En mouvement, du modèle à la puce : Pour des systèmes de vision polyvalents et performants.

Antoine Manzanera

UNIVERSITÉ PIERRE ET MARIE CURIE
HABILITATION À DIRIGER DES RECHERCHES

Spécialité

Sciences pour l'Ingénieur

Présentée par

Antoine MANZANERA

En mouvement, du modèle à la puce :
Pour des systèmes de vision polyvalents et performants

soutenue le 7 décembre 2012

devant le jury composé de :

M. Patrick Bouthémy	Rapporteur
M. Michel Couprie	Rapporteur
M. Michel Paindavoine	Rapporteur
M. Matthieu Cord	Examinateur
M. Jean-Michel Jolion	Examinateur
M. Marc Van Droogenbroeck	Examinateur
M. Bertrand Zavidovique	Examinateur

Remerciements

Mes remerciements s'adressent d'abord aux personnes qui m'ont initié à la Recherche, qui m'ont fait confiance et accueilli dans leur équipe : Jean-Michel Jolion à Lyon, dont l'influence sur mon travail a été décisive et durable, et qui a accepté de participer à mon jury bien qu'il navigue aujourd'hui dans des sphères bien plus hautes. Thierry Bernard à Arcueil, puis à Paris, dont l'esprit et les ideés imprègnent fortement les pages qui suivent ; je le remercie pour son enthousiasme, son exigence et sa bienveillance, ainsi que pour sa relecture attentive du mémoire. Françoise Prêteux à Evry, pour sa disponibilité et son soutien pendant ma thèse, qui m'a fait évoluer sur plusieurs points essentiels, tels que la communication scientifique.

Le fait que la Recherche soit indissociable de l'Enseignement me paraît de plus en plus évident au fil des années. Je remercie chaleureusement les personnes qui m'ont accueilli dans leur équipe d'enseignement : Jean Louchet à l'ENSTA, Georges Stamon à Paris 5, Isabelle Bloch, Henri Maître et Florence Tupin à Télécom.

Je remercie les membres de mon jury pour l'honneur et le plaisir qu'ils m'accordent par leur participation. Merci en particulier aux rapporteurs : Patrick Bouthémy, Michel Couprie et Michel Paindavoine pour leur lecture scrupuleuse, leurs commentaires et leurs critiques. Merci à Matthieu Cord pour le guidage dans le déroulement de l'HDR. Merci à Marc Van Droogenbroeck et à Bertrand Zavidovique, aussi pour l'influence qu'ils ont eue sur ces travaux.

Mes remerciements s'adressent aussi à mes étudiants : élèves Ingénieurs ou en Master, stagiaires de recherche, doctorants et post-doctorants, en particulier bien sûr ceux dont le travail remplit ces pages : Julien, Yahya, Renaud, Taha, Paul, Olivier, Christine, Toby, Gloria, Fabio, Matthieu, Dominique, Phuong...

Je remercie les Directeurs de l'UEI, devenu U2IS : Alain Sibille, puis Bruno Monsuez, de m'avoir accueilli et permis de développer mes recherches avec écoute et confiance. Merci enfin à tous les membres de l'U2IS pour leur sympathie et l'ambiance agréable qui domine dans les murs de ce laboratoire.

Table des matières

Chapitre 1

Introduction

1.1 Parcours Personnel

L'élaboration d'un mémoire d'Habilitation est un événement suffisamment rare dans la vie d'un chercheur pour qu'on s'arrête un instant sur les questions qu'il soulève : le sens qu'on cherche à donner à notre travail, nos idéaux de la Recherche, nos motivations profondes. Si les réponses qu'on en retire demeurent en grande partie sur un plan intime, ce n'est pourtant pas la moindre vertu de cet exercice, ni sa moindre difficulté.

Je crois que c'est d'abord le goût de la Modélisation et de la Théorie qui m'a amené à m'intéresser à l'Analyse d'Images, après une formation initiale en Maths discrètes et Informatique théorique, percevant un domaine qui, étant au carrefour de plusieurs disciplines, se prêterait à une grande variété de représentations et modèles. Par la suite, une expérience de prof de Sciences polyvalent à l'étranger a confirmé mon goût marqué pour la multiculturalité et l'éclectisme scientifique.

Avant ma Thèse, mon expérience pratique de l'Informatique était très faible, et je fus confronté à la programmation SIMD cellulaire des rétines, au langage C et aux scripts Tcl avec à peu près la même ingénuité ; je ne saurai jamais si ce fut un frein ou un avantage. Toujours est-il que je développai un goût certain pour les architectures parallèles et les systèmes non standards, qui obligent à revisiter en profondeur les modèles existants et génèrent souvent une conception tout-à-fait nouvelle du problème abordé.

