du filtre Bayesien récusif (3.6), qui définit la loi de probabilité de l'état d'un système dynamique à l'instant courant, à partir de la loi de probabilité de son état à l'instant précédent, et de l'observation courante. Il produisent une approximation échantillonnée de l'intégrale présente dans l'equation de Bayes. L'efficacité d'un filtre particulaire dépend de sa capacité à exploiter les données antérieures, pour choisir au mieux les échantillons. Les Filtres Particulaires *SIR* (*Sequential Importance Resampling*) s'appuient sur une étape d'*Échantillonnage préférentiel* ou *IS* (*Importance Sampling*).
Supposons que l'on dispose d'une approximation de $p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})$, loi de probabilité *a posteriori* de l'état à l'instant $t-1$ par N échantillons discrets pondérés $\{\mathbf{X}_{t-1}^n, \pi_{t-1}^n\}_{n=1}^N$:

$$p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1}) \approx \sum_{n=1}^{N} \pi_{t-1}^n \delta(\mathbf{X}_{t-1} - \mathbf{X}_{t-1}^n), \tag{3.7}$$

où δ est l'impulsion de Dirac, et π_{t-1}^n est le poids du $n^{\text{ième}}$ échantillon, $n \in 1...N$, tel que $\sum_{n=1}^N \pi_{t-1}^n = 1$.
L'approximation échantillonnée de l'équation (3.5) s'écrit alors :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) \approx \sum_{n=1}^{N} \pi_{t-1}^n p(\mathbf{X}_t|\mathbf{X}_{t-1}^n), \tag{3.8}$$

où $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ est la loi de probabilité d'évolution dynamique du système. La loi définie par l'équation (3.8) est une mixture des N composantes $p(\mathbf{X}_t|\mathbf{X}_{t-1}^n)$, pondérées des poids π_{t-1}^n. L'équation du filtre Bayesien récursif (3.6) peut alors être approximée par :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t}) \approx C^{-1} p(\mathbf{Z}_t|\mathbf{X}_t) \sum_{n=1}^{N} \pi_{t-1}^n p(\mathbf{X}_t|\mathbf{X}_{t-1}^n). \tag{3.9}$$

Comme on ne dispose généralement pas non plus d'expression analytique de $p(\mathbf{Z}_t|\mathbf{X}_t)$, il faut à nouveau échantillonner. On pourrait bien sûr reprendre les anciens échantillons $\mathbf{X}_{t-1}^n$, mais ils ne sont pas forcément les plus pertinents pour représenter efficacement $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$, car la région intéressante de l'espace a pu changer entre $t-1$ et t. Il faut donc rééchantillonner, c'est à dire choisir de nouveaux échantillons $\mathbf{X}_t^n$. Ce choix s'opère par un échantillonnage par importance dans le nuage de particules pondérées $\{\mathbf{X}_{t-1}^n, \pi_{t-1}^n\}_{n=1}^N$, et un tirage dans la loi dynamique appliquée à chacun de ces échantillons. Ces deux opérations réalisées séquentiellement, équivalent à tirer N échantillons d'état $\mathbf{X}_t^n$ selon (3.10) :

$$\mathbf{X}_t^n \sim q(\mathbf{X}_t) = \sum_{n=1}^{N} \pi_{t-1}^n p(\mathbf{X}_t|\mathbf{X}_{t-1}^n) \tag{3.10}$$

Autrement dit, un filtre particulaire produit à chaque pas temporel t un échantillonnage par importance de la loi de probabilité *a posteriori* à l'instant t. Pour chacun des échantillons $\mathbf{X}_t^n$ on calcule enfin la vraisemblance de l'observation : $\pi_t^n = P(\mathbf{Z}_t|\mathbf{X}_t^n)$. Le filtre délivre alors le lot de N échantillons discrets pondérés $\{\mathbf{X}_t^n, \pi_t^n\}_{n=1}^N$, approximant $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$, loi de probabilité *a posteriori* de l'état à l'instant t.
L'algorithme *SIR* est illustré en dimension 1 par la figure 3.6. Il exécute trois étapes à chaque pas temporel, à partir du nuage de particules approximant la loi de probabilité à $t-1$:

- ▷ (a) Tirage par importance : tirage avec remise des particules selon leur « importance » (leur poids) partant de l'observation à $t-1$. Cette opération remplace les particules de poids forts par de nombreuses particules, et les particules de poids faibles par peu de particules. A ce stade, elles sont toutes affectées de poids identiques.

- ▷ (b) Évolution temporelle : remplacer chaque particule selon la fonction de proposition $p(\mathbf{X}^*|\mathbf{X})$. Si on n'a aucune information sur la loi d'évolution du processus à estimer, on peut choisir une loi normale $p(\mathbf{X}^*|\mathbf{X}) = \mathcal{N}(0, \sigma)$, la valeur de σ étant choisie avec soin car c'est elle qui détermine l'échantillonnage de l'espace d'état par les particules. Mais le filtre sera beaucoup plus performant si on inclut à ce stade un modèle d'évolution, prenant en compte la dynamique des objets suivis.

- ▷ (c) Pondération des nouvelles particules par la vraisemblance de l'observation $\mathbf{Z}_t$ à l'instant t, sachant l'échantillon d'état $\mathbf{X}_t^n$: $\pi_t^n = P(\mathbf{Z}_t|\mathbf{X}_t^n)$

La propagation des particules de $t-1$ à t fait la force de l'algorithme *SIR*, et lui donne naturellement les propriétés d'un filtre. A chaque itération, on attire l'exploration vers les régions de forte loi de probabilité, ce qui rend l'échantillonnage adaptatif. Le choix du modèle d'évolution temporelle des particules détermine l'efficacité du rééchantillonnage. Le point faible de cet algorithme est qu'il requiert un nombre de particules, lié exponentiellement à la dimension de l'espace, défaut pointé dans de nombreuses études, parmi lesquelles (46; 100).

3.2.3.1 Choix des échantillons

Un échantillonnage uniforme, couvrant tout l'espace d'état, est déjà peu efficace en dimension 1, comme l'illustre le graphe (a) de la figure 3.7. Un échantillonnage aléatoire est illustré graphe (b). Il ne fait pas mieux, et les deux graphes ont en commun un important gaspillage d'échantillons qui tombent dans des régions de l'espace où il ne se passe rien. L'échantillonnage adaptatif réalisé sur le graphe (c) est beaucoup plus intéressant : il ajuste la densité des échantillons en fonction de l'intérêt de la région. Dans les applications de suivi multi-objets, la dimension de l'espace d'état est élevée : une application typique est le suivi d'un piéton marchant sur un sol plan. Son descripteur le plus simple (un cylindre) nécessite un vecteur de dimension 6 : deux composantes pour la forme (rayon et hauteur), deux composantes pour la position, deux composantes pour la vitesse. Supposons que N échantillons réalisent un bon échantillonnage de son espace d'état. Si l'on souhaite maintenant suivre p piétons, la dimension de l'espace d'état joint représentant la scène devient $6p$. Le nombre d'échantillons nécessaires devient N^{6p}. Cette croissance exponentielle selon la dimension de l'espace rend la méthode inopérante dès qu'il y a plus de deux objets à suivre (46). L'échantillonnage adaptatif réalisé sur le graphe (c) devient alors indispensable. Une méthode d'échantillonnage appelée *SIR Partitionné* a été conçue pour surmonter cette difficulté. L'état est partitionné et l'échantillonnage s'effectue de manière séquentielle selon les partitions. Plus de détails sur cette méthode sont disponibles dans (64). Le *SIR Partitionné* présente malheureusement un défaut, identifié dans (100) qui se traduit par une pauvreté (nombre de particules différentes) des premières partitions re-échantillonnées.

3.2.4 Échantillonneurs *MCMC*

Un échantillonneur permet d'approximer une loi de probabilité stationnaire inconnue notée π définie sur l'espace de la variable aléatoire $\mathbf{X}$, par un nuage d'échantillons non pondérés, tel que défini dans l'équation (3.2). Lorsque l'on n'a pas de modèle simple de $p(\mathbf{X})$, le tirage des échantillons est difficile. Les

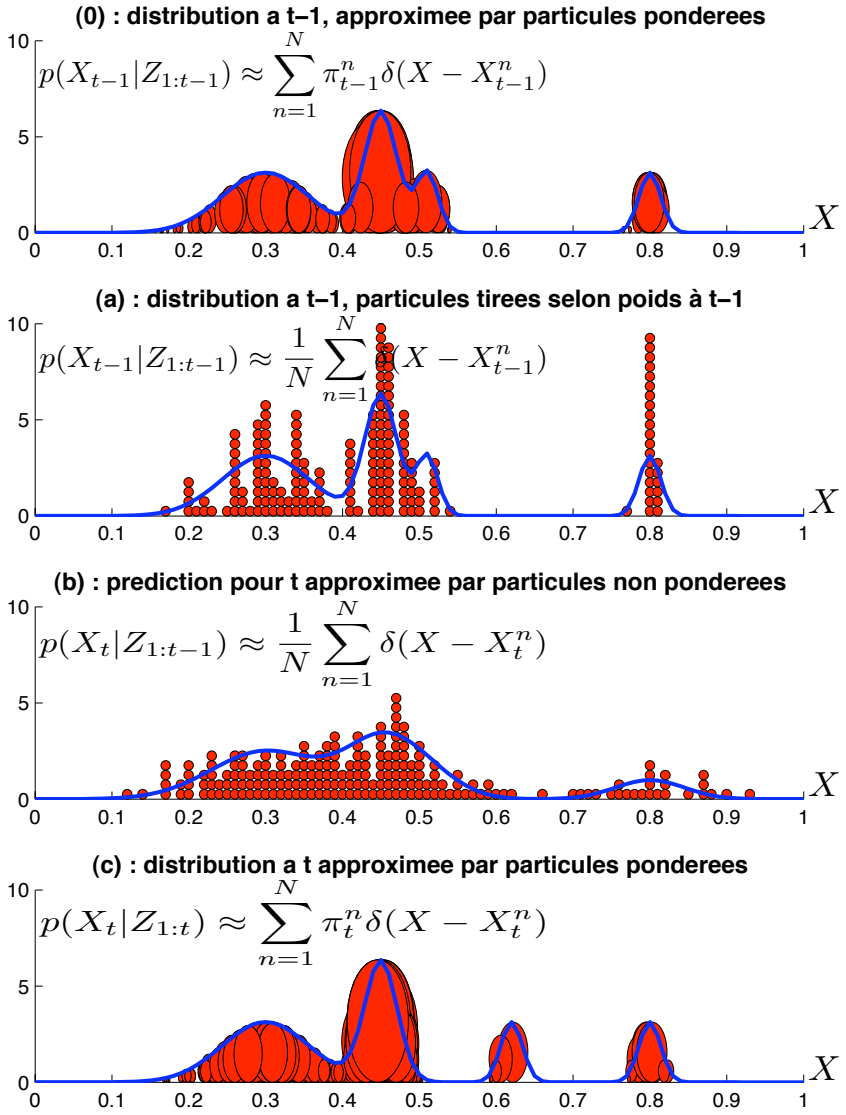


FIGURE 3.6 – Propagation des particules par l'algorithme *SIR* avec évolution dynamique des particules $\mathcal{N}(0, \sigma = 0.06)$. (0) : Loi de probabilité *a posteriori* à $t-1$ (courbe bleue) et son approximation par des particules pondérées (ellipses rouges de surface proportionnelle à leur poids). (a) : La même loi de probabilité *a posteriori* à $t-1$ (courbe bleue) et son approximation par des particules non pondérées, après rééchantillonnage par Importance selon le poids des particules à l'instant $t-1$. (b) : Loi de probabilité *a priori* à t (courbe bleue) et son approximation par des particules non pondérées, générées selon le modèle dynamique $\mathcal{N}(0, \sigma = 0.06)$. (c) : Loi de probabilité *a posteriori* à t (courbe bleue) et son approximation par des particules pondérées selon l'observation à l'instant t.

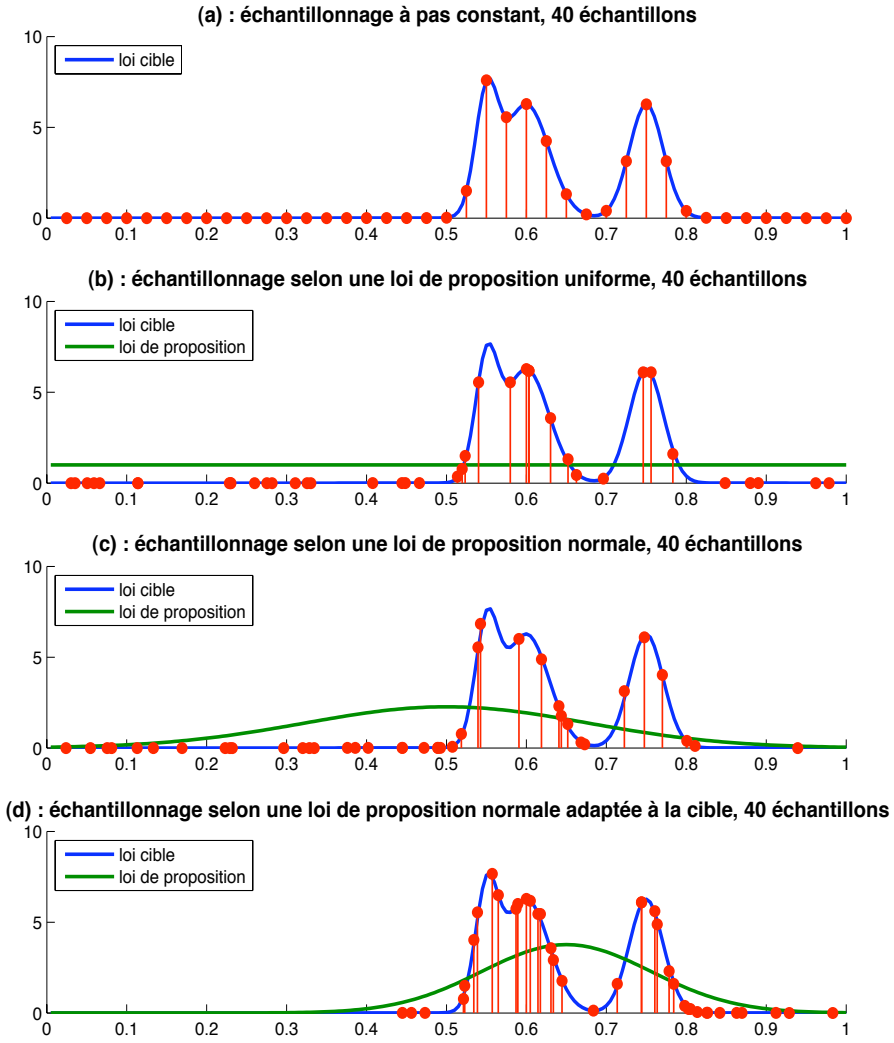


FIGURE 3.7 – Échantillonnage d'une loi stationnaire (courbe continue bleue) par une population de $N = 40$ échantillons (points rouges), tirés d'une loi de proposition inadaptée à la cible ((a), (b), et (c)). On obtient évidemment un échantillonnage plus efficace si la loi de proposition est adaptée à la cible (c'est à dire qu'on a une petite idée de celle-ci), tout en permettant des tirages faciles (ici $\mathcal{N} = (0.65, \sigma = 0.15)$). Illustration en couleur

échantillonneurs *MCMC* (Markov Chain Monte-Carlo) forment une famille de méthodes d'échantillonnage non parallèles mais chaînées, c'est à dire que chaque échantillon est généré par une fonction dite fonction d'importance ou de proposition, à partir de l'échantillon précédent. Nous consacrons cette section aux échantillonneurs *MCMC* car les algorithmes de Filtrage Particulaire présentés plus loin dans ce chapitre mettent en oeuvre à chaque pas temporel un échantillonnage *MCMC* de la loi de probabilité de l'état *a posteriori* du système.

3.2.4.1 Échantillonneur de Metropolis

L'échantillonneur de Metropolis est une méthode itérative ou chaînée : l'exploration de l'espace est menée par une chaîne de Markov d'ordre 1, selon l'algorithme de Métropolis (65), où la transition d'un échantillon $\mathbf{X}^{n-1}$

au suivant $\mathbf{X}^n$, est assurée par l'intermédiaire d'une proposition $\mathbf{X}^*$ tirée de la loi $q(\mathbf{X}^*|\mathbf{X}^{n-1})$. Cette loi n'a rien de dynamique, puisque le temps n'intervient pas ici. C'est une loi d'exploration de l'espace d'état. Le rapport des vraisemblances respectives de l'observation $\mathbf{Z}$ sachant $\mathbf{X}^*$ et $\mathbf{X}^{n-1}$ détermine la probabilité d'acceptation de $\mathbf{X}^*$. En cas de refus $\mathbf{X}^{n-1}$ est dupliqué. Une exploration avec l'échantillonneur de Metropolis est une marche aléatoire conduite par sa loi de proposition $q(\mathbf{X}^*|\mathbf{X})$. Les hypothèses ne sont donc pas indépendantes. La figure 3.8 illustre une itération de l'échantillonneur de Metropolis : la proposition $\mathbf{X}^*$ est acceptée avec la probabilité α calculée selon la règle de Métropolis, sinon on recopie l'état précédent $\mathbf{X}^n = \mathbf{X}^{n-1}$. Metropolis a montré que son échantillonneur converge vers la distribution cible, à condition que

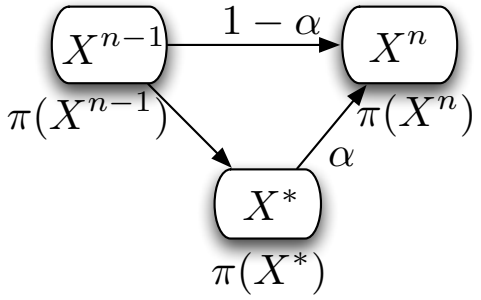


FIGURE 3.8 – Une itération de l'échantillonneur de Metropolis : une proposition $\mathbf{X}^*$ est générée à partir de la loi $q(\mathbf{X}^*|\mathbf{X}^{n-1})$. Elle est acceptée avec la probabilité α calculée à partir des évaluations de la loi de probabilité π pour les deux états : $\pi(\mathbf{X}^{n-1})$ et $\pi(\mathbf{X}^*)$, sinon l'état précédent $\mathbf{X}^{n-1}$ est dupliqué.

la chaîne soit « suffisamment longue », pour rendre les hypothèses pseudo-indépendantes, et que la loi de proposition $q(\mathbf{X}^*|\mathbf{X})$ soit symétrique par rapport à $\mathbf{X}$. En pratique, il n'est pas facile de déterminer la « longueur suffisante » de la chaîne, l'expérimentation est indispensable. Les conditions ci-dessous permettent d'accélérer la convergence :

- ▷ Un échantillon d'initialisation $\mathbf{X}^{ini}$ pas trop décentré par rapport à la cible.
- ▷ La suppression des premiers échantillons de la chaîne, biaisés par leur dépendance à l'échantillon de départ $\mathbf{X^0}$. Cette opération est dénommée *Burn-in*. L'expérience montre qu'un *Burn-in* de 20% de la chaîne, c'est à dire l'élimination des premiers 20% de la chaîne rend le biais d'initialisation négligeable.
- ▷ L'adaptation de la loi de proposition $q(\mathbf{X}^*|\mathbf{X})$ à la fonction cible (a priori inconnue). Si on n'en a aucune idée, on pourra par exemple choisir comme loi de probabilité de proposition une loi normale $\mathcal{N}(0, \sigma)$, mais l'exploration convergera beaucoup plus rapidement si on a une connaissance a priori de la cible. On rend alors la loi de proposition $q(\mathbf{X}^*|\mathbf{X})$ adaptative aux observations (ce sont les *DDMCMC*, ou Data-Driven *MCMC*).

3.2.4.2 Échantillonneur de Metropolis-Hastings MH_D

L'échantillonneur de Metropolis-Hastings est aussi une méthode itérative ou chaînée. Dans l'échantillonneur de Metropolis, le choix d'une loi de proposition $q(\mathbf{X}^*|\mathbf{X})$ adaptative aux observations risque d'entrer en conflit avec la contrainte de parité de $q(\mathbf{X}^*|\mathbf{X})$imposée par l'algorithme de Metropolis. Hastings en a proposé une extension, levant ce conflit : l'algorithme de Metropolis-Hasting (65) accepte une loi de proposition $q(\mathbf{X}^*|\mathbf{X})$ quelconque. Les échantillonneurs de Metropolis-Hastings proposent de nouveaux échantillons en effectuant des mouvements dans toutes les D dimensions de l'espace simultanément, ce qui est spécifié par l'indice D. La progression du processus est illustrée sur la figure 3.9. La figure montre que les premières itérations sont fortement biaisées par la valeur initiale de la chaîne. Il est bénéfique de les éliminer de la chaîne finale (*Burn-in*). Le choix de la loi de proposition $q(\mathbf{X}^*|\mathbf{X})$ est un point clef conditionnant une bonne exploration de l'espace.

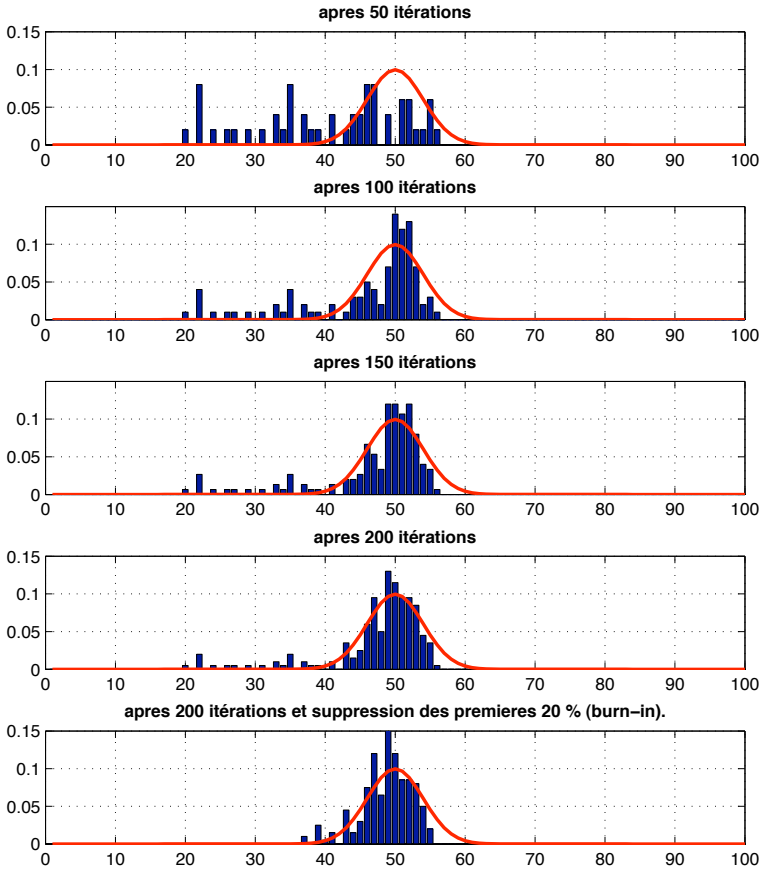


FIGURE 3.9 – Progression de l'échantillonnage par *MCMC* d'une loi de probabilité stationnaire monodimensionnelle unimodale $\mathcal{N}(50, \sigma = 4)$. Chaîne volontairement initialisée à $\mathbf{X} = 20$ pour illustrer la nécessité du *Burn-in*. Loi de proposition $q(\mathbf{X}^*|\mathbf{X}) = \mathcal{N}(0, \sigma = 2)$

3.2.4.3 Échantillonneur de Metropolis-Hastings à propositions marginalisées MH_d

L'échantillonneur de Metropolis-Hastings présenté plus haut utilise une fonction de proposition qui déplace l'échantillon dans toutes les dimensions simultanément. C'est pourquoi il peine à trouver les régions de forte probabilité lorsque la dimension de l'espace augmente. Cette difficulté est parfaitement expliquée dans (99). L'échantillonneur de Metropolis-Hastings à propositions marginalisées (MH_d) est une variante de l' échantillonneur de Metropolis-Hastings, l'indice d signifiant que les nouveaux échantillons sont proposés en effectuant des mouvements marginalisés dans une seule des dimensions de l'espace. Dans cet algorithme, on partitionne le vecteur d'état $\mathbf{X}^n$ de dimension D en S sous-vecteurs $\mathbf{X}^n$, tous de dimension d, tels que $\mathbf{X}^n = [\mathbf{X}^{s,n}]_{s=1}^S$, avec $D = S.d$. A chaque itération de la chaîne, on ne fait évoluer qu'un des vecteurs $\mathbf{X}^n$. Ceci revient à appliquer une succession d'itérations de Metropolis-Hastings dans des sous-espaces de dimensions d de l'espace d'état.

3.2.5 Filtres Particulaires par *MCMC*

Également appelés *Markov Chain Monte-Carlo Particle Filters* (*MCMC PF*), ces filtres furent introduits dans (53) pour le suivi d'un grand nombre d'objets. Le principe proposé est de remplacer l'échantillonnage par importance par un échantillonnage de Metropolis. On a vu qu'un tel échantillonneur produit une approximation d'une loi de probabilité quelconque par un nuage d'échantillons non pondérés. Le Filtre Particulaire *MCMC* applique un échantillonnage de Metropolis pour approximer $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$, loi de probabilité *a posteriori* de l'état à chaque pas temporel t. Dans la section 3.2.6, nous présentons un Filtre Particulaire *MCMC* dont les nouveaux échantillons proposés sont générés en effectuant des mouvements dans toutes les D dimensions de l'espace simultanément. Nous le notons *FP MCMC*$_D$. On examinera dans la section 3.2.6.1 le *FP MCMC*$_d$ une variante du *FP MCMC*$_D$, dans lequel chaque nouvel échantillon proposé est généré en ne perturbant que d dimensions de l'état de l'échantillon précédent, avec $0 \leq d \leq D$, c'est à dire que les propositions se font sur un sous-espace de dimension d. Supposons que l'on dispose d'une approximation de $p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1})$ par N échantillons discrets non pondérés $\{\mathbf{X}_{t-1}^\nu\}_{\nu=1}^N$:

$$p(\mathbf{X}_{t-1}|\mathbf{Z}_{1:t-1}) \approx \frac{1}{N}\sum_{\nu=1}^{N} \delta(\mathbf{X}_{t-1} - \mathbf{X}_{t-1}^\nu). \tag{3.11}$$

L'équation (3.12) délivre $p(\mathbf{X}_t|\mathbf{Z}_{1:t-1})$, loi de probabilité *a priori* de l'état à l'instant t, approximation de l'équation (3.5) par N composantes de prédiction dynamique non pondérées :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) \approx \frac{1}{N}\sum_{\nu=1}^{N} p(\mathbf{X}_t|\mathbf{X}_{t-1}^\nu), \tag{3.12}$$

où $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ est la loi de probabilité d'évolution dynamique du système. La loi définie par l'équation (3.12) est la moyenne des N composantes $p(\mathbf{X}_t|\mathbf{X}_{t-1}^\nu)$. L'approximation échantillonnée de l'équation (3.6) du filtre Bayesien récursif est toujours donnée par l'équation (3.9). Seule la constante C est modifiée, ce qui est sans influence, puisque la loi est normalisée.

3.2.6 Filtre Particulaire *MCMC*$_D$

Dans le Filtre Particulaire *MCMC*$_D$, les nouveaux échantillons proposés sont générés en perturbant simultanément toutes les D dimensions de l'espace. L'échantillonnage s'opère en choisissant pour loi de proposition q, l'approximation échantillonnée de la loi *a priori* :

$$\mathbf{X}_t^n \sim q(\mathbf{X}_t) = \frac{1}{N}\sum_{n=1}^{N} p(\mathbf{X}_t|\mathbf{X}_{t-1}^n) \tag{3.13}$$

La différence est que le rééchantillonnage de la loi de probabilité *a posteriori* réalisé à chaque pas temporel, est mené par un échantillonneur de Metropolis-Hastings. Comme ce dernier, le filtre approxime alors

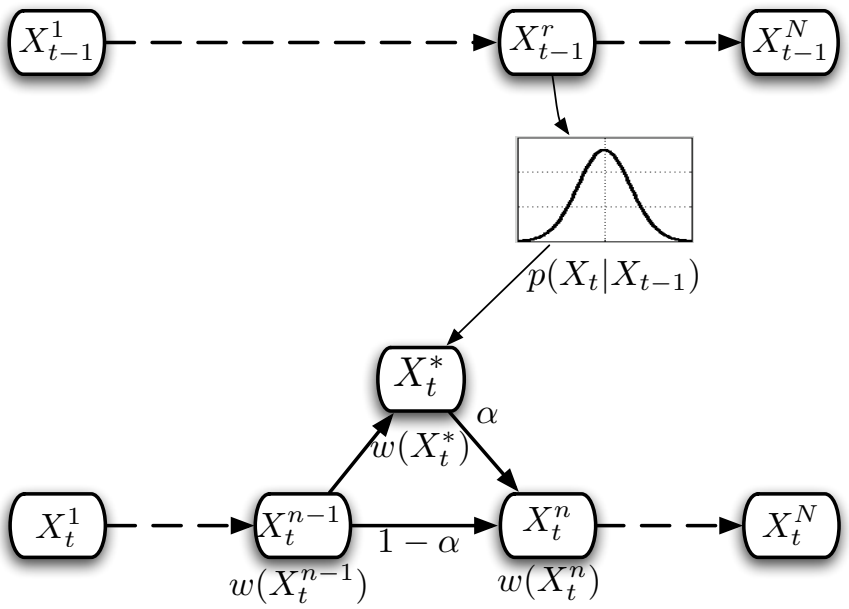


FIGURE 3.10 – Une itération du *Filtre Particulaire MCMC* : une particule $\mathbf{X}_{t-1}^r$ est tirée aléatoirement de la chaîne à l'instant précédent. Une proposition $\mathbf{X}_t^*$ est alors générée à partir de la loi $q(\mathbf{X}_t^*|\mathbf{X}_{t-1}^r)$. Elle est acceptée avec la probabilité α selon l'équation 3.16, sinon l'état précédent $\mathbf{X}_t^{n-1}$ est dupliqué.

$p(\mathbf{X}_t|\mathbf{Z}_{1:t})$, loi de probabilité *a posteriori* de l'état à l'instant t, par le lot de N échantillons discrets non pondérés $\{\mathbf{X}_t^n\}_{n=1}^N$. La figure 3.10 illustre une itération du *Filtre Particulaire MCMC*. L'exploration est menée par une chaîne de Markov selon l'algorithme de Metropolis-Hasting, où la loi cible π est remplacée par $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$, loi de probabilité *a posteriori* de l'état à chaque pas temporel t. La loi de proposition q est définie par (3.13), mixture des lois de prédiction dynamique appliquées au nuage de particules approximant la loi *a posteriori* à l'instant précédent. Le taux d'acceptation à l'itération n est alors :

$$\alpha = min\left(1, \frac{P(\mathbf{X}_t^*|\mathbf{Z}_{1:t})Q(\mathbf{X}_t^{n-1})}{P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t})Q(\mathbf{X}_t^*)}\right). \tag{3.14}$$

L'application de la règle de Bayes à $P(\mathbf{X}_t^*|\mathbf{Z}_{1:t})$ et à $P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t})$ permet de développer α :

$$\alpha = min\left(1, \frac{P(\mathbf{Z}_t|\mathbf{X}_t^*)P(\mathbf{X}_t^*|\mathbf{Z}_{1:t-1})Q(\mathbf{X}_t^{n-1})}{P(\mathbf{Z}_t|\mathbf{X}_t^{n-1})P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1})Q(\mathbf{X}_t^*)}\right) \tag{3.15}$$

Or $q(\mathbf{X}_t)$ défini par l'équation (3.13) n'est autre que l'approximation échantillonnée de $p(\mathbf{X}_t|\mathbf{Z}_{1:t-1})$ défini par l'équation (3.12), donc $P(\mathbf{X}_t^*|\mathbf{Z}_{1:t-1}) \approx Q(\mathbf{X}_t^*)$ et $P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) \approx Q(\mathbf{X}_t^{n-1})$, ce qui simplifie considérablement les calculs :

$$\alpha = min\left(1, \frac{P(\mathbf{Z}_t|\mathbf{X}_t^*)}{P(\mathbf{Z}_t|\mathbf{X}_t^{n-1})}\right) \tag{3.16}$$

Cet algorithme présente deux différences par rapport à l'algorithme *SIR* : 1) la constitution du nuage de particules est séquentielle. 2) l'approximation des lois *a posteriori* se fait ici par nuage de particules non pondérées, alors qu'elles sont pondérées dans le *SIR*. Tel quel, cet algorithme peut être vu comme une réécriture chaînée du Filtre Particulaire parallèle, algorithme *SIR*, et où les poids des particules ont été remplacés par des densités de population de l'espace par les particules. Il présente donc la même faiblesse : il peinera à échantillonner convenablement une loi de probabilité dans un espace d'état de grande dimension.

3.2.6.1 Filtre Particulaire *MCMC*$_d$

Le Filtre Particulaire *MCMC*$_D$ est performant dans les espaces de petites dimensions, mais est inefficace dans les espaces de grande dimension, car l'étape de prédiction mobilise à chaque itération la totalité des composantes de $\mathbf{X}_t^n$. Dans le cas des échantillonneurs, la même difficulté a été mise en exergue avec l'échantillonneur de Metropolis-Hastings MH_D, palliée par l'échantillonneur de Metropolis-Hastings à propositions marginalisées MH_d, une stratégie permettant de la contourner, en effectuant des mouvements sur un sous-espace et non sur la totalité de l'espace d'état. Cet algorithme peut être utilisé pour réaliser le rééchantillonnage des Filtres Particulaires *MCMC* : c'est l'intérêt du filtre présenté par Khan *et al.* (52; 53). Le *FP MCMC*$_d$ est une extension du *FP MCMC*$_D$, dans lequel on partitionne le vecteur d'état $\mathbf{X}_t^n$ de dimension D en S sous-vecteurs $\mathbf{x}_t^{s,n}$, tous de dimension d, tels que $\mathbf{X}_t^n = [\mathbf{x}_t^{s,n}]_{s=1}^S$, avec $D = S.d$. A chaque itération de la chaîne, le *FP MCMC*$_d$ tire l'indice s du sous-espace où va se produire le mouvement. Notons $p(\mathbf{x}_t^s|\mathbf{x}_{t-1}^s)$ la loi de probabilité modélisant la dynamique marginalisée sur le sous-espace s, que nous allons utiliser pour expliciter les facteurs nécessaires au calcul du taux d'acceptation défini en 3.15.

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \prod_{s=1}^{S} p(\mathbf{x}_t^s|\mathbf{x}_{t-1}^s), \tag{3.17}$$

que l'on utilise pour réécrire l'approximation échantillonnée de la loi *a priori* (3.12) :

$$p(\mathbf{X}_t|\mathbf{Z}_{1:t-1}) \approx \frac{1}{N}\sum_{\nu=1}^{N} p(\mathbf{X}_t|\mathbf{X}_{t-1}^{\nu}) = \frac{1}{N}\sum_{\nu=1}^{N}\prod_{s=1}^{S} p(\mathbf{x}_t^s|\mathbf{x}_{t-1}^{s,\nu}) \tag{3.18}$$

A chaque itération n, on propose un nouvel état joint $\mathbf{X}_t^*$, généré par mouvement marginalisé sur un des sous-espaces $s \in \{1, ..., S\}$. Notons s^* ce sous-espace. Pour calculer le taux d'acceptation, il nous faut évaluer cette loi pour l'état proposé $\mathbf{X}_t^*$ et pour l'état précédent $\mathbf{X}_t^{n-1}$:

$$P(\mathbf{X}_t^*|\mathbf{Z}_{1:t-1}) \approx \frac{1}{N}\sum_{\nu=1}^{N} P(\mathbf{x}_t^*|\mathbf{x}_{t-1}^{s^*,\nu}) \prod_{s=1,s\neq s^*}^{S} P(\mathbf{x}_t^*|\mathbf{x}_{t-1}^{s,\nu}) \tag{3.19}$$

$$P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) \approx \frac{1}{N}\sum_{\nu=1}^{N} P(\mathbf{x}_t^{n-1}|\mathbf{x}_{t-1}^{s^*,\nu}) \prod_{s=1,s\neq s^*}^{S} P(\mathbf{x}_t^{n-1}|\mathbf{x}_{t-1}^{s,\nu}) \tag{3.20}$$

Comme les projections de $\mathbf{X}_t^*$ et $\mathbf{X}_t^{n-1}$ sur tout sous-espace $s \neq s^*$ sont confondues, on peut noter :

$$a_t^{\nu} = \prod_{s=1,s\neq s^*}^{S} P(\mathbf{x}_t^*|\mathbf{x}_{t-1}^{s,\nu}) = \prod_{s=1,s\neq s^*}^{S} P(\mathbf{x}_t^{n-1}|\mathbf{x}_{t-1}^{s,\nu}), \forall \nu \in \{1, ..., N\} \tag{3.21}$$

D'autre part, les configurations jointes $\mathbf{X}_t^*$ sont tirées selon la loi de proposition :

$$\mathbf{X}_t^* \sim q(\mathbf{X}_t|\mathbf{X}_t^{n-1}) = \sum_{s=1}^{S} q(s) q(\mathbf{X}_t|\mathbf{X}_t^{n-1}, s), \tag{3.22}$$

où $q(s)$ est la loi de probabilité conditionnant le choix du sous-espace où se fait le mouvement, tandis que $q(\mathbf{X}_t|\mathbf{X}_t^{n-1}, s)$ est la loi de proposition de $\mathbf{X}_t$, conditionnée par le choix du sous-espace s :

$$q(\mathbf{X}_t|\mathbf{X}_t^{n-1}, s) = \frac{1}{N}\sum_{\nu=1}^{N} p(\mathbf{x}_t^s|\mathbf{x}_{t-1}^{s,\nu}) \prod_{s=1,s\neq s^*}^{S} \delta(\mathbf{x}_t^s - \mathbf{x}_t^{s,n-1}) \tag{3.23}$$

L'équation (3.22) peut alors se réécrire :

$$q(\mathbf{X}_t|\mathbf{X}_t^{n-1}) = \sum_{s=1}^{S} q(s) \frac{1}{N}\sum_{\nu=1}^{N} p(\mathbf{x}_t^s|\mathbf{x}_{t-1}^{s,\nu}) \prod_{s=1,s\neq s^*}^{S} \delta(\mathbf{x}_t^s - \mathbf{x}_t^{s,n-1}) \tag{3.24}$$

L'évaluation de la loi (3.24) pour l'état proposé $\mathbf{X}_t^*$ et pour l'état initial $\mathbf{X}_t^{n-1}$, donne respectivement :

$$Q(\mathbf{X}_t^*|\mathbf{X}_t^{n-1}) = Q(s^*)\frac{1}{N}\sum_{\nu=1}^{N} P(\mathbf{x}_t^*|\mathbf{x}_{t-1}^{s^*,\nu}) \tag{3.25}$$

$$Q(\mathbf{X}_t^{n-1}|\mathbf{X}_t^*) = Q(s^*)\frac{1}{N}\sum_{\nu=1}^{N} P(\mathbf{x}_t^{n-1}|\mathbf{x}_{t-1}^{s^*,\nu}) \tag{3.26}$$

Compte tenu des équations (3.19), (3.20), (3.21) (3.25), et (3.26), le taux d'acceptation (3.15) devient alors :

$$\alpha = min\left(1, \frac{P(\mathbf{Z}_t|\mathbf{X}_t^*)\left(\sum_{\nu=1}^{N} a_t^\nu P(\mathbf{x}_t^*|\mathbf{x}_{t-1}^{s^*,\nu})\right)\left(\sum_{\nu=1}^{N} P(\mathbf{x}_t^{n-1}|\mathbf{x}_{t-1}^{s^*,\nu})\right)}{P(\mathbf{Z}_t|\mathbf{X}_t^{n-1})\left(\sum_{\nu=1}^{N} a_t^\nu P(\mathbf{x}_t^{n-1}|\mathbf{x}_{t-1}^{s^*,\nu})\right)\left(\sum_{\nu=1}^{N} P(\mathbf{x}_t^*|\mathbf{x}_{t-1}^{s^*,\nu})\right)}\right) \tag{3.27}$$

Cette équation montre malheureusement que la simplification qui nous avait conduit à l'équation 3.16 dans le cas du filtre non marginalisé noté *FP MCMC*$_D$, n'est plus possible ici.
La marginalisation de l'espace peut être choisie en fonction du besoin : dans le cas du suivi multi-objet, l'approche la plus fréquente consiste à affecter un sous-espace à la description de l'état de chaque objet. La méthode, exposée dans l'échantillonneur de Metropolis-Hastings à propositions marginalisées MH_d, conserve alors ses performances, car on marginalise ainsi les propositions de mouvements des composantes de l'état dans des sous espaces de dimension modeste. Cette opération peut être vue comme une proposition de croisement génétique entre la particule courante et une prédiction générée à partir d'une des particules de la chaîne précédente. L'application typique de cet algorithme au suivi d'objet, consiste à choisir des vecteurs $\mathbf{x}_t^s$ décrivant chacun un objet, S est alors le nombre d'objets suivis. Pour d'autres applications, où les composantes sont indépendamment observables, on peut pousser la logique jusqu'au *FP MCMC*$_1$, l'indice 1 signifiant que les nouveaux échantillons sont proposés en effectuant des mouvements marginaux dans une seule dimension à la fois.

3.2.7 Conclusion

L'approximation d'une densité de probabilité par des techniques de Monte-Carlo peut être réalisée à l'aide de techniques manipulant les hypothèses de manière parallèle (filtre SIR), ou de techniques manipulant les hypothèses de manière chaînée (filtres MCMC). Les méthodes parallèles, plus populaires, ont également l'avantage d'être naturellement parallèlisables sur des ordinateurs multi cœurs. Par contre, leurs performances décroissent rapidement lorsque la taille du vecteur d'état augmente. Les méthodes chaînées sont plus récentes et permettent une exploration plus efficace dans des espaces de dimension élevée. Par contre, ces dernières ne sont pas naturellement parallélisables. La suite de ce chapitre aborde différents travaux de recherche, souvent liés à des applications concrètes, utilisant ces techniques d'exploration.

3.3 Suivi d'objets en utilisant des classifieurs

La première partie de ce manuscrit décrit les techniques basées apprentissage pour l'estimation d'état par vision. Certaines de ces techniques permettent d'apprendre le lien entre un modèle d'état et des observations visuelles. Dans un cadre de suivi séquentiel probabiliste, ce lien définit le modèle d'observation (voir figure 3.4 page 50). Nous proposons donc d'évaluer les performances d'une méthode de suivi de type filtre particulaire utilisant une fonction d'observation basée apprentissage.

3.3.1 Positionnement bibliographique

L'utilisation de méthodes basées apprentissage dans un contexte de suivi visuel a déjà fait l'objet de nombreux travaux. Dans (6), un SVM est appris pour lier le déplacement d'un objet avec la variation de son apparence. Cette méthode a été testée avec succès dans le cadre de suivi de véhicules avec un apprentissage effectué hors

ligne sur la catégorie véhicules (arrières de véhicules). Williams (119) utilise un RVM (relevant vector machine) (107), combiné à un filtre de Kalman pour suivre un objet dont l'apparence est apprise sur la première image de la séquence. Dans (77), Okuma propose une approche basée sur un filtre à particules qui utilise un détecteur de type Adaboost (114) dans la fonction de proposition du filtre afin de guider l'echantillonnage. L'observation s'effectue à l'aide d'histogrammes couleur. L'application résultante est capable de reconnaître et suivre de nombreux joueurs de hockey durant un match. Le but de nos travaux est d'étudier les performance d'une approche de type filtre à particules utilisant uniquement un classifieur comme fonction d'observation.

3.3.2 La méthode

Le principe de la méthode est illustré sur le synoptique de la figure 3.11. Le suivi proposé s'opère dans l'image ; le vecteur d'état est de dimension trois, composé des deux translations dans le plan image et d'un facteur d'échelle. La fonction d'observation est composée d'un classifieur de type Adaboost ou SVM, appris en fonction de la nature de l'objet à suivre. Le modèle d'évolution est une marche aléatoire.
Deux types d'applications peuvent être adressées à l'aide de cette approche. La première concerne le suivi d'une catégorie d'objets (piéton, véhicule, ...). Dans ce cas, les bases utilisées pour l'apprentissage seront identiques à celles employées pour des applications de détection ; c'est à dire composées d'exemple positifs et négatifs. Les positifs comprenant un ensemble d'objets de la classe à suivre. La deuxième application concerne le suivi d'un objet particulier. Dans ce cas, la base de positif est constituée d'une collection de vues de l'objet à suivre, sous différentes illuminations et poses.
Deux types de classifieurs ont été utilisés : les SVM et l'adaboost. Dans le cas du SVM, la sortie du classifieur est une marge dans l'espace des descripteurs. Dans le cas de l'adaboost, il s'agit d'un score. Or, dans un filtre à particules la fonction d'observation doit délivrer une probabilité de présence de l'objet, connaissant l'état. Il est donc nécessaire de convertir la sortie du classifieur en une probabilité. Pour cela, nous avons proposé une méthode basée apprentissage, issue de (82).

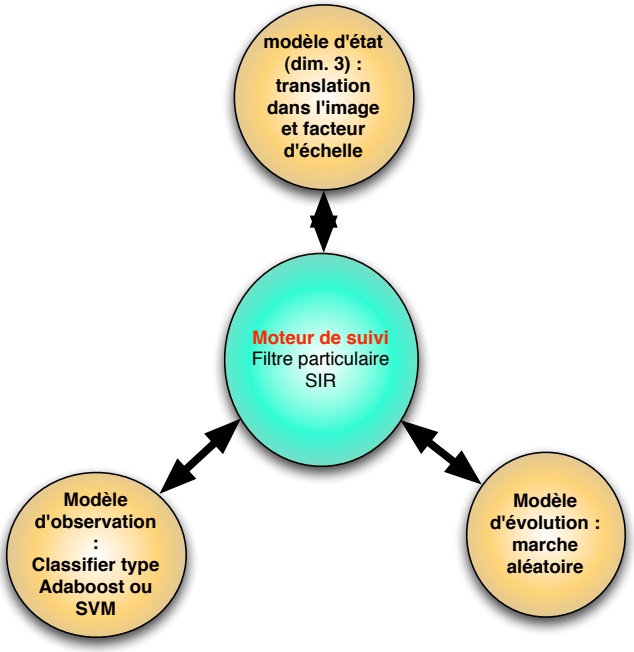


FIGURE 3.11 – Synoptique de l'algorithme de suivi d'objet proposé. Le moteur de tracking est constitué d'un filtre particulaire de type SIR, la fonction d'observation est issue d'un classifieur de type SVM ou Adaboost, et la fonction de prédiction est de type marche aléatoire.