Après ma Thèse, je décidai d'intégrer l'équipe de Thierry Bernard à l'ENSTA, ce qui me permit de continuer à travailler sur les rétines et d'aller plus loin sur un projet initié au cours de ma Thèse. Assez vite néanmoins ma Recherche s'est progressivement éloignée des rétines, d'une part pour se concentrer plus sur l'algorithmique pure, et

d'autre part en diversifiant les architectures cibles. Aujourd'hui je me considère beaucoup plus spécialiste en traitement d'images et algorithmique de vision qu'expert en calcul haute performance. Néanmoins je suis plus que jamais convaincu qu'un système de vision doit être conçu de manière globale, et que la question du «Comment calculer» ne doit pas venir après celle du «Quoi calculer», mais bien en même temps.

1.2 Contexte de la Recherche

Par rapport à ce qu'on en espérait il y a 50 ans, la vision des robots d'aujourd'hui est sans doute décevante car il n'existe toujours pas de système visuel artificiel dont les performances soient comparables à notre vision biologique quant à sa diversité, ses capacités de généralisation et d'adaptation. Et pourtant les progrès réalisés dans ce domaine depuis plusieurs décennies sont considérables, et l'interprétation automatique des images et vidéos a connu de nombreux succès dans des applications diverses et souvent inattendues du point de vue un peu anthropomorphique des débuts.

Le contexte actuel est celui d'un domaine fortement concurrentiel à la fois sur les plans académique et industriel, et qui a sérieusement pénétré le monde de l'Ingénierie. La puissance de calcul disponible sur n'importe quel processeur récent, et la multiplication des bibliothèques et boîtes à outils logicielles libres ou commerciales de traitement d'images permettent même aux amateurs de construire leur application à un niveau fonctionnel. Au delà du prototypage, cette logique modulaire a du sens d'un point de vue «évolutionnaire», la sélection d'une brique algorithmique par un utilisateur plus ou moins expérimenté et l'évaluation de l'application résultante formant le moteur de l'évolution.

Mais cette logique a ses limites, d'abord pour les applications embarquées qui exigent une optimisation des performances aussi globale que possible, mais aussi parce qu'il est indispensable de revisiter les briques existantes et d'en imaginer de nouvelles de façon permanente, en particulier dans un contexte où l'évolution des architectures sur étagère ne se résume plus à une augmentation de la fréquence d'horloge et des tailles mémoire, mais affecte le nombre de cœurs, les hiérarchies mémoire, la topologie, la nature du parallélisme, etc.

1.3 Objectifs et structure du mémoire

Ma recherche vise à améliorer les performances d'un Système de Vision en le considérant dans son ensemble, depuis le modèle de représentation de l'information visuelle, les algorithmes et les structures de don-

nées, jusqu'à l'implantation parallèle sur un système embarqué. Mon objectif est d'améliorer l'autonomie des Systèmes de Vision, aussi bien du point de vue énergétique (efficacité), que fonctionnel (robustesse). Mes contributions ont touché essentiellement au Traitement d'Images et à la Vision précoce, et se répartissent dans les trois thématiques suivantes :

Représentation et Traitement des Images : je m'intéresse aux modèles de représentation de l'information visuelle : géométriques, statistiques, discrets... ainsi qu'aux structures de données et aux algorithmes de traitement associés.
Mots-clefs : Topologie et distances discrètes, Espaces d'échelles, Espaces de caractéristiques, Filtrage, Segmentation...

Analyse du Mouvement : je recherche les algorithmes les plus efficaces pour extraire d'une vidéo les informations de mouvement les plus pertinentes du point de vue de la surveillance ou de la navigation.
Mots-clefs : Détection de mouvement, Flot optique, Poursuite d'objets, détection d'obstacles, caractérisation de mouvements...

Systèmes de Vision Embarqués : je cherche à exploiter de façon optimale une architecture parallèle en adaptant les algorithmes de vision à la puissance de calcul et au flux de données disponibles.
Mots-clefs : Rétines Programmables, Multi-cœurs, Parallélisme vectoriel, GPU,...

La structure du mémoire est fondée sur cette organisation thématique. Il faut toutefois mentionner que les travaux présentés sont souvent à l'intersection de ces trois pôles. La figure 1.1 résume les relations de dépendance, fonctionnelle ou méthodologique, existant dans les travaux présentés. J'ai fait apparaître sur la figure : (1) les mots-clefs ou contributions principales, (2) une sélection des étudiants encadrés, (3) mes projets de recherche co-financés passés et actuels. Dans la suite du rapport, les 3 chapitres suivants forment la synthèse des travaux réalisés selon chacun des axes thématiques, le Chapitre 5 développe mon projet de recherche pour les années à venir.

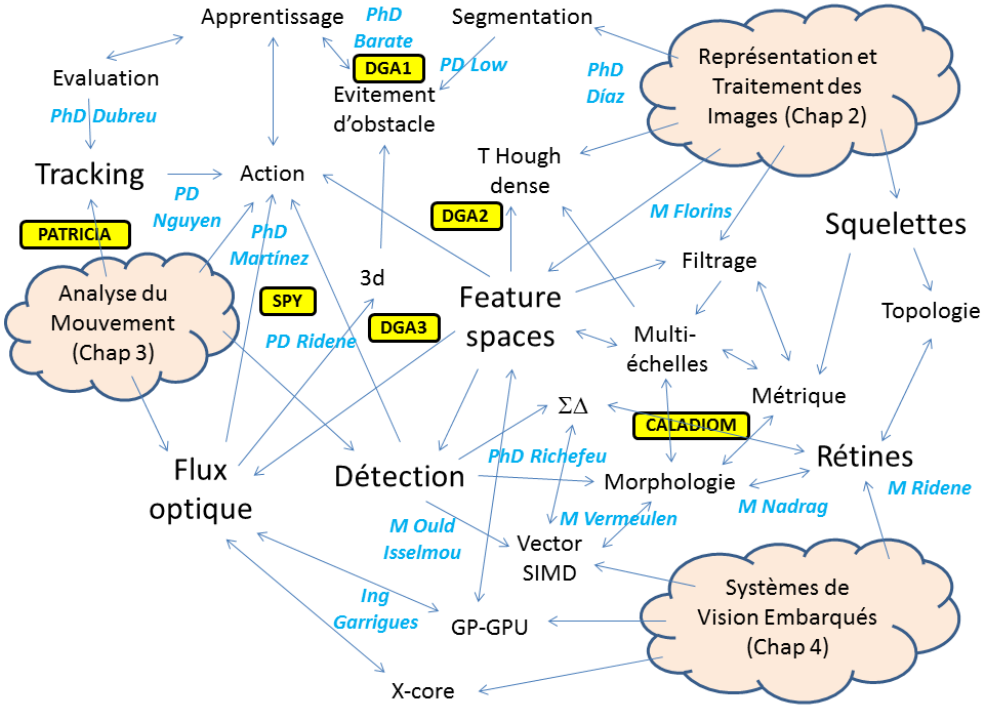


FIGURE 1.1 – Structuration du travail de recherche. Les trois pôles thématiques correspondent aux trois chapitres de bilan du mémoire. En bleu, les étudiants encadrés, PhD pour Doctorant, PD pour Post-Doc, Ing pour Ingénieur de Recherche et M pour Master ou Stage de Recherche (dans ce dernier cas on a fait figurer seulement les étudiants dont le travail a été publié). En jaune, les projets co-financés.

Chapitre 2

Représentations et Traitements

Ce chapitre est consacré à la partie la plus fondamentale de notre travail, la représentation de l'information visuelle et les algorithmes de traitement associés. Mais les modèles et algorithmes proposés ici sont fortement orientés par les préoccupations plus pratiques des chapitres suivants. C'est la conception d'un système de vision qui sous-tend la réflexion : on recherche l'efficacité, par l'économie des calculs et de la mémoire, et la généricité par la mise en facteur de primitives à vocation universelle.

L'information visuelle est associée ici à une fonction à support discret 2d ou nd, et à valeur binaire ou multivaluée. Elle peut concerner un objet physique, auquel cas les propriétés topologiques et géométriques du support sont souvent plus importantes que les valeurs de la fonction, ou bien une texture, où au contraire le support n'a généralement plus d'importance. Mais un objet peut être texturé, et une texture composée d'objets. Dans nos applications de Vidéosurveillance et de Robotique, on doit également caractériser visuellement des choses plus diffuses dans le temps et l'espace : surface, pièce, lieu, contexte ; il est donc nécessaire qu'un système de vision puisse appréhender à la fois des aspects géométriques locaux et statistiques globaux. Il est aussi légitime de chercher à construire un continuum entre ces aspects, continuum dans lequel les mécanismes multiéchelles auront un rôle primordial pour explorer et sélectionner les niveaux de détail adaptés.

La section 2.1 est dédiée aux squelettes, représentations connexes d'un objet binaire fondés sur des critères topologiques ou métriques. La section 2.2 est consacrée aux représentations multiéchelles, qui font le lien dans nos travaux entre les squelettes et les espaces de représentation. Enfin la section 2.3 présente les espaces de caractéristiques dédiées à la représentation visuelle générale et les traitements qui en découlent.

2.1 Représentations connexes

Dans cette section nous nous intéressons aux fonctions binaires, ou ensembles discrets. Un squelette représente une forme binaire quelconque par une variété de dimension inférieure, qui préserve les propriétés de la forme originale en termes de topologie, géométrie et localisation. Dans la continuité de nos travaux de thèse, nous nous sommes intéressés aux squelettes multidimensionnels par amincissement efficaces, puis nous nous sommes concentrés sur les propriétés métriques de ces squelettes définis sur des contraintes topologiques. Enfin, nous avons pris le contre-pied de l'approche précédente en étudiant les squelettes définis de façon purement métrique à partir de transformées en distance, d'abord d'un point de vue algorithmique pour accélérer leur calcul, puis d'un point de vue topologique en établissant les conditions selon lesquels ils préservent la topologie.