Construire des probabilités calibrées. La fonction d'observation ($P(\mathbf{Z}|\mathbf{X})$[1]) définit la vraisemblance d'une mesure $\mathbf{Z}$, sachant un état $\mathbf{X}$. Or, dans de nombreuses méthodes de suivi d'objets, cette probabilité est calculée à partir d'une distance $d(.,.)$, par exemple en appliquant la formule $\exp(-\lambda.d(.,.))$. Le paramètre λ doit alors être ajusté de manière empirique. Nous proposons une méthode dans laquelle les paramètres sont appris, afin que les probabilités générées soient conformes à une distribution cible dont on connaît un ensemble de réalisations (données d'apprentissage). Soit $m(\mathbf{f})$ un classifieur générique qui retourne une valeur réelle non calibrée pour un vecteur de primitives $\mathbf{f}$. Cette valeur est une marge dans le cas d'une classifieur de type SVM ou un score dans le cas d'un algorithme adaboost :
On considère que le classifieur $m(\mathbf{f})$ possède une relation d'ordre entre les mesures fournies par ce dernier et les probabilités d'observer un objet associé : $m(\mathbf{f_1}) < m(\mathbf{f_2}) \rightarrow P(class|\mathbf{f_1}) < P(class|\mathbf{f_2})$. Généralement, $m(\mathbf{f}) \in [a_{min}; a_{max}]$ (où a_{min} et a_{max} dépendent du problème et du classifieur), et il s'agit de remettre à l'échelle les valeurs issues du classifieur dans l'intervalle $[0; 1]$. Si $m_r(\mathbf{f})$ est le score remis à l'échelle, la manière la plus simple de l'obtenir est d'appliquer : $m_r(\mathbf{f}) = (m(\mathbf{f}) - a_{min})/(a_{max} - a_{min})$. Néanmoins, l'estimation de $P(class|\mathbf{f})$ par $m_r(\mathbf{f})$ ne produira pas une distribution de probabilités calibrée correcte (voir (124) pour plus de détails).
Dans (74), trois méthodes utilisées pour la calibration de probabilités en sortie d'un algorithme de Boosting sont comparées : la corrrection logistique (34), la régression isotonique (124) et une méthode de mise à l'échelle proposée par Platt (82) qui fournit des probabilités *a postériori* à partir de la sortie d'un classifieur SVM. Cette dernière a également été utilisée dans (74) pour calibrer des scores en sortie d'un algorithme adaboost. Si $m(\mathbf{f})$ est la sortie du classifieur, un modèle de type sigmoïde est utilisé pour générer les probabilités calibrées :

$$P_{A,B}(m(\mathbf{f})) = (P(\text{positive}|m(\mathbf{f})) = \frac{1}{1 + \exp(A.m(\mathbf{f}) + B)} \tag{3.28}$$

où A et B sont calculés par une estimation de maximum de vraisemblance (MLE) à partir d'une base de calibration (m_i, y_i) ($m_i = m(\mathbf{f}_i)$ et $y_i \in \{0, 1\}$ sont des exemples positifs et négatifs. A et B sont calculés par une optimisation non linéaire du log vraisemblance sur les données d'apprentissage, à partir d'un critère d'entropie croisée :

$$(\hat{A}, \hat{B}) = \arg\min_{(A,B)} \left\{ -\sum_i y_i \log(P_{A,B}(m_i)) + (1 - y_i) \log(1 - P_{A,B}(m_i)) \right\}, \tag{3.29}$$

Les figures 3.12 (a) et (b) montrent les histogrammes pour $p(\mathbf{x}|y = \pm 1)$, sortie du classifieur SVM (fig.3.12(a)) et sortie de l'adaboost (fig.3.12(b)). Les fonctions de densité de probabilité estimées par ces deux histogrammes ne sont pas Gaussiennes. Les figures 3.12 (c) et (d) montrent la forme de la sigmoïde utilisée pour le calcul des probabilités calibrées, et calculées à partir de (a) et (b) avec le méthode de mise à l'échelle proposée par Platt. Ce sont les valeurs fournies en sortie de ces sigmoïdes qui sont utilisées comme vraisemblance dans le filtre à particules.

3.3.3 Résultats

La fonction d'observation décrite se base sur une description de l'image par des ondelettes de Haar, ces dernières présentant l'avantage d'être rapides à calculer moyennant l'utilisation d'une image intégrale. Contrairement à la plupart des méthodes de suivi, l'initialisation (positionnement de l'objet à suivre sur la première image) est ici traité de manière automatique et implicitement dans l'algorithme en échantillonnant régulièrement le filtre dans l'espace d'état.
Plusieurs expérimentations ont été menées :

- ▷ **Etude du bassin de convergence** : il s'agit d'étudier la forme de la courbe de la fonction donnant l'évolution de la sortie du classifieur en fonction du déplacement entre deux images. Cette courbe présente un maximum pour un déplacement nul et décroît de manière continue lorsque le déplacement augmente.

[1] les indices temporels sont ignorés pour alléger la notation

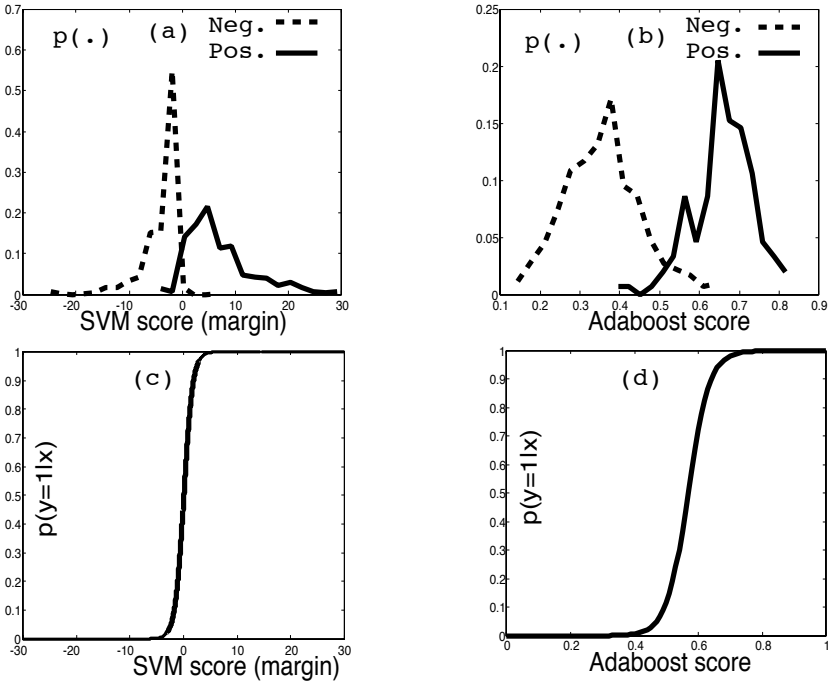


FIGURE 3.12 – Exemples de sigmoïdes utilisées pour calibrer la sortie des classifieurs. (a) et (b) sont les histogrammes pour $p(\mathbf{x}|y = \pm 1)$. La ligne pleine représente $p(\mathbf{x}|y = 1)$, et la ligne en pointillés $p(\mathbf{x}|y = -1)$. (c) et (d) sont les sigmoïdes calculées à partir de la méthode de Platt.

▷ **Illustration du suivi** : il s'agit, d'une part de comparer le comportement des deux types de classifieur testés (adaboost et SVM), dans un cas de suivi réel, et d'autre part, d'illustrer le fonctionnement de la méthode pour quelques séquences typiques (variation d'éclairement, occultation).

La figure 3.13 illustre la relation entre la sortie du classifieur et un mouvement de translation horizontal de l'objet autour de la fenêtre d'intérêt. Les deux courbes représentent deux objets différents : en trait plein, le dalton dont l'apprentissage a été effectué sur une base comportant plusieurs images de cet objet acquise en faisant varier la pose, l'arrière plan et les conditions d'éclairement ; et en pointillés, un piéton dont l'apprentissage a été effectué sur une base de piétons. D'une part, les courbes fournies par les deux types de classifieurs sont proches et possèdent de bonnes propriétés en terme de bassins de convergence. D'autre part, le comportement des courbes dans le cas d'un objet spécifique, et dans le cas d'une catégorie d'objets sont également proches. La figure 3.14 illustre le comportement des deux classifieurs (SVM et Adaboost) dans le cas du suivi d'un piéton. Les variations de la position horizontale (Fig. 3.14.a) et verticale (Fig. 3.14.b) du piéton au cours de la séquence sont présentées. L'analyse des courbes nous informe que les caractéristiques des deux classifieurs sont proches. La figure 3.15 illustre le comportement de l'algorithme dans le cas du suivi d'un objet spécifique. Il s'agit, ici, de détecter et de suivre un robot mobile en milieu extérieur. La base d'apprentissage est constituée d'un ensemble d'images du robot mobile acquises avec différentes caméras, depuis différents points de vue, pour différentes conditions d'éclairement et différents fonds. Le comportement de l'algorithme est satisfaisant, même lorsque l'objet à suivre est partiellement occulté.
La figure 3.16 illustre le comportement de l'algorithme dans le cas de changements d'apparences dus, par exemple à des modifications d'éclairement ou de pose de l'objet, face, profil, arrière.

3.3.4 Conclusion

Cette étude a permis de montrer que l'utilisation de classifieurs récents (l'adaboost ou les SVM) comme fonction de vraisemblance d'un filtre à particules permet d'obtenir des algorithmes de suivi capables d'apprendre, hors ligne, « ce qu'il faut suivre » . L'algorithme résultant est ainsi capable de suivre, soit une catégorie d'objet comme des piétons ou des véhicules, soit un objet particulier appris. Comme toutes les

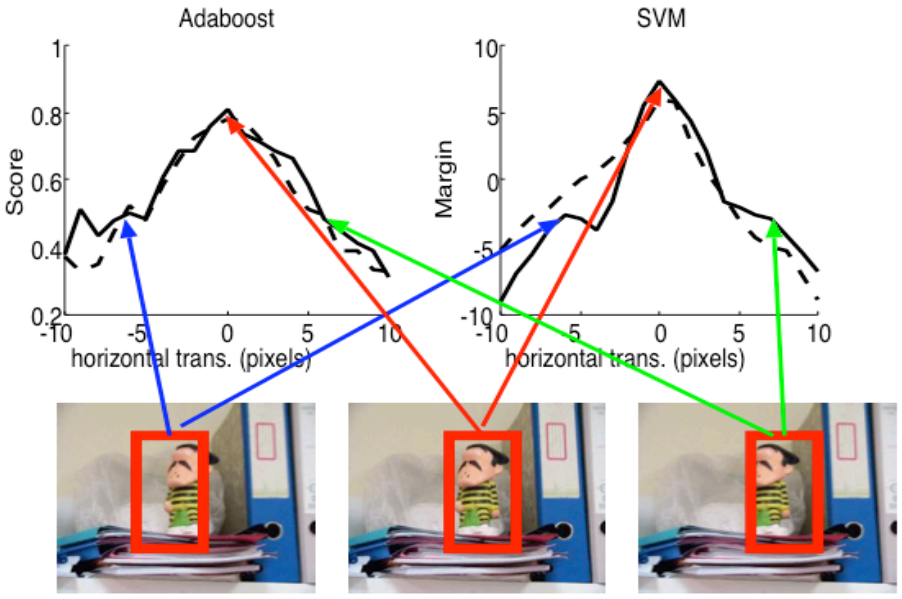


FIGURE 3.13 – Evolution de la sortie du SVM et de l'Adaboost en fonction de la translation horizontale pour le dalton (courbe continue) et une catégorie piéton (courbe pointillée).

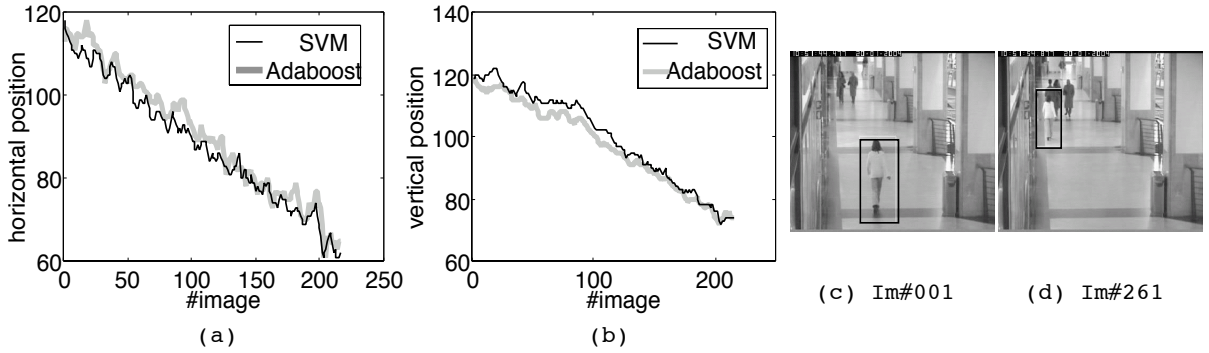


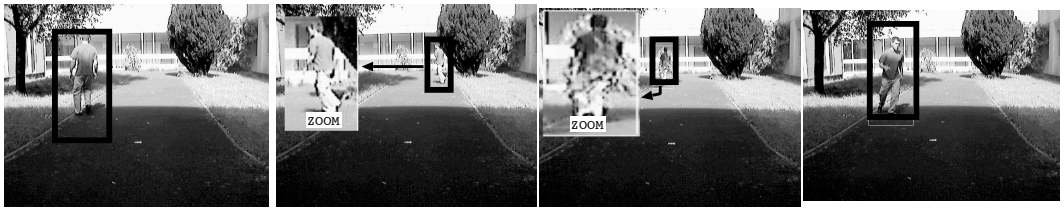
FIGURE 3.14 – positions horizontale (a) et verticale (b) estimées du piéton, pour un classifieur de type SVM et un classifieur de type Adaboost. (c) et (d) sont deux images illustrant la séquence vidéo utilisée dans cette expérience.

méthodes basées apprentissage, le choix de la base d'apprentissage influe de manière importante sur les performances de l'algorithme. D'autre part, la méthode proposée est peu gourmande en terme de temps de calcul (quelques ms) et accepte un fonctionnement temps réel sur des flux classiques allant jusqu'à soixante images par seconde.

3.3.5 Publications associées

1	**Real-time tracking with classifiers** T. Chateau, V. Gay-Belille, F. Chausse, and J. T. Lapresté. *WDV - WDV Workshop on Dynamical Vision at ECCV2006*, Grazz, Austria, May 2006

FIGURE 3.15 – Suivi d'un objet : l'algorithme a été entraîné à détecter et suivre un robot mobile en milieu extérieur. Le rectangle montre la position du robot détecté dans les images sélectionnées dans la séquence. Les images de la troisième ligne illustrent quelques exemples positifs utilisés lors de l'apprentissage.



(a) t=0.3s (b) t=15s (c) t=20s (d) t=38s

FIGURE 3.16 – Illustration du comportement de l'algorithme dans le cas du suivi d'un piéton avec des conditions d'éclairement variables (ombres) et différentes poses de l'objet (face, profil, arrière).

3.4 Suivi et catégorisation d'un nombre variables d'objets

Le suivi visuel en temps réel d'un nombre variable d'objets est d'un grand intérêt pour diverses applications. Au cours des dernières années, plusieurs travaux ont porté sur le suivi de piétons et de véhicules (125). Dans toutes ces applications le temps réel peut être requis, soit parce qu'une information immédiate est nécessaire, soit parce que l'enregistrement des images n'est pas autorisé, ou parce que la quantité de données est tout simplement trop énorme pour qu'elles soient enregistrées et traitées ultérieurement.
Pour pouvoir concevoir une méthode utilisable dans le cadre d'applications réelles comme la surveillance du trafic routier, il faut adresser plusieurs problématiques scientifiques :

- ▷ **Suivre un nombre variable d'objets.** Dans des applications de vidéo-surveillances, les méthodes doivent gérées le suivi de plusieurs objets, ainsi que l'apparition et la disparition des objets dans la scène.

- ▷ **Suivre des objets de catégories différentes.** Dans les applications de surveillance du trafic routier par exemple, les objets cibles appartiennent à des classes diverses, tels que les piétons, cycles, motocycles, véhicules légers, camions légers, ou semi-remorques. Le système de suivi doit donc gérer des cibles de tailles 3D variables, ainsi que des projections 2D de tailles variables, du fait des effets de perspective.

- ▷ **Gérer la variation des conditions d'éclairement et les ombres portées.** En extérieur, les variation d'éclairement et les ombres portées par les objets opaques sont des sources de perturbations pour les algorithmes de traitement d'images. Ces phénomènes doivent donc être pris en compte lors de la modélisation.

Nous adressons le problème du suivi et de la classification conjoints en temps réel d'un nombre variable d'objets par mono-vision en caméra statique. Le cœur de ce traqueur est basé sur un algorithme *FP RJ-MCMC* inspiré de (53; 121) et étendu pour suivre et classifier conjointement des objets et la source lumineuse. En outre, pour éviter la difficile tâche de segmentation de l'ombre, nous avons choisi de la modéliser et de l'inclure dans l'*avant-plan*. Nous nous appuyons alors sur une observation délivrée par une méthode basique de segmentation *arrière-plan/avant-plan*. Dans (59), l'ombre portée est modélisée en utilisant un modèle d'objet 3-D, et une position du soleil initialisée à la main. Nous étendons cette approche pour confier au *FP RJ-MCMC* l'estimation automatique et continuelle de la lumière du soleil, permettant ainsi le suivi à long terme en extérieur. La source de lumière est modelisée et mise à jour au fil du temps dans le filtre à particules, afin de gérer les changements d'illumination lentes mais intenses causées par les nuages et les changements de position du soleil.

3.4.1 Positionnement bibliographique

Ces travaux se positionnent dans la problématique du suivi temps réel d'un nombre d'objets variables, de classes variables, en milieu extérieur. De part la richesse des applications qui en découlent, cette problématique a été abondamment abordée ces quinze dernières années. Les principaux verrous scientifiques associés à cette problématique sont les suivants :

- ▷ **La modélisation de la cohérence temporelle.** Dans une stratégie de suivi en ligne, l'estimation de la position de l'objet suivi dans l'historique des images passées est connue lors de la recherche de sa position dans l'image courante. Cette connaissance est un *a priori* important qu'il faut utiliser au mieux dans l'algorithme de suivi. La plupart des méthodes considèrent que la position courante dépend uniquement de la position précédente. Dans l'algorithme *Meanshift* (25), une recherche de maximum local du critère est lancée, à partir d'une initialisation sur l'estimation de la position précédente. Dans des techniques de filtrage séquentiels, des modèles d'évolution plus complexes sont proposés, comme des modèles auto-régressifs (79) ou des modèles de type bicyclette dans le cas du suivi de véhicules (39).

- ▷ **La modélisation d'un nombre variable d'objets.** Le suivi d'un nombre variable d'objets nécessite, d'une part, de traiter les problèmes liés à la taille du vecteur d'état qui croit avec de manière proportionnelle avec le nombre d'objets à suivre, et d'autre part de gérer les entrées et sorties des objets. La méthode la plus classique consiste à séparer la phase de détection d'objets de la phase de suivi d'objets, le lien entre les deux phases étant assuré par une méthode dite d'association de données (10; 38; 51; 75; 87; 93; 112). Nous avons choisi d'orienter nos recherches sur une autre technique, pour laquelle tous les objets sont regroupés dans un vecteur d'état joint. L'estimation de la densité de probabilité *a posteriori* associée à cet état est assurée par une méthode de Monte-Carlo. Dans (46), un suivi de plusieurs objets à l'aide d'un filtre à particules est proposé. Dans les faits, ce suivi reste toutefois limité à 2 ou 3 objets, le nombre de particules nécessaires pour assurer une bonne exploration de l'état augmentant de manière exponentielle avec le nombre d'objets à suivre. Dans (52), une version chaînée du filtre à particules est proposée, permettant une stratégie d'exploration marginale, ce qui améliore considérablement la performance du filtre dans le cas où certains paramètres du vecteur d'état sont considérés indépendants. Cette méthode a ensuite été étendue dans le cas du suivi d'un nombre variable d'objets (53) (121). Nos travaux s'appuient sur cette méthode.

- ▷ **La modélisation d'objets de différentes classes.** Dans des applications de suivi réelles il est fréquent que différentes classes d'objets cohabitent dans la scène. En milieu autoroutier par exemple, on trouve à la fois des deux roues, des véhicules légers et des véhicules lourds. la détection et le suivi simultanés de ces différentes classes d'objets nécessitent la mise en oeuvre de stratégies adaptées. Nous proposons d'explorer de manière jointe la configuration dynamique de l'objet à suivre et de sa classe.

- ▷ **La prise en compte des phénomènes d'éclairement associés à un environnement extérieur.** Les ombres portées et les variations de conditions d'éclairement sont deux difficultés connues dans les application de vision par ordinateur en milieu extérieur. Toutefois, les ombres portées par les objets sont

également porteuses d'informations pertinentes sur l'objet lui-même, ce qui offre l'opportunité d'accroître son observabilité. Pour ces deux raisons, les ombres portées doivent être prises en compte pour améliorer le suivi des performances visuelles (92). Une étude comparative des algorithmes de détection d'ombre a été publié dans (85). Néanmoins, la segmentation de l'image en trois classes (arrière-plan, les objets, d'ombres portées par les objets) est une étape très difficile, ce qui a obligé les auteurs à incorporer des raisonnements spatiaux et temporels dans leurs méthodes de segmentation. Nos travaux reprennent ce raisonnement et proposent une modélisation de la position du soleil, qui est ensuite explorée de manière similaire aux autres objets.

3.4.2 La méthode

La figure 3.17 est un synoptique représentant les différents choix de modèles. Le moteur de suivi est un algorithme de type PF RJ-MCMC, permettant une exploration d'état sur des espaces de dimension variable. Le vecteur d'état permet de coder la configuration dynamique de chaque objet, sa classe, ainsi que la présence et la position du soleil. Le modèle d'observation est issue d'une extraction fond-forme. Le modèle d'évolution utilise un modèle bicyclette de robot dans le cas du suivi de véhicules.
Nous présentons la formalisation probabiliste séquentielle d'un problème de suivi multi-objets et sa résolution par une méthode de type filtre à particule issu d'une chaîne de Markov construite par une méthode de Monte-Carlo.

3.4.2.1 Vecteur d'état

Dans le contexte du suivi d'objet avec prise en compte des conditions d'éclairement, l'état du système encode à la fois la configuration dynamique des objets présents dans la scène et la configuration de l'illuminant (présence et position du soleil dans un contexte extérieur) : $\mathbf{X}_t^n = \{\mathbf{l}_t^n, J_t^n, \mathbf{x}_t^{j,n}\}$, $j \in \{1, ..., J_t^n\}$, où $\mathbf{l}_t^n = \{\xi_t^n, \phi_t^n, \psi_t^n\}$ définit une hypothèse d'illumination associée à la particule n à l'instant t, $n \in \{1, ..., N\}$, où N est le nombre de particules. Plus précisément, ξ_t^n est une variable binaire aléatoire modélisant la présence du soleil, et ϕ_t^n et ψ_t^n sont des variables aléatoires continues modélisant respectivement les angles d'azimut et d'élévation du soleil, comme illustré figure 3.18. Lorsque le soleil est visible, les ombres portées des objets sont alors visibles sur le sol et doivent être modélisées. J_t^n est le nombre d'objets visibles pour l'hypothèse n à l'instant t, et chaque objet j est défini par : $\mathbf{x}_t^{j,n} = \{c_t^{j,n}, \mathbf{p}_t^{j,n}, \mathbf{v}_t^{j,n}, \mathbf{a}_t^{j,n}, \mathbf{s}_t^{j,n}\}$. Afin de pouvoir suivre des objets de catégories différentes, la variable aléatoire discrète $c_t^{j,n}$ codant la catégorie de l'objet j est définie. Cette dernière appartient à l'ensemble *C={piéton, moto, véhicule léger, camionnette, véhicule lourd}* par exemple.
On suppose que les objets se déplacent sur un monde plan. Par conséquent, la position absolue d'un objet candidat j dans la particule n à l'instant t est définie par $\mathbf{p}_t^{j,n} = (x_t^{j,n}, y_t^{j,n}, \rho_t^{j,n})$, avec un centre de gravité de l'objet $x_t^{j,n}$ et $y_t^{j,n}$, et une orientation $\rho_t^{j,n}$. La vitesse de l'objet j et son accélération sont décrites par $\mathbf{v}_t^{j,n}$ et $\mathbf{a}_t^{j,n}$, respectivement l'amplitude et l'orientation. La forme de l'objet est modélisée par un parallélépipède dont les dimensions sont codées dans $\mathbf{s}_t^{j,n}$. On considère le soleil comme une source ponctuelle, ce qui permet de modéliser de manière simple l'ombre portée des objets sur le soleil.

Suivi d'un nombre variable d'objets Pour permettre aux objets d'entrer et de sortir de la scène, Khan et al. ont étendu l'algorithme *MCMC PF* présenté dans la section 3.2.6.1 page 60 afin de suivre un nombre variable d'objets. Dans ce but, l'échantillonnage est opéré par un algorithme *RJ MCMC* (Reversible Jump Markov Chain Monte Carlo) (41), qui est capable d'explorer des espaces d'état de dimension variable. Cet échantillonneur met en oeuvre une paire de mouvements réversibles *{entrée, sortie}* afin d'étendre la loi de proposition $q(\mathbf{X})$, et ainsi permettre de « sauter » dans un espace de dimension plus élevé ou moins élevé (53; 99). Cet échantillonneur délivre une approximation de $p(\mathbf{X}^*|\mathbf{Z}_{1:t})$. Le calcul du taux d'acceptation α nécessite l'évaluation de la loi de proposition $q(\mathbf{X})$ pour $\mathbf{X}^*$ et $\mathbf{X}_t^{n-1}$. Le calcul du taux d'acceptation dépend donc du mouvement proposé, comme montré dans (53). Afin d'améliorer la cohérence temporelle de la présence des objets, (53) propose une paire de mouvements supplémentaires : *{stay, quit}*. Le mouvement $Stay$ permet de récupérer un objet j présent dans au moins une des particules de la chaîne à $t-1$, et qui ne

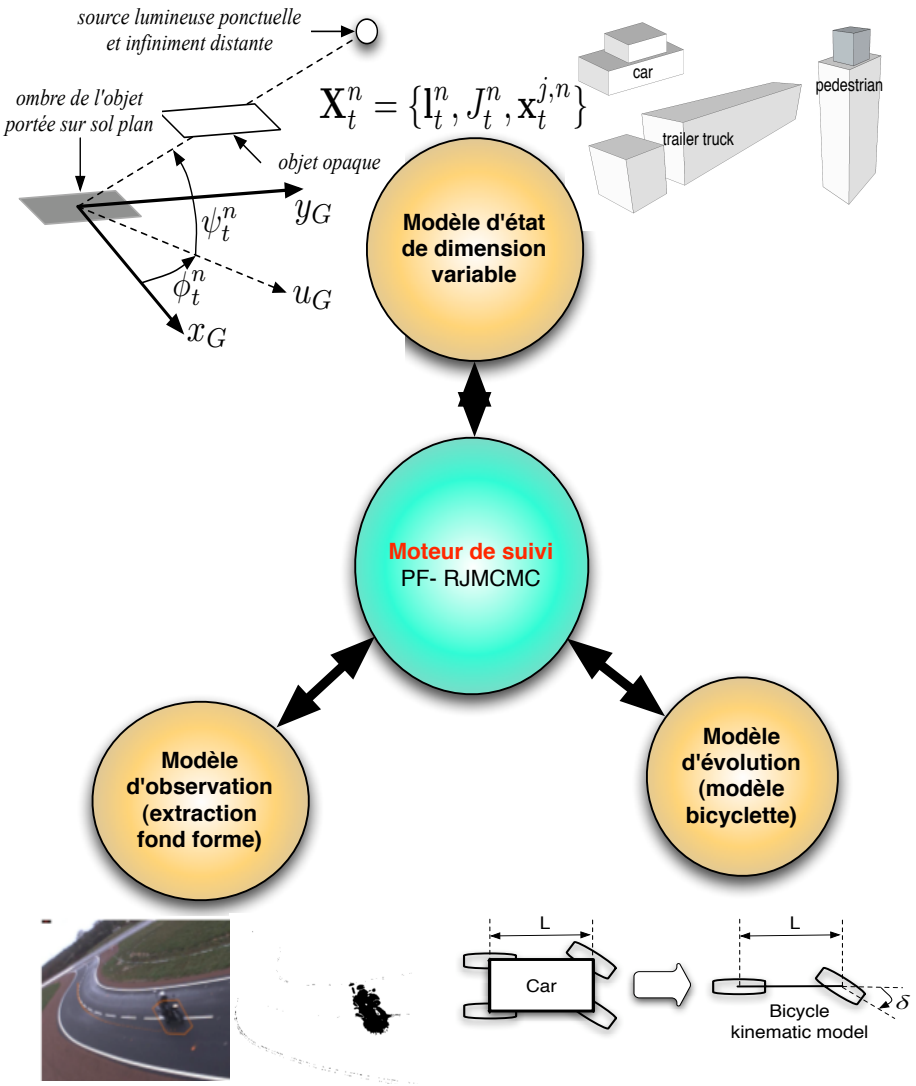


FIGURE 3.17 – Synoptique de l'algorithme de suivi d'objet proposé. Le moteur de tracking est constitué d'un filtre particulaire de type RJMCMC, la fonction d'observation est issue de la comparaison d'une carte fond forme image avec un modèle global de la scène. Le vecteur d'état décrit la scène, en terme de contenu multi-objets (multi-classes) et de conditions d'éclairement.

l'est pas dans la particule courante à l'instant t. Le mouvement $Quit$ propose qu'un objet j présent dans la particule courante de la chaîne à l'instant t, quitte la scène s'il n'était présent dans aucune des particules de la chaîne à $t-1$. L'expérimentation montre que cette paire de mouvements ne suffit pas à maintenir une cohérence temporelle suffisante de la présence des objets lorsque l'observation est mauvaise pendant plusieurs images consécutives. Pour cette raison, nous n'utilisons pas cette paire de mouvements, et introduisons une variable additionnelle : la *vitalité* de chaque objet, une variable intégrant les vraisemblances individuelles de chaque objet candidat au cours des itérations et du temps.

Nous introduisons également deux propositions de mouvements réversibles concernant le soleil et la catégorie des objets, ce qui conduit à l'ensemble de mouvements suivant : $\mathcal{M}$ *={entrée objet, sortie objet, mise à jour objet, entrée soleil, sortie soleil, mise à jour soleil}* notés $\{e, l, u, se, sl, su\}$. La categorie de l'objet est recherchée en proposant qu'elle change au sein de l'ensemble $\mathcal{C}$ *={piéton, moto, voiture, camionette, poids lourd}*, selon une matrice de transition. Cette proposition de saut de catégorie de l'objet étend les fonctionnalités du *RJ-MCMC PF* à la catégorisation des objets. Cette extension permet au *RJ MCMC PF* de

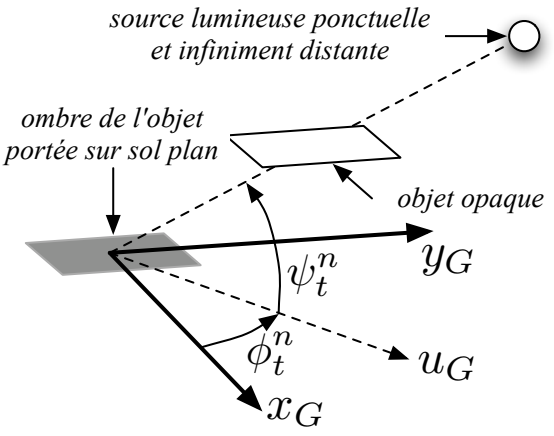


FIGURE 3.18 – Modèle utilisé pour le positionnement du soleil.

discriminer les catégories des objets sur la base de leur géométrie, mais aussi sur la base de leur dynamique. Ce point est très intéressant dans le cas de catégories d'objets présentant des dynamiques bien différentes, telles que semi-remorque contre voiture sur route sinueuse, ou comme un piéton contre un cycliste. En d'autre termes le fait d'integrer la catégorie de l'objet comme une variable aléatoire au sein du *RJ-MCMC PF*, permet à la dynamique de contribuer à la classification des objets, au même titre que leur géométrie.

Fonctions de propositions guidées par les observations Dans le but d'améliorer l'efficacité du filtre, le quota de propositions d'entrées ρ_e est guidé par les observations $\mathbf{Z}$ et la particule $\mathbf{X}_t^{n-1}$ à chaque itération en fonction de l'équation 3.38. De plus, $\rho_l(j)$, quota de sortie de chaque objet j, dépend d'un indicateur appelé vitalité, dont le calcul est détaillé dans (11). La mise à jour de la configuration d'un object j, la mise à jour du soleil, et les propositions d'entrée et de sortie du solail sont des valeurs constantes : $\rho_u(j) = 1$, $\rho_{su} = 0.1$, and $\rho_{se} = 0.02$. Les probabilités de chaque mouvement m, notées P_m, sont alors calculées à partir de ces quota, selon l'équation 3.30, où Jest le nombre d'objets candidats dans la particule $\mathbf{X}_t^{n-1}$.

$$P_m = \frac{\rho_m}{\rho_e + J\rho_u + \sum_{j\in\{1,.,J,s\}} \rho_l(j) + \rho_{se} + \rho_{su}}, \forall m \in \mathcal{M} \tag{3.30}$$

Entrée Objet : propose l'entrée d'un nouvel objet avec la probabilité P_e, produisant la nouvelle configuration jointe $\mathbf{X}^* = \{\mathbf{X}_t^{n-1}, \mathbf{x}^{j*}\}$. Cet objet reçoit un nouvel identifiant j, une catégorie et des dimensions initiales, et une vitalité initiale $\Lambda_t^j = \Lambda_0$. Le taux d'acceptation est calculé selon :

$$\alpha_e = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1}) P_l(j)}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) P_e Q(\mathbf{x}^{j*})}\right) \tag{3.31}$$

où l'objet $\mathbf{x}^{j*}$ est tiré selon la distribution de *faux fond* $\mathbf{I}_{fb}$ (équation 3.37), tel que sa projection coïncide sur l'amas de $\mathbf{I}_{fb}$.

Sortie de l'objet j : propose de supprimer l'objet j de $\mathbf{X}_t^{n-1}$ avec la probabilité $P_l(j)$, produisant la nouvelle configuration jointe $\mathbf{X}^* = \{\mathbf{X}_t^{n-1} \setminus \mathbf{x}_t^{j,n-1}\}$. Le taux d'acceptation est calculé selon :

$$\alpha_l = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1}) P_e Q(\mathbf{x}_t^{j,n-1})}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) P_l(j)}\right) \tag{3.32}$$

Mise à jour de l'object j : propose d'abord de changer la catégorie de $\mathbf{x}_t^{j,n-1}$ selon une matrice de probabilité de transition. Tire ensuite aléatoirement $\mathbf{x}_{t-1}^{j,r}$, une instance de l'object j décrite par la chaîne à $t-1$. Tire enfin $\mathbf{x}_t^{j*}$ de la loi dynamique $p(\mathbf{x}_t^j|\mathbf{x}_{t-1}^{j,r})$ relative à la catégorie de l'objet, et construit $\mathbf{X}^* = \{\mathbf{X}_t^{\setminus j,n-1}, \mathbf{x}_t^{j*}\}$.

$$\alpha_u = min\left(1, \frac{\pi^* w^*}{\pi_t^{n-1} w_t^{n-1}}\right) \tag{3.33}$$

Entrée du soleil : propose que le soleil devienne brillant avec la probabilité P_{se}. Le taux d'acceptation est calculé selon :

$$\alpha_{se} = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1}) P_l(s)}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) P_{se}}\right) \tag{3.34}$$

Sortie du soleil : propose que le soleil soit voilé par des nuages avec la probabilité $P_l(s)$. Le taux d'acceptation est calculé selon :

$$\alpha_{sl} = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1}) P_{se}}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) P_l(s)}\right) \tag{3.35}$$

Mise à jour de position du soleil avec probabilité P_{su}. Tire aléatoirement $\mathbf{l}_{t-1}^r$, une instance de position du soleil décrite par la chaîne à $t-1$. Tire $\mathbf{l}^*$ des lois dynamiques du soleil (3.44) et (3.45). Le taux d'acceptation est calculé selon 3.33.

3.4.2.2 Fonction de vraisemblance de l'observation

Dans cette section, nous calculons $P(\mathbf{Z}|\mathbf{X})$ la probabilité d'observation $\mathbf{Z}$, compte tenu de la configuration jointe multi-objet $\mathbf{X}$. Bien que la méthode autorise l'utilisation de plusieurs caméras, les notations de cette section n'en prendront qu'une en compte, par soucis de simplicité. A partir de l'image actuelle (Fig.3.19-a), et d'un modèle de fond (fig. 3.19-b), une image d'*avant-plan* binaire $\mathbf{I}_F(g)$ est calculée, telle qu'illustrée par la Fig.3.19-d , où g désigne un pixel. Nous utilisons l'algorithme $\Sigma - \Delta$ de (1), qui calcule et met à jour en ligne un modèle d'*arrière-plan* comme approximation de la médiane et de la covariance temporelles de l'image, qui permet de faire face aux variations intenses d'éclairement rencontrées en extérieur. D'autre part, chaque objet candidat de la particule $\mathbf{X}$ est modélisé par un parallélépipède rectangle ayant la forme définie dans la section 3.4.2.1. L'enveloppe convexe des sommets de sa projection est calculée. Si la lumière solaire est directe, l'enveloppe convexe des sommets de son ombre est également calculée. Une image masque binaire $\mathbf{I}_M(g, \mathbf{X})$ est calculée, où le pixel G vaut 1 s'il est à l'intérieur d'au moins l'une des enveloppes convexes, sinon à 0, comme illustré par la Fig. Fig.3.19-c. l'image de similarité $\mathbf{I}_S(g, \mathbf{X})$ est alors calculée (3.36), ainsi que l'image de *faux arrière-plan* (équation 3.37) utilisée pour guider les propositions d'objet selon (3.38), où S_o est la superficie a priori de la projection d'un objet entrant.

$$\mathbf{I}_S(g, \mathbf{X}) = \begin{cases} 1 \text{ if } \mathbf{I}_F(g) = \mathbf{I}_M(g, \mathbf{X}), & \forall g \\ 0 \text{ sinon} \end{cases} \tag{3.36}$$

$$\mathbf{I}_{fb}(g, \mathbf{X}) = \mathbf{I}_F(g) \ \& \ \overline{\mathbf{I}_M(g, \mathbf{X})}, \forall g \tag{3.37}$$

$$\rho_e = \frac{1}{S_o} \sum_g \mathbf{I}_{fb}(g, \mathbf{X}) \tag{3.38}$$

La vraisemblance de l'observation $P(\mathbf{Z}|\mathbf{X})$ est calculée selon (3.39) :

$$p(\mathbf{Z}|\mathbf{X}) = \left(\frac{1}{S} \sum_g \mathbf{I}_S(g, \mathbf{X})\right)^{\beta_j}, \tag{3.39}$$

où β_j est calculé en fonction de la superficie de la projection de l'objet j. Cette méthode est d'un grand intérêt car elle produit une probabilité d'observation qui permet de suivre équitablement les objets quelle que soit leur distance et leur masquage par d'autres objets. Ces deux besoins sont requis par les applications de vidéo-surveillance, comme la surveillance routière ou de lieux publics tels que les stations de métro, où les caméras ne peuvent pas être situées sur un point très élevé, ce qui génère de fortes occultations et de forts changements d'échelle dus à la projection. En outre, cette méthode accroît la performance du *FP MCMC* car elle offre un asservissement du taux d'acceptation α sur une valeur cible.

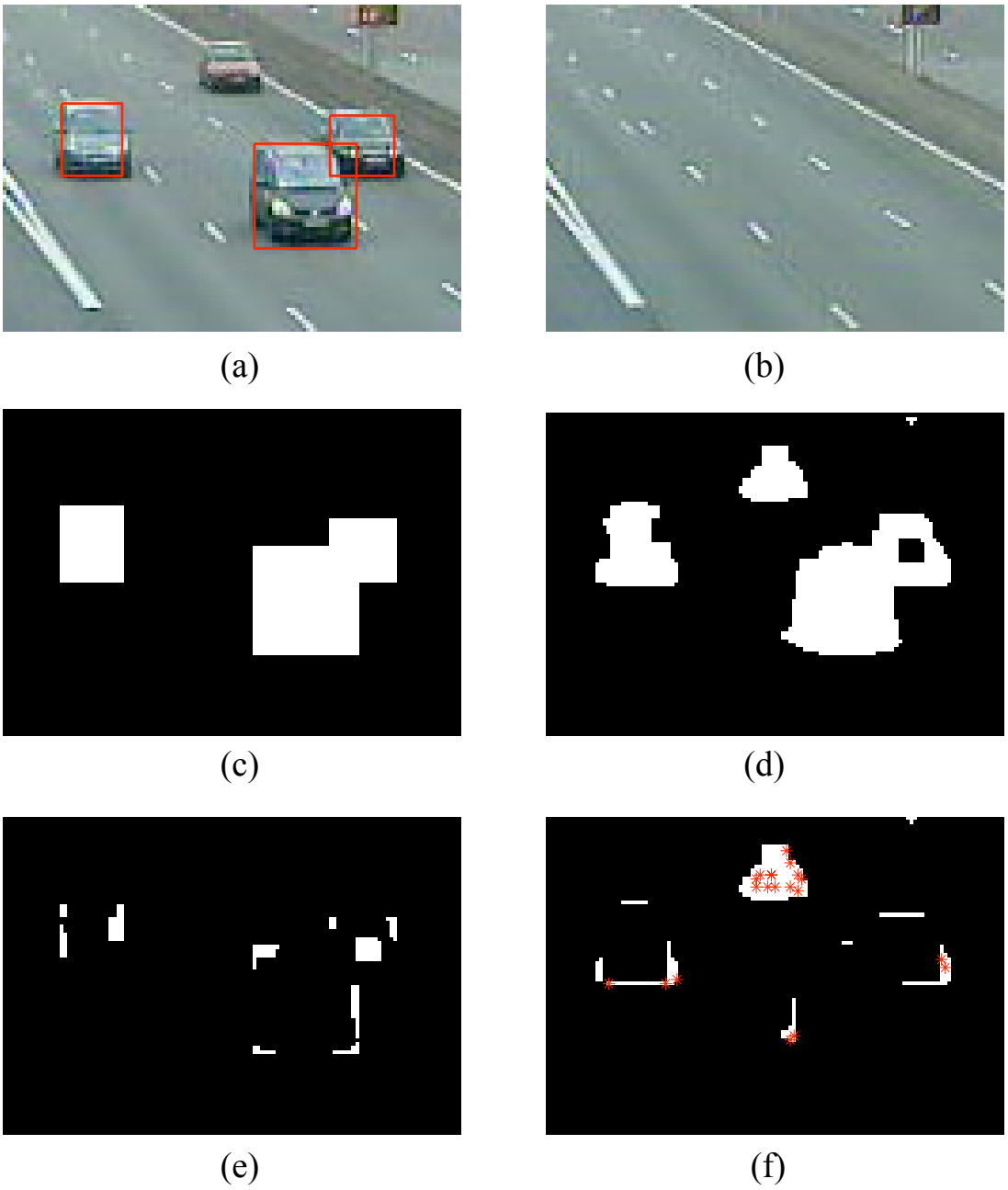


FIGURE 3.19 – Segmentation de l'avant-plan et images résiduelles, avec projection des objets candidats. Pour plus de lisibilité, leurs polygones sont approximés par des rectangles. (a) : images brutes avec rectangles estimant l'objet. (b) : modèle de fond. (c) : image binaire hypothèse $\mathbf{I}_M(g, \mathbf{X})$. (d) : image binaire d'avant-plan $\mathbf{I}_F(g)$. (e) : image binaire de *faux avant-plan*, i.e. pixels couverts par la projection d'au moins un objet, mais classifiés comme *arrière-plan*. (f) : image binaire de *faux arrière-plan* $\mathbf{I}_{fb}(g, \mathbf{X})$, i.e. pixels couverts par aucun objet candidat, mais classifiés comme *avant-plan*. Quelques points sont tirés (étoiles rouges), pour guider les propositions d'entrée d'un nouvel objet.

3.4.2.3 Poids d'Interaction Multi-Objet

Comme la fonction de vraisemblance permet à un objet entièrement masqué de survivre, nous devons l'empêcher de rester «coincé» derrière l'objet qui le masque. Pour le suivi de piétons, (121) propose d'utiliser

une distance de Mahalanobis plutôt qu'une distance euclidienne pour modéliser les distances entre piétons. Nous calculons également un poids inter-objet basé sur cette distance de Mahalanobis. Une distance anisotrope est encore plus nécessaire dans le cas du suivi de véhicules, parce que leurs longueurs sont beaucoup plus grandes que leurs largeurs, et leurs interactions sont aussi fortement anisotropes : 2 véhicules proches sont plus susceptibles de rouler sur 2 voies de circulation adjacentes plutôt que sur la même voie. En outre, l'interaction entre deux véhicules est fonction de leurs dimensions. Ceci est modélisé par le calcul de poids d'interaction d'objets w comme fonction d'une distance anisotrope entre chaque paire d'objets candidats. Ces deux conditions sont remplies en approximant chaque objet par une distribution gaussienne bivariée de masse, dont la matrice de covariance comporte les moments d'ordre 2 de la masse. La distance inter-véhicules est alors : $d_{ij} = (\mathbf{\Delta}_{ij}^T.(\mathtt{C}_i.\mathtt{C}_j)^{-1}.\mathbf{\Delta}_{ij})^{1/2}$, où $\mathbf{\Delta}_{ij}$ est le vecteur des différences de positions 2D entre les objets i et j sur le plan du sol, $\mathtt{C}_i$ et $\mathtt{C}_j$ sont leurs matrices de covariance respectives. Le poids d'interaction de la paire d'objets est alors calculé selon l'équation (3.40) :

$$w_{ij} = \left(1 + e^{-k_s.(d_{ij}-d_s)}\right)^{-1}, \tag{3.40}$$

produisant un poids proche de 1 pour les objets éloignés, et proche de 0 pour des objets matériellement trop proches. d_s est la distance inter-objets correspondant au paramètre d'inflexion de la sigmoïde et K_S permet d'ajuster la pente de la courbe autour de d_s. Le poids d'interaction de la particule $\mathbf{X}$ impliquant J_t^n objets est alors :

$$w(\mathbf{X}) = \prod_{i=1}^{J_t^n-1} \prod_{j=i+1}^{J_t^n} w_{ij}. \tag{3.41}$$

3.4.3 Résultats

Nou décrivons ici les tests réalisés afin de valider l'apport des contributions proposées, à la fois sur des données de synthèse et des données réelles.

3.4.3.1 Données utilisées et méthodologie

Les performances de la méthode proposée ont été évaluées à la fois sur des séquences de synthèse et des séquences réelles. Des bases de données provenant de plusieurs types d'applications ont été récoltées : suivi de piétons et suivi de véhicules dans un contexte autoroutier. Les expérimentations en suivi de piétons ont pour but de valider la capacité de la méthode dans le cadre du suivi d'un nombre d'objets supérieur à 10, tout en gérant les changements d'illumination dus à l'apparition ou la disparition du soleil. Les expériences mettant en scène le suivi de véhicules sur autoroutes ont pour but de tester le fonctionnement simultané du suivi et de la catégorisation des véhicules, tout en gérant également les changements d'illumination. De plus, comme la méthode doit prendre en compte des données bruitées (compression pas exemple), les séquences réelles utilisées sont issues de caméras bas coût compressées. de résolution 320×240 pixels. De plus, la calibration de ces caméras est effectuée de manière approximative, par une heuristique supervisée. Les objets à suivre sont localisés à l'intérieur d'une zone de suivi sélectionnée (définie en 3D et affichée en vert dans les figures 3.20, 3.21 et 3.22). Nous proposons d'évaluer la méthode proposée selon quatre critères :

- ▷ Pourcentage d'objets correctement suivis $\theta_T = \frac{1}{J_t}\sum_{t,j}\delta_T(t,j)$ avec $\delta_T(t,j) = 1$ si l'objet j est correctement suivi à l'instant t, sinon 0. $J_t = \sum_t j_t$, avec j_t le nombre d'objets dans la zone de suivi.
- ▷ Pourcentage d'objets correctement catégorisés $\theta_C = \frac{1}{J_t}\sum_{t,j}\delta_C(t,j)$ où $\delta_T(t,j) = 1$ si la catégorie de l'objet j est correcte à l'instant t, sinon 0.
- ▷ Pourcentage d'objets fantômes $\theta_G = \frac{1}{J_t}\sum_{t,j}\delta_G(t,j)$ où $\delta_G(t,j)$ est le nombre de *fantômes* i.e. hypothèses d'objets qui ne correspondent pas à un objet réel.
- ▷ Erreur moyenne d'estimation de la position $\varepsilon_T = \frac{1}{J_t}\sum_{t,j}(\boldsymbol{\delta}_p^T.\boldsymbol{\delta}_p)^{-1}$, avec $\boldsymbol{\delta}_p = \mathbf{p}_t^{j,e} - \mathbf{p}_t^{j,gt}$, où $\mathbf{p}_t^{j,e}$ est la position de l'objet j à l'instant t, $\mathbf{p}_t^{j,gt}$ est la j position réelle de l'objet.

3.4.3.2 Implémentation

Deux propositions sont calculées en parallèle sur chaque coeur de processeur. L'implémentation de la méthode a été réalisée en utilisant la librairie NT^2 C++[2]. De plus, nous utilisons un processeur Intel E6850 Core 2 Duo cadencé à 3Ghz et muni de 4Go RAM, sous Linux. Toutes les expérimentations présentées ici ont été effectuées à la cadence de la vidéo , soit (i.e. 25 fps), en monovision sur des images de taille 320×240. Le filtre à particules utilisé comporte $N = 200$ particules.

3.4.3.3 Suivi de piétons dans des conditions d'illumination variables

Les séquences vidéos utilisées lors de ce test sont issues de caméras de vidéo surveillances, dans un contexte de suivi de piétons. La dynamique associée aux piétons est de type vitesse constante :

$$p(\mathbf{v}_t|\mathbf{v}_{t-1}^r) = \mathcal{N}\left(\mathbf{v}_{t-1}^r, diag\left(\left[\sigma_m^2, \sigma_a^2\right]\right)\right), \tag{3.42}$$

où σ_m et σ_a sont respectivement l'écart type du bruit associé à l'amplitude de la vitesse et à son orientation. La forme est mise à jour par l'équation (3.43), où σ_s est l'écart type associé à la forme, et I_3 une matrice identité de taille 3×3. La dynamique associée au soleil est définie dans (3.44) et (3.45), où σ_ϕ et σ_ψ sont respectivement l'écart type de l'angle d'azimut et d'élévation.

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}^r) = \mathcal{N}(\mathbf{s}_{t-1}^r, \sigma_s^2 I_3) \tag{3.43}$$

$$p(\phi_t|\phi_{t-1}^r) = \mathcal{N}(\phi_{t-1}^r, \sigma_\phi^2), \forall r \in \{1, ..., N\} \tag{3.44}$$

$$p(\psi_t|\psi_{t-1}^r) = \mathcal{N}(\psi_{t-1}^r, \sigma_\psi^2), \forall r \in \{1, ..., N\} \tag{3.45}$$

Séquences de synthèse : des piétons, approximés par des cubes se déplacent de manière aléatoire sur un sol plan, dans une zone de tracking de taille 12x15m, sous un soleil dont la position évolue avec le temps, d'élévation $\psi = 0.8$ *rad* et d'azimut croissant de $\phi = 0$ à $\phi = \pi$ *rad* sur une durée de 1000 images. C'est une vitesse bien supérieure au déplacement réel du soleil. La figure 3.20 illustre le suivi, montrant le bénéfice de la modélisation de l'ombre. Le tableau 3.1 présente les résultats obtenus pour les méthodes MOT et MOTS, et montre que la modélisation du soleil diminue le taux de fantômes et augmente la précision du suivi.

TABLE 3.1 – Performances de la méthode pour le suivi de piétons dans des séquences de synthèse avec des conditions d'éclairement variables (position et présence de soleil, changeant toutes les 200 images).

	avec soleil		**sans soleil**	
	MOT	MOTS	MOT	MOTS
θ_T **(%)**	84.7	89.7	84.1	87.0
θ_G **(%)**	5.7	4.9	5.1	4.3
error (m)	0.91	0.63	0.82	0.70

Séquences réelles : nous disposons d'une séquence assez courte, mais comportant des apparitions/disparitions rapides de soleil. La figure 3.21 image #786 montre trois piétons suivis alors que le soleil est absent. Quelques images plus tard, le soleil se montre ; il est détecté à l'image #823 et il reste présent jusqu'à la fin. La méthode commet une erreur d'estimation lors du suivi des deux piétons, qui s'occultent sur le reste de la séquence. Les deux personnes sont suivies comme un unique piéton (#14), à cause de la faible observabilité liée à cette configuration.

[2]Numerical Template Toolbox. http ://nt2.sourceforge.net

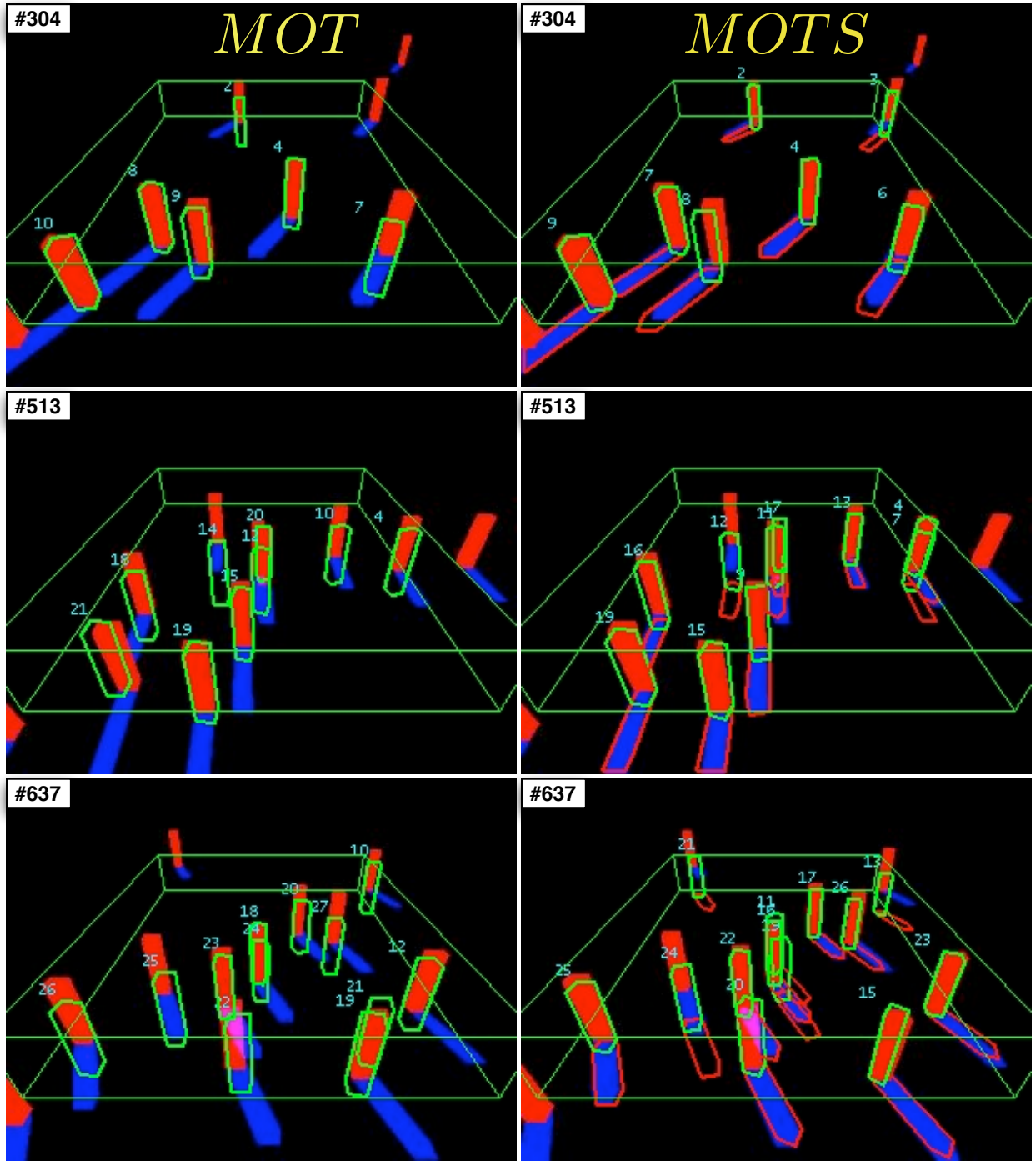


FIGURE 3.20 – Illustration du suivi de piétons sur une séquence de synthèse, avec des conditions d'éclairement variables. Les poitions des objets suivis sont reprojetés dans l'image en vert. Colonne de gauche : sans modélisation de l'ombre dans le processus de suivi. Colonne de droite : ombre portée reprojetée en rouge. Illustration en couleur

3.4.3.4 Suivi et classification de véhicules

Ces expériences visent à évaluer la capacité du traqueur à suivre et classer simultanément les véhicules comme voitures, camionettes et poids lourds. La dynamique des véhicules candidats est contrôlée par des propositions

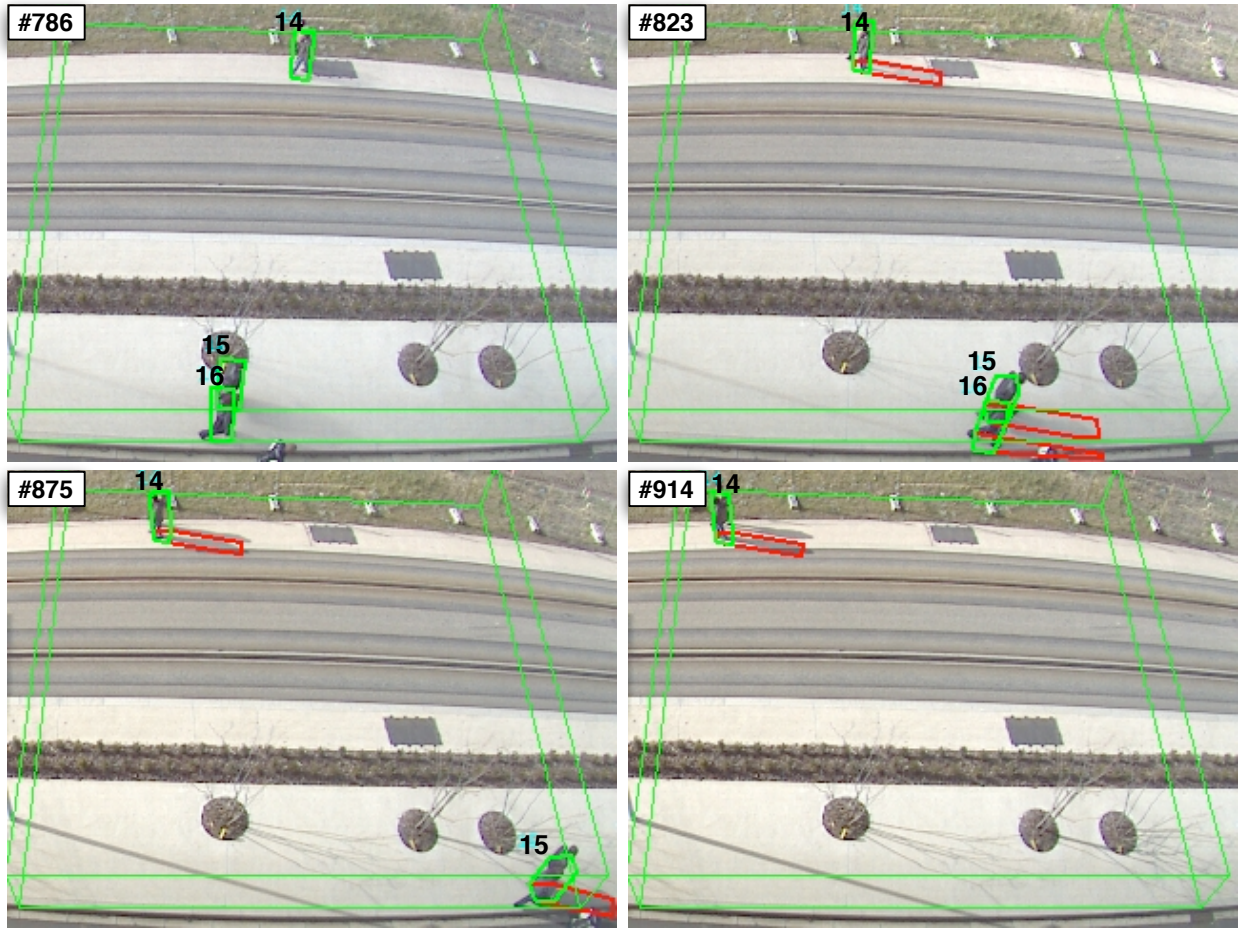


FIGURE 3.21 – Illustration du suivi de piétons dans des conditions variables d'éclairement. Les personnes suivies sont entourées d'une boite englobante verte l'ombre estimée associée à ces personne est entourée d'une boite englobante rouge.

TABLE 3.2 – Performances du suivi et de la classification dans le cas de deux classes. Taux de suivi θ_T (%) / Taux de classification θ_C (%) / Taux de fantômes θ_G(%). Erreur de position moyenne par véhicules en mètres.

	MOT	MOTS	MOTC^2	MOTC^{2}S
light vehicles	.	.	59/54/0	**90/89/11**
trailer trucks	.	.	86/86/0	**90/89/0**
total	52/22/17	51/25/16	58/53/0	**90/89/11**
error (m)	6.17	5.80	2.76	**2.00**

de commande du conducteur établies à partir de (3.46) :

$$p(\mathbf{a}_t|\mathbf{a}^r_{t-1}) = \mathcal{N}\left(0, diag\left(\left[\sigma_l^2, \sigma_t^2\right]\right)\right), \tag{3.46}$$

où σ_l est l'écart-type d'accélération longitudinale demandée par le conducteur, σ_l est l'écart-type d'angle de braquage, conditionnant l'accélération transversale. Les équations d'un modèle bicycle sont ensuite appliquées à l'objet j. Les lois dynamiques donnent alors la vitesse $\mathbf{v}^*_t$ et la position $\mathbf{x}^*_t$.

Séquences de synthèse : elles impliquent des paréllélépipèdes approximant des voitures et des camions sur une autoroute à trois voies, en plein soleil. Le tableau 3.2 résume les résultats, montrant que la classification et la modélisation de l'ombre améliorent toutes deux le suivi. Les meilleurs résultats sont atteints lorsque les deux sont activés.

Séquences réelles : les séquences de trafic réel impliquent des véhicules légers, des camionettes et des poids lourds sur une autoroute à quatre voies, incluant une voie d'entrée autoroute, sous soleil variable. Pour le suivi du trafic réel, une classification en 3 classes est nécessaire afin de prendre en compte les 3 principales catégories de véhicules. En raison de la grande disparité des dimensions des véhicules, les méthodes sans classification (MOT et MOTS) exigent une forte dynamique de la forme du véhicule σ_s. Une telle stratégie ne peut pas fonctionner en présence de fortes occultations. Pour qu'elles puissent néanmoins servir de référence, ainsi que pour permettre d'établir manuellement la vérité terrain, nous avons choisi une séquence avec un trafic faible, mais impliquant toutes les catégories de véhicules. La figure 3.22 illustre le fait que le classement selon plusieurs catégories et la modélisation d'ombre améliorent le suivi. Les échecs typiques du MOT et du MOTS sont : deux objets candidats poursuivant une cible unique (MOTS) et l'imprécision du suivi (MOT). Sans modélisation des ombres, le système peine à suivre des objets de tailles très différentes : il explique l'ombre portée d'un camion par une voiture *fantôme* (# 7 sur MOTC^3 et #8 sur MOTC^{3}S). La modélisation des ombres portées explique ces pixels d'*avant-plan* (MOTC^{3}S). En outre, les voitures lointaines sont suivies avec plus de précision quand l'ombre est modélisée (MOTS et MOTC ^{3}S), car leur ombre fournit des indices quant à leur position longitudinale. Table 3.3 résume les résultats et confirme les analyses de la section 3.4.3.4 : la classification et la modélisation de l'ombre améliorent le suivi, avec de meilleurs résultats lorsque les deux sont activés.

3.4.4 Conclusion

Nous avons proposé un système générique permettant de suivre et de classifier en temps réel un nombre variable d'objets soumis à une illumination variable. Le système peut être exploité en mono-vision ou en multi-vision. Les fonctionnalités de suivi et de classification des objets, ainsi que l'estimation de la source d'éclairement sont prises en charge conjointement par un Filtre Particulaire *RJ-MCMC*. Pour ce faire, l'éclairage est intégré dans l'espace d'état de la configuration, et suivi au même titre que les objets. Les expériences montrent que l'objet et l'estimation conjointe du soleil améliore le suivi, diminuant à la fois les faux positifs et l'erreur d'estimation de la position des objets. Nous avons également proposé d'inclure la catégorie d'objets comme une variable aléatoire discrète estimée par le filtre *RJ-MCMC*, ce qui lui permet de

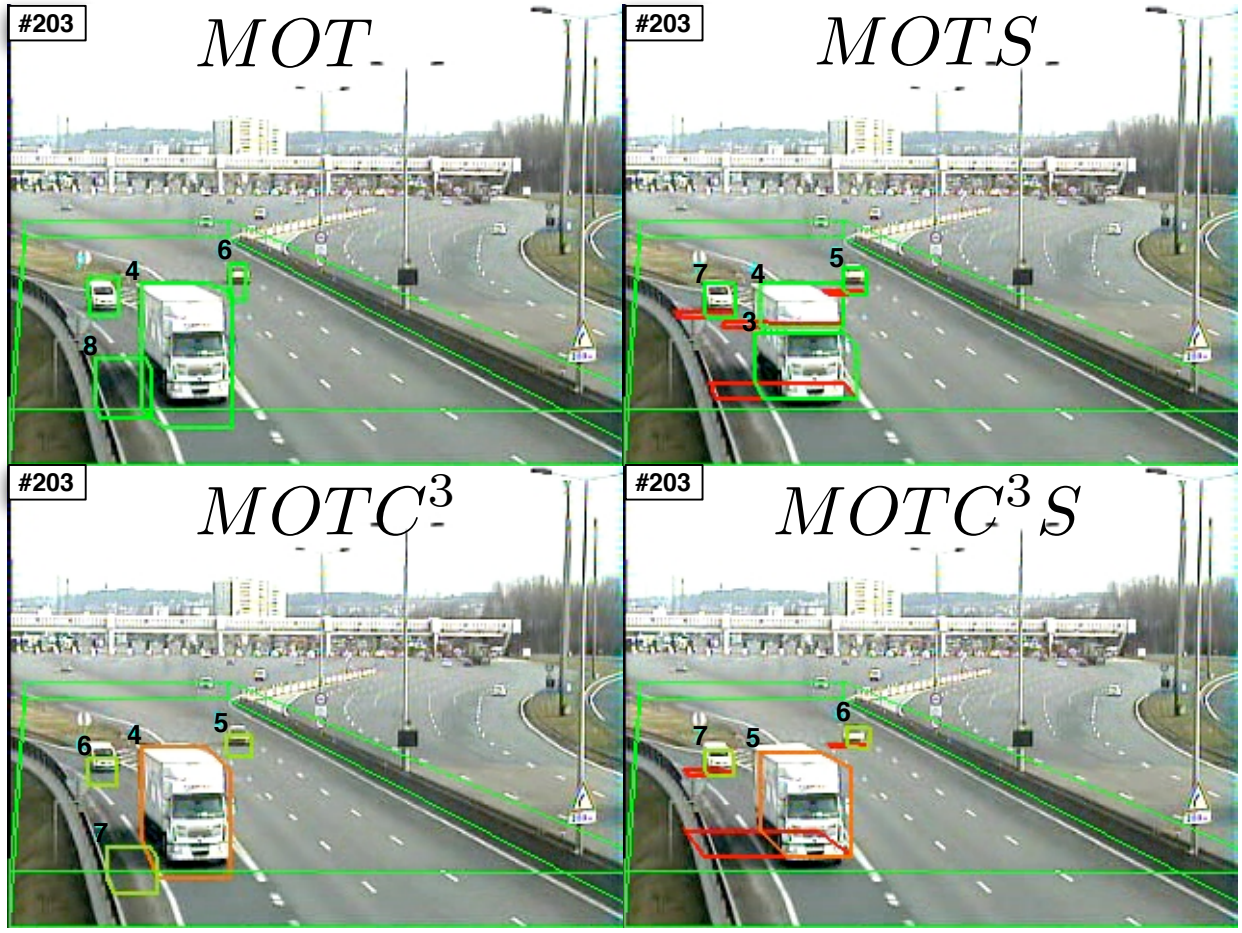


FIGURE 3.22 – Image #203 illustrant le suivi d'une séquence réelle. Ligne du haut : pas de classification, les véhicules détectés sont reprojetés en vert. Ligne du bas : classification en trois catégories en fonction du type de véhicule (voiture, véhicule utilitaire et poids lourd) les véhicules détectés sont reprojetés en vert (resp. magenta et orange). A gauche : sans modèle de soleil. A droite : avec un modèle de soleil (en rouge). Illustration en couleur

	MOT	MOTS	MOTC^3	MOTC^{3}S
light vehicles	.	.	67/64/2.6	**67/67/0.05**
light trucks	.	.	83/36/1.0	**92/86/3.7**
trailer trucks	.	.	93/83/0	**100/100/2**
total	51/45/0	60/51/0	72/62/2.5	**70/70/3.1**
error (m)	6.80	6.22	6.13	**5.40**

TABLE 3.3 – Performances de la méthode dans le cas du suivi et de la classification sur une vidéo de trafic autoroutier (3 classes). Taux de suivi θ_T (%) / Taux de classification θ_C (%) / Taux de fantôme θ_G(%). Erreur de position moyenne par véhicules (en mètres). Illustration en couleur.

classifier les objets conjointement à leur suivi. Les expériences montrent que cette stratégie améliore le suivi, car elle propose plusieurs modèles géométriques, permettant ainsi un meilleur ajustement du modèle. Cette approche unifiée est également d'un grand intérêt car elle permet d'appuyer la classification sur la dynamique propre à chaque classe d'objets. Cette fonctionnalité peut être utilisée pour améliorer le suivi et la classification d'objets ayant des géométries proches mais des dynamiques différentes, tels que les cyclistes et les piétons par exemple. Comme ce traqueur est conçu pour être générique, il est basé sur des informations de bas niveau (segmentation arrière-plan simple), et sur des données d'acquisition de faible qualité. En fonction des applications, les performances seront incontestablement améliorées par l'ajout de données a priori sur les objets et la scène, notamment les zones d'entrée et de sortie des objets. Le travail présenté dans ce document considèrent une source d'éclairage unique, bien adapté au modèle d'éclairement solaire. Il peut facilement être étendu à de multiples sources d'éclairage et à la modélisation des réflets sur le sol, nécessités pour l'éclairage intérieur ou extérieur par temps humide.

1 **Mcmc particle filter for real-time visual tracking of vehicles**
F. Bardet et T. Chateau
In 11th International IEEE Conference on Intelligent Transportation Systems, Beijing, China, octobre 2008

2 **Real time multi-object tracking with few particles**
F. Bardet et T. Chateau
In Visapp, International Conference on Vision Theory and Applications, Lisbonne, Portugal, Fevrier 2009

3 **Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous tracking and classification**
F. Bardet, T. Chateau, and D. Ramadasan
In ICCV 2009, International Conference on Computer Vision, Tokyo, Japon, Septembre 2009

4 **Unifying real-time multi-vehicle tracking and categorization**
F. Bardet, T. Chateau, and D. Ramadasan
In Intelligent Vehicle Symposium, Xi'an's, Chine, Juin 2009

5 **Performances comparées de rééchantillonnage pour filtres de monte-carlo**
F. Bardet et T. Chateau
RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle,Amiens, France, Janvier 2008

6 **Suivi et classification visuels temps réel d'un nombre variable d'objets : application au suivi de véhicules**
F. Bardet, D. Ramadasan, et T. Chateau
ORASIS - Congrès francophone des jeunes chercheurs en vision par ordinateur, Tregastel, June 2009

7 **Suivi et classification conjoints de multiples objets et de la source lumineuse par filtre particulaire MCMC**
F. Bardet, T. Chateau, and D. Ramadasan
RFIA : 17e congrés francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, Caen, France, Janvier 2010

3.5 Estimation précise de la trajectoire d'un véhicule

L'ensemble des travaux présentés dans ce chapitre sont issus d'une problématique scientifique liée au suivi d'objets dans une séquence d'images, avec la contrainte de générer une estimation de l'objet à l'instant présent (*Online Tracking*) en fonction de l'historique des mesures présentes jusqu'à cet instant, et de l'état précédent (hypothèse Markovienne d'ordre un).
Lorsque le suivi peut se faire hors ligne, la totalité de la séquence d'observation est disponible au moment du traitement. Dans ce cas, des techniques non séquentielles peuvent être préférables, et le problème de suivi est alors vu comme un problème d'estimation d'état stationnaire, où l'état est un vecteur de paramètres décrivant la trajectoire du mobile à suivre. Nous proposons de comparer les performances d'une méthode de suivi de type filtre particulaire, avec une méthode d'estimation de trajectoire hors ligne, elle aussi basée sur une méthode de Monte-Carlo. Cette comparaison a été effectuée dans le cadre d'une application d'estimation de la trajectoire de véhicules à partir d'observations issues d'un capteur fixé sur le bord de la chaussée et composé d'un ensemble de caméras et d'un télémètre laser à balayage 1D. D'autre part, l'observation étant constituée de deux capteurs, nous proposons une variante de l'échantillonnage d'importance capable de prendre en compte la multimodalité des observations, et nous comparons cet algorithme avec des opérateurs classiques de fusion (produit, somme).

3.5.1 Positionnement bibliographique

On propose de reformuler un problème de suivi de trajectoire où l'état, indexé par le temps, code la configuration dynamique de l'objet à suivre à chaque instant, par un problème d'estimation où l'état est une représentation paramétrique de la trajectoire du mobile. Plusieurs travaux utilisent ce type d'approche dans le cadre du suivi d'objets. Dans (33), les auteurs proposent de résoudre le problème du suivi de plusieurs personnes vues par un système multi-caméras à l'aide d'une approche par optimisation. La trajectoire de chaque personne est estimée sur un horizon d'une centaine d'images dans la séquence. Une formalisation probabiliste est préférée dans (123) avec une approche stochastique de type RJMCMC pour suivre un nombre variable de véhicules sur une autoroute, à partir d'une caméra statique. L'approche que nous proposons est assez proche de cette dernière. Nous proposons d'intégrer la dynamique du véhicule afin de contraindre les trajectoires générées à être conformes aux contraintes de non-holonomie du mobile. Cette contribution a pour conséquence de restreindre l'exploration à un ensemble de trajectoires admissibles.
D'autre part, comme les observations sont multimodales (vision et télémètrie), il est nécessaire, de combiner les vraisemblances issues de chaque capteur au niveau du filtre à particules. Dans (80) différentes observations sont fusionnées dans un contexte audiovisuel. Les filtres à particules sont très populaires pour adresser des problèmes de fusion de capteurs associés à des problématiques de suivi. Klein (47) propose d'introduire des fonctions de croyance ainsi que différentes règles d'association dans le calcul du poids des particules. L'application finale est du suivi d'obstacles dans un contexte routier. In (26), Le processus de fusion permet la sélection d'hypothèses vraisemblables. Dans un contexte multi caméras, Wang (122) propose d'adapter la méthode d'échantillonnage en fonction de la qualité des données.
Dans le cadre de travaux dont la finalité applicative porte sur l'estimation de la trajectoires de véhicules, nous avons donc proposé de formaliser le problème du suivi d'objets par une technique d'exploration de type MCMC (*Monte-Carlo Markov Chain*), appelée *méthode globale* dans la suite, et de comparer les performances de cette approche avec un suivi récursif par filtrage particulaire SIR, appelé *méthode séquentielle* dans la suite.

3.5.2 La méthode

3.5.2.1 Suivi séquentiel

La figure 3.23 illustre l'algorithme de suivi séquentiel. Basé sur un filtre à particules de type SIR, le modèle d'état décrit la configuration dynamique du mobile à suivre à un instant donné. Le modèle d'évolution prend en compte un modèle de véhicule simple appelé modèle bicyclette présenté figure 3.24.

Soient (x, y) les coordonnées du centre du véhicule dans le repère « monde », c'est-à-dire dans le repère de la scène observée, les relations cinématiques qui régissent ce modèle peuvent donc s'écrire :

$$\begin{aligned} \dot{x_t} &= v_t . \cos \beta_t \\ \dot{y_t} &= v_t . \sin \beta_t \\ \dot{\beta} t &= \tfrac{v_t}{L} . \tan \delta_t \end{aligned} \tag{3.47}$$

avec v_t la vitesse et δ_t l'angle de braquage de la roue avant dans le repère du véhicule (à l'instant t). x_t et y_t représentent la position du centre de gravité du modèle cinématique et β_t représente l'orientation du véhicule dans le repère monde [3].
D'après la théorie d'Ackerman, en comportement à basse vitesse, le centre instantané de rotation est à l'intersection du prolongement de l'axe de la roue arrière et de la perpendiculaire au plan de la roue avant tirée du centre de celle-ci. Le braquage idéal de la roue avant se déduit de la construction illustrée sur la Figure 3.24 et son angle peut alors s'écrire : $\tan \delta = \frac{L}{R}$, avec L désignant l'empattement du véhicule et R le rayon de virage au centre du véhicule. En faisant l'hypothèse des petits angles : $\delta = \dfrac{L}{R}$.
Le vecteur d'état du système est alors :

$$\mathbf{X}_t \doteq (x_t, y_t, \beta_t, \delta_t, v_t)^t \tag{3.48}$$

avec (x_t, y_t) représentant la position du centre du véhicule et v_t la vitesse dans l'axe du véhicule, dans un repère monde plan.
Le modèle cinématique, de type bicyclette, est appliqué lors de la phase de prédiction du filtre :

$$\begin{aligned} x_{t+1} &= x_t + T.v_t . \cos(\beta_t) \\ y_{t+1} &= y_t + T.v_t . \sin(\beta_t) \\ \beta_{t+1} &= \beta_t + T.\tfrac{v}{L} . \tan \delta_t \\ \delta_{t+1} &= \delta_t + T.b_{\dot{\delta}} \\ v_{t+1} &= v_t + T.b_a \end{aligned} \tag{3.49}$$

avec $b_{\dot{\delta}} \sim \mathcal{N}(0, \sigma_{\dot{\delta}})$ et $b_a \sim \mathcal{N}(0, \sigma_a)$. Les deux termes $\dot{\delta}$ et a sont distribués de manière aléatoire (selon une Gaussienne). Les termes $\sigma_{\dot{\delta}}$ et σ_a représentent respectivement les plages d'écarts de vitesse d'angle de braquage de la roue avant et d'accélération, effectués par un véhicule standard, durant une période d'échantillonnage T.
Le modèle d'observation qui utilise des cartes fond forme issues des capteurs, et un modèle 3D du véhicule, identique à celui de la méthode globale, est décrit plus bas. Elle associe à chaque particule un poids vision et un poids télémètrique.
Après observation, le filtre peut être représenté par un ensemble de N particules avec un vecteur poids associé : $\{\mathbf{X}_t^{(i)}, \boldsymbol{\pi}_t^i\}_{i=1,\ldots,N}$. Le vecteur de poids $\boldsymbol{\pi}_t^i$, de taille M égale au nombre de sources[4], est constitué du poids de la particule estimé par chaque source. Pour des raisons de lisibilité, les notations figurant dans la suite de ce paragraphe omettent l'indice temporel t.
L'échantillonnage multi-sources consiste à générer un nouveau jeu de particules, selon une approche divisée en trois étapes :

1. M échantillons sont tirés (un pour chaque source) selon une stratégie d'échantillonnage d'importance associé à chaque source (*importance sampling*). La sortie de cette étape est alors un ensemble de M échantillons candidats avec leur vecteur poids associé $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^{(i)}\}_{i=1,\ldots,M}$

2. Un vecteur de confiance de taille M est construit à partir de ratios de vraisemblance estimés pour chaque échantillon candidat (calcul détaillé ci-dessous).

[3] Le référentiel choisi pour le "monde" est le système géodésique NTF avec la projection en Lambert II étendu.

[4] De manière générale, une source est définie comme une modalité. Dans cette application, nous disposons de deux sources qui sont les deux capteurs présents (caméra et télémètre laser).

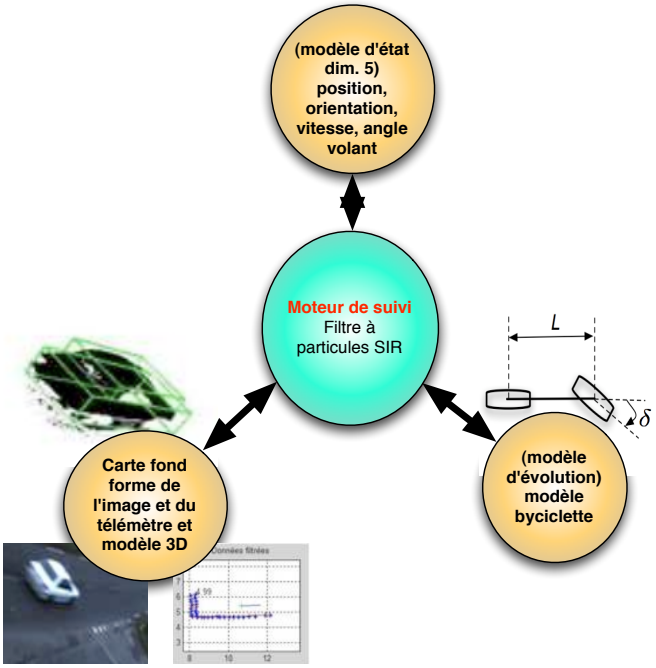


FIGURE 3.23 – Synoptique de l'algorithme de suivi d'objet proposé. Le moteur de *tracking* est constitué d'un filtre particulaire de type SIR, la fonction d'observation est issue de la comparaison d'une carte fond forme image et télémètrique avec la projection d'un modèle 3D générique de l'objet à suivre et la fonction de prédiction utilise un modèle géométrique générique du véhicule. Illustration en couleur

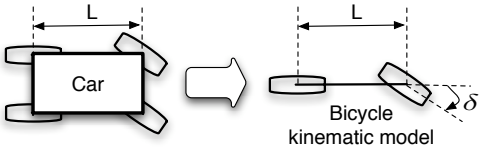


FIGURE 3.24 – Le modèle bicyclette synthétise le déplacement d'un véhicule à quatre roues.

3. L'échantillon « gagnant » est issu d'un choix effectué parmi les échantillons candidats en appliquant une stratégie de « prélèvement d'importance » sur le vecteur de confiance.

Les trois étapes ci-dessus sont répétées N fois pour obtenir le jeu complet. Le schéma explicatif de cette fusion est proposé à la figure 3.25.
Nous détaillons ici la deuxième étape de l'échantillonnage multi-sources dont le but est de construire un "vecteur de confiance" associé à l'ensemble des particules candidates. Le principe consiste à calculer le produit de rapports de vraisemblance entre les poids de même source, pour des couples de particules candidates.
Par exemple, dans le cas de deux capteurs, deux échantillons candidats sont tirés. Pour chaque particule, un produit de rapport de vraisemblance est calculé, ce qui donne, pour la première particule candidate :

$$r_1 \doteq \frac{\pi_1^1}{\pi_1^2} \tag{3.50}$$

où π_j^i représente la $j$ème composante du vecteur $\boldsymbol{\pi}^i$, i étant la source.
Dans le cas où une source est aveugle (les valeurs retournées par la fonction d'observation associée à cette source sont constantes), les rapports de vraisemblances dans lesquels la source intervient valent un et n'ont pas d'influence sur le calcul des termes r_i. Dans un cas général, il est préférable d'utiliser des log-ratios, ce qui

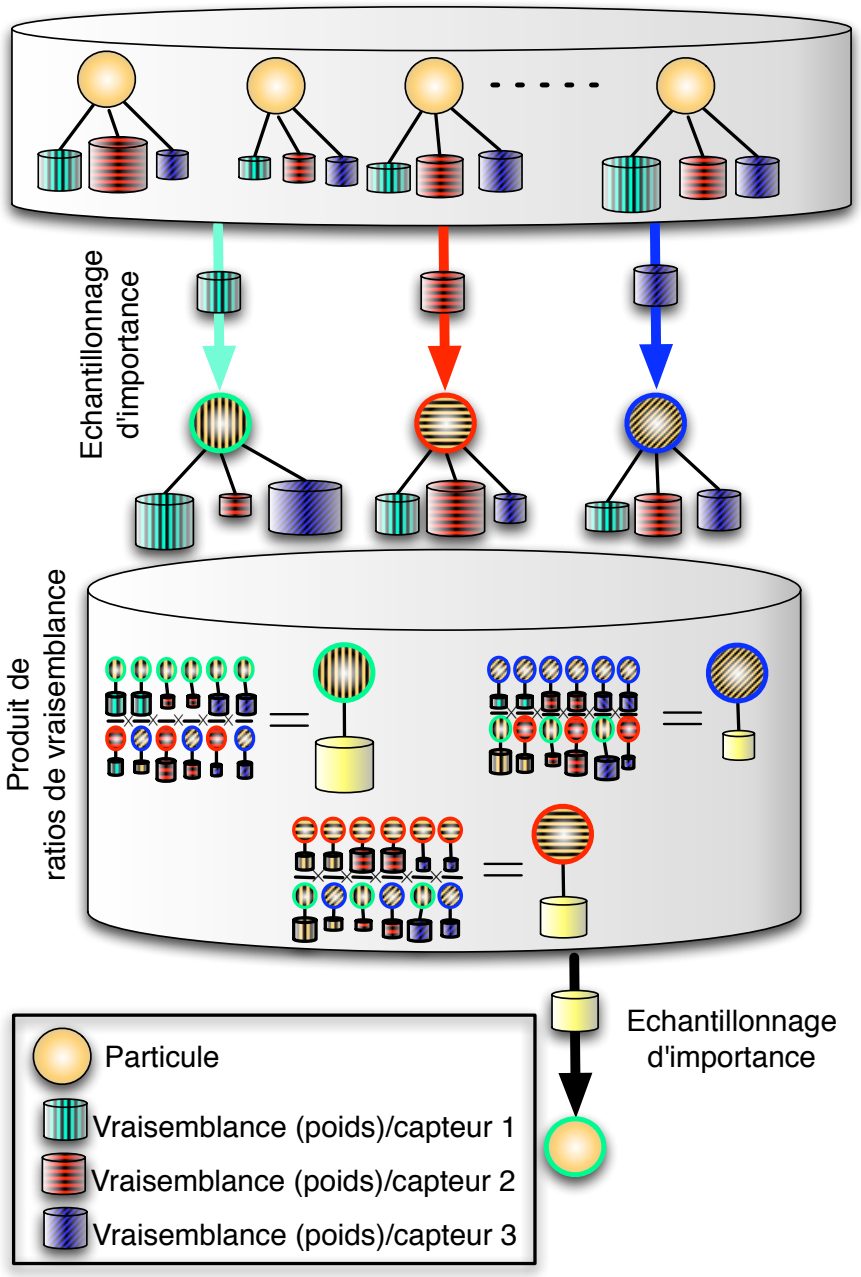


FIGURE 3.25 – Synoptique de l'algorithme M2SIR dans le cas de 3 sources : 1)Trois particules sont tirées par importance (en fonction de la vraisemblance de chaque capteur). 2) Des ratio de vraisemblance sont calculés pour chacune de ces trois particules. 3) La particule finale est tirées par importance à partir de ces trois ratios.

permet de calculer un vecteur $\mathbf{l}_r$, log de $\mathbf{r}$, constitué des coefficients r_i, par l'expression suivante :

$$\mathbf{l}_r \doteq M \begin{pmatrix} \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_1} - \frac{1}{M} \sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i} \right) \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_2} - \frac{1}{M} \sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i} \right) \\ \dots \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_M} - \frac{1}{M} \sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i} \right) \end{pmatrix} \tag{3.51}$$

où $\mathbf{l}_{\boldsymbol{\pi}_i}$ est le log du vecteur $\boldsymbol{\pi}_i$ et $\mathbf{1}_{(1\times M)}$ est une matrice de une ligne et M colonnes composée de un. En posant $\mathbf{C}_\pi \doteq \frac{1}{M}\sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i}$, $\mathbf{l}_r$ peut s'écrire :

$$\mathbf{l}_r = M \begin{pmatrix} \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_1} - \mathbf{C}_\pi\right) \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_2} - \mathbf{C}_\pi\right) \\ \dots \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_M} - \mathbf{C}_\pi\right) \end{pmatrix} \tag{3.52}$$

Le vecteur de confiance $\mathbf{c}$ est obtenu en normalisant $\mathbf{r}$ par un coefficient C_c pour que la somme de ses éléments soit unitaire.

$$\mathbf{c} \doteq C_c.\exp\left(\mathbf{l}_r\right) \tag{3.53}$$

Algorithm 1 échantillonnage multi-sources

Entrée : jeu de particules et vecteur de poids associé $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^i\}_{i=1,\dots,N}$, M sources
for $n = 1$ to N **do**
- Choisir M particules candidates à partir de $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^{(i)}\}_{i=1,\dots,N}$ et construire $\{\mathbf{X}^{*(j)}, \boldsymbol{\pi}^{*(j)}\}_{j=1,\dots,M}$ où $\mathbf{X}^{*(j)}$ est issue d'un tirage par *importance sampling* sur les poids de la source j.
- Calculer le vecteur $\mathbf{l}_r$ à partir de l'equation 3.52, puis le vecteur de confiance $\mathbf{c} \doteq C_c.\exp\left(\mathbf{l}_r\right)$
- Choisir la particule élue $\mathbf{X}^{e(n)}$ parmi les particules candidates selon un tirage par *importance sampling*.

end for
Sortie : jeu de particules $\{\mathbf{X}^{e(i)}\}_{i=1,\dots,N}$ formé des particules élues.

La figure 3.26 illustre le fonctionnement de la méthode d'échantillonnage multi-sources, pour deux scénarios différents, en comparant son comportement, à celui d'un échantillonnage issu d'une fonction de poids composée du produit des poids de chaque source, et d'une fonction de poids composée de la somme des poids de chaque source. Dans le premier scénario (colonne de gauche), la source 1 est aveugle (elle renvoie une mesure constante), et la source 2 est unimodale. Dans ce cas, la source aveugle ne doit pas venir perturber l'échantillonnage et le nouveau jeu doit coller au jeu de la source 2. On peut constater que dans le cas d'un échantillonnage de type somme des poids des deux sources, la source aveugle pollue le jeu généré. Par contre, les deux autres types d'échantillonnage se comportent bien. Le deuxième scénario illustre le cas où deux sources sont dissonantes (unimodales mais centrées sur un point différent). Le jeu de particules généré à partir de ces deux sources doit permettre de créer deux modes, autour des deux hypothèses dissonantes. On peut constater que dans le cas d'un échantillonnage de type produit, aucun des modes n'est conservé. Par contre, les deux autres types d'échantillonnage fournissent des résultats cohérents avec la distribution souhaitée.

3.5.2.2 Suivi global

Lorsque la totalité des mesures est disponible au moment de l'analyse, une méthode globale peut être envisagée. Cette dernière définit l'état comme le vecteur de paramètres d'une fonction paramétrique modélisant la trajectoire de mobile à suivre. Dans une logique de suivi séquentiel, la trajectoire globale est construite à partir de la concaténation des états à chaque instant. La période utilisée pour construire la séquence d'état étant basée sur la période du capteur le plus rapide, il faut $K \times M$ paramètres pour déterminer une trajectoire si la séquence d'observation est de dimension K et le vecteur d'état de dimension M. Par exemple, estimer la trajectoire planaire d'un mobile (position 2D et orientation) dans une séquence de 100 observations nécessite l'estimation de 300 paramètres. Dans le cas d'une approche globale, il n'est plus possible de conserver la même paramétrisation. En effet, effectuer une recherche dans un espace de dimension très important n'est pas réalisable avec les moyens de calcul actuels. Pour réduire la dimension du problème, nous avons proposé de coder la trajectoire par une fonction paramétrique, à partir de laquelle il est possible de générer une séquence d'état.

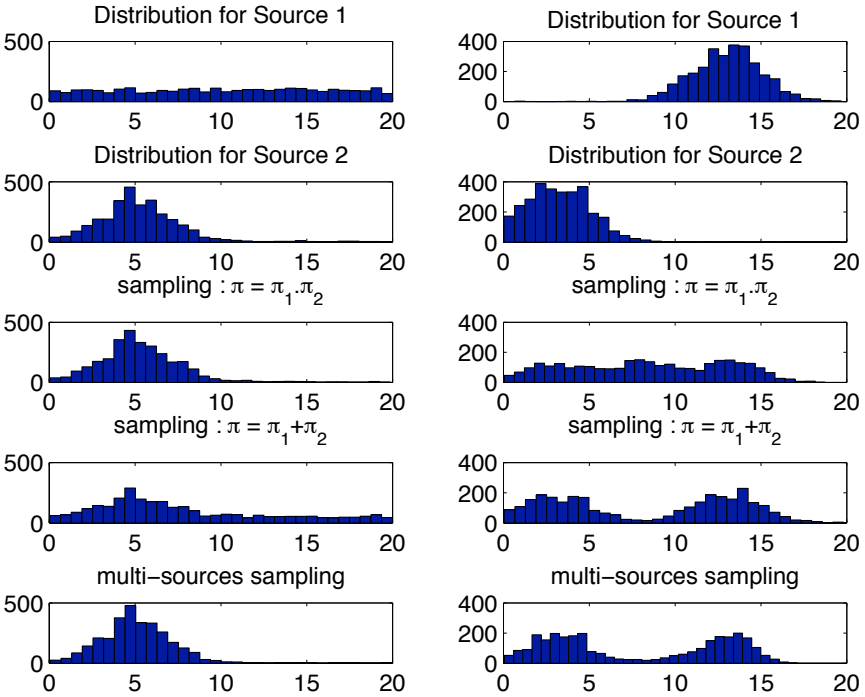


FIGURE 3.26 – Illustration du fonctionnement de la méthode d'échantillonnage multi-sources, pour 2 scénarios différents (1 par colonne). Sur la colonne de gauche, de haut en bas : les deux premières courbes représentent la réponse de la source (observation). La courbe suivante représente le résultat d'un échantillonnage d'importance basé sur un poids qui est le produit des poids des deux sources. L'avant dernière courbe représente le résultat d'un échantillonnage d'importance basé sur un poids qui est la somme des poids des deux sources. La dernière courbe représente le résultat de l'échantillonnage multi-sources proposé.

Les commandes du conducteur sont l'angle de braquage des roues et la vitesse longitudinale (déduite de l'accélération) du véhicule. Les lois de commande pour un véhicule léger peuvent être modélisées par des sigmoïdes, sachant que la vitesse de l'angle de braquage admissible est comprise entre 1.5 et 4 degrés par seconde, et que la gamme d'accélération longitudinale pratiquée par un conducteur "normal" est comprise entre 1 $m.s^{-2}$ et 3 $m.s^{-2}$. Pour prendre en compte une large proportion de conducteurs et pour s'affranchir de la non linéarité du système (contacts pneumatiques-chaussée ...), nous simulons chacune des commandes de braquage et d'accélération avec une sigmoïde. Cette dernière permet de coder de nombreuses configurations routières (entrée dans le virage, partie centrale d'un virage, sortie de virage, ligne droite, ...) :

$$f_\delta(\boldsymbol{\theta}_\delta, t) \doteq \frac{\theta_{\delta,2}}{1 + \exp\left[\dfrac{\theta_{\delta,3}(\theta_{\delta,4} - t)}{|\theta_{\delta,2}|}\right]} + \theta_{\delta,1}; \tag{3.54}$$

La figure 3.27 montre l'effet des différents paramètres utilisés pour coder la sigmoïde. Les plages de variation des différents paramètres sont définis à partir d'a priori sur les comportements des conducteurs ou sur le profil du lieu d'observation.
Une sigmoïde est également utilisée pour modéliser les profils de vitesse $f_v(\boldsymbol{\theta}_v, t)$, calculée de manière identique à $f_\delta(\boldsymbol{\theta}_\delta, t)$, en substituant l'index δ par l'index v.
Pour modéliser le véhicule, le modèle cinématique simple de type bicyclette, décrit précédemment, est utilisé. Ce modèle génère itérativement x_t, y_t et β_t, pour chaque pas de temps t, à partir d'une position initiale est des sigmoïdes de vitesse et d'angle au volant.

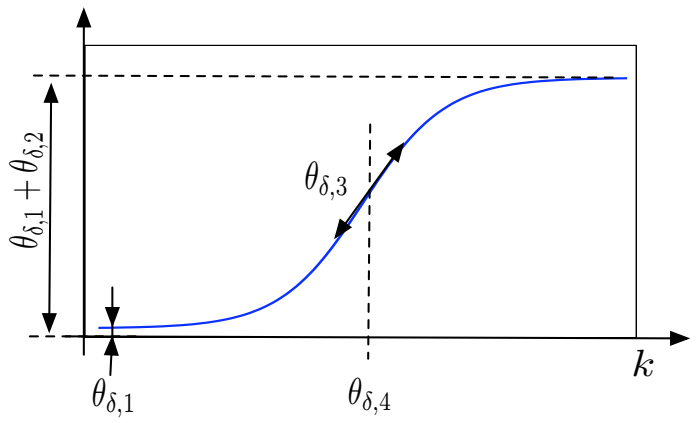


FIGURE 3.27 – Représentation graphique des paramètres de la sigmoïde utilisée pour coder l'angle au volant et la vitesse.

La trajectoire du véhicule est donc représentée par un vecteur d'état aléatoire de dimension 11 :
$\mathbf{X} \doteq (x_0, y_0, \beta_0, \boldsymbol{\theta}_\delta, \boldsymbol{\theta}_v)^T$ avec :

- ▷ (x_0, y_0, β_0) représentant la position initiale et l'orientation du véhicule (dans un référentiel monde).
- ▷ $\boldsymbol{\theta}_\delta \doteq (\theta_{\delta,1}, ..., \theta_{\delta,4})$ sont les paramètres de la fonction sigmoïde $\delta_k = f_\delta(\boldsymbol{\theta}_\delta, k)$ représentant l'évolution temporelle discrète de l'angle de braquage.
- ▷ $\boldsymbol{\theta}_v \doteq (\theta_{v,1}, ..., \theta_{v,4})$ sont les paramètres de la fonction sigmoïde $v_k = f_v(\boldsymbol{\theta}_v, k)$ représentant l'évolution temporelle discrète de la vitesse du véhicule.

Nous voulons estimer $p(\mathbf{X}|\mathbf{Z})$, la densité de probabilité du modèle paramétrique $\mathbf{X}$, connaissant les données observées $\mathbf{Z}$. La technique de Monte-Carlo suppose qu'on peut faire une approximation de la distribution grace à un jeu de N échantillons $\{\mathbf{X}^n\}_{n=1}^N$ de la façon suivante :

$$p(\mathbf{X}|\mathbf{Z}) \approx \sum_{n=1}^{N} \pi_n \delta(\mathbf{X} - \mathbf{X}^{(n)}), \tag{3.55}$$

où δ est une fonction de dirac et π_n représente le poids associé à $\mathbf{X}^{(n)}$ de manière à avoir $\sum_{n=1}^N \pi_n = 1$.
La figure 3.28 représente une illustration de l'exploration par **MCMC**. Ce processus est itératif : d'une trajectoire donnée ($\mathbf{X}^{(n-1)}$), un paramètre est modifié, permettant la proposition d'une nouvelle trajectoire ($\mathbf{X}^*$). Le générateur de trajectoires fournit l'ensemble des positions du véhicule dans chaque capteur, à partir desquelles une vraisemblance (poids), issue des mesures, est calculée. Le principe est alors de comparer ce nouveau poids (proposition) au poids de la trajectoire précédemment choisie, par le calcul du ratio d'acceptation Metropolis-Hasting. Le processus s'arrête quand il atteint le nombre d'itérations fixé au départ.
A l'itération n, le **MCMC** génère une nouvelle proposition en échantillonnant une distribution de propositions $q(\mathbf{X}^*|\mathbf{X}^{(n-1)})$ définie par :

$$q(\mathbf{X}^*|\mathbf{X}^{(n-1)}) = \sum_{m \in \{1;...;M\}} q'(m) q(\mathbf{X}^*|\mathbf{X}^{(n-1)}, m), \tag{3.56}$$

où $q'(m)$ est une distribution *a priori* permettant de sélectionner l'index du paramètre de $\mathbf{X}$ à modifier (M représente la taille de $\mathbf{X}$). Une distribution de proposition d'un paramètre est alors définie par :

$$q(\mathbf{X}^*|\mathbf{X}, m) \doteq p(X_m^*|X_m^{(n-1)}) \prod_{j \neq m} \delta(X_j^* - X_j^{(n-1)}) \tag{3.57}$$

Ici, seulement le m^ième^ composant (m est sélectionné grâce à la distribution *a priori* $q'(m)$) du vecteur d'état est modifié à l'itération n ; les autres paramètres restent inchangés. La distribution $q'(m)$ est construite en fonction des connaissances *a priori* des conditions initiales (par exemple la position).

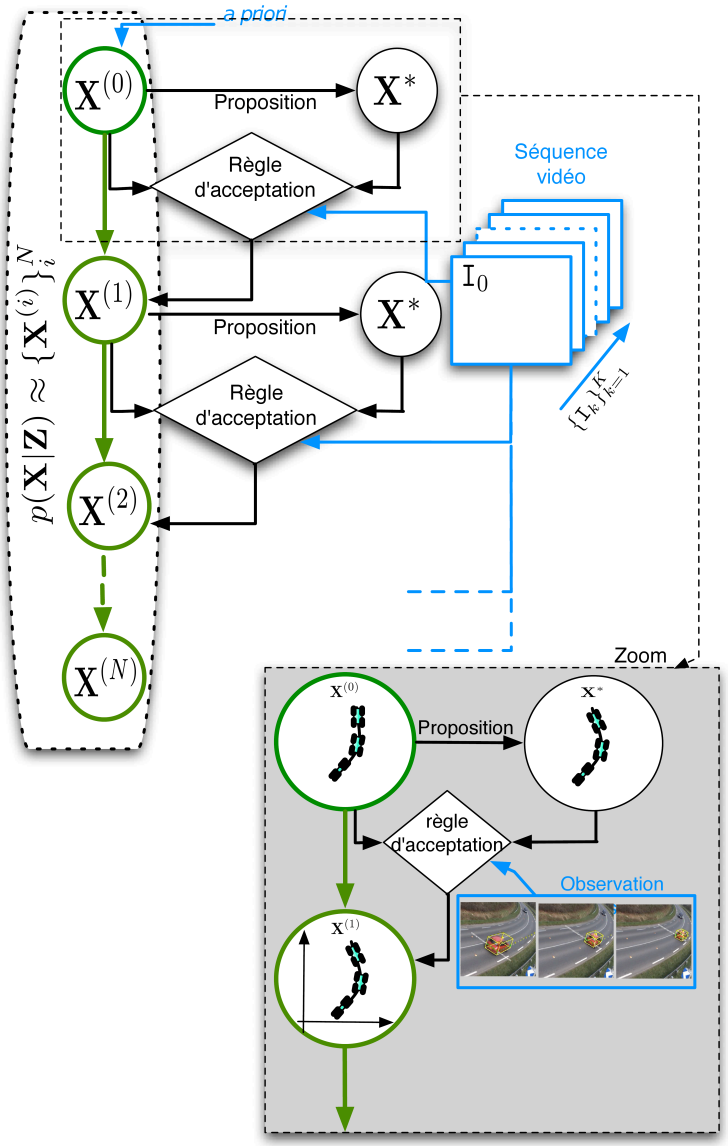


FIGURE 3.28 – Illustration du fonctionnement de l'algorithme d'exploration MCMC pour l'approximation de la distribution de probabilité associée à la trajectoire d'un véhicule.

3.5.2.3 Fonction de vraisemblance

La fonction de vraisemblance est basée sur une extraction fond forme des données capteurs, puis une comparaison de données de forme avec une projection d'un modèle 3D générique du véhicule à suivre dans les espaces capteur. Pour l'extraction fond forme associée à la caméra, nous avons proposé un algorithme adaptatif de modélisation de la distribution des couleurs de chaque pixel de l'image. Ce dernier est basé sur l'approximation marginalisée de la vraisemblance en chaque pixel par une discrétisation. Cette vraisemblance est alors remise à jour de manière temporelle, selon une constante de temps réglable de manière très intuitive. Dans le cas de la méthode globale, pour chaque proposition, une séquence d'état temporelle est reconstruite et la vraisemblance globale est obtenue en intégrant les vraisemblances à chaque instant. La figure 3.29 illustre le

calcul de cette vraisemblance globale, à partir d'une hypothèse de trajectoire.

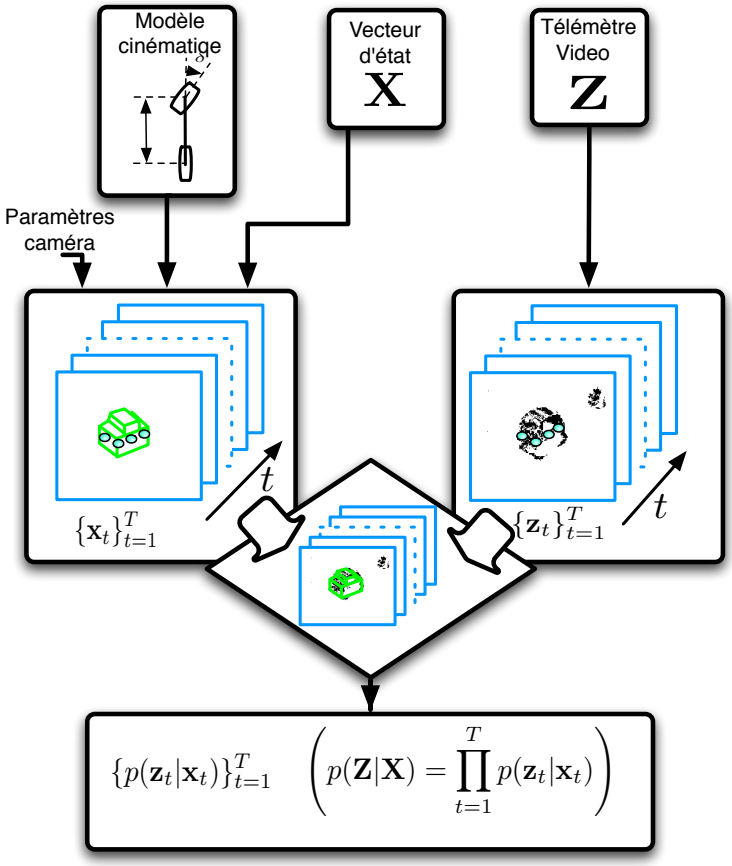


FIGURE 3.29 – Illustration du la fonction de vraisemblance dans le cas de la méthode globale.

3.5.3 Résultats

Toutes les expérimentations ont été effectuées sur des données réelles, représentant un virage d'environ 100m. La vérité terrain est obtenue à l'aide d'un GPS cinématique qui délivre la position du mobile dans le référentiel absolu. L'erreur entre l'estimation et la vérité terrain a été définie entre la moyenne des distances euclidiennes entre chaque estimation et la droite issue des deux points GPS les plus proches de l'estimation. Parmi les tests réalisés, le plus important concerne la comparaison entre l'approche séquentielle et l'approche globale, pour une vingtaine de passages du véhicule.
Le tableau 3.4 présente l'erreur moyenne d'estimation de la position et de l'orientation, ainsi que l'écart type associé, pour les deux méthodes proposées. On constate que l'erreur moyenne sur la position est environ 25% inférieure dans le cas de la méthode globale. Le gain est encore plus important dans le cas de l'estimation de l'orientation où l'erreur est 70% plus faible dans le cas de la méthode globale.

Méthode	Position erreur (m)	Position écart type (m)	Orientation erreur (deg.)	Orientation écart type (deg.)
Méthode séquentielle	0.27	0.26	3.67	3.36
Méthode globale	0.20	0.22	1.12	0.97

TABLE 3.4 – Erreurs de position et d'orientation (la position réelle est donnée par un GPS cinématique) pour la méthode séquentielle et la méthode globale

La figure 3.30 présente, pour la position et l'orientation, le pourcentage d'estimations correctes en fonction

d'une tolérance maximale sur l'erreur, pour les deux méthodes. Ce type de courbe se lit comme une courbe ROC en classification, la courbe idéale étant celle qui s'approche le plus du point $(0, 1)$. La méthode globale présente de meilleures performances que la méthode séquentielle. La différence est encore plus importante dans le cas de l'estimation de l'orientation.
La figure 3.31 illustre le comportement des deux méthodes sur une séquence réelle. Les courbes présentées sur la gauche de la figure sont des zooms sur une portion de la trajectoire. Les colonnes du milieu et de droite illustrent la projection du résultat de l'estimation dans quelques images de la séquence, respectivement pour la méthode séquentielle et la méthode globale. On peut noter, dans le cas de la méthode globale, des variations plus importantes dans la trajectoire reconstruite, due à des bruits d'observation locaux (ici, présence d'ombre portée).

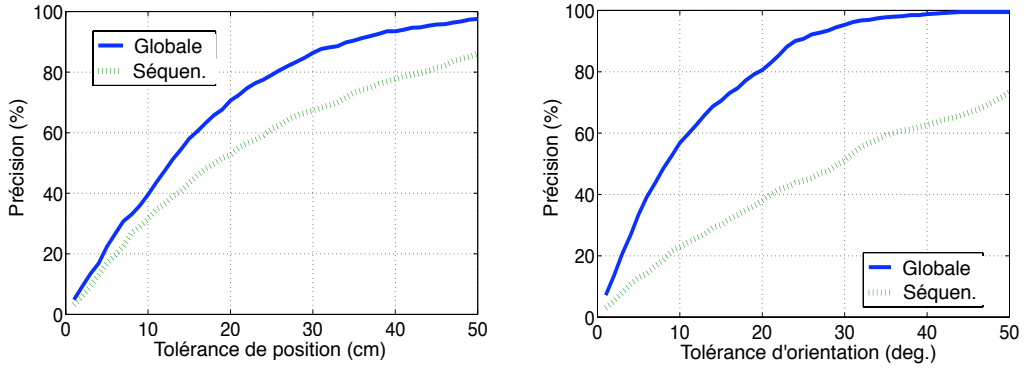


FIGURE 3.30 – Performances comparées des deux méthodes d'estimation : pourcentage d'estimations correctes en fonction d'une tolérance maximale sur l'erreur, pour les deux méthodes : à gauche, estimation de la position et à droite, estimation de l'orientation.

3.5.4 Conclusion

L'intérêt principal de cette étude a été de montrer le gain de précision obtenu par une méthode globale par rapport à une méthode séquentielle. Pour arriver à cette conclusion, dans un cadre applicatif bien défini (l'estimation de trajectoires de véhicules), nous avons développé deux méthodes, et proposé un certain nombre de contributions :

- ▷ pour la méthode séquentielle, nous avons proposé de prendre en compte un modèle cinématique du véhicule dans le modèle de prédiction du filtre séquentiel. D'autre part, nous avons proposé un algorithme de ré-échantillonnage multi-sources permettant de gérer de manière intrinsèque la multi-modalité dans le filtre à particules.

- ▷ pour la méthode globale, nous avons proposé une paramétrisation de la trajectoire d'un véhicule sous la forme d'une courbe de vitesse et d'une courbe d'angle au volant. D'autre part, les fonctions de propositions utilisées dans l'algorithme d'exploration tiennent compte à la fois des comportements moyens des conducteurs et des spécificités géométriques du lieu de l'expérimentation.

- ▷ pour la fonction d'observation, nous avons proposé une méthode d'extraction fond forme adaptative.

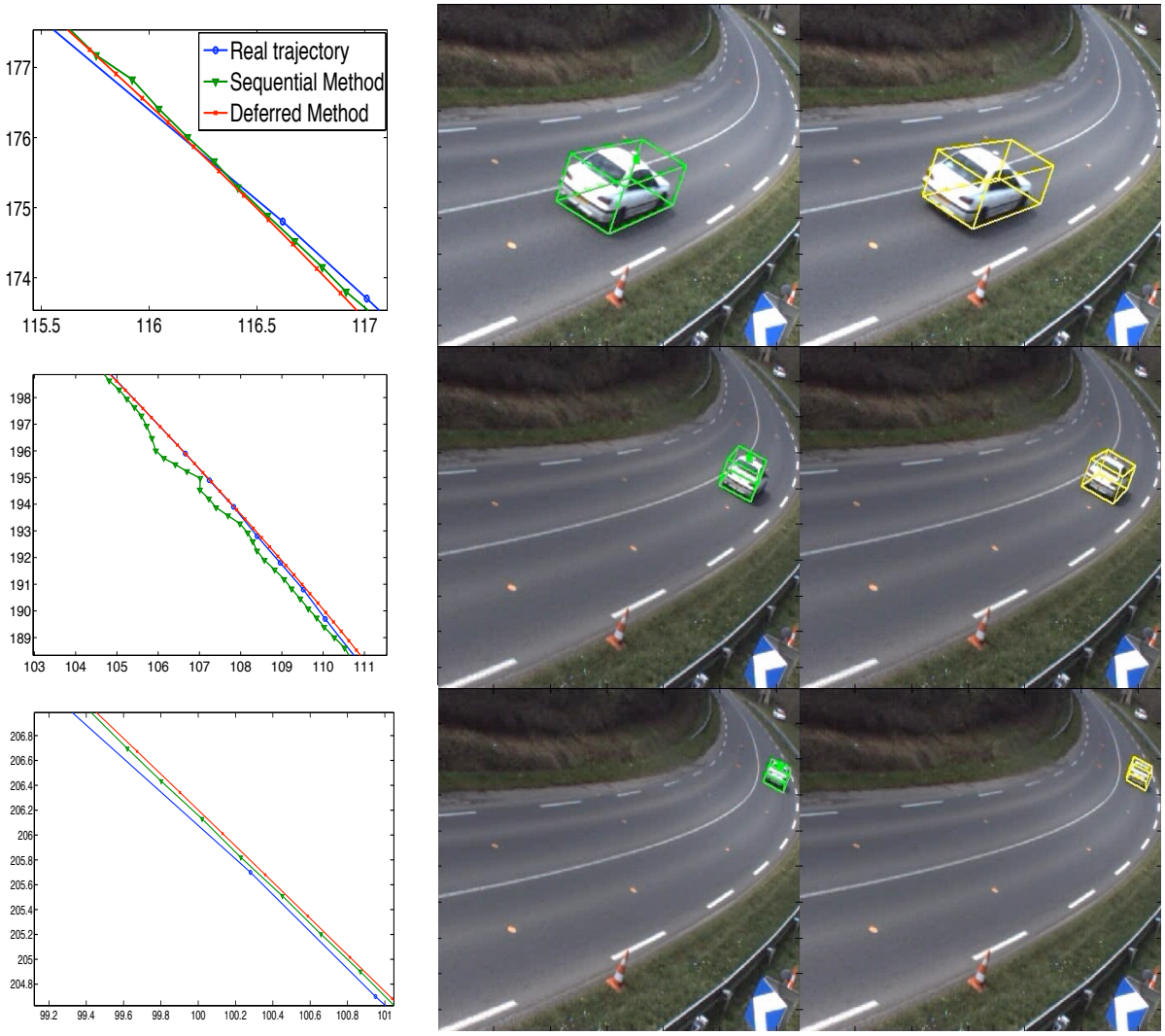


FIGURE 3.31 – Quelques images illustrant le comportement des deux méthodes. sur la colonne de gauche, zoom sur certaines portions locales de la trajectoire, sur la colonne du milieu et la colonne de droite, reprojection du résultat de l'estimation respectivement pour la méthode séquentielle et la méthode globale.

3.5.5 Publications associées

Ces travaux ayant été réalisés dans un cadre applicatif, il ont donné lieu à des publications, à la fois dans le domaine de la vision par ordinateur, et dans le domaine des transports intelligents (*Intelligent Tranportation Systems*). Ces publications décrivent uniquement la méthode par filtrage séquentielle ; la méthode globale faisant l'objet, d'une procédure de dépôt de brevet, aucune publication n'a été effectuée concernant cette dernière.

1	**M2SIR, a Multi Modal Sequential Importance Resampling Algorithm for Particle Filters** T. Chateau, Y. Goyat et L. Trassoudaine *ICIP IEEE International Conference on Image Processing*, Le Caire, Egypte, Novembre 2009
1	**Trajectory Measurement of Vehicles : A New Observation** Y. Goyat, T. Chateau, L. Trassoudaine et L. Malaterre *Advances in Transportation Studies*, Vol. 8, pp 5-17, Juillet 2009
2	**Tracking of Vehicle Trajectory by Combining a Camera and a Laser Rangefinder** Y. Goyat, T. Chateau et L. Trassoudaine *Springer MVA : Machine Vision and Application*, online, Mars 2009
3	**Métrologie des trajectoires de véhicules** Y. Goyat, T. Chateau et L. Trassoudaine *Conférence Internationale Francophone d'Automatique*, Bucarest, Roumanie, Septembre 2008
4	**Estimation Précise de la Trajectoire d'un Véhicule par Fusion Vision Télémètre Laser** Y. Goyat, T. Chateau, L. Malaterre et L. Trassoudaine *RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*, Amiens, France, Janvier 2008
5	**Un observatoire de trajectoires en virages fondé sur la vision artificielle** Y. Goyat, T. Chateau, L. Malaterre, L. Trassoudaine et F. Menant *RTS : Recherche, Transport et Sécurité*,Vol 98, pp 73–88, Mars 2008
6	**Trajectographie des véhicules en vision monoculaire** Y. Goyat, T. Chateau, L. Malaterre et L. Trassoudaine *GRETSI - 11eme Colloque de traitement du signal et des images*, Troyes, France, Septembre 2007
7	**Vehicle Trajectories Evaluation by Static Video Sensors** Y. Goyat, T. Chateau, L. Malaterre et L. Trassoudaine *ITSC06 2006 - 9th International IEEE Conference on Intelligent Transportation Systems*, Toronto, Canada, Septembre 2006

CHAPITRE

4

CONCLUSION ET PERSPECTIVES DE RECHERCHES

Les deux chapitres précédents synthétisent les travaux de recherche que j'ai menés autour de la problématique de l'estimation d'état par vision. Ce chapitre conclut et expose, autour de cette problématique, des perspectives scientifiques qui pourraient donner lieu à des actions articulées à différents niveaux des structures de recherche (Projets, Thèses, Postdoc, ...)

4.1 Perspectives de recherche

Les travaux présentés dans ce manuscrit s'articulent autour de l'estimation d'état par vision. Certaines des méthodes proposées sont suffisamment performantes pour déboucher sur des applications réelles. Plus précisément, nous nous intéressons à des problèmes d'estimation dont les données d'entrées sont des séquences d'images décrivant de manière spatio-temporelle le contenu de la scène à analyser. Dans ce contexte, il reste de nombreux verrous scientifiques à lever pour rendre les méthodes d'estimation d'état plus performantes devant la variabilité des objets à représenter et la complexité des états à expliquer. :

- ▷ la variation spatio-temporelle de l'apparence des objets dans des applications de suivi visuel ;
- ▷ la représentation et la manipulation des densités de probabilités d'états complexes.

Dans la suite de ce chapitre, je présente la problématique associée à chacun de ces verrous scientifiques et je donne des pistes de recherches qui me semblent pertinentes.

4.1.1 La variation spatio-temporelle de l'apparence des objets dans des applications de suivi visuel

Dans une problématique d'estimation d'état à partir d'une séquence d'images, la solution la plus courante consiste à expliquer la configuration dynamique de l'objet, à l'aide d'un filtre temporel. Dans le cas du suivi d'un objet par exemple, l'état est défini par la position de l'objet dans un référentiel monde à l'instant courant. L'algorithme filtre donc cette position, à partir d'un modèle d'observation souvent figé. Lorsque l'observation utilise un modèle d'apparence de l'objet, souvent issu de la première image de la séquence, on fait l'hypothèse que ce modèle reste valide durant toute la durée du suivi. Or ce n'est pas le cas car la pose de l'objet par rapport aux sources d'éclairement (souvent variables) et à la caméra change, ce qui fait varier l'apparence de l'objet durant la séquence. La figure 4.1 illustre ce phénomène dans le cas du suivi de visages.
On se pose alors la question de la mise à jour du modèle d'apparence de l'objet. C'est un problème difficile car le modèle d'apparence fait parti de la fonction de vraisemblance utilisée pour estimer la configuration spatiale de l'objet ; et la configuration spatiale est souvent la base de la mise à jour du modèle d'apparence. Une méthode d'exploration jointe de l'apparence et de la configuration spatiale dériverait très rapidement. Pour éviter ce phénomène de dérive, il est important d'injecter une information supplémentaire au système, indépendante des deux autres. Certains travaux utilisent une mise à jour du modèle basé sur une analyse d'une couronne de l'image (fond) autour de l'objet (58) (7). D'autres travaux combinent un détecteur générique d'objets avec un détecteur dont le modèle se met à jour en ligne (40).
Au delà des méthodes proposées, il est important de considérer que le problème du suivi d'un objet doit être formalisé dans son ensemble. L'état doit à la fois englober la configuration spatiale de l'objet ainsi que son modèle d'apparence. Le suivi de l'objet par filtrage temporel doit donc s'appliquer à la fois sur la position et sur le modèle d'apparence de l'objet. Cette solution ne devrait pas permettre de régler le problème de dérive sans y injecter une information complémentaire telle que le modèle de fond ou un détecteur de la classe d'objet que l'on suit. Cette information complémentaire doit être globale. Une première hypothèse peut être de considérer que l'objet suivi appartient à une classe générique d'objet (piéton, voiture, ...) ; et l'utilisation d'un classifieur générique de cette classe apporte une information de recalage spatial indépendante du modèle d'apparence de l'objet. Une seconde hypothèse consiste à considérer que l'évolution du fond proche de l'objet est indépendante de l'évolution de l'apparence de l'objet ; et une mise à jour conjointe des deux modèles atténue le phénomène de dérive.
Les majorité des méthodes de détection d'objets (piétons, visages) sont ciblées pour fonctionner à partir d'une image statique. Or, dans les applications de vidéo-surveillance, les données d'entrée sont composées de séquences d'images. L'idée est alors d'introduire, dans les descripteurs, des informations issues d'une analyse de la séquence. Dans (115), les auteurs proposent de prendre en compte des informations de déplacement des pixels, en plus de l'apparence, dans un algorithme de détection de piétons. D'autre part, dans (121), les auteurs proposent un descripteur basé sur la covariance de paramètres mélangeant des informations spatiales, d'apparence, et d'extraction fond/forme. L'algorithme de détection de piétons ainsi généré est particulièrement

FIGURE 4.1 – Illustration de la variation de l'apparence d'un visage suivi dans une séquence. chaque colonne représente, pour un visage, la collection des vues sur quelques images consécutives.

performant. La figure 4.2 illustre les résultats de cette méthode dans le cas d'une application de vidéo-surveillance.

La combinaison d'informations issues d'extraction fond/forme, d'informations de flot optique ou de gradient temporel, avec des informations spatiales et d'apparences devrait permettre d'obtenir des descripteurs plus

performants et particulièrement adaptés à des applications de vidéo-surveillance. Ces descripteurs pourraient servir dans le cas d'applications de détection et de suivi d'objets.

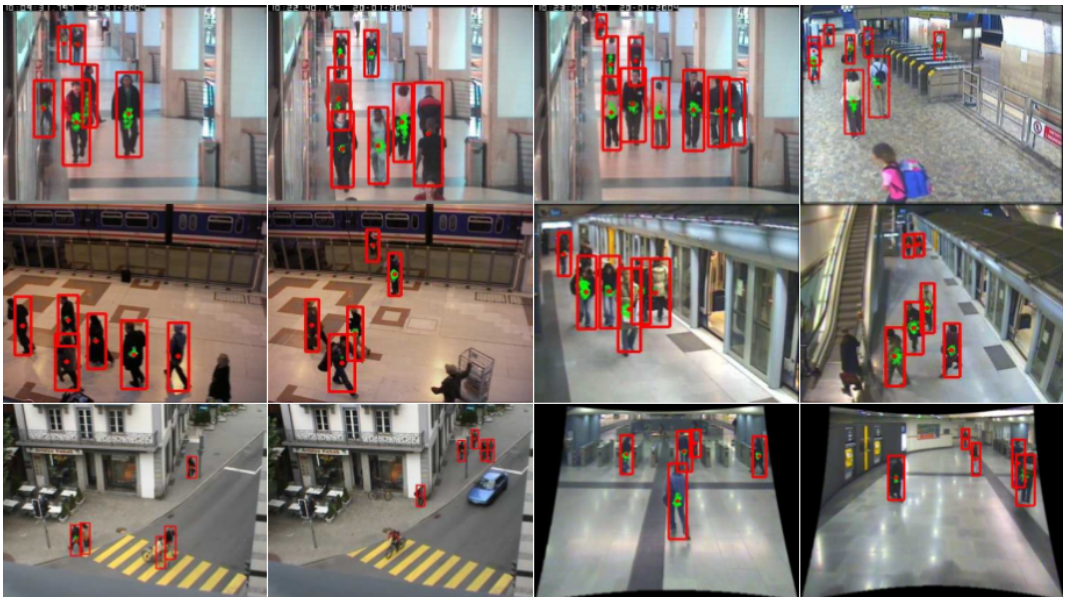


FIGURE 4.2 – Exemples de détections de piétons, à l'aide d'un descripteur mêlant à la fois des informations spatiales, d'apparence, et d'extraction fond/forme, issues de (121).

4.1.2 la représentation et la manipulation des densités de probabilités d'états complexes

Un problème récurrent en vision par ordinateur consiste à modéliser et manipuler des distributions de probabilités pour lesquelles on ne dispose que d'un ensemble de réalisations. Dans le cas de la modélisation, on distingue deux problématiques principales :

1. On connaît un ensemble de réalisations d'une loi (appelée loi cible), éventuellement de grande taille et l'on souhaite une expression analytique de la vraisemblance de cette loi. Une approximation non paramétrique de la loi est possible par des techniques à noyau comme les fenêtres de Parzen (KDE[1]). Néanmoins, dans le cas où le nombre de réalisations est très important, cette technique devient coûteuse en temps de calcul. Une alternative consiste alors à utiliser des modèles paramétriques comme des combinaisons linéaires de fonctions de bases. Les mixtures de Gaussiennes peuvent être vues comme un cas particulier de ces modèles. Les techniques les plus utilisées pour estimer les paramètres du modèles utilisent des algorithmes EM (*Expectation Maximisation*. La méthode SKDA (*Sequential Kernel Density Approximation*), proposée dans (44), est une alternative intéressante car elle permet de construire la mixture en insérant les réalisations une par une. Dans les méthodes paramétriques à base de mixtures de Gaussiennes, les paramètres à estimer sont les poids des mixtures, leur centre et leur matrice de covariance. Il pourrait être intéressant d'étudier les modèles tels que ceux utilisés dans des machines de régression parcimonieuses telles que les RVM.

2. On cherche à représenter, de manière efficace, par une méthode de Monte-Carlo, une loi de probabilité complexe, éventuellement expliquant un état de dimension variable, avec un faible nombre de réalisations par rapport à la dimension de la loi. Pour cela, on dispose d'une fonction permettant d'évaluer la vraisemblance de la loi pour une hypothèse donnée. Un des points clefs des méthodes MCMC concerne la fonction de proposition. Une majorité de méthodes utilise une marche aléatoire pour explorer l'état. Cette stratégie est loin d'être optimale et dans l'optique où l'on dispose de peu d'explorations possibles pour expliquer la loi cible, il me paraît important d'explorer de nouvelles

[1] Kernel Density Estimation

fonctions de proposition. Il est par exemple possible d'utiliser une estimation du gradient de la loi lors de l'exploration pour rendre plus efficace la fonction de proposition (65) . D'autres techniques utilisent un *a priori* issu des observations pour guider la fonction de proposition. Certaines méthodes de suivi d'objets introduisent par exemple des fonctions de proposition basées sur des observations (53). Dans le cas d'une proposition d'apparition d'objet, la position de ce dernier est initialisée en fonction des données de l'image non expliquées par les objets déjà présents. Je pense que la stratégie de proposition des méthodes de Monte-Carlo est un problème ouvert et que de nombreuses contributions peuvent encore être réalisées, dans l'optique d'explorer efficacement avec peu de particules.[2].

Dans le cas de la manipulation de distributions de probabilités, la problématique principale consiste à estimer une distance entre deux lois de probabilités pour lesquelles on ne dispose que d'un ensemble de réalisations de ces lois. La divergence de Kullback-Leibler, aussi appelée entropie relative, est une mesure de ressemblance entre deux distributions de probabilités. Dans (16), l'auteur propose une expression de cette divergence à partir d'un ensemble de réalisations suivant les deux distributions. Cette expression permet donc de comparer deux distributions à partir d'un ensemble de réalisations issu de chacune de ces distributions. Cette mesure peut ensuite être utilisée dans des applications telles que du suivi ou de la détection d'objets.

4.2 Conclusion

Ce manuscrit dresse une synthèse de mes activités de recherche sur la période 2001-2009. J'ai proposé, dans le domaine de la vision par ordinateur, des contributions autour de deux axes principaux :

- ▷ les méthodes d'apprentissage pour l'estimation d'état. J'ai évalué l'utilisation de machines d'apprentissage pour adresser des problèmes classiques en vision par ordinateur tels que la détection d'objets, l'estimation de posture ou le suivi de motifs planaires.

- ▷ les méthodes de Monte-Carlo pour l'estimation d'état. Ces méthodes ont été évaluées dans le cadre de problèmes d'estimation de trajectoire et du suivi d'un nombre variable d'objets.

Les principales retombées applicatives de mes travaux sont autour des systèmes de transport intelligents. D'un point de vue infrastructure, les travaux réalisés dans le cadre de la thèse de Yann Goyat ont montré qu'il été possible d'estimer la trajectoire de véhicules dans un virage avec une précision de l'ordre de 10cm. D'autre part, les travaux réalisés dans le cadre de la thèse à F. Bardet ont montré qu'il était possible de suivre et de catégoriser des véhicules en temps réel. D'un point vue systèmes embarqués, les travaux de thèse de Laétitia Leyrit et de Samuel Gidel ont permis d'évaluer les performances d'un système de détection de piétons à partir de différents capteurs positionnés dans un véhicule. Les travaux de thèse d'Eric Royer et ceux de Guillaume Blanc ont montré la faisabilité d'un système de localisation par mémoire d'images dans des applications de robotique mobile.
De manière secondaire, mes travaux peuvent aussi être utilisés dans le cadre d'applications de vidéo-surveillance. La thèse de Francois Bardet a montré la faisabilité d'un système de suivi temps réel d'un nombre variable d'objets comme des piétons. La thèse de Laetitia Gond, qui adresse la problématique de l'estimation de postures peut servir de base à des applications de vidéo-surveillance dans des lieux tels que des appartement de personnes fragiles (séniors).
Au delà des contributions méthodologiques, tous ces travaux ont été menés avec la volonté d'aboutir à des démonstrateurs réels ou des produits. C'est une volonté personnelle forte[3], en cohérence avec la politique scientifique du Laboratoire. Trois dépôts de codes APP et un dépôt de brevet ont été effectués. D'autre part, les travaux réalisés autour de l'estimation de trajectoires font actuellement l'objet d'un transfert de technologie

[2] Au delà de l'approximation précise des lois de probabilité, on cherche avant tout à connaître les différents modes de la loi ; ces derniers traduisant les différentes hypothèses que peut prendre l'état. Aussi, ce qui est important, c'est que l'exploration conserve ces modes

[3] Cette volonté d'aboutir à des réalisations concrètes a toujours guidée mes travaux : les contributions réalisées lors ma thèse ont donné lieu à un produit (guidage automatique d'une moissonneuse batteuse) primé par une médaille au salon de Agro-technica en 1999

auprès d'une société Auvergnate, dans l'optique d'estimation d'origines/destinations sur des giratoires[4]. Les travaux réalisés dans le cadre de la thèse d'Eric Royer sont actuellement en cours de transfert dans un projet de création de Véhicule Autonome nommé VIPA[5].

[4]ce transfert, financé par la région auvergne et l'europe, d'une durée d'un ans, a débuté au premier mars 2010

[5]ce projet, issu d'une consorsium composé du Lasmea et des sociétés Ligier et Apogée, se propose de construire un véhicule autonome présenté lors du salon de l'automobile en 2010. Il est financé par la région Auvergne et des fond Européens

Bibliographie

[1] A., M. & Richefeu, J. (2004). A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'04)*.. Kolkata, 46–51.

[2] Agarwal, A. (2006). *Machine Learning for Image Based Motion Capture*. Ph.D. thesis, Institut National Polytechnique de Grenoble.

[3] Agarwal, A. & Triggs, B. (2006). A local basis representation for estimating human pose from cluttered images. In *Proceedings of the Asian Conference on Computer Vision*. Hyderabad, India.

[4] Agarwal, A. & Triggs, B. (2006). Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(1), 44–58.

[5] Ahonen, T., Hadid, A., & Pietikäinen, M. (2006). Face Description with Local Binary Patterns : Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 2037–2041.

[6] Avidan, S. (2001). Support vector tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*. Hawaii.

[7] Avidan, S. (2007). Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(2), 261–271.

[8] Baker, S. & Matthews, I. (2004). Lucas-kanade 20 years on : A unifying framework. *(IJCV), Internationnal Journal on Computer Vision*, *56*(3), 221–255.

[9] Bandouch, J., Engstler, F., & Beetz, M. (2008). Accurate human motion capture using an ergonomics-based anthropometric human model. In *Proceedings of the International Conference on Articulated Motion and Deformable Objects*. Mallorca, Spain.

[10] Bar-Shalom, Y. & Fortmann, T. (1988). *Alignement and Data Association*. New-York : Academic.

[11] Bardet, F., Chateau, T., & Lapresté, J. (2009). Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous tracking and classification. In *ICCV 2009, International Conference on Computer Vision*. Tokyo, Japan.

[12] Barron, C. & Kakadiaris, I. (2001). Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, *81(3)*, 269–284.

[13] Bégard, J., Allezard, N., & Sayd, P. (2008). Détection de piétons temps-réel en milieu urbain. In *RFIA : 16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle*. Amiens, France.

[14] Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 509–522.

[15] Benhimane, S. & Malis, E. (2004). Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJ IROS*. Japan.

[16] Boltz, S., Debreuve, E., & Barlaud, M. (2008). High-dimensional statistical measure for region-of-interest tracking. *IEEE Transactions on Image Processing*.

[17] Bottino, A. & Laurentini, A. (2001). A silhouette based technic for the reconstruction of human movement. *Computer Vision and Image Understanding*, *83*, 79–95.

[18] Bowden, R., Mitchell, T., & Sahardi, M. (2000). Non-linear statistical models for the 3d reconstruction of human pose and motion from monocular image sequences. *Image and Vision Computing*, *18 (9)*, 729–737.

[19] Bregler, C. & Malik, J. (1998). Tracking people with twists and exponential maps. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Santa Barbara, CA, USA, 8–15.

[20] Canterakis, N. (1997). Fast 3D Zernike Moments and Invariants. Tech. rep., Institute of Informatics, University of Freiburg, Germany.

[21] Cham, T. & Rehg, J. (1999). A multiple hypothesis approach to figure tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Ft. Collins, CO, USA, II : 239–245.

[22] Chateau, T., Jurie, F., Dhome, M., & Clady, X. (2002). Real-time tracking using Wavelets Representation. In *Symposium for Pattern Recognition, DAGM'02*. Zurich : Springer, 523–530.

[23] Cohen, I. & Li, H. (2003). Inference of Human Postures by Classification of 3D Human Body Shape. In *Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures*. Nice, France.

[24] Cootes, T., Edwards, G., & C.J., T. (2001). Active appareance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 681–685.

[25] D. Comaniciu, V. Ramesh, & P. Meer (2000). Real-time tracking of non-rigid objects using mean shift. *Conference on Computer Vision and Pattern Recognition*, *2*, 142–149.

[26] D. Marimon, Y. Maret, Y. Abdeljaoued, & T. Ebrahimi (2007). Particle filter-based camera tracker fusing marker and feature point cues. In *IS&T/SPIE Conf. on visual Communications and image Processing*. vol. 6508, 1–9.

[27] Dalal, N. & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. San Diego, California, USA, 886–893.

[28] Delamarre, Q. & Faugeras, O. (2001). 3D articulated models and multiview tracking with physical forces. *Computer Vision and Image Understanding*, *81(3)*, 328–357.

[29] Duy Dinh, L. & Shin'ichi, S. (2004). Feature Selection by AdaBoost for SVM-based Face Detection. *Forum on Information Technology*, 183–186.

[30] Elgammal, A. & Lee, C. (2004). Inferring 3D Body Pose from Silhouettes Using Activity Manifold Learning Using Activity Manifold Learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA.

[31] Felzenszwalb, P. & Huttenlocher, D. (2000). Efficient matching of pictorial structures. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Hilton Head, SC, USA.

[32] Fischler, M. & Elschlager, R. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, *22*, 67–92.

[33] Fleuret, F., Berclaz, J., Lengagne, R., & Fua, P. (2008). Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(2), 267–282.

[34] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression : a statistical view of boosting. *Tge Annals of Statistics*, *38*(2).

[35] Gavrila, D. M. (1999). The visual analysis of human movement : A survey. *Computer Vision and Image Understanding*, *73*, 82–98.

[36] Gavrila, D. M. & Davis, L. S. (1995). 3-D model-based tracking of human upper body movement : a multi-view approach. In *Proceedings of the International Symposium on Computer Vision*. Coral Gables , Florida , USA.

[37] Geronimo, D., Lopez, A., Ponsa, D., & Sappa, A. D. (2006). Haar Wavelets and Edge Orientation Histograms for On-Board Pedestrain Detection. In *Proceedings of the 3rd Iberian Conference on Computer Vison and Pattern Recognition*. New-York, USA.

[38] Gorji, A., Shiry, S., & Menhaj, B. (2007). Multiple Target Tracking For Mobile Robots using the JPDAF Algorithm. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. Greece.

[39] Goyat, Y., Chateau, T., Malaterre, L., & Trassoudaine, L. (2006). Vehicle trajectories evaluation by static video sensors. In *9th International IEEE Conference on Intelligent Transportation Systems Conference (ITSC 2006)*. Toronto, Canada.

[40] Grabner, H., Leistner, C., & Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision (ECCV), 2008*. Marseille, France.

[41] Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, *4*(82), 711–732.

[42] Guo, F. & Qian, G. (2006). Dance posture recognition using wide-baseline orthogonal stereo cameras. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*. Southampton, UK.

[43] Hager, G. & Belhumeur, P. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(10), 1025–1039.

[44] Han, B., Comaniciu, D., Zhu, Y., & Davis, L. S. (2007). Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*, 1186–1197.

[45] Isard, M. & Blake, A. (1998). Condensation – conditional density propagation for visual tracking. *IJCV : International Journal of Computer Vision*, *29*(1), 5–28.

[46] Isard, M. & MacCormick, J. (2001). Bramble : A bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision, vol. 2 34-41*. Vancouver, Canada.

[47] J. Klein, C. Lecomte, & P. Miche (2008). Preceding car tracking using belief functions and a particle filter. In *ICPR08, International Conference on Pattern Recognition*. Tampa, Florida, USA, 1–4.

[48] Ju, S., Black, M., & Yacoob, Y. (1996). Cardboard people : a parameterized model of articulated image motion. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*. Killington, Vermont, USA, 38–44.

[49] Jurie, F. & Dhome, M. (2001). Real time template matching. In *International Conference on Computer Vision*. Vancouver, Canada, 544–549.

[50] Kakadiaris, I. & Metaxas, D. (1998). Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, *30(3)*, 192–218.

[51] Karlsson, R. & Gustafsson, F. (2001). Monte carlo data association for multiple target tracking. In *In IEEE Target tracking : Algorithms and applications*.

[52] Khan, Z., Balch, T., & Dellaert, F. (2004). An MCMC-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision (ECCV)*. Prague, Czech Republic, 279–290.

[53] Khan, Z., Balch, T., & Dellaert, F. (2005). MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1805 – 1918.

[54] Kira, K. & Rendell, L. (1992). A practical approach to feature selection. In *Proceedings of the 9th International Workshop on Machine Learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 249–256.

[55] Kohavi, R. & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1-2), 273–324.

[56] Körtgen, M., Park, G.-J., Novotni, M., & Klein, R. (2003). 3d shape matching with 3d shape contexts. In *The 7th Central European Seminar on Computer Graphics*. Budmerice, Slovakia.

[57] Lee, M. W. & Cohen, I. (2004). Human upper body pose estimation in static images. In *Proceedings of the European Conference on Computer Vision*. Prague, Czech Republic.

[58] Lehuger, A., Lechat, P., & Pérez, P. (2006). An adaptive mixture color model for robust visual tracking. In *Proc. Int. Conf. on Image Processing (ICIP'06)*. Atlanta, USA, 573–576.

[59] Leotta, M. & Mundy, J. (2006). Learning background and shadow appearance with 3-d vehicle models. In *British Machine Vision Conference (BMVC)*. Edinburgh, Scotland, vol. 2, 649–658.

[60] Lerasle, F., Rives, G., & Dhome, M. (1999). Tracking of human limbs by multiocular vision. *Computer Vision and Image Understanding*, *75(3)*, 229–246.

[61] Leyrit, L., Chateau, T., Tournayre, C., & Lapresté, J.-T. (2008). Association of AdaBoost and Kernel Based Machine Learning Methods for Visual Pedestrian Recognition. In *IEEE Intelligent Vehicles Symposium (IV 2008)*. Eindhoven, Netherlands.

[62] Lo, C.-H. & Don, H.-S. (1989). 3-d moment forms : their construction and application to object identification and positioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11(10)*, 1053–1064.

[63] Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(20), 91–110.

[64] MacCormick, J. & Blake, A. (1999). A probabilistic exclusion principle for tracking multiple objects. In *Int. Conf. Computer Vision, 572-578*. Kerkyra, Corfu, Greece.

[65] MacKay, D. (2003). *Information Theory, Inference and Learning Algorithms.*. Cambridge University Press.

[66] Mikic, I., Trivedi, M., Hunter, E., & Cosman, P. (2001). Articulated body posture estimation from multi-camera voxel data. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Hawaii.

[67] Moeslund, T. & Granum, E. (2001). A comprehensive survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, *81(3)*, 231–268.

[68] Mori, G. & Malik, J. (2006). Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28 (7)*, 1052–1062.

[69] Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations : Combining segmentation and recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA.

[70] Morris, D. & Rehg, J. (1998). Singularity analysis for articulated object tracking. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. 289–296.

[71] Moutarde, F., Stanciulescu, B., & Breheret, A. (2008). Real-time Visual Detection of Vehicles and Pedestrians with New Efficient AdaBoost Features. In *International Conference on Intelligent RObots and Systems (IROS) workshop*. Nice, France.

[72] Munder, S. & Gavrila, D. (2006). An Experimental Study on Pedestrian Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, *28*(11).

[73] Negri, P., Clady, X., Hanif, S. M., & Prevost, L. (2008). A Cascade of Boosted Generative and Discriminative Classifiers for Vehicle Detection. *Eurasip Journal on Advances in Signal Processing*.

[74] Niculescu-Mizil, A. & Caruana, R. (2005). Obtaining calibrated probabilities from boosting. In *Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI '05)*. Edinburgh, Scotland : AUAI Press.

[75] Oh, S., Russell, S., & Sastry, S. (2004). Markov chain monte carlo data association for multiple-target tracking. In *IEEE Conference on Decision and Control*. Island.

[76] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution Gray-Scale and Rotation invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

[77] Okuma, K., Taleghani, A., de Freitas, N., Little, J., & Lowe, D. G. (2004). A boosted particle filter : Multitarget detection and tracking. In *IEEE, ECCV, 8th European Conference on Computer Vision*. Prague, Czech Republic, vol. 1, 28–39.

[78] Ong, E. & Gong, S. (2002). The dynamics of linear combinations : tracking 3d skeletons of human subjects. *Image and Vision Computing*, *20*, 397 – 414.

[79] P. Perez, C. Hue, J. Vermaak, & M. Gangnet (2002). Color-Based Probabilistic Tracking. In *European Conference on Computer Vision (ECCV)*. Copenhagen, Denmark, vol. 1, 661–675.

[80] P. Pérez, J. & A. Blake (2004). Data fusion for visual tracking with particles. *Proceedings of the IEEE*, *92*(2), 495–513.

[81] Papageorgiou, C. & Poggio, T. (2000). A trainable system for object detection. *International Journal of Computer Vision*, *38*(1), 15–33.

[82] Platt, J. (1999). *Advances in Large Margin Classifiers*, MIT Press, chap. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. 61–74.

[83] Poppe, R. (2007). Evaluating example-based pose estimation : Experiments on the humaneva sets. In *Proceedings of the Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM), at the International Conference on Computer Vision and Pattern Recognition*. Minneapolis, Minnesota, USA.

[84] Poppe, R. & Poel, M. (2006). Comparison of silhouette shape descriptors for example-based human pose recovery. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*. Southampton, UK.

[85] Prati, A., Mikic, I., Trivedi, M., & Cucchiara, R. (2003). Detecting moving shadows : Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 918–923.

[86] Ramanan, D. & Forsyth, D. (2003). Finding and tracking people from the bottom up. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Nice, France.

[87] Read, D. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automation and Control*, *24*, 84–90.

[88] Ripley, B. (1996). *Pattern Recognition and Neural Network*. Cambridge University Press.

[89] Ronfard, R., Schmid, C., & Triggs, B. (2002). Learning to parse pictures of people. In *Proceedings of the European Conference on Computer Vision*. Copenhagen, Denmark.

[90] Rosales, R. & Sclaroff, S. (2000). Inferring body pose without tracking body parts. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Hilton Head, SC, USA, II : 721–727.

[91] Rosales, R., Siddiqui, M., Alon, J., & Sclaroff, S. (2001). Estimating 3d body pose using uncalibrated cameras. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Hawaii, I :821–827.

[92] Salvador, E., Cavallaro, A., & Ebrahimi, T. (2004). Cast shadow segmentation using invariant color features. *Computer Vision and Image Understanding*, *95*(2), 238 – 259.

[93] Sarkka, S., Vehtari, A., & Lampinen, J. (2004). Rao-blackwellized particle filter for multiple target tracking. In *7th International Conference on Information Fusion*. Italy.

[94] Shakhnarovich, G., Viola, P., & Darrell, T. (2003). Fast pose estimation with parameter-sensitive hashing. In *Proceedings of the International Conference on Computer Vision*. Nice, France.

[95] Shen, D. G. & Ip, H. H. S. (1999). Discriminative wavelet shape descriptors for recognition of 2-d patterns. *Pattern Recognition*, *32(2)*, 151–165.

[96] Sigal, L., Bhatia, S., Roth, S., Black, M., & Isard, M. (2004). Tracking loose-limbed people. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA.

[97] Sminchisescu, C., Kanaujia, A., Li, Z., & Metaxas, D. (2005). Discriminative density propagation for 3d human motion estimation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*. San Diego, CA, USA.

[98] Sminchisescu, C. & Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, *22(6)*, 371–391.

[99] Smith, K. (2007). *Bayesian Methods for Visual Multi-Object Tracking with Applications to Human Activity Recognition*. Ph.D. thesis, EPFL, Lausanne, Suisse.

[100] Smith, K. & Gatica-Perez, D. (2004). Order matters : A distributed sampling method for multi-object tracking. In *British Machine Vision Conference (BMVC)*. London, UK.

[101] Suard, F., Rakotomamonjy, A., Bensrhair, A., & Broggi, A. (2006). Pedestrian detection using infrared images and histograms of oriented gradients. In *Proceedings of the IEEE Conference of Intelligent Vehicles (IV)*. Tokyo, Japan, 206–212.

[102] Sun, Y., Bray, M., Thayananthan, A., Yuan, B., & Torr, P. H. S. (2006). Regression-based human motion capture from voxel data. In *Proceedings of the British Machine Vision Conference*. Edinburgh, Scotland.

[103] Tangkuampien, T. & Suter, D. (2006). Real-time human pose inference using kernel principal component pre-image approximations. In *Proceedings of the British Machine Vision Conference*. Edinburgh, Scotland.

[104] Taylor, C. J. (2000). Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Computer Vision and Image Understanding*, *80*, 349–363.

[105] Teague, M. (1980). Image analysis via the general theory of moments. *Journal of the Optical Society of America*, *70(8)*, 920–930.

[106] Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P., & Cipolla, R. (2006). Multivariate relevance vector machines for tracking. In *European Conference on Computer Vision*. Graz, Austria.

[107] Tipping, M. (2000). *Advances in Neural Information Processing Systems*, MIT Press, chap. The relevance vector machine.

[108] Tipping, M. (2004). Bayesian inference : An introduction to principles and practice in machine learning. In *Advanced Lectures on Machine Learning*. 41–62.

[109] Tresadern, P. A. & Reid, I. D. (2007). An evaluation of shape descriptors for image retrieval in human pose estimation. In *Proceedings of the British Machine Vision Conference*. Warwick, UK.

[110] Tuzel, O., Porikli, F., & Meer, P. (2007). Human detection via classifiction on riemannian manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, USA.

[111] Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

[112] Vermaak, J., Godsill, J., & Pérez, P. (2005). Monte carlo filtering for multi-target tracking and data association. *IEEE Transactions on Aerospace and Electronic Systems*, *41*, 309–332.

[113] Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Hawaii, vol. 1, 511–518.

[114] Viola, P. & Jones, M. (2001). Robust Real-time Object Detection. In *Second International Workshop on statistical and computational theories of vision-modeling, learning, computing, and sampling*.

[115] Viola, P., Jones, M., & Snow, D. (2003). Detecting Pedestrians Using Patterns of Motion and Appearance. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*. Nice, France.

[116] Wachter, S. & Nagel, H.-H. (1999). Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, *74(3)*, 174–192.

[117] Wang, J. J. & Singh, S. (2003). Video analysis of human dynamics - a survey. *Real Time Imaging*, *9*, 321–346.

[118] Werghi, N. (2005). A discriminative 3d wavelet-based descriptors : Application to the recognition of human body postures. *Pattern recognition letters*, *26(5)*, 663–677.

[119] Williams, O., Blake, A., & Cipolla, R. (2003). A sparse probabilistic learning algorithm for real-time tracking. In *Int. Conf. of Computer Vision*. Nice, France, 353–361.

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

[121] Yao, J. & Odobez, J.-M. (2008). Multi-camera multi-person 3D space tracking with mcmc in surveillance scenarios. In *European Conference on Computer Vision workshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCV-M2SFA2)*. Marseille, France.

[122] Y.D. Wang, J. & A. Kassim (2007). Adaptive particle filter for data fusion of multiple cameras. *The Journal of VLSI Signal Processing*, *49*(3), 363–376.

[123] Yu, Q., Medioni, G., & Cohen, I. (2007). Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, Minnesota, USA, 1 – 8.

[124] Zadrozny, B. & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proc. 18th International Conf. on Machine Learning*. Williamstown, MA, USA : Morgan Kaufmann, San Francisco, CA, 609–616.

[125] Zhan, B., Monekosso, D. N., Remagnino, P., Velastin, S. A., & Xu, L.-Q. (2008). Crowd analysis : a survey. *Machine Vision and Applications*, *19*(5-6), 345–357.

ANNEXE

A

LISTE D'ARTICLES LES PLUS REPRÉSENTATIFS

Ce chapitre est un recueil de quelques articles que j'ai écrits ou co-écrits sur la période 2001 – 2009. Il accompagne les deux chapitres précédent en reprenant, en détails, certains travaux effectués.

A.1 Estimation de postures

A regression-based approach to recover human pose from voxel data
L. Gond, P. Sayd, T. Chateau and M. Dhome.
Second IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS2009), Septembre 2009, Kyoto, Japon

A regression-based approach to recover human pose from voxel data

Laetitia Gond Patrick Sayd
CEA LIST, Embedded Vision Systems Laboratory,
Point Courrier 94, Gif-sur-Yvette, F-91191 France
first.last@cea.fr

Thierry Chateau Michel Dhome
LASMEA CNRS
Clermont-Ferrand, France
first.last@lasmea.univ-bpclermont.fr

Abstract

This paper deals with human body pose recovery from multiple cameras, which is a key task in monitoring of human activity. This regression-based approach relies on a 3D description of a body voxel reconstruction, combined with a decomposition of the estimation, which allows to recover a wide range of poses using synthetic training data. The precision of the proposed shape descriptor is quantitatively evaluated on synthetic data for a ground truth comparison, while the effectiveness of the whole system is qualitatively demonstrated on various real sequences.

1. Introduction

Human pose analysis from images is a challenging task due to both the complexity of human body (as a result of the high number of degrees of freedom of the body and the variability of human appearance) and the visual ambiguities inherent to the use of image projection (lack of depth information, self occlusions,etc.). However, the number of potential applications, such as virtual reality, human-computer interaction or athletes' gesture analysis, has intensified the interest for this topic within the computer vision community.

The pose recognition process presented in this paper addresses applications such as visual surveillance, video monitoring, smart Human Machine Interface, telehealthcare, etc. One important goal of such systems is to achieve human behavior analysis and automatically detect potential alarm situations such as unusual motion, falls, immobility or some particular gestures. More precisely, we describe here a method to recover the pose of a standing person moving inside a room equipped with a calibrated multi-camera setup. The output of the system is the configuration of the body, in the form of the set of angular parameters of an articulated body model. This method comes as a complement to systems such as the one described in [14] , which classifies a set of basic postures (as standing, sitting, lying down,etc.), and it represents a first step towards automatic motion interpretation. Indeed a further analysis of the output parameters and their time evolution could allow the recognition of some particular gestures or actions. This is a learning-based method, which is static and model-free. Without any prepocessing step, the system would automatically recognizes the pose of any subject entering the room; neither pose initialization (in the first frame of the sequence), nor body model adjustment are required. In contrast with some other learning-based approaches, we assume no restriction on the motion being performed, except that the person is standing.

The main contributions of this work are:

- a 3D shape descriptor, which enables the system to recover various complex poses. This descriptor will be compared to the 3D Shape Context proposed in [20] on walking synthetic data,
- a method to decompose the estimation of body degrees of freedom (DOF), allowing to extend the amount of poses that can be recognized by the system and to use synthetic databases general enough to analyse a large set of motions on real sequences.

1.1. Related works

Over the last two decades, the problem of recovering human pose from images has received a growing attention and two main approaches have emerged. *Model-based approaches* define an explicit body model and maximize a likelihood function measuring how well a model configuration fits with image observations. Due to the high dimensionality of the problem (number of DOF of the body) and the presence of local maxima inherent to visual ambiguities, the function to be optimized is very complex. Model-based methods are then generally accurate but also computationally expensive, and often restricted to a tracking framework, and hence encounter the problem of pose initialization in the first frame of the sequence (or re-initialization in case of lost tracks). These limitations motivate the use of *model-free approaches*, which generally rely on a pre-constructed database containing image-pose examplars. Example-based approaches explicitly store the collection of examplars and, given a new input, search for similar samples in the database

to interpolate a pose estimate [18, 15]. In contrast, learning-based approaches use an off-line training stage to generalize database properties, for example by learning a manifold of admissible poses [6, 11] or a prior distribution [12]. In particular, regression-based methods learn a compact mapping from image features to pose space. Many of them rely on a prior silhouette segmentation through background subtraction [17, 3, 20], but recently some works extended their use to unsteady environments and cluttered background using local features such as histograms of oriented gradients [2, 13] or Haar features [5].

As the main part of the modeling computation is done during an off-line training phase, regression-based methods allow a direct prediction of the pose from low-level image features, without any prediction on a high-level body model. As a result, they are generally faster and work on static images. However, these approaches suffer from their lack of generality, in the sense that they are limited to poses similar to the training data. The database must contain any type of pose that has to be recognized. A lot of previous works on regression-based pose estimation are restricted to some predefined motion such as walking [3, 20, 5, 13] or upper body gestures of a static person facing the camera [2], or other actions learnt on very similar sequences (as proposed in HumanEva [19]). Thus they assume a prior knowledge of the type of movement being observed. The number of DOF of the body is too high to construct a training database for general motion type. This restriction makes them inappropriate for the kind of application we aim at. In a surveillance system, the construction of a database containing all the actions to be detected would be an intractable computational task. Moreover, if the type of action to be recognized has to be known in advance, the method becomes unsuitable for motion interpretation.

In this paper we propose a method to reduce this limitation. We increase the number of recognized poses by decomposing the estimation into several subproblems. The orientation of the body is first estimated, and we then separate pose estimation of different body parts (the upper and lower body pose). The estimation does not assume any prior knowledge of the motion being performed.

1.2. Outline of the system

Our system handles the case where a person is standing (walking, performing gestures), and moving inside the common field of view of several calibrated cameras (we use a 4-camera setup in our experiments). Our objective is to robustly estimate the pose, i.e. the angular configuration of body joints, with an accuracy good enough to animate an avatar and reproduce its motion along time.

The pipeline of our system is given in figure 1. For the training phase, examples are generated from synthetic data in order to get ground truth on the body configuration (joint angles). Avatars are animated using the software POSER 6, and binary silhouettes images from several viewpoints are rendered using 3D Studio Max. A 3D reconstruction of the body shape is then computed, and encoded with our 3D shape descriptor. The regression process learns the mapping from the descriptor to the body pose. In the testing phase (red lines on figure 1), silhouettes are extracted through background subtraction, and the 3D shape descriptor is directly given as input to the trained regressor, which predicts a pose estimate.

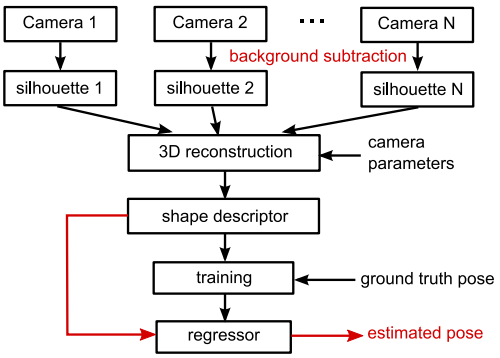


Figure 1. Overview of the system.

Silhouettes can be relatively robustly extracted from images when camera setup and background are reasonably static, as in our case. As the cameras are calibrated, we choose to combine multi-view information by reconstructing the 3D visual hull of the body. Working on this 3D shape makes the estimation more independent of the camera setup (number and position of cameras). In particular, the regressor needs not necessarily to be relearnt each time the camera setup is changed [20]. A 3D voxel-based reconstruction is achieved through a shape-from-silhouettes algorithm similar to [7].

This paper is organized as follows: section 2 describes our 3D Shape Description. Our method to estimate human pose from this descriptor is presented in section 3. Section 4 reports quantitative comparison with a relevance reference [20] on synthetic data, and qualitative results on real sequences of several motion types.

2. Shape description

Shape description is of crucial importance in our method. Its role is to encode the geometry of the voxel reconstruction in a compact vector, that will given as input to the regressor. We require the representation to be translation and scaling-invariant, but rotation-dependent as the body orientation has to be estimated by the system. In the particular case of the human body, it has to cope with the high variability of people appearance, as a result of differences in size, corpulence,

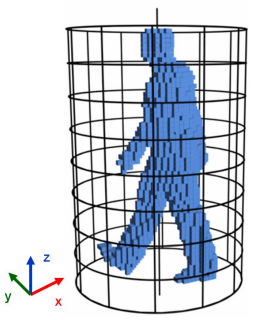


Figure 2. Reference cylinder of a 3D voxel reconstruction.

morphology, clothing, etc. Many 2D shape descriptors were proposed in the literature to estimate the pose the from 2D silhouettes (Hu moments, [17], histograms of Shape Context [3], Fourier descriptors [16]), and some of them have been adapted to 3D shape description, such as histograms of 3D Shape Context [20], wavelet-based descriptors [22] or Cohen's descriptor proposed in [8].

We use in our study an intuitive shape description based on the encoding of voxel distribution in a vertical cylinder centered on the body center of mass. Cohen et al. [8] make use of a cylindric reference shape to construct their shape descriptor, but in a different way (in this work, shape description is used for classification of a predefined set of postures and did not appear to be adapted to our regression problem). Given a 3D voxel silhouette, we define a reference cylinder (see fig. 2), with main axis being the vertical axis passing through the center of mass of the shape, and with radius being proportional to the height of the body. For each horizontal cross section of the shape, the circle defined by this cylinder is split into a grid consisting of shell-sector bins (see fig. 3(a)). A 2D shape histogram is computed by counting the number of voxels falling inside each bin (fig. 3(b)). The height of the voxel shape is then divided into n_{Slices} equal slices and we compute the mean histogram of the sections contained in each slice. The concatenation of these mean histograms gives the 3D feature vector. The descriptor is then normalized by dividing the vector by the total number of voxels of the reconstruction, so that each component represents the proportion of voxels localized in a region of the cylinder. To smooth the effects of spatial quantization and ensure the continuity of the description with regards to poses changes, a soft voting process is employed, so that a voxel lying near a sector boundary also votes for neighbor sectors.

We use 3 divisions along the radial direction (which was experimentally proven to be sufficient), and adjust the size of bin-graduations to the morphology of the body as follows:

- the inner radius is set proportional to the estimated corpulence (the ratio between the volume of the voxel reconstruction and the size of the body), and chosen to approximate waist half-width,
- the external radius is proportional to the height of the body in order that all voxels can be contained in the cylinder if legs or arms are spread,
- the intermediate radius is chosen to be halfway between the inner and the external radius.

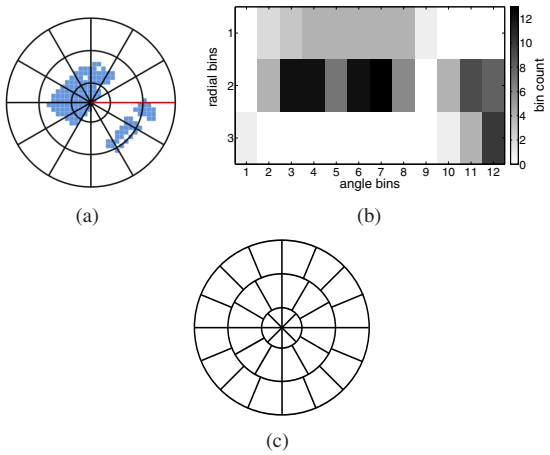


Figure 3. 2D shape histograms. *(a)* a horizontal voxel layer and *(b)* its corresponding 2D shape histogram. *(c)* optimal layout of angular bins.

The optimal choice for the number of angular and vertical divisions was experimentally evaluated on synthetic data. As the area of angular sectors differs depending on the radial division, we kept the possibility of fixing different numbers of angular sectors on each radial division. The optimal number of vertical divisions was found to be $n_{Slices} = 8$, and for angular divisions resp. 8, 12 and 16 along the 3 radial divisions (fig. 3(c)).

3. Human pose inference

3.1. Pose representation

As explained in section 1.2, our method is based on a modeling of the relationship between a feature vector encoding the geometry of the 3D reconstruction and a vector describing body pose. The pose is represented by a vector containing the angular parameters of the body skeleton. We use the kinematic structure of POSER 6 default avatar as shown on figure 4, and consider in our experiments the following DOF (see figure 4): shoulders (3 DOF each), forearms (1 DOF each), hips (3 DOF each) and knees (1 DOF each). As we assume the person is standing, body orientation is encoded with one overall azimuth angle (torso orientation around the world vertical axis z).

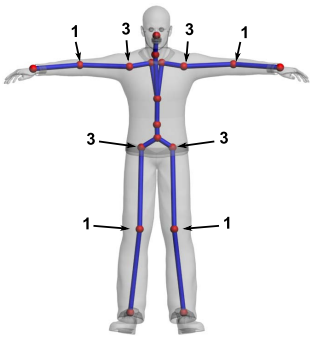


Figure 4. DOF of the skeleton. The arrows indicate the joints employed in the body model and the number of DOF per joint.

3.2. Regression

We use linear models to relate an image descriptor $\mathbf{x}$ to the 3D joint configuration $\mathbf{y}$: the mapping from feature to pose space is approximated by a weighted linear combination of (possibly non-linear) basis functions:

$$\mathbf{y} = \sum_{m=1}^{M} \mathbf{w_m} \phi_m(\mathbf{x}) = \mathbf{W}\mathbf{f}(\mathbf{x}) \tag{1}$$

where the weight vectors $\mathbf{w_m}$ are optimized from training data $\{(\mathbf{x_n}, \mathbf{y_n})\}_{n=1}^{N}$ during the learning stage. A regularization term $R(.)$ is commonly added in the error function to control overfitting:

$$\mathbf{W} = \arg\min_{\mathbf{W}} \left\{ \sum_{n=1}^{N} \|\mathbf{y_i} - \mathbf{f}(\mathbf{x_i})\|^2 + R(\mathbf{W}) \right\} \tag{2}$$

In our study we used gaussian kernels for the basis functions: $\phi_m(\mathbf{x}) = K(\mathbf{x}, \mathbf{x_m}) = e^{-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x_m}\|^2}$.

Sparse learning algorithms have been proposed to select a small subset of the training data as the informative supports to the basis functions. Support Vector Machine (SVM) regression makes use of an ϵ-insensitive loss function to achieve sparsity. Relevance Vector Machine (RVM) is a Bayesian method that results in taking a prior of the form $p(\mathbf{W}) \sim \prod_l \|\mathbf{w_l}\|^{-\nu}$ and using the Automatic Relevance Determination (ARD) principle to select relevant vectors in the linear model. Some formulations have been introduced to handle multi-dimensional outputs: Multivariate Relevance Rector Machine (MVRVM) in [20] and the MAP approximation proposed in [3].

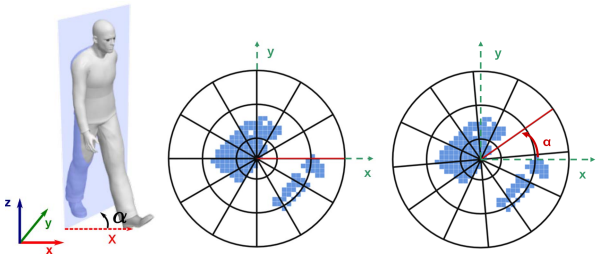
We conducted experiments with several learning algorithms on synthetic data and compared their accuracies. As in [3], the best performance was obtained with a SVM regression (we used SVM-Torch [9] for implementation), while RVM models give more sparsity. Random selecting of a subset of training examples (about 20%) gave a reasonable accuracy while having a very low computational cost for training (consisting in the inversion of a $M \times M$ matrix, where M is the number of basis functions).

3.3. Decomposing the estimation to recognize more poses

As said in introduction, the main drawback of learning-based methods is the restriction on the type of poses that can be recognized. The training data must account for all the configurations of the body that will be considered in the estimation. As the approximated mapping is complex and highly non-linear, the pose space must also be densely sampled. The number of samples needed to describe the space of an object configurations grows exponentially with its dimension, i.e. the number of DOF of this object. In the case of the human body, the set of all feasible configurations is extremely large, and the construction of a database encounters a combinatorial problem. The idea we propose is to decompose the estimation into several subproblems, reducing significantly the amount of training data required to recognize a large set of body poses.

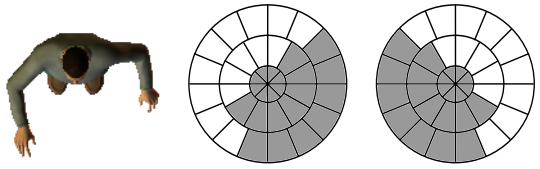
The first problem is the case of body orientation. As we want to recover joint configuration for any body orientation, the database should in theory contain an example of each pose with every possible orientation. If the body orientation was known, we could compute a kind of "rotation-normalized" descriptor, i.e. align the shape descriptor with body orientation, and consider only in the regression process body internal angles, as we would do if the orientation was fixed. In our system, the shape descriptor is therefore computed in two steps. The first shape descriptor is aligned with the world coordinate frame: its reference line is taken parallel to the x-axis (in red on figure 5(b)). This descriptor is employed to estimate the body orientation α (torso orientation with regards to the world vertical axis, see figure 5(a)). A second descriptor is then computed, this time aligned with body orientation (figure 5(c)). This descriptor is used to estimate body joint angles. Finally, two regression steps are employed: one with a world-aligned descriptor to estimate body orientation, and another one with a body-aligned descriptor to estimate joint angles. To take account for errors in the estimation of the orientation, we add a random noise in the orientation of the descriptor of training examples in the second step ($\pm 10^\circ$).

In the general case of an unconstrained motion (see last paragraph of section 4.2), we also learn separate regressors for arm and leg motions. Leg and arm movements of a standing person can indeed be considered independently. As they are localized in different part of the reference cylinder (legs are the lower part and arms in the upper), they are represented by disjoint components in the feature vector. This process allows to extend the set of poses that can be recognized by the system. This set is not only limited to the poses stored in the database, but all the poses com-



(a) (b) (c)

Figure 5. Alignment of the shape descriptor with body orientation α. *(a)* torso orientation in the world coordinate frame. *(b)* descriptor aligned to the world x-axis. *(c)* descriptor aligned with torso orientation.



(a) (b) (c)

Figure 6. Components of the shape descriptor employed to estimate arm position. *(a)* top view of an avatar and the corresponding components employed to estimate *(b)* left arm position and *(c)* right arm position.

posed of the combinations of leg and arm movements are taken into account. As the descriptor (in the second step) is aligned with body orientation, we can also identify relevant components to estimate the left /right arm position. These components are represented on figure 6. Although they cannot be considered as really independent (since the selected sectors are overlapping), we experimentally noticed an improved accuracy by separating left and right arm estimation. This process removes uninformative components that can be seen a noise in the regression process.

4. Experimental results

4.1. Comparison with 3D Shape Context on synthetic data

Histograms of Shape Context were first employed in [3] to recover body pose from 2D binary silhouettes extracted from monocular images, and also adapted in [10] to estimate hand pose from multiple views. This representation consists in assigning to each sampled point along the contour of the silhouette a histogram representing the distribution of other contour points in a local neighborhood. Clustering is then performed in the Shape Context space to reduce the distributions of all points of a silhouette to a second histogram forming the final feature vector (see [4] and [3] for more details). According to the authors, this descriptor takes advantage of its locality, leading to a better robustness to noise and segmentation errors. Nevertheless, the experiments conducted in [21] tend to prove that, despite its computational complexity, it offers very little benefits over some alternative simpler methods such as Discrete Cosine Transform or Lipschitz embeddings.

	full body	body heading angle	left shoulder	right hip
[3]	6.0	17	7.5	4.2
[20]	5.2	8.8	6.3	3.2
ours	3.0	4.6	3.7	2.8
two step	2.5	-	3.4	2.1

Table 1. RMS errors (in degrees) over the 418 examples of the test sequence. The first and second rows show the results obtained resp. in [3] with monocular estimation and in [20] with a 6 camera setup and 3DSC. The third and fourth rows show comparative results between our method with single step or two step estimation.

An extension of this description to 3D shapes was recently proposed in [20] (3D Shape Context - 3DSC) and employed to estimate body pose from voxel data in a regression-based framework very similar to ours. Here we propose to compare the performance of our shape descriptor with the 3DSC on synthetic data. We used the same spiral-walking MOCAP data as [20] (publicly available at [1]) to animate POSER 6 default avatar, and reconstructed the 3D visual hull with a similar camera setup, consisting in 6 circularly distributed viewpoints. In our case, the pose was estimated with a SVM regressor. Experiments were conducted both with a single step regression (all angles are computed at the same time from a world-aligned shape descriptor) or with a two step estimation (orientation is computed first and body internal angles are estimated with the oriented shape descriptor). Table 1 shows comparative results on the 418 test examples. We report *Root Mean Square* (over time) absolute difference errors between the true and estimated joint angles, for both the mean over all body DOF and some key body angles (body orientation, left shoulder and right hip). A significative improvement is achieved with our method. The comparison between the last two rows also shows that the two-step estimation is more accurate.

These experiments were conducted on perfect synthetic data (i.e. 3D reconstructions obtained from clean silhouette images). As the 3DSC description is based on the surface voxels, we believe that its performance could significantly degrade on noisy voxel reconstructions (as confirmed by the experiments of [21] in the 2D case).

4.2. Real images

Experiments on real data were carried out on sequences captured by a 4-camera system. The main difficulty in our approach is to achieve an accurate pose estimation from noisy silhouette reconstructions of real persons, despite

Figure 7. The 8 avatars used to generate synthetic training data.

building our training database on perfect 3D synthetic silhouettes. However, this approach allows in return more flexibility on the building of training databases. We included in our databases several avatars (shown in figure 7) to train the regressor to be more robust to clothing and corpulence variations. In the following experiments, frames are processed independently, i.e. no use is made of any temporal coherence between successive images of a sequence. For each test, a linear model (with randomly selected support vectors) was trained on a synthetic database.

Walking motion. A dataset of 2952 training examples was synthesized using the same MOCAP data as in section 4.1. Examples of training poses are given in the first row of figure 8. For this test, the pose was estimated through a two-step regression: torso orientation was estimated first, and the other angles were computed from a rotation-normalized descriptor. Some sample results are shown on figure 8. An avatar was animated with the estimated angles, and rendered with the same viewpoint as one of the cameras. Figure 12 shows the estimated angles for orientation and left hip over 200 frames of a circular walking sequence. Simple walking motion is relatively easy to process because of the correlation between arm and leg motions. Common errors are imprecisions in the estimated orientation, and rarely a $\pm 180^\circ$ turnaround (see frame 74 on the graph 12(a)), due to the symmetry ambiguity of the visual hull.

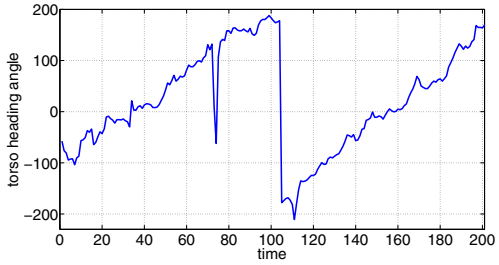
Upper body motion. In the second experiment, we considered the motion of a fixed person (with a fixed and known orientation) performing arm gestures. In the case of upper body motions, MOCAP data cannot easily be used to synthesize a database that is general enough to recognize any arm movement. We therefore constructed with POSER training examples by randomly fixing angle values in some reasonable intervals (5000 examples). The 8 degrees of freedom and their possible values are summarized in table 2. Some examples of generated poses are given in figure 8 (second row). As the orientation is not estimated, left/right arm positions are computed through a single step regression, using separate components of the upper part of the shape descriptor (as in figure 6). In this case, the assumption of a known orientation allows the system to achieve a good accuracy. In the figure 10, the estimated skeleton is overlayed on some sample test images.

rotation	angle values
right shoulder twist	$[-90, 80]$
right shoulder bend	$[-30, 80]$
right shoulder front-back	$[-40, 90]$
left shoulder twist	$[-90, 80]$
left shoulder bend	$[-80, 30]$
left shoulder front-back	$[-90, 40]$
right forearm bend	$[-20, 120]$
left forearm bend	$[-120, 20]$

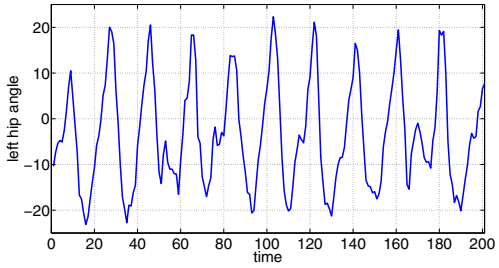
Table 2. Degrees of freedom and intervals of angle values (in degrees) used in the training database for upper body poses.

Combination of both movements. This paragraph handles the general case where the subject can perform any motion, i.e. can have any body orientation, and freely move its arms and legs. Training data were synthesized combining the arm and leg motion data of the two previous paragraphs, and fixing a random orientation for each example. Some examples are shown in the third row of figure 8. The database contains about 8000 training examples. Here the pose was estimated with the whole method described in section 3.3. The orientation is estimated first, and used to compute a body-aligned descriptor, and leg and arm poses are estimated independently with separate components of the shape descriptor. Figure 11 illustrates the results on real images. Our system captures the general appearance of the subject pose quite well, while working on static images at a low computational cost. The precision could be improved using the result as an initialisation to a model-based refinement.

Figure 8. Example poses rendered in training data. *First row*: walking motion. *Second row*: gestures. *Third row*: combination of both.



(a)



(b)

Figure 12. Estimated values (in degrees) of *(a)* torso orientation angle and *(b)* left hip angle along a real walking sequence.

5. Conclusion

This paper proposed a regression-based process to recover human pose from the voxel data reconstructed from a multi-camera setup. This method relies on the use of a 3D shape descriptor, combined with a decomposition of the estimation into several subproblems, allowing to recognize a wide range of poses on real sequences using synthetic training data. The accuracy of our shape descriptor was demonstrated on synthetic walking data, and we presented promising results on various real sequences, even with complex motions. However, the performance of the system remains highly dependent on the quality of the extracted silhouette images. The method proved its effectiveness on simple real sequences, in which silhouette pixels can easily be discriminated from the background, but may lack of robustness in more realistic situations. Future work will therefore focus on improving the quality of the (2D and 3D) silhouettes in difficult conditions, mainly by considering multi-camera cues.

References

[1] www.ict.usc.edu/graphics/animweb/humanoid.

[2] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *ACCV*, 2006.

[3] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *PAMI*, 28(1):44–58, January 2006.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24:509–522, 2002.

[5] A. Bissacco, M.-H. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *CVPR*, 2007.

[6] M. Brand. Shadow puppetry. In *ICCV*, pages 1237–1244, 1999.

[7] K. M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *CVPR*, 2000.

[8] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *Int. Workshop on Analysis and Modeling of Faces and Gestures*, 2003.

[9] R. Collobert and S. Bengio. Svmtorch: Support vector machines for large-scale regression problems. *JMLR*, 1:143–160, 2001.

[10] T. E. de Campos and D. W. Murray. Regression-based hand pose estimation from multiple cameras. In *CVPR*, 2006.

[11] A. Elgammal and C. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, 2004.

[12] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV*, 2003.

[13] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *ECCV*, 2008.

[14] Q.-C. Pham, Y. Dhome, L. Gond, and P. Sayd. Video monitoring of vulnerable people in home environment. In *Proc. of the 6th Int. Conf. On Smart homes and health Telematics, Ames, IOWA, June 28-July 2*, 2008.

[15] R. Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. In *EHuM, CVPR*, 2007.

[16] R. Poppe and M. Poel. Comparison of silhouette shape descriptors for example-based human pose recovery. In *FG*, 2006.

[17] R. Rosales and S. Sclaroff. Inferring body pose without tracking body parts. In *CVPR*, 2000.

[18] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *ICCV*, 2003.

[19] L. Sigal and M. J. Black. Humaneva : Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical report, Brown University, Department of Computer Science, 2006.

[20] Y. Sun, M. Bray, A. Thayananthan, B. Yuan, and P. H. S. Torr. Regression-based human motion capture from voxel data. In *BMVC*, 2006.

[21] P. A. Tresadern and I. D. Reid. An evaluation of shape descriptors for image retrieval in human pose estimation. In *BVMC*, 2007.

[22] N. Werghi. A discriminative 3d wavelet-based descriptors: Application to the recognition of human body postures. *PRL*, 26(5):663–677, 2005.

Figure 9. Pose reconstructions on real images: walking motion. *First row:* test images from one of the 4 cameras. *Second row:* examples of voxel reconstructions. *Third row:* estimated poses from the same viewpoint.

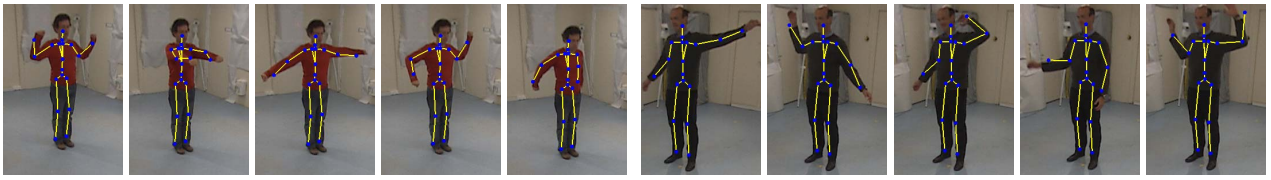


Figure 10. Pose reconstructions on real images: upper body motion.

Figure 11. Pose reconstructions on real images: free motion.

A.2 Suivi d'objets planaires basé apprentissage

Realtime Kernel based Tracking
T. Chateau et J.T. Lapresté
Electronic Letters on Computer Vision and Image Analysis, Vol. 8 (1), pp 27-43, 2009

Electronic Letters on Computer Vision and Image Analysis 8(1):27-43, 2009

Realtime Kernel based Tracking

T. Chateau* and J.T. Lapresté*

* *Lasmea, CNRS/Blaise-Pascal University, Clermont-Ferrand, France*

Received 26 May 2008; accepted 13 January 2009

Abstract

We present a solution for realtime tracking of a planar pattern. Tracking is seen as the estimation of a parametric function between observations and motion and we propose an extension of the learning based approach presented simultaneously by Cootes and al. and by Jurie and Dhome. We show that the hyperplane classic algorithm is a specific case of a more generic linearly-weighted sum of fixed non-linear basis functions model. The weights associated to the basis functions (kernel functions) of the model are estimated from a training set of perturbations and associated observations generared in a synthetic way. The resulting tracker is then composed by several iterations on trackers learned with coarse to fine magnitude of perturbations. We compare the performance of the method with the linear algorithm in terms of accuracy and convergence frequency. Moreover, we illustrate the behaviour of the method for several real toy video sequences including different patterns, motions and illumination conditions, and for several real video sequences sampling from rear car tracking databases.

Key Words: realtime planar tracking, kernel based regression functions, rear-car tracking

1 Introduction

Tracking a planar textured pattern is a popular topic in computer vision [2]. The aim is to estimate the unknown motion (generally expressed by a homography) of the pattern observed into a video sequence. Solutions to this problem are classified according to the video information available at the estimation time. First approaches, called *offline* use all the video information to solve the motion estimation problem. In this case, global optimization techniques can be applied. In [6], the author presents an optimization solution to pedestrian trajectories estimation from a video sequence. Moreover, in [16], a multiple target tracking system is proposed, into a global Monte-Carlo framework. Second approaches are called *online* and only past and present video information are available. Real-time tracking are *online* methods where the result of the motion estimation must be produced before the next video frame. The method presented in this paper is a real-time method.

Generally, real-time tracking can be modelized with a regression function between observations (the image) and a motion model of the template to be tracked. Some methods (called model based approaches) rely on an analytical model of the regression function. In [8], the authors compute a first order approximation of the relation between the observations (luminance) and the motion model and propose to use the Jacobian matrix

Correspondence to: <Thierry.CHATEAU@univ-bpclermont.fr>

Recommended for acceptance by <Thomas Breuel>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

to define the regression function. Recently, Benhimane et al. [3] extend the method to a second order approximation of the Hessian matrix. Other methods (called learning based approaches), use a parametric form for the regression function where the parameters are estimated from a training set of motions and associated observations. Jurie and Dhome and [10] and Cootes and al. [4] have simultaneously proposed a first order hyperplane model for the regression function where the parameters of the linear matrix between motion and observations are estimated using a least-square error criteria computed from a training set. Generally, image features used are normalized luminance of pixels. However, In [11], Chateau et al. use Haar-like wavelets to describe the image. We propose to extend the learning based approach presented simultaneously by Jurie and Dhome [10] and Cootes and al. [4] to a parametric regression model using non linear basis functions. Kernel based regression functions have been used recently for tracking objects but using simple motion models (translation/scale). The pioneering work is the one of Avidan [1] (*support vector tracking*) which uses the output of an SVM based regression function to perform a tracking task. The idea is to link the SVM scores with the motion of the pattern between two images shots. This method provides a way to track classes of objects. No model of the current object is learnt but the classifier uses a generic model learnt offline. Williams [15] proposes a probabilistic interpretation of SVM. He presents a solution based on RVM (relevance vector machine) ([13]), combined with a Kalman temporal filtering. RVM is used to link the image luminance measure to the relative motion of the object with a regression relation. Recently, Thayananthan and al. [12] have presented a learning based approach to track articulated human body motion extending the RVM kernel based machine to multivariate MVRVM.

This paper is organized as follows. Section two deals with the general framework of visual tracking. Section three presents the regression function proposed: a kernel based parameter model where parameters are estimated from a training set. The resulting solution is compared with the classical one (hyperplane approximation) in section four, in terms of accuracy and convergence frequency related to several relevant parameter such as the magnitude of perturbations chosen for the learning step or noisy obsevations. Moreover, we illustrate the behaviour of the method for several real toy video sequences including different patterns, motions and illumination conditions, and for several real video sequences sampling from rear car tracking databases.

2 Visual Tracking

We present a generic solution for pattern tracking based on the definition on a regression function between a motion model and the variation of the template appearance.

Let us define $\mathtt{I}_k$ be an image extracted from a video sequence at time k, and $\mathcal{W}$ a planar pattern to be tracked, defined by four corner points into the image. Let us define a state associated to the image position of the pattern by:

$$\mathbf{p}_k \doteq (\mathbf{p}_k^1, \mathbf{p}_k^2, \mathbf{p}_k^3, \mathbf{p}_k^4),$$

a vector composed by the four corners of the pattern, where $\mathbf{p}_k^i = (u_k^i, v_k^i)^t$ denotes the position of $\mathbf{p}_k^i$ expressed into the image based reference frame.

Temporal matching of $\mathcal{W}$ can be seen as the estimation $\hat{\mathbf{p}}_k$ of the state system, for each new image of the video sequence. It can be realized in a iterative way, from the motion estimation $\boldsymbol{\delta}_{\mathbf{p}k}$ of the four template corners between two successive images:

$$\hat{\mathbf{p}}_k = \mathbf{p}_{k-1} + \boldsymbol{\delta}_{\mathbf{p}k} \tag{1}$$

Let $\mathbf{z}(\mathtt{I}_k, \mathcal{W}, \mathbf{p}_k)$, be an observation function which provides a feature vector associated to $\mathcal{W}$ for the position defined by the state $\mathbf{p}_k$ in the image at time k (for example the luminance computed on a sub-sampling grid of $\mathcal{W}$). A direct consequence of the so-called image constancy assumption can be expressed as follows:

$$\forall i, j \in \{1, .., K\},\ \mathbf{z}(\mathtt{I}_i, \mathcal{W}, \mathbf{p}_i) = \mathbf{z}(\mathtt{I}_j, \mathcal{W}, \mathbf{p}_j) = \mathbf{z}^*_{\mathcal{W}} \tag{2}$$

with K is the number of images of the video-sequence. $\mathbf{z}^*_{\mathcal{W}}$ is a feature vector extracted to the pattern in first image.

Now, the variation of the observations between two successive images, given the previous state, is defined by:

$$\boldsymbol{\delta}_{\mathbf{z}k} \doteq \mathbf{z}(\mathtt{I}_k, \mathcal{W}, \mathbf{p}_{k-1}) - \mathbf{z}(\mathtt{I}_{k-1}, \mathcal{W}, \mathbf{p}_{k-1}) \tag{3}$$

Using (2), the previous equation becomes:

$$\boldsymbol{\delta}_{\mathbf{z}k} = \mathbf{z}(\mathtt{I}_k, \mathcal{W}, \mathbf{p}_{k-1}) - \mathbf{z}^*_{\mathcal{W}} \tag{4}$$

We propose to link motion $\boldsymbol{\delta}_{\mathbf{p}k}$ and observation variation $\boldsymbol{\delta}_{\mathbf{z}k}$ by a regression function:

$$\boldsymbol{\delta}_{\mathbf{p}k} = \mathbf{f}(\boldsymbol{\delta}_{\mathbf{z}k}; \mathbf{w}_k) + \boldsymbol{\epsilon}_k \tag{5}$$

where $\boldsymbol{\epsilon}_k$ denotes a random noise and $\mathbf{w}_k$ is the vector of parameters of the regression function. Since this formulation depends on time (k), parameters must be estimated at each iteration. The idea is to find a new relation, in which the parameters of the regression function have to be estimated only for the first image of the sequence.

Let $\mathtt{H}_k$ be the homography between the position of the four corners of the pattern to be tracked and a canonical reference frame. $\mathbf{p}_k$ is projected into $\mathbf{P}_k = \mathbf{P}_0 = \left((0,0)^t, (0,1)^t, (1,1)^t, (1,0)^t\right)$ with :

$$\tilde{\mathbf{P}}_0 = \tilde{\mathbf{P}}_k \propto \mathtt{H}_k.\tilde{\mathbf{p}}_k \tag{6}$$

Notation $\tilde{\mathbf{p}}_k$ is introduced to define $\mathbf{p}_k$ with homogeneous coordinates. $\mathtt{H}_k$ is computed simply from matching $\mathbf{P}_0$ with $\mathbf{p}_k$ and solving the resulting linear system [9]. The variation of the state vector can be expressed in the canonical reference frame by:

$$\tilde{\boldsymbol{\delta}_{\mathbf{P}k}} = \mathtt{H}_{k-1}.\tilde{\boldsymbol{\delta}_{\mathbf{p}k}} \tag{7}$$

$\boldsymbol{\delta}_{\mathbf{P}k}$ is the variation of the projection of $\mathbf{p}_{k-1}$ into the canonical reference frame using the previous homography $\mathtt{H}_{k-1}$. We propose to link $\boldsymbol{\delta}_{\mathbf{P}k}$ with the variation of the observation by the following regression function:

$$\boldsymbol{\delta}_{\mathbf{P}k} = \mathbf{F}(\boldsymbol{\delta}_{\mathbf{z}k}; \mathbf{w}) + \varepsilon_k \tag{8}$$

where ε_k is a random noise. In this expression, the parameter vector $\mathbf{w}$ to be estimated does not depend on time. So, it can be estimated once for all frames of the sequence.

The resulting template tracking method is summarized into the algorithm 1. and, Figure 1 illustrates the principle of the method.

3 Kernel based Regression Functions

3.1 Learning

A key point associated with the method proposed in the previous section concerns the model of the regression function $\mathbf{F}$ and the estimation of the associated vector of parameters $\mathbf{w}$. $\mathbf{F}$ can be either linear [8] [10], or non-linear [3].

The vector of parameters $\mathbf{w}$ can be estimated either in an analytical way (estimation of the Jacobian matrix [8] or a second order matrix in [3]), or using machine learning techniques [10].

Let $\mathcal{V} \doteq \{\boldsymbol{\delta}^{(n)}_{\mathbf{P}}, \boldsymbol{\delta}_{\mathbf{z}}{}^{(n)}\}$ be a learning set built from random motions $\{\boldsymbol{\delta}^{(n)}_{\mathbf{P}}\}_{n=1}^N$ of the pattern (projected into a reference frame), associated to the variations of the feature vector (observation) $\{\boldsymbol{\delta}_{\mathbf{z}}{}^{(n)}\}_{n=1}^N$. The learning motions are applied to the reference image (generally the first image of the video sequence).

Jurie and Dhome propose to to use a hyperplane model for the regression function:

$$\mathbf{F}(\boldsymbol{\delta}_{\mathbf{z}}; \mathtt{W}) = \mathtt{W}\boldsymbol{\delta}_{\mathbf{z}} \tag{9}$$

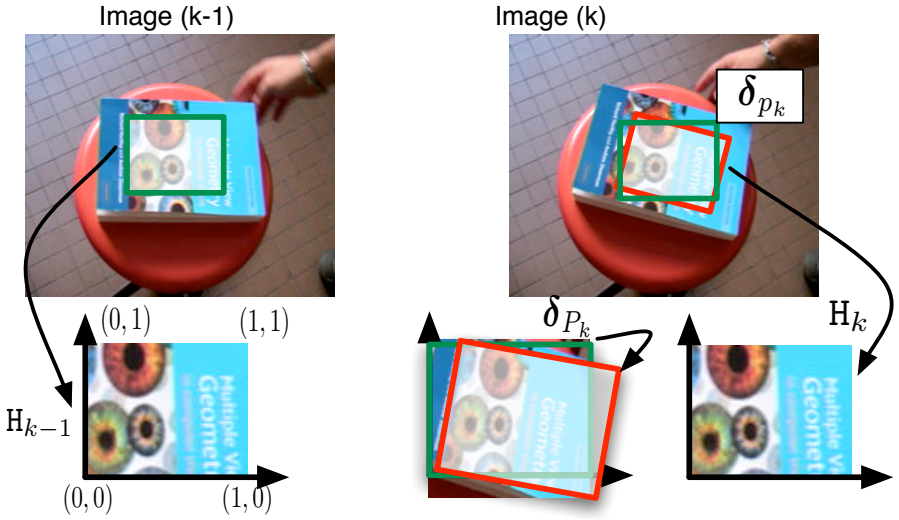


Figure 1: Illustration of the tracking method. The reference pattern, at time k is projected into a canonical reference frame using the homography $\mathtt{H}_{k-1}$. A regression function estimates the motion ($\boldsymbol{\delta}_{\mathbf{p}_k}$) between $k-1$ and k according to the observed variation. The motion of the pattern is then simply given by $\tilde{\boldsymbol{\delta}}_{\mathbf{P}k} = \mathtt{H}_{k-1}.\tilde{\boldsymbol{\delta}}_{\mathbf{p}_k}$ and the homography is updated.

The parameter matrix $\mathtt{W}$ is learnt from the training set $\mathcal{V}$ by minimizing a "least-square" error measure.

We propose to use models which are a linearly-weighted sum of M fixed non-linear basis functions:

$$\mathbf{F}(\boldsymbol{\delta}_{\mathbf{z}};\mathtt{W}) = \mathtt{W}.\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}) = \sum_{m=1}^{M} \mathbf{w}_m \boldsymbol{\phi}_m(\boldsymbol{\delta}_{\mathbf{z}}) \tag{10}$$

$\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}})$ denotes a vector of M basis functions:

$$\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}) = [\phi_1(\boldsymbol{\delta}_{\mathbf{z}}), \phi_2(\boldsymbol{\delta}_{\mathbf{z}}), ..., \phi_M(\boldsymbol{\delta}_{\mathbf{z}})]^T \tag{11}$$

and with $\phi_m(\boldsymbol{\delta}_{\mathbf{z}}) = k(\boldsymbol{\delta}_{\mathbf{z}}, \boldsymbol{\delta}_{\mathbf{z}}^{*\mathbf{m}})$, a kernel function, applied to $\boldsymbol{\delta}_{\mathbf{z}}$ and a basis vector denoted $\boldsymbol{\delta}_{\mathbf{z}}^{*\mathbf{m}}$.

Let $\mathtt{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M)$ be a matrix of parameters associated to the M basis functions. The objective is to find values for $\mathtt{W}$ such that $\mathbf{F}(\boldsymbol{\delta}_{\mathbf{z}};\mathtt{W})$ makes good predictions for new data: i.e. it models the underlying generative function.

A classic approach to estimating $\mathtt{W}$ is "least- square", minimization of the error measure:

$$E_{\mathcal{D}}(\mathtt{W}) = \frac{1}{2}\sum_{n=1}^{N} \left|\left|\delta_{\mathbf{P}}^{(n)} - \mathbf{W}.\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(n)})\right|\right|^2 \tag{12}$$

This can be rewritten in the following system:

$$\left(\boldsymbol{\delta}_{\mathbf{P}}^{(1)}, \boldsymbol{\delta}_{\mathbf{P}}^{(2)}, ..., \boldsymbol{\delta}_{\mathbf{P}}^{(N)}\right) = \mathtt{W}\left(\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(1)}), \boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(2)}), ..., \boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(N)})\right) \tag{13}$$

Let denote $\Phi \doteq \left(\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(1)}), \boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(2)}), ..., \boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(N)})\right)$ and $\Delta_P \doteq \left(\boldsymbol{\delta}_{\mathbf{P}}^{(1)}, \boldsymbol{\delta}_{\mathbf{P}}^{(2)}, ..., \boldsymbol{\delta}_{\mathbf{P}}^{(N)}\right)$; the system can be expressed under a more compact form:

$$\Delta_P = \mathtt{W}.\Phi \tag{14}$$

Algorithm 1 Tracking

Input : state $\mathbf{p}_0$, image $\mathtt{I}_0$ and regression function $\mathbf{F}$
Output : set of states $\{\mathbf{p}_k\}_{k=1}^{K}$
Initialisation : $k = 0$, extraction of the reference feature vector $\mathbf{z}^*_{\mathcal{W}} \doteq \mathbf{z}(\mathtt{I}_0, \mathcal{W}, \mathbf{p}_0)$ and estimation of the canonical homography $\mathtt{H}_0$
for $k = 1$ to K (loop on the images of the video sequence) **do**
Observation : Extraction of the feature vector $\boldsymbol{\delta}_{\mathbf{z}k} = \mathbf{z}(\mathtt{I}_k, \mathcal{W}, \mathbf{p}_{k-1}) - \mathbf{z}^*_{\mathcal{W}}$
Estimation : Estimation of the motion into the canonical reference frame, then into the image reference frame :

$$\boldsymbol{\delta}_{\mathbf{P}k} = \mathbf{F}(\boldsymbol{\delta}_{\mathbf{z}k}; \mathbf{w})$$

$$\tilde{\boldsymbol{\delta}}_{p_k} \propto \mathtt{H}_{k-1}^{-1}.\tilde{\boldsymbol{\delta}}_{Pk}$$

Update : state vector and homography

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \boldsymbol{\delta}_{\mathbf{p}k}$$

$$\mathtt{H}_k, \text{ Homography between } \mathbf{P}_0 \text{ and } \mathbf{p}_k$$

end for

and the estimation of the parameter matrix $\mathtt{W}$ using (12) is given by:*

$$\mathtt{W}_{LS} = \Delta_P \Phi^+, \tag{15}$$

Alternative methods based on the "least-square" criterion can be used to estimate the parameter matrix $\mathtt{W}$. A solution is to place a prior over $\mathtt{W}$ in order to set many weights to zero. The resulting model is then called sparse linear model. SVM (*Support Vector Machine*) [14] is a sparse linear model where the weights are estimated by the minimization of a Lagrange multipliers based functional. Other sparse linear models, like RVM (*Relevance Vector Machines*) [13] or multivariate RVM [12] may also be employed.

Generally, vectors used in basis functions ($\boldsymbol{\delta}^{*\mathbf{m}}_{\mathbf{z}}$) can be chosen from the training set. It is also possible to use the entire training set and in this case $N = M$. Moreover, we can notice that the hyperplane model presented in eq. (9) is a special case of the model (10) with linear basis functions $\phi_m(\boldsymbol{\delta}_{\mathbf{z}})$.

We make the common choice to use Gaussian data-centred basis functions:

$$\phi_m(\boldsymbol{\delta}_{\mathbf{z}}) = \exp\left[-\left(\boldsymbol{\delta}_{\mathbf{z}} - \boldsymbol{\delta}^{*\mathbf{m}}_{\mathbf{z}}\right)^2 / \sigma^2\right], \tag{16}$$

which gives us a "radial basis function" (RBF) type model from which the parameter σ must be adjusted. On one hand, if σ is too small, the "design matrix" Φ is mostly composed of zeros. On the other hand, if σ is too large, Φ is mostly composed of ones ($\exp(0)$). We propose to adjust σ using a non-linear optimization maximizing a cost function based on the sum of the variances computed for each line of Φ:

$$\sigma = \arg\max_{\sigma}[C(\sigma)] \tag{17}$$

with

$$C(\sigma) = \sum_{n=1}^{N}\sum_{m=1}^{M}\left(\phi_m(\boldsymbol{\delta}^{(n)}_{\mathbf{z}}) - \overline{\phi}(\boldsymbol{\delta}^{(n)}_{\mathbf{z}})\right)^2 \tag{18}$$

and

$$\overline{\phi}(\boldsymbol{\delta}^{(n)}_{\mathbf{z}}) = \frac{1}{M}\sum_{m=1}^{M}\phi_m(\boldsymbol{\delta}^{(n)}_{\mathbf{z}}) \tag{19}$$

*Φ^+ denotes the pseudo-inverse of Φ.

An overview of the learning method proposed in this section, called KBT (*Kernel based Machine Learning Tracker*) is given in algorithm 2. The learning method is called L times, for different values of b, following a coarse to fine scheme [5, 7]

Algorithm 2 KBT: learning step

Input : Size of the training set N, training parameter b, number of basis functions M, initial homography $\mathtt{H}_0$, initial std. of the kernel function σ_0.
Output : parameter matrix $\mathtt{W}$ and std. of the kernel function σ.
Pattern motion generation : A set of N motion vectors is drawn according to a uniform law on the interval $[-b, b]$: $\{\boldsymbol{\delta}_{\mathbf{P}}^{(n)}\}_{n=1}^{N}, \mathbf{P}^{(n)} \sim \mathcal{U}(-b, b)$.
Observation : Compute $\{\boldsymbol{\delta}_{\mathbf{z}}{}^{(n)}\}_{n=1}^{N}$ such as :

$$\boldsymbol{\delta}_{\mathbf{z}}{}^{(n)} = \mathbf{z}(\mathtt{I}_0, \mathcal{W}, \mathbf{p}^{(n)}) - \mathbf{z}^*_{\mathcal{W}},$$

with

$$\tilde{\mathbf{p}}^{(n)} \propto \mathtt{H}_0^{-1}\tilde{\mathbf{P}}^{(n)}.$$

Draw basis functions: draw $\{\boldsymbol{\delta}_{\mathbf{z}}^{*\mathbf{m}}\}_{m=1}^{M}$, using an uniform law from $\{\boldsymbol{\delta}_{\mathbf{z}}{}^{(n)}\}_{n=1}^{N}$.
estimation of the kernel function parameter σ : non-linear optimization of σ, such as:

$$\sigma = \arg\max_{\sigma} \left\{ \sum_{n=1}^{N} \sum_{m=1}^{M} \left(\phi_m(\boldsymbol{\delta}_{\mathbf{z}}^{(n)}) - \overline{\phi}(\boldsymbol{\delta}_{\mathbf{z}}^{(n)}) \right)^2 \right\}$$

estimation of the weight (parameter) matrix $\mathtt{W}$:

$$\mathtt{W} = \Delta_P.(\Phi^T(\Phi\Phi^T)^{-1})$$

with $\Phi \doteq \left(\boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(1)}), \boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(2)}), ..., \boldsymbol{\phi}(\boldsymbol{\delta}_{\mathbf{z}}{}^{(N)})\right)$

3.2 Tracking

The learning step provides, for each training level l:

1. The weight matrix $\mathtt{W}_l$
2. The set of basis functions $\{\boldsymbol{\delta}_{z,l}^{*m}\}_{m=1}^{M}$
3. The width parameter associated with the kernel function: σ_l.

These parameters are then used to build the regression function $\mathbf{F}(\boldsymbol{\delta}_{\mathbf{z}}; \mathtt{W}_l) = \sum_{m=1}^{M} \mathbf{w}_{m,l}\boldsymbol{\phi}_{m,l}(\boldsymbol{\delta}_{\mathbf{z}})$. Moreover, it is possible to call the regression function several times in order to increase the tracking precision. The resulting method is presented in algorithm 3.

4 Experiments

This section presents the experiments achieved in order to compare the proposed method to the reference linear algorithm. After a description of the datasets and the methodology, experimental results are presented and then discussed.

Algorithm 3 KBT: tracking step

Input: Regression function parameters: $\mathtt{W}_l, \{\boldsymbol{\delta}^{*m}_{z,l}\}_{m=1}^{M}$, initial state $\mathbf{p}_0$ and reference image $\mathtt{I}_0$
Output: set of the states $\{\mathbf{p}_k\}_{k=1}^{K}$
Initialisation: $k = 0$, extract reference measure vector $\mathbf{z}^*_{\mathcal{W}} \doteq \mathbf{z}(\mathtt{I}_0, \mathcal{W}, \mathbf{p}_0)$ and compute the initial homography $\mathtt{H}_0$
for $k = 1$ to K (loop on the images) **do**
  $\mathbf{p}' = \mathbf{p}_{k-1}$ and $\mathtt{H}' = \mathtt{H}_{k-1}$
  for $l = 1$ to L (loop on the levels) **do**
    for $i = 1$ to I (loop for each level) **do**
      Observation: features vector extraction $\boldsymbol{\delta}_{\mathbf{z}} = \mathbf{z}(\mathtt{I}_k, \mathcal{W}, \mathbf{p}') - \mathbf{z}^*_{\mathcal{W}}$
      Estimation : Motion estimation into the canonical reference frame, then into the image reference frame:

$$\boldsymbol{\delta}_{\mathbf{P}}{}' = \sum_{m=1}^{M} \mathbf{w}_{m,l}\boldsymbol{\phi}_{m,l}(\boldsymbol{\delta}_{\mathbf{z}})$$

$$\tilde{\boldsymbol{\delta}_{\mathbf{p}}}' \propto (\mathtt{H}')^{-1}.\tilde{\boldsymbol{\delta}_{\mathbf{P}}}'$$

      Update: state vector and homography

$$\mathbf{p}' = \mathbf{p}' + \boldsymbol{\delta}_{\mathbf{p}}{}'$$

      Estimate $\mathtt{H}'$, solving $\tilde{\mathbf{P}}_0 \propto \mathtt{H}'.\tilde{\mathbf{p}}'$
    end for
  end for
  Final update: $\mathbf{p}_k = \mathbf{p}'$ and $\mathtt{H}_k = \mathtt{H}'$
end for

4.1 Datasets and Methodology

Methodology for evaluation of region based tracking methods has been already proposed for both rigid [10, 3, 11, 8] and non rigid objects [4, 7]. Experiments done can be classified in three categories:

1. **Accuracy of motion parameters estimation**. The aim of this experiment is to show the estimated variation of motion parameters related to the true variation. Experimental data used for this test are generated from a static image. Virtual motion parameters variation and a synthetic resulting region are achieved and stored in a ground truth database. Motions are usually generated in a marginalized way (the variation is applied for one parameter, a translation coordinates for example). The tracking algorithm is then tested using the generated dataset and the motion parameter estimation is compared with the true estimation. Several tests are usually achieved related to the most relevant parameters of the tracking algorithm.

2. **convergence frequency**. The aim of this test is to compute the convergence frequency of tracking algorithms. In [3], the algorithm diverges when the final SSD motion error is bigger than the initial SSD. A database containing motions and the resulting synthetic region is generated and then used to compute the rate of convergence of the algorithm. This frequency is then presented compared to the amplitude variation of the motion parameters , or compared to the appearance perturbations like illumination variations (affine or gaussian for example).

3. **illustration on real sequences**. The aim of the experiment is to illustrate the behaviour of the algorithm for several real sequences with illumination variation, clutter background, or partial occlusion. Since ground truth can not be known for such sequence, key samples of the video are extracted and presented with the superimposed tracking region.

The methodology proposed here is close to the one already proposed to evaluate similar methods. We compare two algorithms in terms of accuracy and convergence rate for a synthetic image. Moreover three algorithms have been compared on real videos. Two classical algorithms:

- LBT: the Linear Based Tracker algorithm proposed in [10].

- ESM: The ESM visual tracking software is based on a fast optimization technique called ESM (Efficient Second-order Minimization). The ESM technique has been proposed for improving standard visual servoing techniques. Thanks to its generality, it has been extended to improve template-based visual tracking techniques [3]. Since we have used the ESM MATLAB (TM) toolkit provided by the author, the method has been used with default parameters.

The proposed algorithm:

- KBT : the Kernel Based Tracker.

Since LBT and KBT are very close, several experiments compare the two algorithms in relation with their most relevant parameters. The same learning dataset is used for both algorithms. The latter is generated from random (gaussian law) disturbance of motion parameters (variation of the position of the four corners or the patterns). In the following, we define b as the standard deviation of the gaussian law in percentage of the tracked pattern width.

The learning step of KBT can be achieved by two methods: least square linear optimization or multivariate relevant vector machine learning. The two algorithms have been tested and results obtained are very close ; so the following results are achieved with the second approach because the learning step is much longer for MVRVM.

The observation function is based on the luminance of pixels extracted from a regular sampling of the pattern to track. In the following experiments, the size of the sampling grid is 15×15 pixels (225 points). The resulting

feature vector is then zero-normalized in order to be invariant according to affine luminance transformations. Moreover, all the tests presented here have been realized with a coarse to fine learning strategy using three levels and three loops per level.

4.2 Results

In order to assess the algorithms in different controlled conditions, we synthesize images from a template. Moreover, LBT and KBT algorithms have been implemented on a PC Desktop (PIV 3.2GHz) and run at a 6 ms by image for KBT and 3 ms for LBT.

The first test shows the ability of the method to estimate a known variation (cf. fig. 2). A horizontal displacement of the pattern is generated from a static image. The training step is realized for three levels ($L = 3$), with $N = 400$ and $M = N$ basis functions. Moreover, three iterations are applied to the tracker ($I = 3$). The figure shows the mean translation error related to the translation amplitude (% of the pattern width), for four values of the training parameter b (disturbance magnitude used to build the training set for the first level). The figure shows the accuracy of LBT and KBT algorithm.

The second test shows the convergence frequency of for LBT and KBT, in relation with the disturbance magnitude used for the training set. We define a successful convergence with a threshold on the quadratic error between the estimated position (the four corners of the pattern to track) and the real position. A random perturbation (gaussian law) is applied to the four corners of the pattern and a resulting synthetic dataset is generated using one thousand realisations of this process. Figure 3 shows the convergence rate according to the quadratic motion generated, and for $b = 0.1$ (training parameter) for the left sub figure and $b = 0.2$ for the right sub figure.

The third test compares the convergence frequency for LBT and KBT, in relation with noisy observations. A subset of the observation vector $\mathbf{z}(\mathtt{I}_k, \mathcal{W}, \mathbf{p}_{k-1})$ (randomly selected) has been replaced by a random value drawn from an uniform distribution between 0 and 255 (the gray level range). Figure 6 shows the convergence rate for LBT and KBT related to the percentage of noisy features.

The two above described algorithms have been compared on three sets of real data. For the first set (see fig. 7), the two methods have been compared with ESM. The figure shows the output of the three algorithms for key images of the video. The toy video selected provides high rotations and scale variations. The pattern to be tracked is selected on the first image. The second set is composed by five other toy sequences, with several patterns, motions and illumination conditions. Figure 6 shows the results of the tracking process for 4 sampled images of each sequence. The "Box" sequence is quite simple, with no specularity, slow motion and scale variation. "Crisp 1" and "Crisp 2" sequences contain strong rotations with blur and specularities. "Lipton 1" and "Lipton 2" sequences provides large scale variations. Moreover, for "Lipton 2", the learning step is achieved using a low resolution pattern. For each sequence, we measure the number of successfully tracked frames by each method before divergence of the algorithm. Table 1 summarizes the resulting frequency rates. The third set is composed by four real video sequences extracted from the PETS[†] dataset which provides traffic videos from a camera embedded within a vehicle. One of the related application is collision avoidance or automatic cruise control using rear car vision based tracking. Figure 7 shows the results of the tracking process for 4 sampled images of each sequence and table 1 summarizes the computed frequency rates.

4.3 Discussion

The accuracy test (2) shows that for small training amplitudes $b = 0.1$ and $b = 0.2$, motion translation estimation is correct for values within the training area (lower than the training amplitude b) and accuracy is the similar for LBT and KBT. For high translation, the error increases for both the LBT and the KBT method. For a training amplitude $b = 0.3$, the LBT fails while the KBT still provides correct translation estimation. The

[†]Performance Evaluation of Tracking and Surveillance

Seq. name (#nb. frames)	LBT Nb. (%) of successfully tracked frames	KBT Nb. (%) of successfully tracked frames
Box (#198)	191 (96%)	198 (100%)
Crisp 1 (#200)	15 (7%)	200 (100%)
Crisp 2 (#249)	165 (66%)	249 (100%)
Lipton 1 (#144)	17 (12%)	144 (100%)
Lipton 2 (#341)	107 (31%)	267 (78%)

Table 1: Convergence frequency of LBT and KBT for five toy sequences.

Seq. name (#nb. frames)	LBT Nb. (%) of successfully tracked frames	KBT Nb. (%) of successfully tracked frames
Vehicle. 1 (#495)	60 (12%)	495 (100%)
Vehicle. 2 (#458)	57 (12%)	198 (43%)
Vehicle. 3 (#765)	0 (0%)	180 (23%)
Vehicle. 4 (#97)	7 (7%)	97 (100%)

Table 2: Convergence frequency of LBT and KBT for four sequences related to rear car tracking.

first order approximation used for the LBT learning step is only correct for small motions. Since the KBT is a non-linear model, it is possible to learn the non linear regression function for large motions (until $b = 0.4$).

The convergence rate test presented in fig. 3 shows that KBT has higher convergence rates than the LBT algorithm. The convergence rate of the LBT decreases because the first order approximation made in the method is correct only for small motion. These results are linked to the accuracy test. Since KBT can learn higher motions than LBT, the convergence area is higher for KBT than for LBT. It results a higher convergence basin for KBT.

Fig. 3 shows that the convergence rate of the LBT decreases to 50% with only 0.5% of noisy features. The KBT provides a higher convergence rate than the LBT with 50% of convergence for 4.4% of noisy features. The reason is that when a noisy observation occurs, the vector $\boldsymbol{\delta}_{\mathbf{z}}$ produces high values. For the LBT, $\boldsymbol{\delta}_{\mathbf{z}}$ is directly multiplied with the interaction matrix and generates high displacements. For the KBT,$\boldsymbol{\delta}_{\mathbf{z}}$ is used in kernel functions and the resulting scalar vector is often small; so the motion generated is only slightly modified by the noisy observation. This is an important property of the KBT because if the features are far from the features used in the learning dataset, the estimated motion is near zero.

Both the accuracy and the convergence tests show that KBT has good performances compared to LBT. In order to illustrate these results on real data, six video sequences have been chosen, with different conditions. Table 1 reports the number of successfully tracked frames by each method before divergence. KBT algorithm give a higher successfully tracked frames rate than LBT.

The last experiment illustrates the performance of the method for a rear vehicles tracking application. Figure 7 shows the results of the tracking process for 4 sampled images of each sequence. Table 2 shows the number of successfully tracked frames by each method before divergence of the algorithm. For all the sequences, the frequency rate of the KBT method is higher than the frequency rate of LBT.

5 Conclusion

We have proposed an extention of the linear based planar pattern tracking algorithm to non-linear models. The learning based framework provides an efficient way to build a regression function between observation and motion. Moreover, an empirical rule is proposed to estimate the parameters of the kernel function in order to give an informative design matrix.

This method has been implemented using simple gray-level features and experiments have been performed to compare it to the linear algorithm in terms of accuracy and frequency convergence. For small learning magnitudes, accuracy is the similar for LBT and KBT. Moreover, frequency basin is higher for KBT than for LBT. However, further tests on a large number of patterns are needed to establish this firmly.

The algorithm has been also tested for realtime rear car tracking, a necessary subtask for applications like vision based automatic cruise control or vision based collision avoidance. Results indicates that KBT has better convergence frequency than LBT for the sampled video sequences.

Future works will be done in order to improve robustness against noisy data provided by partial occlusion or specularities. Since the method works at 6 ms for one image, more complex kernel function may be used (robust kernel functions).

References

[1] S. Avidan. Support vector tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001)*, Hawaii, December 2001.

[2] S. Baker and I. Matthews. Lucas-Kanade 20 years on : A unifying framework. *IJCV International Journal on Computer Vision*, 56(3):221–255, 2004.

[3] S Benhimane and E Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *IEEE/RSJIROS*, Japan, October 2004.

[4] T.F. Cootes, G.J. Edwards, and Taylor C.J. Active appareance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[5] C. Dehais, M. Douze, V. Charvillat, and G. Morin. Augmented reality through real-time tracking of video sequences using a panoramic view. *Int. Conf. on Pattern Recognition*, 2004.

[6] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-camera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[7] V. Gay-Bellile, A. Bartolli, and P. Sayd. Feature-driven non-rigid image registration. In *BMVC, British Machine Vision Conference*. Warwick, United Kindom, September 2007.

[8] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.

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

[10] F. Jurie and M. Dhome. Real time template matching. In *International Conference on Computer Vision*, pages 544–549, Vancouver, Canada, July 2001.

[11] T. Chateau, F. Jurie, M. Dhome, and X. Clady. Real-time tracking using Wavelets Representation. In *Symposium for Pattern Recognition, DAGM'02,*, pages 523–530, Zurich, September 2002. Springer.

[12] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. In *ECCV, European Conference on Computer Vision*, 2006.

[13] M.E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Larning Research*, 1:211–244, 2001.

[14] V.N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[15] O. Williams, A. Blake, and R. Cipolla. A sparse probabilistic learning algorithm for real-time tracking. pages 353–361, Nice, France, 2003.

[16] Q. Yu, G. Medioni, and I. Cohen. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

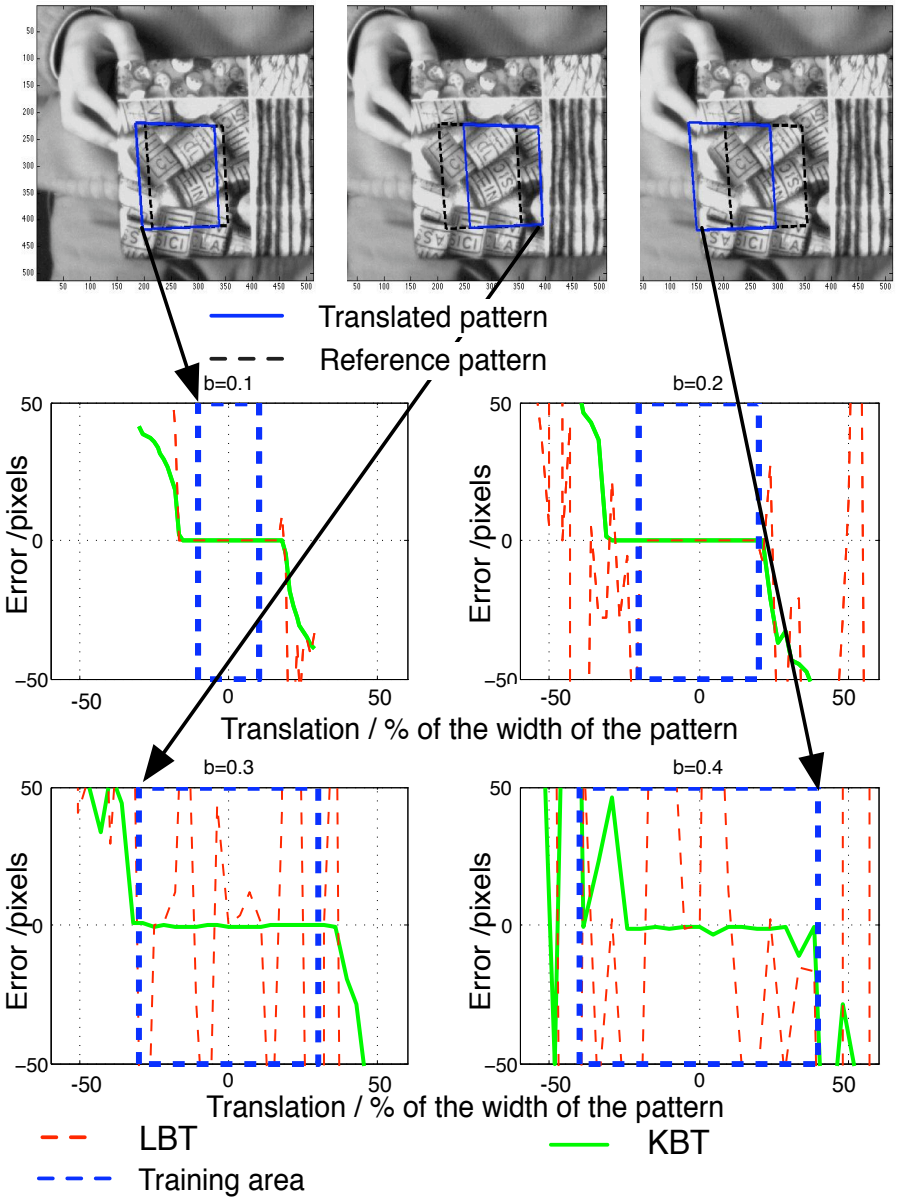


Figure 2: Comparison of the LBT and the KBT in terms of accuracy, against horizontal displacement magnitude, and for four different values of the training parameter b.

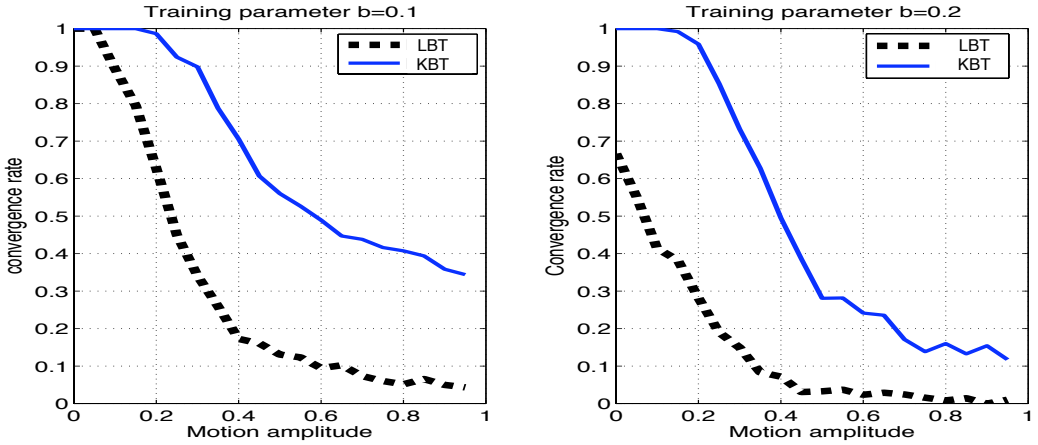


Figure 3: Comparison of the LBT and the KBT method in terms of convergence frequency, against the displacement magnitude

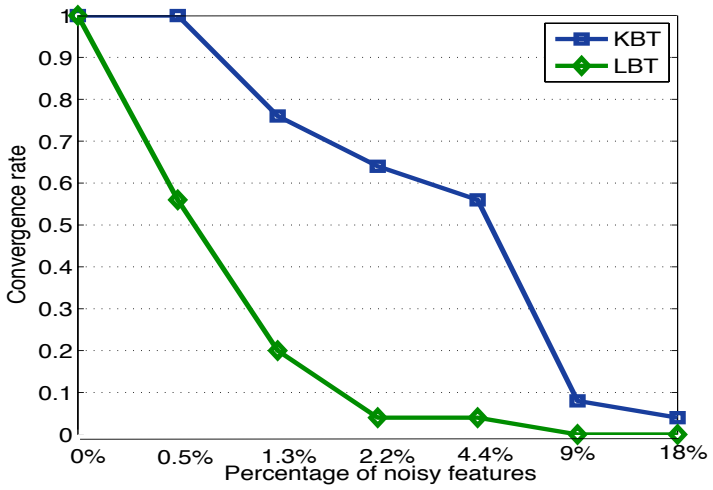


Figure 4: Comparison of the LBT based tracker and the KBT method against noisy observations.

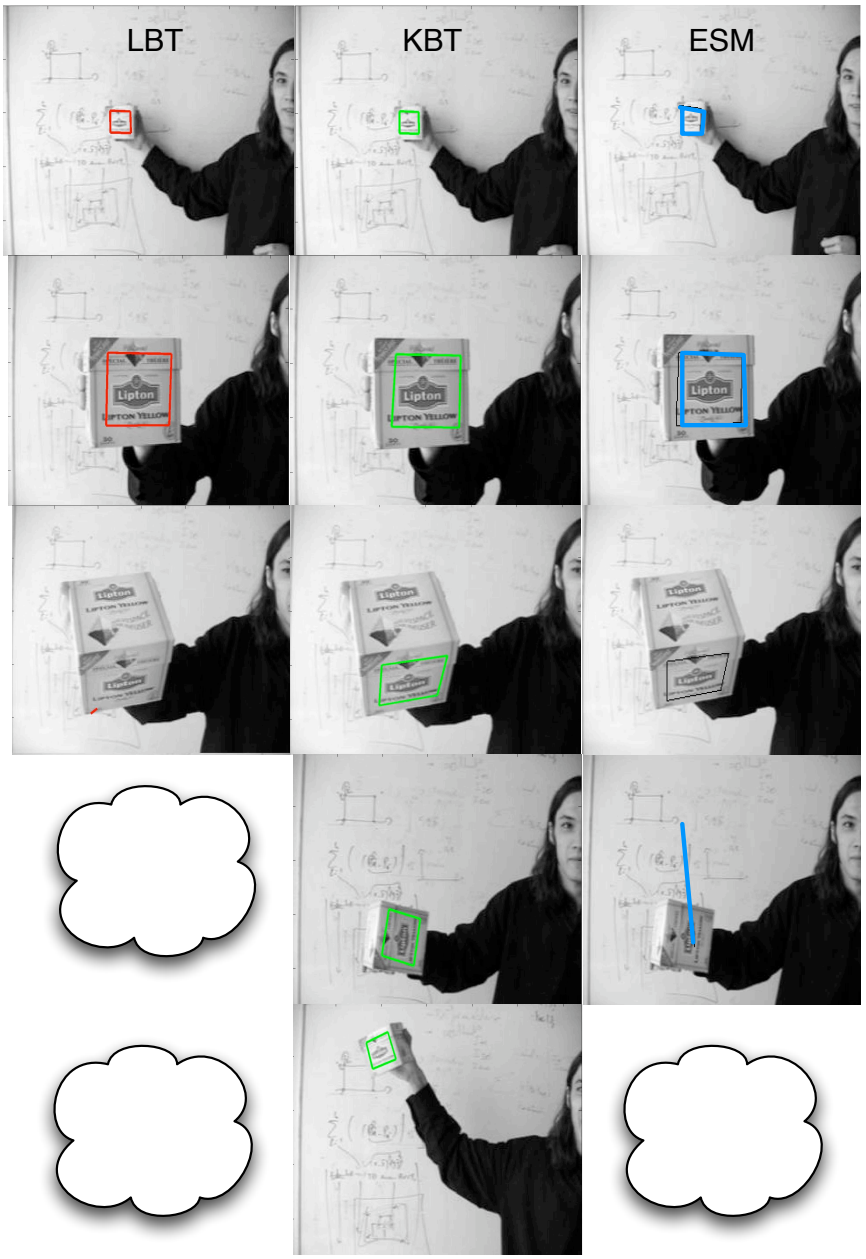


Figure 5: Comparison of three tracking algorithms, on a real video sequence: LBT for the left column, KBT for the median column and ESM for the right column. Default parameters have been used for ESM. Morevover 1000 samples have been generated for the training step of KHT and KBT and 100 basis functions have been used for KBT.

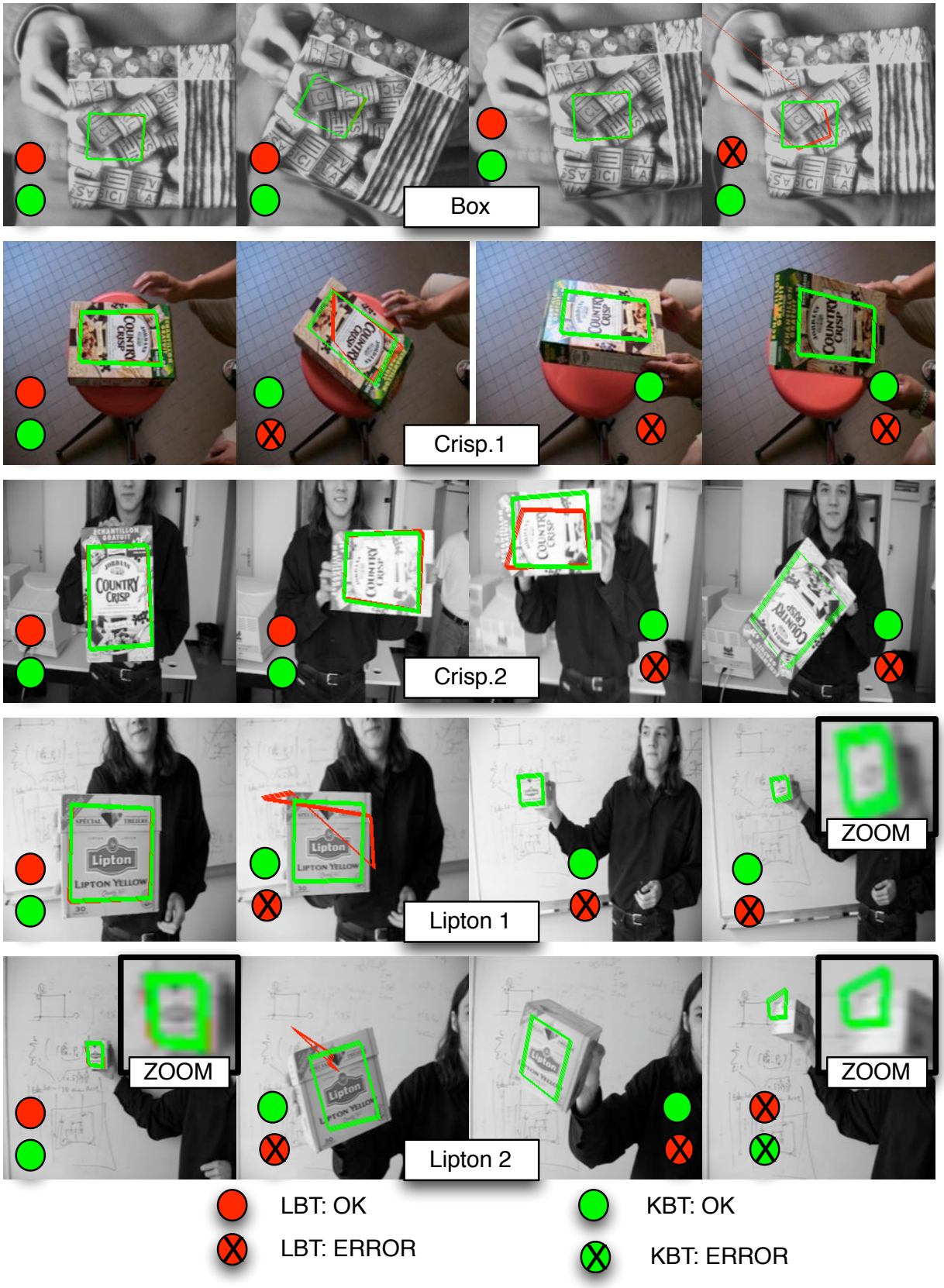


Figure 6: Comparison of the LBT and the KBT for five sequences related to rear car tracking.

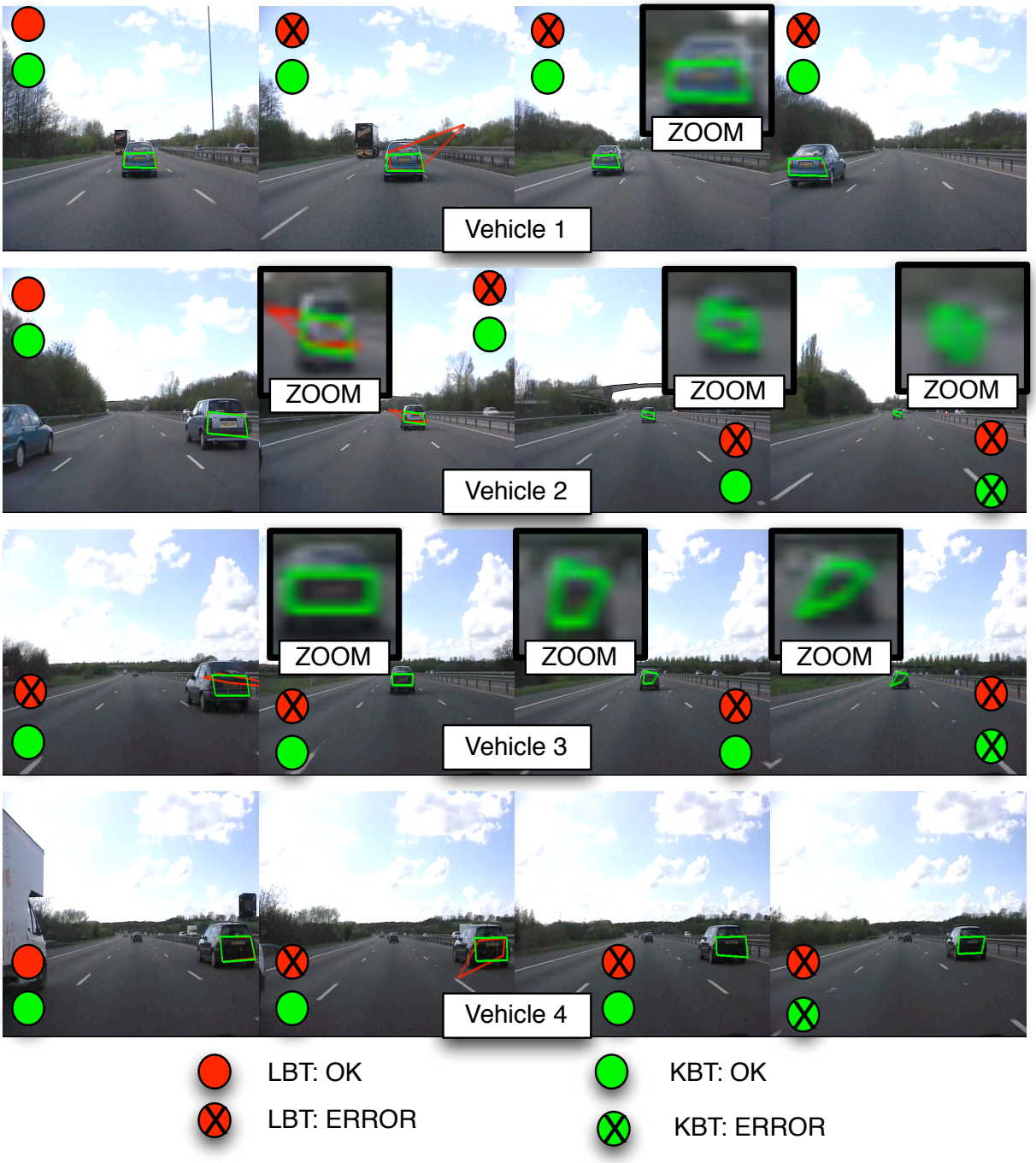


Figure 7: Comparison of the LBT and the KBT for five sequences related to rear car tracking.

A.3 Suivi et catégorisation d'un nombre variable d'objets

Illumination aware mcmc particle filter for long-term outdoor multi-object simultaneous tracking and classification
F. Bardet, T. Chateau, and D. Ramadasan
In ICCV 2009, International Conference on Computer Vision, Tokyo, Japon, Septembre 2009

Illumination Aware MCMC Particle Filter for Long-Term Outdoor Multi-Object Simultaneous Tracking and Classification

François Bardet, Thierry Chateau, Datta Ramadasan
LASMEA, Université Blaise Pascal
24 avenue des Landais, F-63177 Aubière cedex, FRANCE
{bardet,chateau,ramadasan}@lasmea.univ-bpclermont.fr

Abstract

This paper addresses real-time automatic visual tracking, labeling and classification of a variable number of objects such as pedestrians or/and vehicles, under time-varying illumination conditions. The illumination and multi-object configuration are jointly tracked through a Markov Chain Monte-Carlo Particle Filter (MCMC PF). The measurement is provided by a static camera, associated to a basic foreground / background segmentation. As a first contribution, we propose in this paper to jointly track the light source within the Particle Filter, considering it as an additionnal object. Illumination-dependant shadows cast by objects are modeled and treated as foreground, thus avoiding the difficult task of shadow segmentation. As a second contribution, we estimate object category as a random variable also tracked within the Particle Filter, thus unifying object tracking and classification into a single process. Real time tracking results are shown and discussed on sequences involving various categories of users such as pedestrians, cars, light trucks and heavy trucks.

1. Introduction

Real-time visual tracking of a variable number of objects is of high interest for various applications. In the recent years, several works have addressed multiple pedestrian and vehicle tracking [14]. In all these applications, real time may be needed either because an immediate information is required, or because recording images is not allowed, or because the amount of data is simply too huge to be recorded and processed later. Vision has been chosen as it offers a large measuring range, required by several surveillance applications: about 200 meters for traffic surveillance. Unfortunately, this benefit also causes deep object appearance scale changes. In addition, in traffic surveillance, target objects belong to various classes, such as pedestrians, cycles, motorcycles, light vehicles, light trucks, or heavy trucks. The tracker is thus required to deal with various target 3D sizes, and with various target projection 2D sizes, due to heavy perspective effect.

In outdoor environment, shadows cast by opaque objects interfere with object segmentation and description. This decreases tracking accuracy as object estimate may be shifted towards its shadow. Moreover, it yields tracking failures as the tracker may instantiate a *ghost* candidate object upon a cast shadow. Both failures have been described in the literature, [10, 11] among others. However, shadows cast by objects also feature relevant information about object itself, offering the opportunity to increase its observability. For these two reasons, cast shadows have to be taken into account to improve visual tracking performance [11]. A survey and benchmark of moving shadow detection algorithms has been published in [10]. Nevertheless, segmenting the image into three classes (background, objects, shadows cast by objects) is a very challenging step, yielding authors to incorporate spatial and temporal reasonning into their segmentation methods.

Reversible Jump Markov Chain Monte-Carlo Particle Filter (*RJ MCMC PF*) has become a popular algorithm for real-time tracking of a varying number of interacting objects, as it allows to smartly manage object interactions as well as object enter and leave. The benefit of *MCMC PF* is that the required number of particles is a linear function of the number of tracked objects, when they do not interact. More computation is only required in case of object interaction (*i.e.* occlusion). This technique has been proposed and successfully used for jointly tracking up to 20 ants from a top view [5], or for tracking several pedestrians in a multi-camera subway surveillance setting [13].

In this paper, we address a mono-vision infrastructure-located real-time multi-object joint tracker and classifier. The core of the tracker is based on a *RJ MCMC PF* algorithm inspired of [5, 13] and extended to jointly track and classify objects and light source. Moreover, considering the difficulty to compute a reliable low level shadow segmentation, we choose to use a basic *background / foreground*

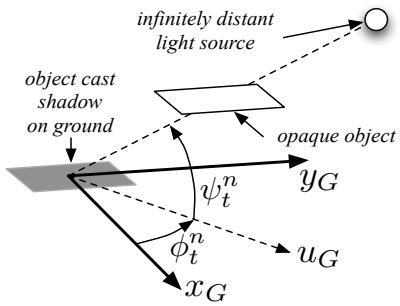


Figure 1. Infinitely distant light source and object cast shadow over the ground, assumed to be horizontal and defined by x_G and y_G. Light source position angles (azimut ϕ_t^n, and elevation ψ_t^n) are relative to the local ground reference.

segmented image as an observation. In [9], cast shadow is modeled using a 3-D object model, with a hand-defined sun position. We extend this approach to allow the *RJ MCMC PF* to automatically and continuously track sunlight estimate, allowing long-term outdoor tracking. The light source is modeled and updated over time within the particle filter, in order to manage slow but strong illumination changes caused by clouds and sun position dynamics. In section 2, we introduce joint light source and multi-object tracking. The observation likelihood is described in section 3, focusing on cast shadow representation. Object interaction weight is described in section 4. Finally, tracking results are reported and discussed in section 5.

2. Multi-Object *MCMC PF*

2.1. State Space

In an illumination aware visual object tracking, the system state encodes the configuration of the perceptible objects as well as illumination data: $\mathbf{X}_t^n = \{\mathbf{l}_t^n, J_t^n, \mathbf{x}_t^{j,n}\}$, $j \in \{1, ..., J_t^n\}$, where $\mathbf{l}_t^n = \{\xi_t^n, \phi_t^n, \psi_t^n\}$ defines the illumination hypothesized by particle n at time t, $n \in \{1, ..., N\}$, where N is the number of particles. More precisely, ξ_t^n is a binary random variable hypothesizing sunlight to be broken by a cloud or not, while ϕ_t^n and ψ_t^n are continuous random variables respectively standing for sun azimut and elevation angles, as illustrated on figure 1. When sunlight is bright (unbroken), object shadows are assumed to be cast onto the ground or other objects. J_t^n is the number of visible objects for hypothesis n at time t, and each object j is defined by: $\mathbf{x}_t^{j,n} = \{c_t^{j,n}, \mathbf{p}_t^{j,n}, \mathbf{v}_t^{j,n}, \mathbf{a}_t^{j,n}, \mathbf{s}_t^{j,n}\}$. Object j category at iteration n is given by $c_t^{j,n}$, a discrete random variable belonging to object category set, *C*={*pedestrian, motorcycle, light vehicle, light truck, heavy truck*} for instance.

Objects are assumed to move on a planar ground. Absolute position of candidate object j in particle n at time step t is defined by $\mathbf{p}_t^{j,n} = (x_t^{j,n}, y_t^{j,n}, \rho_t^{j,n})$, with object center of gravity position $x_t^{j,n}$ and $y_t^{j,n}$, and yaw angle $\rho_t^{j,n}$. Object j velocity and acceleration are described by $\mathbf{v}_t^{j,n}$ and $\mathbf{a}_t^{j,n}$, with magnitude and orientation. Object shape is modeled by a cuboid with dimension vector $\mathbf{s}_t^{j,n}$. Considering the sun to be a unique infinitely distant light source allows to very simply cast hypothesis object shadows over the ground. Nevertheless, the method can be extended to one or more finitely distant light sources.

2.2. MCMC PF for Multi-Object Tracking

Let $\mathbf{Z}_{1:t}$ the past observation sequence. Particle Filters approximate the posterior $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$ with N samples $\mathbf{X}_t^n$, $n \in \{1, ..., N\}$ at each time step. As the posterior is dynamic, samples have to be moved at each time step. Isard *et al.* [6] proposed a sampling strategy known as *SIR PF* (Sequential Importance Resampling Particle Filter), and a monocular Multi-Object Tracker (*MOT*) based on it [3], where the posterior is resampled at each time step by an importance sampler. This method draws new samples by jointly moving along all the state space dimensions. The required number of samples and the computation load thus grow as an exponential of the space dimension, as focused in [12]. As a result, it cannot track more than 3 persons. To overcome this limitation, it is necessary to draw samples by only moving within a subspace of the state space. Khan *et al.* proposed the *MCMC PF* [4], replacing the importance sampler with a *MCMC* sampler, according to Metropolis-Hastings algorithm [7]. The chain is built by markovian transitions from particle $\mathbf{X}_t^{n-1}$ to particle $\mathbf{X}_t^n$ via a unique new proposal $\mathbf{X}^*$, which may be accepted with probability α defined in 1. If refused then $\mathbf{X}_t^n$ is a duplicate of $\mathbf{X}_t^{n-1}$.

$$\alpha = min\left(1, \frac{\pi^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1}) Q(\mathbf{X}_t^{n-1})}{\pi_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1}) Q(\mathbf{X}^*)}\right) \quad (1)$$

In eq. 1, $\pi^* = P(\mathbf{Z}_t|\mathbf{X}^*)$ and $\pi_t^{n-1} = P(\mathbf{Z}_t|\mathbf{X}_t^{n-1})$ are likelihoods for observation $\mathbf{Z}_t$ under states $\mathbf{X}^*$ and $\mathbf{X}_t^{n-1}$, as detailed in section 3, $q(\mathbf{X})$ is the proposal law for a joint configuration $\mathbf{X}$, $w^* = w(\mathbf{X}^*)$ and $w_t^{n-1} = w(\mathbf{X}_t^{n-1})$ are interaction weights detailed in section 4. As real objects do not behave independantly from each other, Khan *et al.* proposed to include it within the dynamics model, and showed that it can be moved out of the prior mixture: $p(\mathbf{X}|\mathbf{Z}_{1:t-1}) \approx w(\mathbf{X}) \sum_n \prod_j p(\mathbf{x}_t^j|\mathbf{x}_{t-1}^{j,n})$, where $p(\mathbf{x}_t^j|\mathbf{x}_{t-1}^j)$ is object j dynamics model. As *MCMC* sampler is an iterative strategy, Khan *et al.* proposed to draw new samples by only moving one object $\mathbf{x}^j$ at a time, according to 2. This is the keypoint of the method: at each iteration, it lets the filter operate within object j subspace.

$$Q(\mathbf{X}^*|\mathbf{X}_t^{n-1}) \propto \begin{cases} Q(\mathbf{x}_t^{j*}) \text{ if } \mathbf{X}^{\backslash j*} = \mathbf{X}_t^{\backslash j,n-1} \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

where $\mathbf{X}^{\setminus j}$ is joint configuration $\mathbf{X}$ without object j, and $q(\mathbf{x}_t^{j*})$ is object j proposal law, whose approximation is:

$$q(\mathbf{x}_t^j) \approx \frac{1}{N}\sum_{n=1}^{N} p(\mathbf{x}_t^j|\mathbf{x}_{t-1}^{j,n}), \forall j \in \{1,..,J_t^{n-1}\} \quad (3)$$

The required number of particles thus is only a linear function of the number of tracked objects, when they do not interact. We adopt all the previous features.

2.3. Variable Number of Objects

To allow objects to enter or leave the scene, Khan *et al.* extended their *MCMC PF* to track a variable number of objects. For that purpose, the sampling step is operated by a *RJ MCMC* sampler (Reversible Jump Markov Chain Monte Carlo) [2], which can sample over a variable dimension state space, as the number of visible objects may change. This sampler involves the pair of discrete reversible moves {*enter, leave*} in order to extend the proposal law $q(\mathbf{X})$, thus allowing the state to jump towards a higher or lower dimension subspace [5, 12]. This sampler can approximate $p(\mathbf{X}^*|\mathbf{Z}_{1:t})$ if the acceptance ratio α is computed according to 1, involving evaluations of the proposal law $q(\mathbf{X})$ for $\mathbf{X}^*$ and $\mathbf{X}_t^{n-1}$. This leads to move-specific acceptance ratio computations, as shown in [5], and we use the same computations. In order to get time consistency, they also propose the pair of discrete reversible moves {*stay, quit*}. *Stay* allows to recover an object j which was present in the time $t-1$ particle set, and no more is in the current particle at time t. *Quit* proposes an object j which was not present in the time $t-1$ particle set, and is in the current particle at time t, to quit the scene. Though this pair of moves is devoted to object presence time consistency, it cannot cope with long duration occlusions or poor observation. For that reason, we do not use it and introduce object *vitality*, a continuous variable collecting the past likelihoods of each object, along iterations and time steps. It is integrated over all iterations of each time step, as detailed in appendix, and is used to drive object *leave* moves detailed in section in section 2.4. We extend the approach to reversible sun parameters and object category updates, yielding the following move set $\mathcal{M}$ ={*object enter, object leave,object update, sun enter, sun update*} denoted $\{e, l, u, se, su\}$ (*sun leaves* are treated with *object leaves*). Object category is tracked by proposing it to changes among set $\mathcal{C}$ ={*pedestrian, motorcycle, light vehicle, light truck, heavy truck*} according to a transition matrix. This move extends the *MCMC PF* framework to object classification functionality. In addition to processing a geometry-based classification within the *RJ MCMC PF*, it is of high interest when object classes have obviously different dynamics such as a trailer versus a light vehicle on a windy road or a pedestrian versus a vehicle. In other words integrating object class as a random variable within the *RJ MCMC PF* allows object time dynamics to contribute to object classification as well as object shapes.

2.4. Data-Driven Proposal Moves

In order to improve filter efficiency, object enter quota ρ_e is driven by observation $\mathbf{Z}$ and particle $\mathbf{X}_t^{n-1}$ at each iteration, according to eq. 12. Each object j leave quota $\rho_l(j)$ depends on its vitality, according to eq. 25. Object j update, sun update, and sun enter quota are set to constant values : $\rho_u(j) = 1$, $\rho_{su} = 0.1$, and $\rho_{se} = 0.02$. Move m probabilities P_m are computed from these quota, according to eq. 4, where J is the number of objects in particle $\mathbf{X}_t^{n-1}$.

$$P_m = \frac{\rho_m}{\rho_e + J\rho_u + \sum_{j\in\{1,..,J,s\}}\rho_l(j) + \rho_{se} + \rho_{su}}, \forall m \in \mathcal{M} \quad (4)$$

Object Enter: proposes a new object to enter with probability P_e, yielding joint configuration $\mathbf{X}^* = \{\mathbf{X}_t^{n-1}, \mathbf{x}^{j*}\}$. It is given a unique index j, initial dimensions, and initial vitality $\Lambda_t^j = \Lambda_0$. Acceptance rate is:

$$\alpha_e = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1})P_l(j)}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1})P_e Q(\mathbf{x}^{j*})}\right) \quad (5)$$

where object $\mathbf{x}^{j*}$ is drawn from the *false background* distribution $\mathbf{I}_{fb}$ (eq. 11), such that its projection fits $\mathbf{I}_{fb}$ blob.

Object j Leave: proposes to withdraw object j from $\mathbf{X}_t^{n-1}$ with probability $P_l(j)$, yielding the new joint configuration $\mathbf{X}^* = \{\mathbf{X}_t^{n-1} \setminus \mathbf{x}_t^{j,n-1}\}$. Acceptance rate is:

$$\alpha_l = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1})P_e Q(\mathbf{x}_t^{j,n-1})}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1})P_l(j)}\right) \quad (6)$$

Object j Update with probability P_u. Proposes to change $\mathbf{x}_t^{j,n-1}$ class according to a transition probability matrix. Randomly choose $\mathbf{x}_{t-1}^{j,r}$, an instance of object j from time $t-1$ chain. Draw $\mathbf{x}_t^{j*}$ from dynamics model $p(\mathbf{x}_t^j|\mathbf{x}_{t-1}^{j,r})$ and build $\mathbf{X}^* = \{\mathbf{X}_t^{\setminus j,n-1}, \mathbf{x}_t^{j*}\}$. Object dynamics model is relative to object category (see section 5 for examples).

$$\alpha_u = min\left(1, \frac{\pi^* w^*}{\pi_t^{n-1} w_t^{n-1}}\right) \quad (7)$$

Sun Enter: proposes sunlight to become bright with probability P_{se}. Acceptance rate is:

$$\alpha_{se} = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1})P_l(s)}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1})P_{se}}\right) \quad (8)$$

Sun Leave: proposes sunlight to become cloudy with probability $P_l(s)$. Acceptance rate is:

$$\alpha_{sl} = min\left(1, \frac{\pi^* w^* P(\mathbf{X}^*|\mathbf{Z}_{1:t-1})P_{se}}{\pi_t^{n-1} w_t^{n-1} P(\mathbf{X}_t^{n-1}|\mathbf{Z}_{1:t-1})P_l(s)}\right) \quad (9)$$

Sun Update with probability P_{su}. Randomly chooses a sun position instance $\mathbf{l}_{t-1}^r$. Draw $\mathbf{l}^*$ from sun dynamics laws (18) and (19). Acceptance rate is given by 7.

3. Observation Likelihood Function

In this section, we compute $P(\mathbf{Z}|\mathbf{X})$, the likelihood for observation $\mathbf{Z}$, given the joint multi-object configuration $\mathbf{X}$. Though we commonly use a multi-camera setting, a mono-vision setting will be considered in this section, for sake of simplicity. From the current image (Fig.2-a), and a background model (Fig.2-b), a foreground binary image $\mathbf{I}_F(g)$ such as in Fig.2-d is computed, where g denotes a pixel location. We use $\Sigma - \Delta$ algorithm [8], which efficiently computes an on-line adaptive approximation of background image temporal median and covariance, thus coping with outdoor illumination changes and noises for a low computational cost. On the other hand, each object hypothesized by particle $\mathbf{X}$ is modeled as a cuboid with shape defined in section 2.1. The convex hull of its vertice projections is computed. If sunlight is unbroken, its cast shadow vertices are computed, and the corresponding convex hull also is computed. A binary mask image $\mathbf{I}_M(g, \mathbf{X})$ is computed, with pixel g set to 1 if it is inside at least one of the convex hulls, else to 0, as drawn in Fig.2-c. Similarity image $\mathbf{I}_S(g, \mathbf{X})$ is then computed (10), as well as *false background* image (11) used to drive object enter proposals through (12), where S_o is object projection prior area.

$$\mathbf{I}_S(g, \mathbf{X}) = \begin{cases} 1 \text{ if } \mathbf{I}_F(g) = \mathbf{I}_M(g, \mathbf{X}), \\ 0 \text{ otherwise} \end{cases} \quad \forall g \tag{10}$$

$$\mathbf{I}_{fb}(g, \mathbf{X}) = \mathbf{I}_F(g) \& \overline{\mathbf{I}_M(g, \mathbf{X})}, \forall g \tag{11}$$

$$\rho_e = \frac{1}{S_o} \sum_g \mathbf{I}_{fb}(g, \mathbf{X}) \tag{12}$$

The observation likelihood $P(\mathbf{Z}|\mathbf{X})$ is computed as (13):

$$p(\mathbf{Z}|\mathbf{X}) = \left(\frac{1}{S} \sum_g \mathbf{I}_S(g, \mathbf{X}) \right)^{\beta_j}, \tag{13}$$

where β_j is computed according to object j projection area, according to the method detailed in [1]. This method is of high interest as it produces an observation likelihood that fairly tracks objects whatever their distance, and that fairly compares occluded and unoccluded objects. Both are highly demanded by video surveillance applications, such as highway or subway surveillance, where cameras cannot be located on a very elevated point, yielding deep occlusions and scale changes due to projection. Moreover, this method allows *MCMC PF* to operate with acceptance rate α to be tuned to a target value, thus improving its efficiency.

4. Multi-Object Interaction Weight

As the foreground likelihood function allows fully occluded objects to survive, we must prevent them from getting stuck behind another object. For pedestrian tracking,

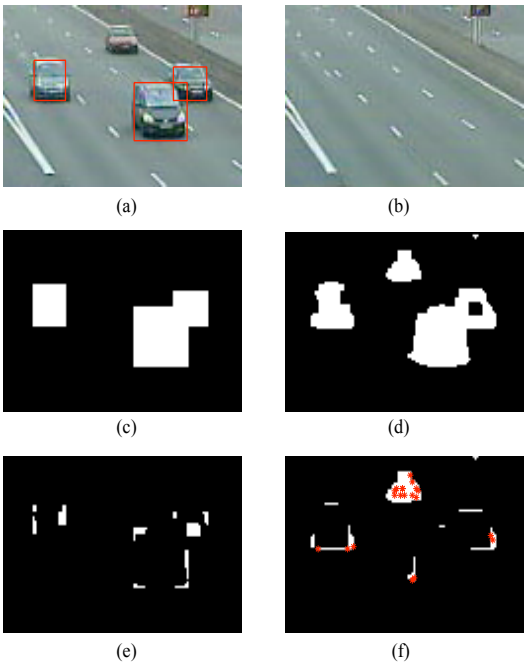


Figure 2. Background subtraction and residual images, with projected candidate objects. For readability, their projective polygons are approximated as rectangles. (a): raw color image with bounding object rectangles. (b): background model. (c): binary hypothesis image $\mathbf{I}_M(g, \mathbf{X})$. (d): binary foreground image $\mathbf{I}_F(g)$. (e): binary *false foreground* image, *i.e.* pixels covered by the projection of at least one object, but classified as background. (f): binary *false background* image $\mathbf{I}_{fb}(g, \mathbf{X})$, *i.e.* pixels not covered by any candidate object, but classified as foreground. Few points randomly sampled (red stars), to drive new object *enter* proposals.

[13] proposes to use a Mahalanobis distance rather than an Euclidean distance to model distances between pedestrians. We also compute an inter-object anisotropic weight based on Mahalanobis distance. This is mostly required in the case of vehicle tracking, because their lengths are much larger than their widths, and their interactions also are highly anisotropic, as 2 nearby vehicles are more likely to ride on 2 adjacent lanes rather than on the same lane. Moreover, the interaction between two vehicles depends on their dimensions. This is modeled by computing object interaction weight w as a function of an anisotropic distance between every pair of hypothesized vehicles. Both conditions are met approximating each object as a bivariate gaussian mass distribution, with covariance matrix featuring second order mass moments. Inter-vehicle distance then is: $d_{ij} = (\mathbf{\Delta}_{ij}^T.(\mathtt{C}_i.\mathtt{C}_j)^{-1}.\mathbf{\Delta}_{ij})^{1/2}$, where $\mathbf{\Delta}_{ij}$ is the 2D position difference vector between vehicles i and j, $\mathtt{C}_i$ and $\mathtt{C}_j$ their respective covariance matrices. Object pair interaction

weight then is computed according to equation (14):

$$w_{ij} = \left(1 + e^{-k_s.(d_{ij}-d_s)}\right)^{-1}, \tag{14}$$

yielding a weight near 1 for far objects, and near 0 for materially impossibly close objects. d_s is the inter-vehicle distance corresponding to the sigmoid inflection parameter, and k_s is adjusted to tune curve slope around d_s. Interaction weight for particle $\mathbf{X}$ involving J_t^n objects then is:

$$w(\mathbf{X}) = \prod_{i=1}^{J_t^n-1} \prod_{j=i+1}^{J_t^n} w_{ij}. \tag{15}$$

5. EXPERIMENTS AND RESULTS

5.1. Datasets and Methodology

Tracker performance is assessed over both synthetic and real sequences. Datasets have been sampled from two different fields of applications: pedestrian tracking and highway vehicle tracking. Pedestrian tracking experiments are devoted to assessing the tracker ability to track more than 10 objects while coping with variable sunlight conditions. Highway vehicle tracking experiments are devoted to assessing the tracker ability to simultaneously track and classify vehicles such as cars, light trucks and trailer trucks, while also complying with time-evolving sunlight. As we want our tracker to comply with poor acquisition data, real sequences are provided by low-quality non-calibrated webcams with a 320×240 pixel resolution and a high compression rate. Moreover, projection matrices have been approximated by hand. Target objects located within a selected tracking area (defined in the 3-d world and overplotted with green lines on figures 3, 4 and 5) are to be tracked and classified. We propose to assess the proposed method performance over four criteria:

- Tracking rate $\theta_T = \frac{1}{J_t}\sum_{t,j}\delta_T(t,j)$ with $\delta_T(t,j) = 1$ if target j is tracked at time t, else 0. $J_t = \sum_t j_t$, with j_t the number of objects in the tracking area.
- Classification rate $\theta_C = \frac{1}{J_t}\sum_{t,j}\delta_C(t,j)$ where $\delta_T(t,j) = 1$ if target j class is correct at time t, else 0.
- Ghost rate $\theta_G = \frac{1}{J_t}\sum_{t,j}\delta_G(t,j)$ where $\delta_G(t,j)$ is the number of *ghosts i.e.* candidate objects over no target.
- Position average error $\varepsilon_T = \frac{1}{J_t}\sum_{t,j}(\boldsymbol{\delta}_p^T.\boldsymbol{\delta}_p)^{-1}$, with $\boldsymbol{\delta}_p = \mathbf{p}_t^{j,e} - \mathbf{p}_t^{j,gt}$, where $\mathbf{p}_t^{j,e}$ is object j estimated position at time t, $\mathbf{p}_t^{j,gt}$ is object j position ground truth.

Four methods are assessed according to $\theta_T, \theta_C, \theta_G, \varepsilon_T$:

- MOT - Multi Object Tracker: an implementation of *RJMCMC* algorithm with one category (object size noise has been increased in order to match different size objects) and with no light estimation.
- MOTS - Multi Object Tracker and Sun: an implementation of *RJMCMC* algorithm with one category (object size noise has been increased in order to match different size objects) and with light estimation.
- MOTC^n - Multi Object Tracker and Classifier: an implementation of *RJMCMC* algorithm with n categories and with no light estimation.
- MOTC^{n}S - Multi Object Tracker and Classifier with Sun: an implementation of *RJMCMC* algorithm with n categories and with light estimation.

5.2. Implementation

Two configuration proposals and their likelihoods are computed in parallel on each processing core, through threads supplied by the Boost C++ Libraries[1]. Code is written using NT^2 C++ Library[2]. We use a 3GHz Intel E6850 Core 2 Duo processor PC, with 4Go RAM, running Linux. All experiments presented below have been done at video real time (*i.e.* 25 fps), over mono-vision 320×240 frames. The filter number of particles is set to $N = 200$.

5.3. Pedestrian tracking under variable sunlight

Datasets are sampled from pedestrian tracking sequences. Candidate pedestrians are controlled in velocity:

$$p(\mathbf{v}_t|\mathbf{v}_{t-1}^r) = \mathcal{N}\left(\mathbf{v}_{t-1}^r, diag\left(\left[\sigma_m^2, \sigma_a^2\right]\right)\right), \tag{16}$$

where σ_m and σ_a are the respective velocity magnitude and orientation standard deviations. Acceleration is not used. Dynamics laws then yield position $\mathbf{x}_t^*$. Shape is updated according to equation (17), where σ_s is object shape standard deviation, and I_3 the 3-dimension identity matrix. Sun dynamics is defined by (18) and (19), where σ_ϕ and σ_ψ respectively are sun azimut and elevation standard deviations.

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}^r) = \mathcal{N}(\mathbf{s}_{t-1}^r, \sigma_s^2 I_3) \tag{17}$$

$$p(\phi_t|\phi_{t-1}^r) = \mathcal{N}(\phi_{t-1}^r, \sigma_\phi^2), \forall r \in \{1, ..., N\} \tag{18}$$

$$p(\psi_t|\psi_{t-1}^r) = \mathcal{N}(\psi_{t-1}^r, \sigma_\psi^2), \forall r \in \{1, ..., N\} \tag{19}$$

Synthetic Sequences: Cuboid approximated pedestrians randomly move on a 12x15 meter wide tracking area, under a simulated time-evolving bright sunlight with elevation $\psi = 0.8$ *rad* and azimut increasing from $\phi = 0$ to $\phi = \pi$ *rad* in 1000 frames. This is much faster than real world sun moves. Figure 3 illustrates tracking operation, showing the benefit of shadow modeling. Table 1 reports results for MOT and MOTS, and shows that modeling cast shadows decreases ghost rate and improves tracking accuracy.

[1] http://www.boost.org
[2] Numerical Template Toolbox. http://nt2.sourceforge.net

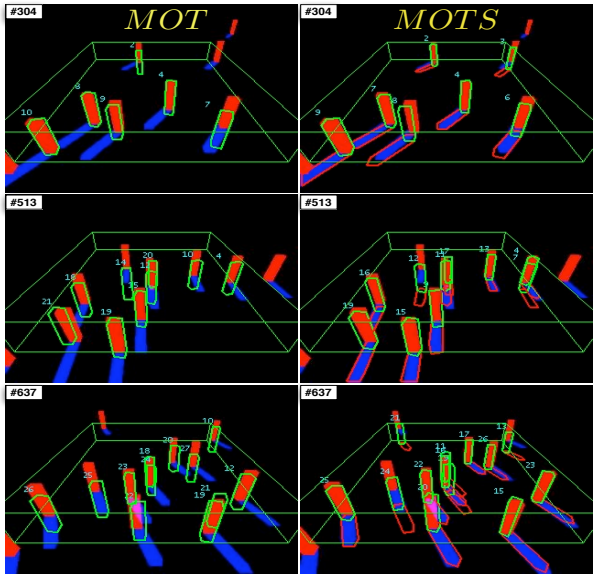


Figure 3. Excerpts from synthetic pedestrian tracking under time-evolving sunlight azimut. Estimated object cuboids overplotted in green lines. Left column: no shadow model. Right column: estimated cast shadow overplotted in red.

Table 1. Tracking cuboid approximated pedestrians on synthetic scenes, under time-evolving sunlight, and under an alternation of bright and cloudy sunlight: sun state changes every 200 frames.

	bright sun		**sun & clouds**	
	MOT	MOTS	MOT	MOTS
θ_T **(%)**	84.7	89.7	84.1	87.0
θ_G **(%)**	5.7	4.9	5.1	4.3
error (m)	0.91	0.63	0.82	0.70

Real Sequence: A short sequence with clouds and sun yielding fast illumination changes. Figure 4 frame #786 illustrates three pedestrians being tracked while sunlight is estimated to be cloudy (no estimated cast shadow). Few frames later, as sunlight becomes brighter the tracker estimates it to appear at frame #823 and to remain bright until the end. The tracker fails at estimating the two targets walking side by side and occluding each other over the whole sequence: it tracks both people as a unique pedestrian (#14), due to lack of observability.

5.4. Vehicle tracking and classification

These experiments aim at assessing the tracker ability to simultaneously track and classify vehicles such as cars, light trucks and trailer trucks. Vehicles are controlled through driver command proposals drawn from (20):

$$p(\mathbf{a}_t|\mathbf{a}^r_{t-1}) = \mathcal{N}\left(0, diag\left(\left[\sigma_l^2, \sigma_t^2\right]\right)\right), \qquad (20)$$

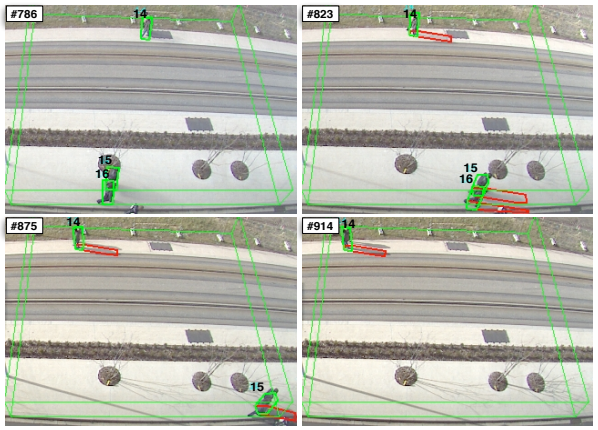


Figure 4. Excerpts from pedestrian tracking under time-evolving sunlight conditions. Estimated cuboids overplotted in green lines with estimated cast shadow in red when sunlight is bright.

Table 2. Two-class synthetic highway vehicle tracking and classification. Tracking rate θ_T (%) / Classification rate θ_C (%) / Ghost rate θ_G(%). Average position error per vehicle in meters.

	MOT	MOTS	MOTC^2	MOTC^{2}S
light vehicles	.	.	59/54/0	**90/89/11**
trailer trucks	.	.	86/86/0	**90/89/0**
total	52/22/17	51/25/16	58/53/0	**90/89/11**
error (m)	6.17	5.80	2.76	**2.00**

where σ_l is driver longitudinal acceleration standard deviation, σ_t is driver steer angle standard deviation, conditionning transversal acceleration. Bicycle model equations as defined in [1] then are applied to object j. Dynamics laws then yield velocity $\mathbf{v}^*_t$ and position $\mathbf{x}^*_t$.

Synthetic Sequences: They involve car and truck cuboid approximates on a three-lane highway, under bright sunlight. Table 2 reports results, showing that both classification and shadow modeling independently improve tracking. Best results are reached when both are activated.

Real Sequences: Real traffic sequences involving light vehicles, light trucks, and trailer trucks on a four-lane highway, including a highway entry lane, under variable sunlight. For real traffic tracking, a 3-class classification is necessary to take into account the three major classes of vehicles. Due to tracked object size wide range, methods without classification (MOT and MOTS) require vehicle shape dynamics σ_s to be dramatically increased, to let objects fit targets. Such a strategy cannot operate in presence of deep occlusions. To let them serve as reference anyway, as well as to let hand-made ground truth be affordable,

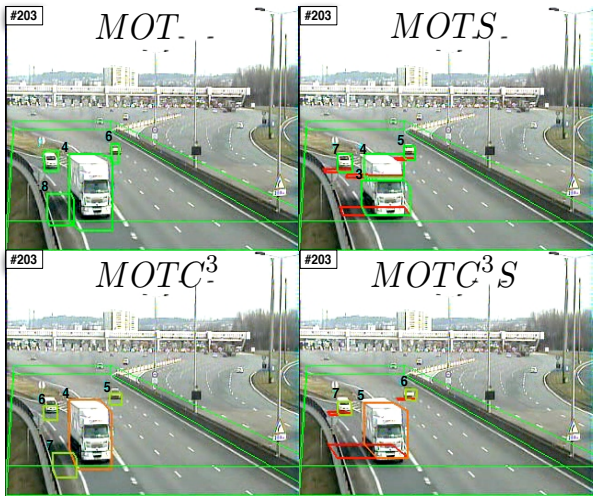


Figure 5. Frame #203 from real traffic tracking. Top row: no classification, vehicle estimated cuboids overplotted in green lines. Bottom row: classification in 3 categories with light vehicle (resp. light trucks and trailer trucks) estimated cuboids overplotted in green lines (resp. magenta and orange). Left: without cast shadow model. Right: with cast shadow modeling, overplotted in red lines.

Table 3. Three-class real highway vehicle tracking and classification. Tracking rate θ_T (%) / Classification rate θ_C (%) / Ghost rate θ_G(%). Average position error per vehicle in meters.

	MOT	MOTS	MOTC^3	MOTC^{3}S
light vehicles	.	.	67/64/2.6	**67/67/0.05**
light trucks	.	.	83/36/1.0	**92/86/3.7**
trailer trucks	.	.	93/83/0	**100/100/2**
total	51/45/0	60/51/0	72/62/2.5	**70/70/3.1**
error (m)	6.80	6.22	6.13	**5.40**

we choose a sequence with light traffic, but involving all categories of vehicles. Figure 5 illustrates that both multi-category classification and cast shadow modeling improve tracking. MOT and MOTS typical failures are: two objects tracking a unique target (MOTS) or poor tracking accuracy (MOT). Without cast shadow modeling, the system fails at tracking very differently sized objects: it explains truck cast shadow pixels classified as foreground with a *ghost* car (#7 on MOTC^3 and #8 on MOTC^{3}S). Modeling cast shadow explains these foreground pixels (MOTC^{3}S). Moreover, further cars are more accurately located when shadow is modeled (MOTS and MOTC^{3}S), as their shadows provide clues concerning their longitudinal position. Table 3 reports results and confirms section 5.4 analysis: both classification and shadow modeling improve tracking, with best results when both are activated.

6. Conclusion and Future Works

We have proposed a generic illumination-aware framework to simultaneously track and classify multiple objects into various classes in real time. The system can be operated in monovision or with a multi-camera setting. It is wholy integrated within a RJ MCMC Particle Filter framework. To make this possible, illumination is integrated into the global configuration state-space, and tracked as well as objects. Experiments show that joint object and sunlight estimation improves tracking, both decreasing false positives and object position error. We also have proposed to include object category as a discrete random variable to be estimated by the filter, extending *RJ MCMC PF* framework to object classification functionnality. Experiments show that simultaneously tracking and classifying improves tracking as it proposes multiple geometric models, thus allowing better model fitting. This unified approach also is of high interest as it allows tracking and classification to cooperate through object class specific dynamics. This functionality might be used to improve tracking and classification of objects with similar geometric models, but with different dynamics models, such as cyclists and pedestrians for instance. As this tracker is designed to be generic, it is based on low level information (simple background segmentation), and complies with low-quality acquisition data. There is undoubtedly room for improvement, adding object ad-hoc features in the likelihood computation. The work presented in this paper deals with a unique illumination source, well suited to model sun illumination. It can easily be extended to multiple illumination sources, and to ground reflection modeling suitable for indoor lighting or outdoor wet conditions.

Appendix: Vitality-Driven Leave Moves

Object and sun vitalities, ranging from 0 to 1, are updated by the same process. At iteration n of time t, we compute object j *false foreground* ratio $f_t^{j,n}$:

$$f_t^{j,n} = \frac{1}{|\mathcal{R}_t^{j,n}|} \sum_{g \in \mathcal{R}_t^{j,n}} \overline{\mathbf{I}_F(g)}, \forall j \in \{1, ..., J_t^n, s\} \quad (21)$$

where s denotes sun as an object. $\mathcal{R}_t^{j,n}$ denotes image region covered by the projection convex hull of each object but the sun ($\forall j \in \{1, ..., J_t^n\}$). For the sun ($j = s$), $\mathcal{R}_t^{j,n}$ is the region covered by the union of all object cast shadow projections. Object j vitality increment λ_t^j is computed (22) over the whole particle set, as a sum of sigmoids of $f_t^{j,n}$:

$$\lambda_t^j = k_d \sum_{n=1}^{N} \frac{e^{-k_r.(f_t^{j,n} - r_f)} - 1}{e^{-k_r.(f_t^{j,n} - r_f)} + 1}, \forall j \in \{1, .., J_t^n, s\} \quad (22)$$

where r_f is the inflection parameter of false foreground rate curve (*i.e.* the value of $f_t^{j,n}$ yielding an increment equal to

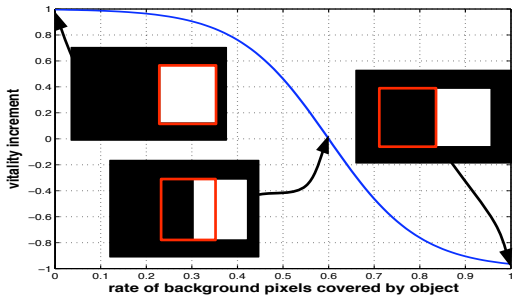


Figure 6. Vitality increment λ_t^j versus object j false foreground rate $f_t^{j,n}$, in monovision, with $k_d = 1$, $r_f = 0.6$ and $k_r = 10$.

0), and k_r is the curve steepness parameter. Equation (22) produces a positive increment if $f_t^{j,n} < r_f$, else negative, allowing object vitality to compile the history of object j likelihoods along past iterations and time steps. Vitality dynamics coefficient k_d is computed in equation (23):

$$k_d = (1 - \Lambda_0)(n_s.C.N)^{-1}, \tag{23}$$

where Λ_0 denotes object initial vitality, n_s denotes the number of frames an object with maximal vitality can survive total invisibility (generally due to total occlusion by the background). This parameter allows the user to adjust vitality dynamics, depending on the duration of possible occlusions. Each object vitality is finally updated for time $t+1$:

$$\Lambda_{t+1}^j = \begin{cases} min(\Lambda_t^j + \lambda_t^j, 1) \text{ if } (j = s \text{ or } z_j) \\ max(\Lambda_t^j + \lambda_{out}, 0) \text{ otherwise} \end{cases}, \tag{24}$$

where z_j is a binary variable set to 1 if object j is in the tracking area, else 0. In the latter case, its vitality is updated by λ_{out}. The values chosen for experiments, reported in table 4, yield the vitality increment illustrated on Fig. 6. At each time step t, object j *leave* proposal rate $\rho_l(j)$ is driven by its own vitality, according to equation (25):

$$\rho_l(j) = \left(1 + e^{k_v.(\Lambda_t^j - \Lambda_0)}\right)^{-1}, \forall j \in \{1, .., J_t^n, s\}. \tag{25}$$

Sigmoid inflection parameter is chosen equal to Λ_0, yielding object *enter* and *leave* reversibility. Less $leave$ proposals appear as object j vitality grows higher than Λ_0, preventing it from leaving the scene at once when poorly segmented from background or deeply occluded. In this case, vitality allows it to survive several images. k_v is sigmoid steepness parameter. The same mechanism stands for sun, with slower dynamics driven by a higher n_s (see table 4).

Table 4. Object vitality computation parameters. $n_s = 25$ frames means 1 second long total occlusion survival at 25 fps.

	Λ_0	λ_{out}	r_f	n_s	k_r	k_v
object	0.2	-0.1	0.6	10	10	10
sun	0.2	.	0.4	50	10	10

References

[1] F. Bardet and T. Chateau. MCMC particle filter for real-time visual tracking of vehicles. In *International IEEE Conference on Intelligent Transportation Systems*, pages 539 – 544, 2008. 4, 6

[2] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 4(82):711–732, 1995. 3

[3] M. Isard and J. MacCormick. Bramble: A bayesian multiple-blob tracker. In *Proc. Int. Conf. Computer Vision, vol. 2 34-41*, 2001. 2

[4] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. *ECCV*, 3024:279–290, 2004. 2

[5] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1805 – 1918, 2005. 1, 3

[6] M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. *IJCV : International Journal of Computer Vision*, 29(1):5–28, 1998. 2

[7] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. 2

[8] A. Manzanera and J. Richefeu. A robust and computationally efficient motion detection algorithm based on sigma-delta background estimation. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'04).*, pages 46–51, 2004. 4

[9] J. L. M. Matthew J. Leotta. Learning background and shadow appearance with 3-d vehicle models. In *British Machine Vision Conference (BMVC)*, volume 2, pages 649–658, september 2006. 2

[10] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:918–923, 2003. 1

[11] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *Computer Vision and Image Understanding*, 95(2):238 – 259, August 2004. 1

[12] K. Smith. *Bayesian Methods for Visual Multi-Object Tracking with Applications to Human Activity Recognition*. PhD thesis, EPFL, Lausanne, Suisse, 2007. 2, 3

[13] J. Yao and J.-M. Odobez. Multi-camera multi-person 3D space tracking with mcmc in surveillance scenarios. In *European Conference on Computer Visionworkshop on Multi Camera and Multi-modal Sensor Fusion Algorithms and Applications (ECCV-M2SFA2)*, 2008. 1, 4

[14] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu. Crowd analysis: a survey. *Machine Vision and Applications*, 19(5-6):345–357, 2008. 1

A.4 Estimation précise de la trajectoire d'un véhicule

Tracking of Vehicle Trajectory by Combining a Camera and a Laser Rangefinder
Y. Goyat, T. Chateau et L. Trassoudaine
Springer MVA : Machine Vision and Application, online, Mars 2009

Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Y. Goyat · T. Chateau · L. Trassoudaine

Tracking of Vehicle Trajectory by Combining a Camera and a Laser Rangefinder

Received: date / Accepted: date

Abstract This article presents a probabilistic method for vehicle tracking using a sensor composed of both a camera and a laser rangefinder. Two main contributions will be set forth in this paper. The first involves the definition of an original likelihood function based on the projection of simplified 3D vehicle models. We will also propose an efficient approach to computing this function using a line-based integral image. The second contribution focuses on a sampling algorithm designed to handle several sources. The resulting modified particle filter is capable of naturally merging several observations functions in a straightforward manner. Many trajectories of a vehicle equipped with a kinematic GPS[1] have been measured on actual field sites, with a video system specially developed for the project. This field input has made it possible to experimentally validate the result obtained from the algorithm. The ultimate goal of this research is to derive a better understanding of driver behavior in order to assist road managers in their effort to ensure network safety

Keywords Visual tracking · particle filter · sensor fusion

Y. Goyat
LCPC
Route de Bouaye
44341 Bouguenais
FRANCE
Tel.: +33-240845852
Fax: +33-240845992
E-mail: yann.goyat@lcpc.fr

T. Chateau, L. Trassoudaine
LASMEA
24, av. des Landais
63177 Aubire Cedex
FRANCE
Tel.: +33-473407660
E-mail: thierrry.chateau@lasmea.univ-bpclermont.fr

[1] This device is an absolute localization sensor with a level of accuracy on the order of one centimeter

1 Introduction

We present an online vehicle trajectory tracking method using a sensor composed of a color camera and a one-dimensional scanning laser rangefinder. The objective of this system is to accurately estimate the trajectory of a vehicle traveling through a curve. The research reported herein lies within the scope of an French ANR-PREDIT project [2]

The sensor, installed in a curve, is composed of three cameras placed on a tower approximately 5 m high to cover the beginning, middle and end of the curve, in addition to a scanning laser rangefinder laid out parallel to the ground. Since the cameras offer only limited coverage, their observations do not overlap and we will be

[2] See acknowledgments section

considering in the following discussion that the system can be divided into three subsystems, each composed of a camera-rangefinder pair, with the recalibration between each pair performed by means of rigid transformations, which will not be addressed in the present article. The object-tracking procedure is intended to estimate the state of an object at each moment within a given scene, based on a scene observation sequence. Tracking methods can be broken down into two major categories: the first concerns off-line or non-causal tracking, for which the state estimation at a given point in time uses the entire observation sequence [5]. The second category relates to online or causal tracking, for which the state of the object at a given point in time has been estimated as a function of the record of past and current observations and the record of past states [9]. This second category may also encompass the notion of realtime when the period necessary to estimate a state is shorter than the sensor acquisition frequency. The method described herein would be classified as a causal method.

In this article, we are proposing a solution based on a probabilistic formalization of the tracking problem with a stochastic framework (particle filter). The literature, which contains many references in the areas of vision and data merging, proceeds with the recursive time estimation of a state through application of Monte-Carlo methods [1].

The rest of the article has been divided into four parts. The first will discuss the tracking principle, on the basis of a probabilistic model. The next part will focus on the likelihood functions proposed for estimating the weights associated with the particle set. The third part will provide a detailed description of our proposed sampling method (called multi-source sampling). The last section will then present the measurement campaign conducted for the purpose of quantifying method accuracy and robustness. A large number of trajectories could be estimated and a kinematic GPS was used to determine the actual field values associated with each trajectory.

2 Related Work

Recently, traffic video surveillance has become an important topic in the Intelligent Transport Systems (ITS), so vehicle detection, description, and/or recognition have been an active research field. Most of the solutions assume that the camera is at a high angle. Tracking vehicles from a static camera is challenging for several reasons:

- Outdoor vision tracking solutions must handle with variation of illumination and with shadows.
- Generic model based approaches are complex solutions due to the huge appearance variability of the vehicle class.
- Tracking several vehicles simultaneously implies to be able to handle with mutual occlusions.

Tracking objects from a static camera often uses a background/foreground subtraction [17]. In outdoor environment such method must handle with variations of illumination conditions and shadow. In [23], Stauffer and al. propose a parametric model with a temporal update. The resulting solution deals efficiently with small illumination variations. In [27], Sheikh and Shah propose a Bayesian based approach for object detection in dynamic scenes using correlation that exists between neighborhood pixels within non-parametric distribution models. Moreover, detecting shadow can be done by color based analysis [10] and injecting temporal information into the models [18].

Many object descriptions can be used in order to obtain a model of a vehicle. Some approaches are based on a generic model, uses to detect, then track the vehicle, from a luminance [2] or Haar Wavelets based model [3,21]. 3D wireframes models have been also used but they require to define many models associated to different types of vehicles [26,14,15,24,8]. In [19] a non-rigid 3D model is used into a EM based contour tracking. Recently, Kanhere and al. [12] have proposed an interest point based method to segment and track vehicles. The method presented here uses a $3D$ simplified model of a vehicle, projected into the image, and then compared to the background/foreground subtraction map to provide and efficient observation function.

In [11], Kamijo and al. propose a probabilistic model to track multiple vehicles with spatio-temporal Markov random fields. Recently, stochastic methods [13,22] have been presented to handle with realtime multi object tracking. In [25] Yu and al. propose a Monte Carlo Markov Chain method to estimate, from a video, global trajectories of the vehicles.

In this paper, we propose a probabilistic framework to solve the problem of online vehicle trajectory estimation. The trajectory is modeled by a random state vector and the distribution associated to this random vector is approximated in a sequential scheme with a particle filter. Since the trajectory is highly driven by the kinematic model of the vehicle, we propose to inject this model within the dynamics associated to the filter. Moreover, an original data fusion sampling algorithm is proposed to handle with several observation functions.

3 Principle of the Method

This section will set forth the principle behind the trajectory tracking method.

The core of this proposed method (see Fig. 1) consists of a recursive filter that has been formalized stochastically (using a particle filter). The vehicle state is represented by its corresponding bicycle model. The prediction function adopts the hypothesis of a constant driving angle acceleration and speed. A 3D geometric model of the vehicle, projected onto the image, is then compared with image data described by a probabilistic shape assimilation map. The observation function also comprises a likelihood function that reflects the consistency of telemetric data with respect to the hypothesis. The filter is able to produce, at each iteration, an estimation of the vehicle state (position, heading direction, speed, steering angle).

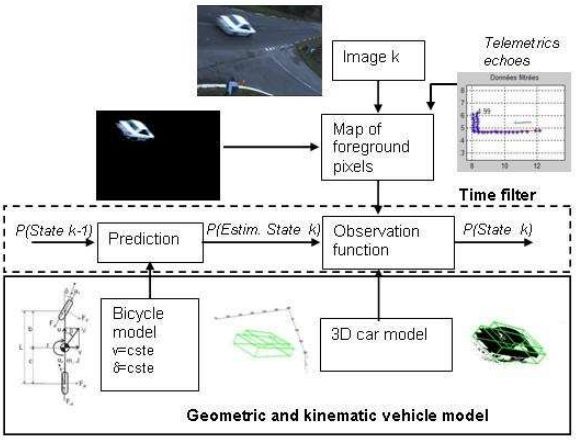


Fig. 1 Block diagram of the proposed method

3.1 Background Extraction

The majority of methods for tracking objects within a static scene introduce a background extraction step, which

consists in a binary case of ascribing each image pixel a "Background/Foreground" class. Most of these assignments are statistical, which forwards the hypothesis that each pixel may be modeled by a random variable capable of assuming either the "Background" or "Shape" state. Stauffer and Grimson [23] proposed employing a parametric Gaussian Mixture Model (GMM) in order to depict the probability density associated with each pixel. It is also possible to use a nonparametric probability density model, such as the one in [9]. This will be the approach adopted herein.

Let $\mathtt{I}_t$ be an image acquired at time t, and $\mathbf{y}_t(\mathbf{u}) \doteq \{y_t(i,\mathbf{u})\}_{i=1,2,3}$ the function that provide information on pixel $\mathbf{u}$ of the CCD sensor according to the three colorimetric components: Red, Green and Blue. Each pixel constitutes a discrete random variable capable of assuming either one of the two following states: 1) "Background" (ω_1), and 2) "Foreground" (ω_2).

3.1.1 Background Model

We propose a discrete model for the likelihood $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$ by marginalizing the three color planes, given that: 1) parametric methods in most cases use a Gaussian modeling approach; and 2) the space discretization of parameters is very costly in terms of computation time should the parameters be considered as dependent upon one another. A discretization of $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$ actually corresponds to N^3 elements if each parameter of $\mathbf{y}_t \in I\!R^3$ has been sampled using N points. If parameters were independent, $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$ could be discretized with just $3N$ elements. The hypothesis of independent variables in this instance is not treated rigorously since the three color components are correlated among each other. Nonetheless, this hypothesis often gets adopted in order to reduce computation time and memory space requirements. Now, let's express $\mathbf{b}_t(\mathbf{u}) \doteq \{b_t(i;\mathbf{u})\}_{i=1,..,3}$ as the histogram associated with color vector $\mathbf{y}_t(\mathbf{u})$ of position $\mathbf{u}$ in the image at time t. It then becomes possible to approximate the likelihood function $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$ by:

$$p(\mathbf{y}_t(\mathbf{u})|\omega_1) = \prod_{i=1}^{3} p(y_t(i;\mathbf{u})|\omega_1) \tag{1}$$

with :

$$p(y_t(i;\mathbf{u})|\omega_1) \approx K \sum_{j=1}^{N} q_t^b(i,j;\mathbf{u}) d(b_t(i;\mathbf{u}) - j), \tag{2}$$

where d represents the Kronecker function and K a constant assigned to standardize the term (dependent on i and $\mathbf{u}$). $\sum_{j=1}^{N} q_t^b(i,j,\mathbf{u}) = 1$. $q_t^b(i,j;\mathbf{u})$ is a weight function associated with the discrete model of the likelihood function $p(\mathbf{y}_t(\mathbf{u})|\omega_1)$. The model evolves in sync with this weight function. $q_{t+1}^b(i,j;\mathbf{u})$ is updated with each image by application of the following AR equation:

$$q_{t+1}^b(i,j;\mathbf{u}) = \frac{1}{1+\alpha}.[q_t^b(i,j;\mathbf{u}) + \alpha.d(b_{t+1}(i;\mathbf{u}) - j)] \tag{3}$$

3.1.2 Shape Model

The shape model is defined by the likelihood of assimilation in the shape category $p(\mathbf{y}_t(\mathbf{u})|\omega_2)$. Even though statistically speaking, some pixels (e.g. those positioned above the horizon line) display a smaller probability of belonging to this shape, we are considering herein that the likelihood of assimilation to the shape category does not depend on the position of the observed pixel. It thus becomes possible to approximate the likelihood function $p(\mathbf{y}_t(\mathbf{u})|\omega_2)$ by :

$$p(y_t(i;\mathbf{u})|\omega_2) \approx K \sum_{j=1}^{N} q_t^f(i,j) d(b_t(i;\mathbf{u}) - j), \quad (4)$$

The weights $q_t^f(i,j)$ represent a distribution discretization, which has been marginalized *a priori* from the color of the shape object being sought. When no information is available on the model of objects present in the foreground, an equal probability hypothesis is to be considered. In this particular case, the terms $q_t^f(i,j)$ stem from a color histogram of the tracked object, as extracted during an initialization phase.

3.1.3 Probabilistic Shape Assimilation Map

A probabilistic shape assimilation map is generated from the likelihood ratios $p(y_t(i;\mathbf{u})|\omega_2)/p(y_t(i;\mathbf{u})|\omega_1)$. By expressing this ratio in log-likelihood form, it becomes possible to build a pseudo-image of the log-likelihood ratio in which the value associated with the pixel located at coordinates $\mathbf{u}$ is calculated using the following $l_{m,t}(\mathbf{u})$ function :

$$l_{m,t}(\mathbf{u}) \doteq \log\left(p(y_t(i;\mathbf{u})|\omega_2)\right) - \log\left(p(y_t(i;\mathbf{u})|\omega_1)\right) \quad (5)$$

3.2 Particle Filter

Vehicle trajectory is estimated recursively using a nonlinear filter, whose resolution entails a widespread stochastic method for vision applications: the particle filter. This choice is dictated by the nonlinear nature of the system. Particle filtering [1,16] is based on an estimation of the a posteriori probability density $p(\mathbf{X}_t|\mathbf{Z}_{0:t})$ of state $\mathbf{X}_t$ conditioned by the historical record of measurements $\mathbf{Z}_{0:t}$, at time t, by a set of N weighted particles $\{(\mathbf{X}_t^n, \pi_t^n)\}_{n=1}^N$ with their associated weights. In the case of an observation stemming from several sources, we are proposing to merge observations intrinsically during the re-sampling phase. The algorithm derived (see Algorithm 1) is a variant of the CONDENSATION algorithm [16]. A weight vector (composed of weights generated from observations of each source) is to be associated with each particle. This multi-source re-sampling method (called MSS) will be further developed in Section 5, page 11.

Algorithm 1 CONDENSATION in the multi-source case

Init : particles $\{(\mathbf{X}_0'^n, \mathbf{1}/N)\}_{n=1}^N$ according to the initial distribution $\mathbf{X}_0$
for $t = 1, ..., T_{end}$ **do**
  Prediction : generation of $\{(\mathbf{X}_t^n, \mathbf{1}/N)\}_{n=1}^N$ from $p(\mathbf{X}_t|\mathbf{X}_{t-1} = \mathbf{X}_{t-1}'^n)$
  Observation : estimation of the weight vector according to the various sources $\{(\mathbf{X}_t^n, \boldsymbol{\pi}_t^n)\}_{n=1}^N$ with $\boldsymbol{\pi}_t^n \propto \mathbf{p}(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{X}_t^n)$
  Sampling : build $\{(\mathbf{X}_{t-1}'^n, \mathbf{1}/N)\}_{n=1}^N$ from $\{(\mathbf{X}_0^n, \boldsymbol{\pi}_0^n)\}_{n=1}^N$ using Multi Source Sampling (MSS)
  Estimation : $\hat{\mathbf{X}}_t \doteq \frac{1}{N}\sum_{n=1}^N \mathbf{X}_t^n$
end for
Output : The set of estimated states during the video sequence $\{\hat{\mathbf{X}}_t\}_{t=1,...,T_{end}}$

Implementing a particle filter necessitates defining three models: 1) a state model that serves to define the kinematic characteristics of the object to be tracked; 2) an evolution model that defines the state of an object at a given point in time depending on the state at the previous point in time; and 3) an observation model that defines a measurement between a state hypothesis and the observations. The state and evolution model are presented in the following sections and the observation model (the core of the method) is detailed in section 4.

3.2.1 State Model

The state model, which identifies the trajectory characteristics to be tracked, must integrate the constraints

related to vehicle kinematics. We are proposing to use a bicycle type of model and then recognize and focus on many behavioral and stability properties. The hypotheses inherent in this model are as follows:

- no transfer of lateral load; the vehicle is thus compressed onto a single path,
- no longitudinal transfer,
- no roll or pitch motion,
- tires in a linear configuration,
- constant forward speed V,
- no aerodynamic effects,
- position control, and
- no effect from suspension and chassis flexibility.

These hypotheses all imply that: vehicle acceleration at all times remains below 0.4 g (i.e. linear operating regime for tires); the steering angle, drift angle, etc. are small; and the ground surface is smooth (no suspension displacement). As an initial approach, it has been considered that vehicles are negotiating the turn at low speed; centrifugal forces are thus negligible and the tires must not develop any lateral forces, i.e.:

- rolling without sliding or drift,
- in order for tires not to slide laterally, the instantaneous center of rotation (ICR) of each tire must lie in the middle of the curve.

Let (x, y) be the selected coordinate system; the kinematic relations governing this model can then be written as follows:

$$\begin{aligned} \dot{x} &= v.\cos\beta \\ \dot{y} &= v.\sin\beta \\ \dot{\beta} &= \tfrac{v}{L}.\tan\delta \end{aligned} \tag{6}$$

with β representing the vehicle direction and δ the steering angle of the front wheel within the world coordinate system (the extended Lambert II system [3]) along the vehicle axis.

According to Ackerman's theory at low-speed behavior, the ICR lies at the intersection of the extension to the rear wheel axis and the perpendicular to the front wheel plane [6]. The ideal front wheel deviation can be deduced from the construct illustrated in Figure 2 and its angle then written: $\tan\delta = \frac{L}{R}$, with L denoting the vehicle wheelbase and R the curve radius of the road. By adopting the small angle hypothesis $\delta = \frac{L}{R}$, the system state vector is expressed as follows:

$$X_t \doteq (\mathbf{P}_t, \beta_t, \delta_t, v_t)^t \tag{7}$$

with $\mathbf{P}_t \doteq (x_t, y_t)$ representing the vehicle position and v_t the vehicle speed within the world coordinate system.

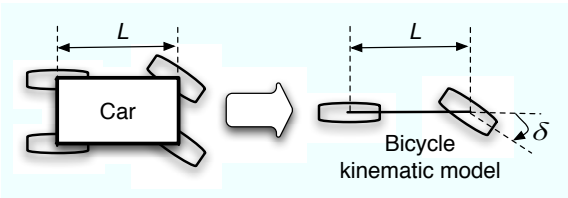


Fig. 2 The bicycle model synthesizes the displacement of a four-wheel vehicle, through the displacement of two wheels whose centers are connected to a rigid axis of length L. Ackerman's theory serves to estimate the steering angle of the front axis of a vehicle traveling at low speed.

[3] The absolute reference frame used here is the NTF geodesic system in extended Lambert II projection.

3.2.2 Prediction Model

The bicycle-type kinematic model applied to each particle evolves according to the following nonlinear form:

$$\begin{aligned} x_{t+1} &= x_t + T.v_t.\cos(\beta_t) \\ y_{t+1} &= y_t + T.v_t.\sin(\beta_t) \\ \beta_{t+1} &= \beta_t + T.\tfrac{v}{L}.\tan\delta_t \\ \delta_{t+1} &= \delta_t + T.b_{\dot{\delta}} \\ v_{t+1} &= v_t + T.b_a \end{aligned} \tag{8}$$

with $b_{\dot{\delta}} \sim \mathcal{N}(0, \sigma_{\dot{\delta}})$ and $b_a \sim \mathcal{N}(0, \sigma_a)$. The two terms $\dot{\delta}$ and a are randomly distributed (according to a Gaussian), while the terms σ_δ and σ_a represent respectively the speed deviation range in the front wheel steering angle and the acceleration range, as performed by a standard vehicle during sampling period T.

3.3 Initialization

The filter initialization process consists of ascribing a hypothesis to each filter particle, such that the entire set of particles offers a stochastic representation of the density associated with the particular state, at the initial time. The vehicle position on the road pavement is initialized based on observations generated from the first image. The steering angle is initialized from an *a priori* distribution calculated as a function of characteristics describing the target curve. The heading direction is initialized using an *a priori* distribution calculated with respect to curve position and characteristics. The speed is calculated by the rangefinder at initial position of the vision process, the two sensors being synchronized temporally.

Determining the *a priori* distribution on the vehicle position relies upon a method for detecting cluster centers, which associates an assimilation probability at the cluster center with each pixel labeled "Shape".

To proceed with this method we take the hypothesis $\alpha << 1$ and a pixel becomes "Background" should its value remains stable for an image number $k >> \alpha$. Then, equation 2 can be simplified and the set of shape pixels $\mathcal{F}_t$ are built by means of thresholding the likelihood function associated with the background assimilation :

$$\mathcal{F}_t \doteq \bigcup_{\mathbf{u}\in\mathrm{I}_t} \{\mathbf{u} | p(\mathbf{y}_t(\mathbf{u}) | \omega_1) < k.\alpha\} \tag{9}$$

Typically $k = 25$ and $\alpha = 0.01$. This probability is obtained from a non-parametric model based on Parzen estimators [4]:

$$\pi_t^n \propto p(\mathbf{Z}_t | \mathbf{X}_t = \mathbf{X}_t^n) \approx \frac{1}{|\mathcal{F}_t|} \sum_{\mathbf{u}\in\mathcal{F}_t} \varphi(\mathbf{p}_t^n, \mathbf{u}) \tag{10}$$

where $\varphi(\mathbf{p}_t^n, \mathbf{u})$ is a Gaussian kernel defined by:

$$\varphi(\mathbf{p}_t^n, \mathbf{u}) = \frac{1}{(2\pi).|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{p}_t^n - \mathbf{u})^t \Sigma^{-1} (\mathbf{p}_t^n - \mathbf{u})\right] \tag{11}$$

Given a calibrated camera, projection of a 3D simple model of the vehicle on the ground plane is achieved. The resulting projection defines a closed shape into the image (approximated by a convex hull). Σ is then defined as the covariance matrix computed from all points within the convex hull. In consequence, Σ depends on the estimated vehicle projection size in the image plane ; as the vehicle approaches the camera, its size grows in the image. Fig-

ure 3 shows the assimilation probability distribution at the cluster center for the background/shape extraction example. In order to decrease the computation time, a look up table given Σ according to the position of the vehicle is built offline. Moreover, this offline step is possible by approximating that Σ is a diagonal matrix (x-position and y-position of the points are independent). This is a rapid and rough method to approximate the kernel parameter Σ.

The complexity of this method lies in $O(n^2)$. In [7], we have proposed a stochastic approach in order to reduce the cost of running this algorithm.

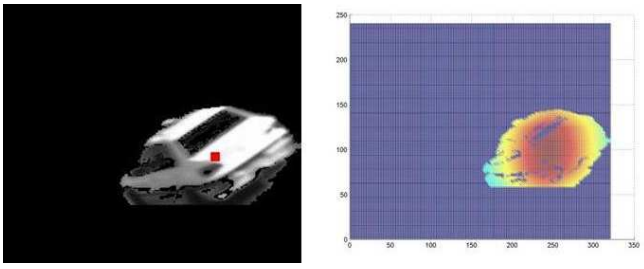


Fig. 3 Illustration of the method for searching the "Shape" point cluster center: Each pixel of the set of "Shape" points displays an assimilation probability at the cluster center. The right-hand side figure is an image of the probability of belonging to the cluster center; as the pixel color becomes redder, the associated weight rises.

The position part ($\mathbf{P}_t^n$) of the filter is initialized as a function of Equation (10). Figure 4 displays an example of an initialized particle set. To improve this initialization, the filter is iterated on the first image, by noising just the position and heading as well as by applying the observation model described in the next section.

The shape model (marginalized color histogram $\mathbf{q}_t^f$) can then be built from the pixels belonging to the estimated vehicle.

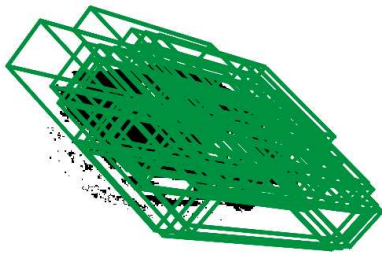


Fig. 4 Example of particle distribution during the initialization phase (shown in green).

4 Observation Function

The proposed observation function utilizes the measurements provided by the two available sensors (i.e. color camera and 1D scanning laser rangefinder). A likelihood function must be defined for each sensor.

4.1 Vision Likelihood Function

The likelihood function proposed herein relies upon a simplified three-dimensional geometric model of the vehicle, as depicted in Figure 5. This model is composed of two nested parallelepipeds. In a general case, the model may be more complex and contain $P_{\mathcal{M}}$ parallelepipeds. Let $\mathcal{M}^{(R_0)} = \{M_i^{(R_0)}\}_{i=1,..,N_{\mathcal{M}}}$ represent the model's set of cube vertices ($N_{\mathcal{M}} = 8 \times P_{\mathcal{M}}$), expressed within a coordinate system associated with model R_0. This coordinate system is selected such that the 3 axes all lie in the same direction as that of the coordinate system associated with scene R_w. For a given particle $\mathbf{X}_t^n$, the likelihood (weighting) calculation is determined as the product of the shape/background likelihood ratios located inside the vehicle model projection in the image.

This computation performed for each particle consumes valuable processing time, and we are proposing herein a fast likelihood calculation method based on an approximation of the 3D model projection in the image

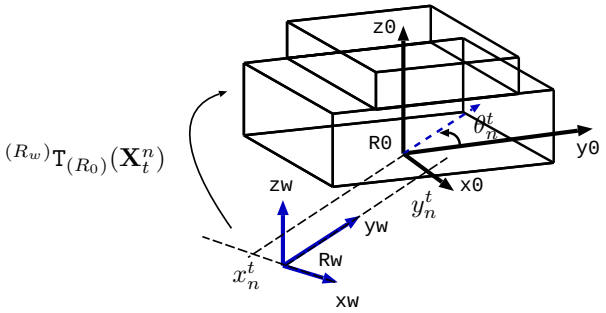


Fig. 5 Example of the simple three-dimensional geometric model used for a vehicle: it is composed of two cubes. The coordinate system associated with the cube and the other system associated with the scene are related according to pure translation. The plane (Oxy) of the world coordinate system and component axes are merged with the GPS coordinate system (Lambert II).

through its convex hull. Each model point is projected onto the image via the following equation:

$$\tilde{\mathbf{m}}_i \propto \mathtt{C}_c.^{(R_w)}\mathtt{T}_{(R_0)}(\mathbf{X}_t^n).\tilde{\mathbf{M}}_i^{(R_0)} \tag{12}$$

with $\tilde{\mathbf{M}}$ homogeneous coordinates associated with point M; $\mathtt{C}_c$ is the camera projection matrix, and $^{(R_w)}\mathtt{T}_{(R_0)}(\mathbf{X}_t^n)$ the homogeneous transformation matrix between the world coordinate system and the system associated with the 3D model (cf. figure 5). This matrix, which depends on $\mathbf{X}_t^n$, may be simply written as:

$$^{(R_w)}\mathtt{T}_{(R_0)}(\mathbf{X}_t^n) = \begin{pmatrix} \cos(\theta_t^n) & -\sin(\theta_t^n) & 0 & x_t^n \\ \sin(\theta_t^n) & \cos(\theta_t^n) & 0 & y_t^n \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

The set $\mathcal{M}^{(R_i)} = \{\mathbf{m}_i\}_{i=1,..N_{\mathcal{M}}}$ is thus built based on the projection of 3D model points within the image.

Let $\mathcal{E}(\mathcal{M}^{(R_0)}; \mathbf{X}_t^n) = \{\mathbf{e}_i\}_{i=1,..,N}$ ($\mathbf{e}_i = (x_i^e, y_i^e)$ as coordinates of $\mathbf{e}_i$ in the image plane) be the list of convex hull points [4]. We will now define $\mathcal{E}_t^n \doteq \mathcal{E}(\mathcal{M}^{(R_0)}; \mathbf{X}_t^n)$ in order to streamline notations. The likelihood calculation may be performed efficiently by use of a line-by-line integral image derived from the log-likelihood ratios calculated in Equation (5) page 5. Since the concept of integral image is often used in computer vision to increase performances, it can not be used directly here because the shapes are not combination of rectangles. So, we propose to extend the concept of integral image to line-by-line integral image. The resulting image is build using integration along the current raw of the image (each raw is independent) and the likelihood calculation is:

$$l_{I,t}((x,y)^T) = \sum_{i=1}^{x} l_{m,t}((i,y)^T) \tag{14}$$

Points $\mathbf{e}_i$ are categorized by pairs featuring the same y-coordinate values, such that:

$$\begin{aligned} \mathcal{E}_t^n = \{ & (x_1^e, y^e), (x_2^e, y^e), \\ & (x_3^e, y^e+1), (x_4^e, y^e+1), ... \\ & (x_{N-1}^e, y^e+N/2), (x_N^e, y^e+N/2)\} \end{aligned} \tag{15}$$

Convex hull coding within the set $\mathcal{E}_t^n$ necessitates a shape discretization along the image lines. Moreover, special attention needs to be paid to coding the upper and lower extremities. On the other hand, it is not at all necessary to sort points positioned on the same line. A compliance measurement relative to a convex envelope is calculated in the integral image by application of the following relation:

[4] Calculation of the convex hull will not be developed in this article; the calculation procedure is conducted using a classical algorithm with a complexity expressed in $O(N.\log N)$.

$$a(\mathcal{E}_t^n) = \sum_{j=1}^{N/2}[2.(l_{i,t}(\mathbf{e}_{2j}) - l_{i,t}(\mathbf{e}_{2j-1})) - (x_{2j}^e - x_{2j-1}^e)] \quad (16)$$

Figure 6 describes the principle behind the likelihood calculation method using the integral image. A line-by-line scanning is performed as part of this method.

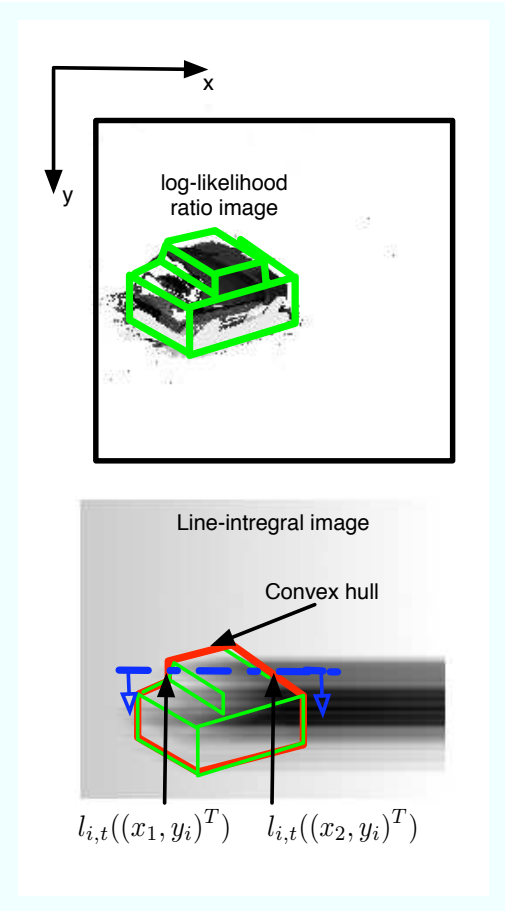


Fig. 6 Illustration of the vision likelihood computation. The 3D model of the vehicle (shown in green) is re-projected onto the image generated from the background-shape extraction. This projection is approximated by its convex hull (shown in red on the lower image). The likelihood calculation proceeds in a line-by-line integral image of the log-likelihood ratio.

The vision weight associated with each particle is directly correlated with $a(\mathcal{E}_t^n, \mathtt{I}_I)$ by means of the following expression:

$$\pi_{v,t}^n \propto p(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{X}_t^n) \doteq \max(0, a(\mathcal{E}_t^n)) \quad (17)$$

4.2 Telemetric Likelihood Function

The telemetric sensor provides, on a horizontal plane, the distance from the first obstacle with a resolution of 1 degree. The intersection of the laser beam with a vehicle yields a set of points; for each particle, a model is now available for calculating the laser echoes generated from the intersection of the beam with the simplified 3D model of the projected vehicle in the world coordinate system. The likelihood associated with the telemetric observation can then be calculated from a modified Hausdorff distance (denoted d_h) between the actual and simulated echoes (cf figure 7) using the following expression:

$$\pi_{l,t}^n \propto p(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{X}_t^n) \doteq \exp(-\lambda_t d_h) \quad (18)$$

where λ_t is an adjustment parameter (typical value is 20).

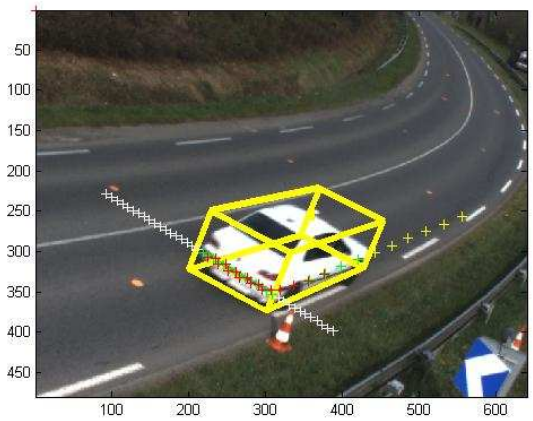


Fig. 7 Illustration of the distance computation between simulated echoes on the 3D model of the cube (shown in green) and the actual echoes (red).

5 Multi-Source Sampling

When the observation function is composed of terms stemming from multiple sources, it becomes necessary to lay out a strategy for combining these terms. A classical approach consists of assembling a weighting function composed of a combination of observations generated from each source. The choice of merge operator often takes place empirically. We propose an alternative to this approach, which calls for performing the combination step as part of the particle filter re-sampling step.

Upon observation, the filter may be represented by a set of N particles with an associated weight vector: $\{\mathbf{X}_t^{(i)}, \boldsymbol{\pi}_t^i\}_{i=1,\ldots,N}$. The weight vector $\boldsymbol{\pi}_t^i$, of size M in correspondence with the number of sources, is composed of the individual particle weight estimated by each source. To facilitate comprehension, the notations contained in the remainder of this section will omit the time index t.

5.1 Principle

Multi-source sampling entails generating a new particle set, by implementing a three-step approach:

1. M particles are sorted (one for each source) according to an importance sampling strategy associated with each source (*importance sampling*). The output of this step consists of a set of M candidate particles along with their respective weight vector $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^{(i)}\}_{i=1,\ldots,M}$:
2. A confidence vector, of dimension M, is built using likelihood ratios estimated for each candidate particle (this calculation will be detailed in the following discussion).
3. The chosen particle is derived from a selection performed among candidate particles by applying an importance sampling strategy on the confidence vector.

These three steps are then repeated a total of N times in order to obtain the complete set.

5.2 Confidence Vector

In the following section, we will describe in detail the second step of this multi-source sampling procedure, which is aimed at building a confidence vector associated with the set of candidate particles. The underlying principle consists of calculating the product of likelihood ratios between weights of the same source, for each pair of candidate particles.

As an example, in the three-sensor case, three candidate particles are drawn; for each particle, a likelihood ratio product is calculated, which for the first candidate particle yields:

$$r_1 \doteq \frac{\pi_1^1}{\pi_1^2} \cdot \frac{\pi_1^1}{\pi_1^3} \tag{19}$$

where π_j^i represents the jth component of vector $\boldsymbol{\pi}^i$. In the case of a blind source (i.e. the values returned by the observation function associated with this source are constant), those likelihood ratios in which the source is present equal one and do not exert any influence on the calculation of terms r_i. In a general case, it is preferable to introduce log ratios, which makes it possible to compute a vector $\mathbf{l}_r$, i.e. the log of $\mathbf{r}$, featuring coefficients r_i, as indicated in the following expression:

$$\mathbf{l}_r \doteq M \begin{pmatrix} \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_1} - \frac{1}{M}\sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i} \right) \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_2} - \frac{1}{M}\sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i} \right) \\ \dots \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_M} - \frac{1}{M}\sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i} \right) \end{pmatrix} \tag{20}$$

where $\mathbf{l}_{\boldsymbol{\pi}_i}$ is the log of vector $\boldsymbol{\pi}_i$ and $\mathbf{1}_{(1\times M)}$ is a matrix composed of a single line and M columns all containing one.

By setting $\mathbf{C}_\pi \doteq \frac{1}{M}\sum_{i=1}^{M} \mathbf{l}_{\boldsymbol{\pi}_i}$, $\mathbf{l}_r$, the following may be written:

$$\mathbf{l}_r = M \begin{pmatrix} \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_1} - \mathbf{C}_\pi \right) \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_2} - \mathbf{C}_\pi \right) \\ \dots \\ \mathbf{1}_{(1\times M)} \left(\mathbf{l}_{\boldsymbol{\pi}_M} - \mathbf{C}_\pi \right) \end{pmatrix} \tag{21}$$

Confidence vector $\mathbf{c}$ is obtained by normalizing $\mathbf{r}$ through a coefficient C_c that results in the sum of its elements equaling one.

$$\mathbf{c} \doteq C_c.\exp\left(\mathbf{l}_r\right) \tag{22}$$

Figure 8 demonstrates how the multi-source sampling method works for two distinct scenarios, by comparing method behavior with that displayed when sampling from both a weight function composed of the weight product from each source and another weight function composed of the sum of weights from each source. In the first scenario (left-hand column), source 1 is a blind source (i.e. returns a constant measurement) and source 2 is unimodal. In this case, the blind source must not intervene to disturb the sampling, and the new set must fit the source 2 set. It should be noted that in the case of sampling using the sum of weights from two sources, the blind source actually deteriorates the resultant set. On the other hand, the other two types of sampling behave quite well. The second scenario illustrates the situation of two dissonant sources (unimodal, yet centered on a different point). The set of particles generated from these two sources must serve to create two separate modes around the two dissonant hypotheses. It can be stated that in the case of product-type sampling, neither of the modes actually gets conserved. In contrast, the other two types of sampling yield results consistent with the desired distribution.

Algorithm 2 Multi-source sampling

Input : Particle set and associated weight vector $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^i\}_{i=1,\dots,N}$, M sources
for $n = 1$ to N **do**
- Choose M candidate particles on the basis of $\{\mathbf{X}^{(i)}, \boldsymbol{\pi}^{(i)}\}_{i=1,\dots,N}$ and build $\{\mathbf{X}^{*(j)}, \boldsymbol{\pi}^{*(j)}\}_{j=1,\dots,M}$ where $\mathbf{X}^{*(j)}$ is derived from an *importance sampling* drawn on source j weights;
- Calculate vector $\mathbf{l}_r$ based on Equation 21, and then calculate confidence vector $\mathbf{c} \doteq C_c.\exp\left(\mathbf{l}_r\right)$
- Select the designated particle $\mathbf{X}^{e(n)}$ from among the candidate particles by proceeding with an *importance sampling* drawing.

end for
Output : Particle set $\{\mathbf{X}^{e(i)}\}_{i=1,\dots,N}$ composed of the selected particles.

6 Experimental Validation

This section will present the experimental campaigns carried out in order to validate the trajectory tracking method.

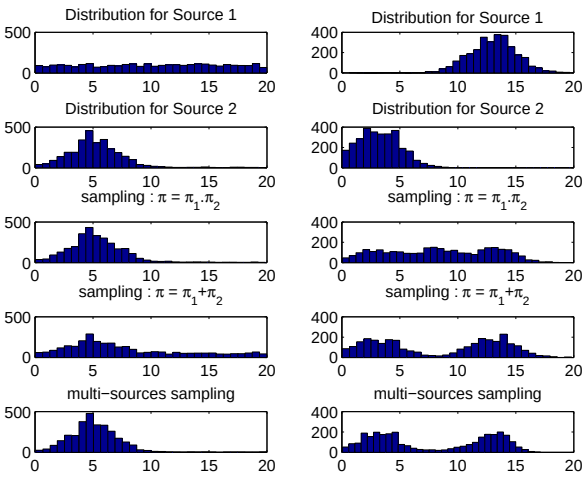


Fig. 8 Illustration of multi-source sampling method operations for 2 different scenarios (one in each column). In the left column, from top to bottom: the first two curves represent the source response (observation). The third shows the result of a weight-based importance sampling, calculated as the product of weights from the two sources. In the fourth curve, this weight-based importance sampling result is generated by summing weights from the two sources. The last curve then displays the result of our proposed multi-source sampling method.

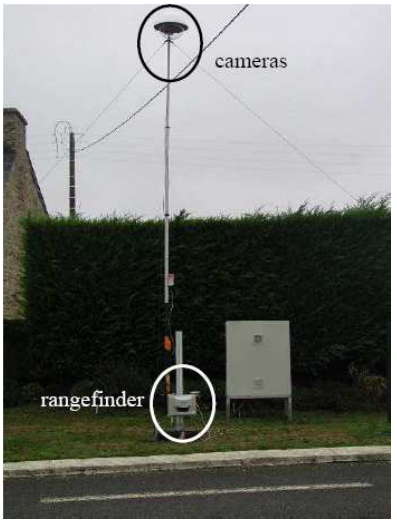


Fig. 9 The video system is composed of three color cameras, capable of being easily deployed onsite.

Speed km/hr	Vision **ave**/std	Rangefinder **ave**/std	Sensor merge **ave**/std
40	**0.25**/0.18	**0.65**/0.54	**0.17**/0.10
60	**0.19**/0.16	**0.72**/0.67	**0.09**/0.06
80	**0.18**/0.15	**0.33**/0.22	**0.14**/0.10

Table 1 Precision for the right camera (error and standard deviation of error between the estimation and actual field value output by a kinematic GPS) of the proposed method, for several travel speeds. It may be remarked that merging the two sensors serves to considerably improve estimation precision. Actual field values have been furnished by a kinematic GPS type of sensor.

The video system developed as part of the SARI project has made it possible to continuously record three color 640 x 480 video streams (cf. Fig. 9), at a frequency of 30 fps, over a several-day period along with data recorded by a scanning telemetry sensor. The calibration between sensors enables expressing all measurements within a common absolute coordinate system tied to the GPS sensor coordinate system, which provides the actual situation in the field.

In order to assess the level of measurement accuracy, a Peugeot 406 type vehicle, equipped with a kinematic GPS accurate to within a centimeter, has been used.

This vehicle traveled through the test section 20 times along various trajectories at speeds ranging from 40 to 80 km/hr. The results listed here are aimed at examining system accuracy as a function of travel speed. Computed trajectories were then compared with trajectories estimated by a vision-only approach, a rangefinder-only approach and an approach merging the two sources. The error was quantified as the average distance between each estimated vehicle position and the straight line passing through the two closest GPS points. For each test, at least five vehicle passes were carried out, which enabled deriving a very rough statistic on the recorded measurements. For the tests actually conducted, the vehicle has been tracked in a curve over a distance of approximately 100 m. The vehicle model used for these purposes comprises a single cube. Finally, according to the empirical results of the Table 2, for every tracking, we decide to use 150 particules.

num. of particles	50	100	150	200	300
average	0.61	0.22	0.13	0.14	0.14
std.dev.	0.61	0.24	0.14	0.14	0.14

Table 2 Precision for the right camera (error and standard deviation of error between the estimated trajectory and the GPS one) of the proposed method, for several dimensions of the particle set.

Figure 10 shows a tracking example from the test campaign carried out as part of this research. The model corresponding to vehicle localization has been re-projected onto the current image. For the left-hand column, the method makes exclusive use of data stemming from the vision sensor. The middle column reflects results based solely on telemetric data. The green crosses correspond with the simulated laser firings re-projected onto the image, while red crosses indicate the actual re-projected laser data. Lastly, for the right-hand column, the method employs both sensors. It is observed that at the beginning of the curve, the vision estimation tends to be rather poor; this is due to the presence of another vehicle crossing the tracked vehicle, thus producing noise in the background/shape extraction. As the vehicle moves away from the sensor, the telemetric approach reveals its flaws, caused by a decrease in the number of laser echoes returned by the vehicle. The approach merging these two sensors allows taking advantage of the accurate information provided by the laser sensor upon entering the curve and then that provided by the vision sensor when exiting the curve.

Table 1 presents the average error and related standard deviation vs. vehicle speed. It can be noticed that the level of accuracy varies only slightly with respect to the tested speed range. Moreover, the two-sensor merge serves to significantly improve this accuracy, which lies on the order of 15 cm with a standard deviation of approximately 10 cm.

One of the most important parameters of Monte-Carlo methods concerns the sampling size. Table 2 shows the precision of the tracking according to the number of particles used into the stochastic filter. In a first step, the precision decreases when the number of particles increases from 50 to 150. Then, in a second step, the precision remains constant when the number of particles is up to 150.

Note that these results come from the right camera, but are extensive to the others cameras. The calibration phase is same for all the system and the reference frame is absolute. Figure 11 shows the three sensors merging with the ground truth.

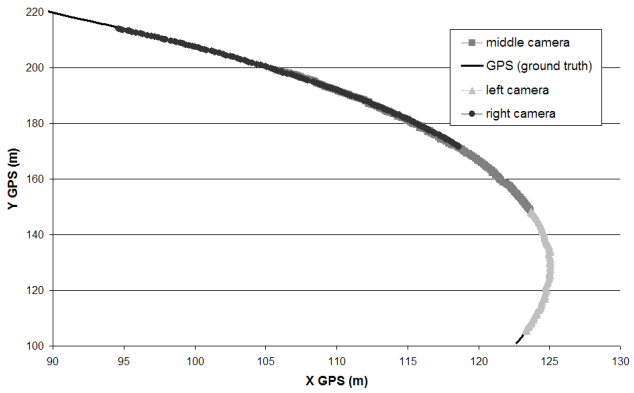


Fig. 11 representation of the three trajectories calculated by each camera and the ground truth associated.

7 Conclusion - Prospects

This article has set forth a method for estimating vehicle trajectories by use of a sensor composed of a color camera and a 1D scanning laser rangefinder. Based on a time monitoring approach formalized by a particle filter, the algorithm has output, at every point in time, an estimation of vehicle state (position, direction, speed, driving angle). We have proposed an original likelihood function

that can be evaluated efficiently through use of a line-by-line integral image. We also introduced an alternative to importance sampling, as classically employed in particle filters. This method has made it possible to implicitly merge observations stemming from different sources. The resulting behavior proves to be better than the classical techniques that rely upon combining weights using algebraic operators.

The experiments conducted have enabled quantifying the precision associated with the proposed approach, thanks to generating actual field values (using a kinematic GPS accurate to within 1 cm). The accuracy obtained lies on the order of 20 cm. These tests have also served to demonstrate the contribution offered by merging vision with telemetry.

Upcoming research will primarily be focused on improving the background subtraction method, which has the potential to introduce bias due to the presence of a vehicle shadow projected onto the pavement [20].

The method discussed in this article is currently operating within the scope of an ANR-PREDIT project. For the purpose of covering the entire curve, the system is composed of three color cameras with very little overlap plus a scanning laser rangefinder, which has successfully analyzed the observations recorded under actual traffic conditions over several-day periods.

Acknowledgements This research work was conducted as part of the PREDIT project entitled "SARI" (French acronym for Automated Road Monitoring to Inform drivers and facility managers). This project is intended to help lead to significant reductions in the number of accidents due to loss of contact with the roadway or loss of vehicle control, by means of better informing drivers of the upcoming difficulties facing them on the road (http://www.sari.prd.fr).

References

1. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing **50**(2), 174–188 (2002). URL citeseer.nj.nec.com/arulampalam02tutorial.html
2. Avidan, S.: Support vector tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2001). Hawaii (2001)
3. Chateau, T., Gay-Belille, V., Chausse, F., Lapresté, J.: Real-time tracking with classifiers. In: WDV 2006 - WDV Workshop on Dynamical Vision at ECCV2006, Grazz (2006)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, Second Edition. Wiley-Interscience (2001)
5. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence (2007)
6. Gillespie, T.: Fundamentals of vehicle dynamics. Society of Automotive Engineers (SAE) (1992)
7. Goyat, Y., Chateau, T., Malaterre, L., Trassoudaine, L.: Vehicle trajectories evaluation by static video sensors. In: ITSC06 2006 - 9th International IEEE Conference on Intelligent Transportation Systems (2006)
8. Haag, M., Nagel, H.: Combination of edge element and optical flow estimate for 3d-model-based vehicle tracking in traffic image sequences. International Journal of Computer Vision **35**(9), 295–319 (1999)
9. Isard, M., MacCormick, J.: Bramble: A bayesian multiple-blob tracker. In: Int. Conf. Computer Vision, vol. 2, pp. 34–41 (2001)
10. Ivanov, Y., Bobick, A., Liu, J.: Fast Lighting Independant Background Subtraction. MIT Media Laboratory (1997)
11. Kamijo, S., Ikeuchi, K., Sakauchi., M.: Vehicle tracking in low-angle and front view images based on spatio-temporal markov random fields. In: 8th World Congress on Intelligent Transportation Systems (ITS) (2001)
12. Kanhere, N.K., Pundlik, S.J., Birchfield, S.T.: Vehicle segmentation and tracking from a low-angle off-axis cam-

era. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). San Diego, USA (2005)

13. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets. IEEE Transactions on Pattern Analysis and Machine Intelligence **(in press)** (2005)
14. Kim, Z.W., Malik, J.: Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In: International Conference on Computer Vision (ICCV), pp. 521–528 (2003)
15. Koller, D., Dandilis, K., Nagel, H.H.: Model based object tracking in monocular image sequences of road traffic scenes. International Journal of Computer Vision **10**(3), 257—281 (1993)
16. M. Isard, A. Blake: Condensation – conditional density propagation for visual tracking. IJCV : International Journal of Computer Vision **29**(1), 5–28 (1998)
17. Magee, D.R.: Tracking multiple vehicles using foreground, background and motion models. Image and Vision Computing **22**(2), 143—155 (2004)
18. Mikik, I., Cosman, P., Kogut, G., Trivedi, M.: Moving shadow and object detection in traffic scenes. In: International Conference on Pattern Recognition. Barcelona, Spain (2000)
19. Pece, A., Worrall, A.: Tracking with the em contour algorithm. In: ECCV European Conference on Computer Vision, vol. 1, pp. 3–17. Copenhagen (2002)
20. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. Comput. Vis. Image Underst. **95**(2), 238–259 (2004). DOI http://dx.doi.org/10.1016/j.cviu.2004.03.008
21. Schneiderman, H., Kanade, T.: A statistical model for 3d object detection applied to faces and cars. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2000)
22. Smith, K., Gatica-Perez, ., , Odobez, J.: Using particles to track varying numbers of interacting people. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
23. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (2000)
24. Tan, T.N., Baker, K.D.: Efficient image gradient based vehicle localization. IEEE Transactions on Image Processing **9**(8), 1343—1356 (2000)
25. Yu, Q., Medioni, G., Cohen, I.: Multiple target tracking using spatio-temporal markov chain monte carlo data association. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
26. Zhao, T., Nevatia, R.: Car detection in low resolution aerial image. In: ICCV, pp. 710—717 (2001)
27. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(11), Nov. 2005 pp 1778 - 1792

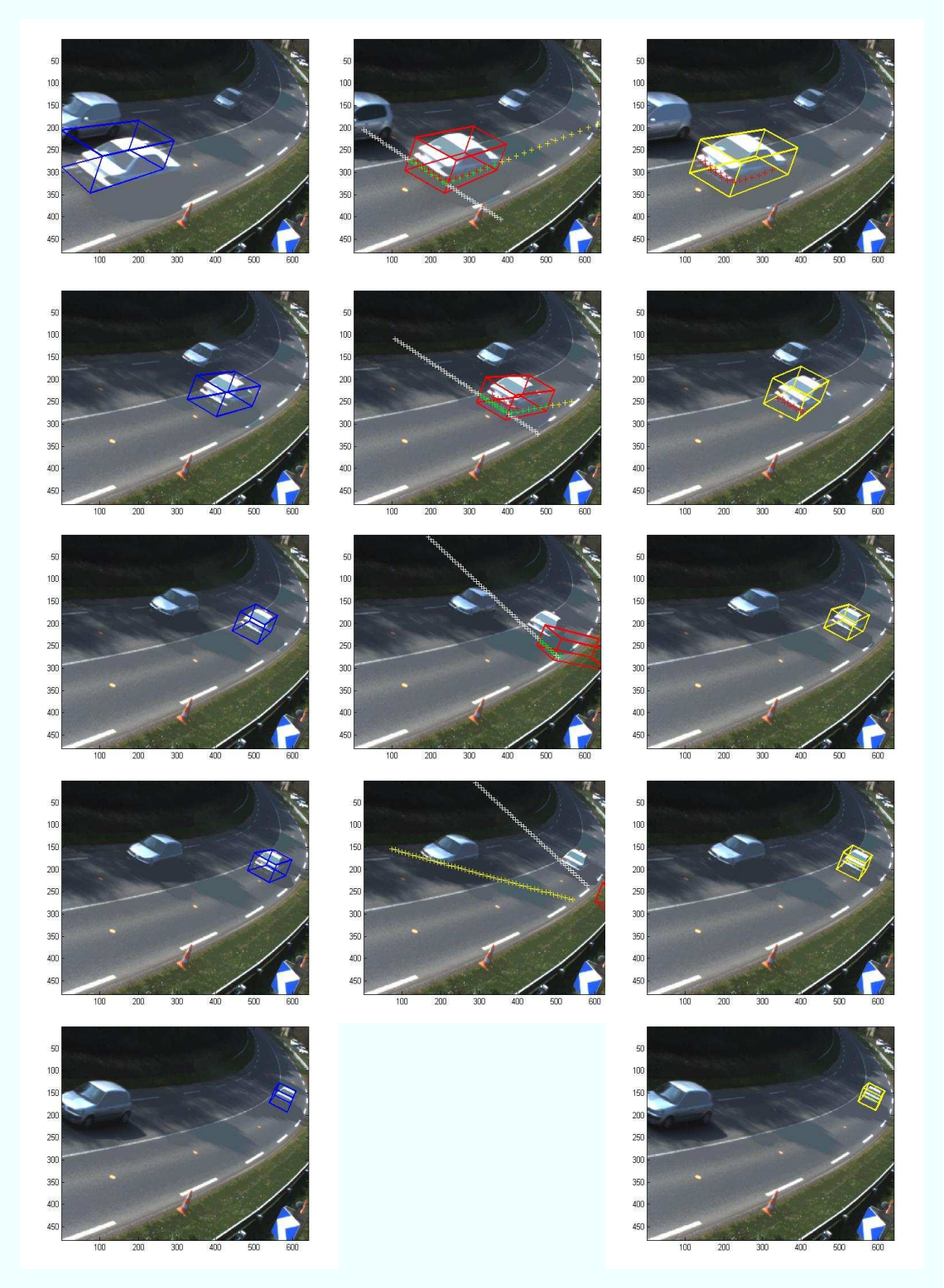


Fig. 10 Tracking example drawn from the testing campaign performed as part of this research (for the right camera). The model corresponding to vehicle localization is re-projected onto the current image. For the left-hand column, the method is confined solely to data stemming from the vision sensor. For the middle column, the method uses only the telemetric data, with the green crosses corresponding to simulated laser firings re-projected onto the image and red crosses indicating actual re-projected laser data. The right-hand column depicts method application by combining the two sensors.

ANNEXE

B

NOTATIONS ET CONVENTIONS

Tout au long de ce manuscrit, les vecteurs sont représentés par des lettres en gras alors que les variables monodimensionnelles sont en fonte normale. Les acronymes suivants sont précédés de leur équivalent français lorsqu'ils sont issus de la langue anglaise.

B.1 Acronymes

ANR	Agence Nationale de la Recherche
CNRS	Centre National de la Recherche Scientifique
GRAVIR	Groupe Automatique : Vision et Robotique
LASMEA	Laboratoire des Sciences de Matériaux pour l'Electronique et d'Automatique
TIMS	Technologies de l'Information, de la Mobilité et de la Sûreté
UBP	Université Blaise Pascal, Clermont II
COMSEE	*Computer that see* Equipe de recherche du groupe *GRAVIR* du *LASMEA*
CEMAGREF	Centre National du Machinisme Agricole, du Génie Rural, des Eaux et Forêts
IV	Véhicules Intelligents, Intelligent Vehicles.
ITS	Systèmes de Transports Intelligents, Intelligent Transportation Systems.
LIDAR	Télémètre par LASER, Light Detection and Ranging.
FP	Filtre Particulaire, Particle Filter.
SIS	Échantillonage pondéré séquentiel, Sequential Importance Sampling.
SIR	Échantillonage pondéré séquentiel avec rééchantillonnage, Sequential Importance Resampling.
FP SIR	Filtre Particulaire basé sur un Échantillonage *SIR*, Sequential Importance Sampling Particle Filter.
FP Partition	Filtre Particulaire Partitionné, Partitionned Particle Filter.
MCMC	Méthode de Monte-Carlo par Chaîne de Markov, Markov Chain Monte Carlo.
FP MCMC	Filtre Particulaire échantillonné par Méthode de Monte-Carlo par Chaîne de Markov, Markov Chain Monte Carlo Particle Filter.
FP MCMC$_1$	Filtre Particulaire *MCMC*, dans lequel chaque particule n de la chaîne est proposée en ne perturbant qu'une des composantes de la particule $n-1$.
FP MCMC$_D$	Filtre Particulaire *MCMC*, dans lequel chaque particule n de la chaîne est proposée en perturbant simultanément toutes les D composantes de la particule $n-1$.
FP MCMC$_d$	Filtre Particulaire *MCMC*, dans lequel chaque particule n de la chaîne est proposée en perturbant simultanément $d \leq D$ composantes de la particule $n-1$ (D est la dimension de l'espace d'état).
FP MCMC^P	Filtre Particulaire *MCMC*, dans lequel chaque particule de la chaîne est tirée parmi P propositions générées en parallèle.
FP MCMC$_d^P$	Filtre Particulaire *MCMC*, à P propositions parallèles, chacune perturbant d composantes du vecteur d'état.
FP RJ-MCMC	Filtre Particulaire échantillonné par Méthode de Monte-Carlo par Chaîne de Markov à Mouvements Reversibles, Reversible Jump Markov Chain Monte Carlo Particle Filter.
PPV	Méthode du Plus Proche Voisin.
MSE	Erreur quadratique moyenne, Mean Squared Error.
MLE	Estimateur par maximum de vraisemblance, Maximum Likelihood Estimator.
MAP	Maximum A Posteriori.
KDE	Estimation de densité par la méthode du noyau, Kernel Density Estimation.
RVM	*Revelant Vector Machine* Machine d'apprentissage à vecteurs pertinents.
SVM	*Support Vector Machine* Machine à Vaste Marge.
LS	*Least Square* Moindres carrés.
PLS	*Penalised Least Square* Moindres carrés pénalisés.
GMM	Modèle de mélanges gaussiens, Gaussian Mixture Model.
MOT	Suivi Multi-Objets, Multiple Object Tracking.
MOTS	Suivi Multi-Objets avec modélisation des Ombres,
MOTC^k	Suivi et Classification Multi-Objets en k classes,
MOTC^k*S*	Suivi et Classification Multi-Objets en k classes, modélisation des Ombres.

B.2 Notations

p	Loi de probabilité d'une variable continue.
P	Probabilité d'une variable discrète.
t	Lettre utilisée en indice pour désigner le temps discret.
j	Lettre utilisée en exposant pour désigner un des J objets.
n	Lettre utilisée en exposant pour désigner une des N particules.
$\mathbf{X}$	Vecteur d'état Markovien à estimer.
$\mathbf{Z}$	Vecteur de mesures.
$\mathbf{X}_t$	État joint de la configuration multi-objets à l'instant t.
$\mathbf{X}_t^n$	Particule n décrivant l'état joint multi-objets à l'instant t.
$\mathbf{x}_t^j$	État de l'objet j à l'instant t.
$\mathbf{x}_t^{j,n}$	État de l'objet j décrit par la particule n à l'instant t.
$\mathcal{X}$	Séquence d'états.
$\mathcal{Z}$	Séquence d'observations.
Φ	Matrice de *design*.
$\mathcal{L}$	Ensemble d'apprentissage composé de couples $\{$état, observation$\}$
$\mathcal{N}(\mu, \Sigma)$	Distribution Gaussienne de moyenne μ et de covariance Σ.
$\mathcal{U}(a, b)$	Distribution uniforme sur l'intervalle $[a, b]$.
δ	Impulsion de Dirac.
$\mathtt{A}^T$	Transposée de la matrice $\mathtt{A}$
$\mathtt{A}^{-1}$	Inverse de la matrice $\mathtt{A}$
$p(\mathbf{Z}\|\mathbf{X})$	Loi de vraisemblance de l'observation $\mathbf{Z}$ étant donné l'état $\mathbf{X}$.
$p(\mathbf{X}_t\|\mathbf{Z}_{1:t-1})$	Loi de probabilité *a priori* de l'état $\mathbf{X}_t$ étant données les observations passées $\mathbf{Z}_{1:t-1}$.
$p(\mathbf{X}_t\|\mathbf{Z}_{1:t})$	Loi de probabilité *a posteriori* de l'état $\mathbf{X}_t$ étant données les observations passées et courante $\mathbf{Z}_{1:t}$.
$p(\mathbf{X}_t\|\mathbf{X}_{t-1})$	Modèle d'évolution dynamique.
$\mathcal{A} \cap \mathcal{B}$	Intersection de 2 ensembles $\mathcal{A}$ et $\mathcal{B}$ (ensemble des élements qui sont dans $\mathcal{A}$ et dans $\mathcal{B}$).
$\mathcal{A} \cup \mathcal{B}$	Réunion de 2 ensembles $\mathcal{A}$ et $\mathcal{B}$ (ensemble des élements qui sont dans $\mathcal{A}$ ou dans $\mathcal{B}$).
$\mathcal{A} \subset \mathcal{B}$	Inclusion de l'ensemble $\mathcal{A}$ dans l'ensemble $\mathcal{B}$.
$\bar{\mathcal{A}}$	Complément de $\mathcal{A}$ (ensemble des élements qui ne sont pas dans $\mathcal{A}$).
$\mathcal{A} \setminus a$	Ensemble $\mathcal{A}$ privé de l'élément a.