2.1.1 Squelettes multidimensionnels

Les squelettes par amincissement sont définis par un algorithme qui consiste à retirer des points d'un ensemble discret sous des conditions diverses liées d'abord à la préservation de la topologie, et ensuite à la forme et au bon centrage du squelette, conditions calculables localement en chaque point, et aussi indépendantes que possible d'un point à l'autre de façon à retirer le maximum de points en parallèle pour accélérer le traitement. Etant donné la multiplicité de contraintes imparfaitement compatibles, beaucoup d'algorithmes différents ont été proposés, depuis les travaux précurseurs de la fin des années 60 [118, 57, 126] jusqu'au début des années 2000. La thèse de Christophe Lohou [73] constitue sans doute un des guides les plus complets de ce monde peu étendu mais extrêmement peuplé.

Conçu au départ sur un objectif de performance (cf Chap. 4), nous avons proposé un algorithme de squelettisation par amincissements entièrement parallèles, c'est à dire retirant des points de la frontière de l'ensemble de façon simultanée dans toutes les directions cardinales de l'espace, à partir de la minimisation de l'expression logique des contraintes liées à : (1) la préservation de la topologie, (2) la conservation des points terminaux, et (3) la symétrie. Outre ses performances, et son adéquation particulière au parallélisme cellulaire, ce squelette dit MB présente la remarquable propriété de pouvoir être exprimé en maille cubique de n'importe quelle dimension : voir Figure 2.1. La propriété de préservation parallèle de la topologie a été prouvée par nos soins [89] sur les critères de Ronse en 2d [114], et de Ma en 3d [77]. Elle a aussi été prouvée par Lohou et Bertrand en utilisant le critère

de P-simplicité de Bertrand [15]. Ces différents critères expriment des conditions suffisantes pour qu'un algorithme de suppression parallèle de points préserve la topologie.

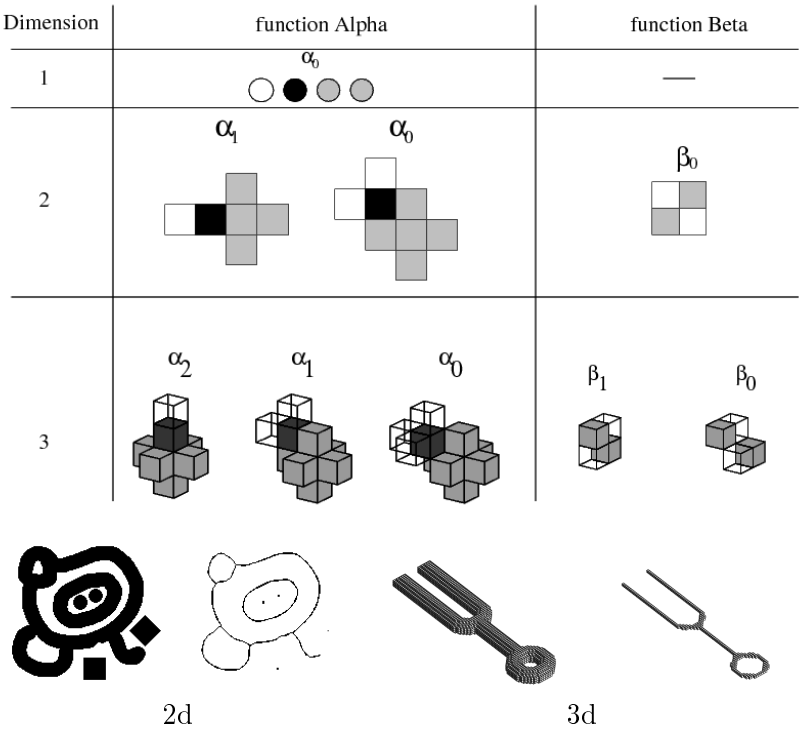


FIGURE 2.1 – Squelette par amincissement MB nd (dimensions 1 à 3). La fonction Alpha correspond aux configurations de suppression, et la fonction Beta aux configurations à préserver. En dessous, exemples de squelettes obtenus en 2d et en 3d.

2.1.2 Métriques des squelettes par amincissement

Contrastant avec la profusion des travaux sur les aspects topologiques, les propriétés métriques des algorithmes d'amincissement n'ont pas fait l'objet de beaucoup d'attention. Ces propriétés sont pourtant essentielles à la représentation, puisqu'en tant que descripteur de formes, le squelette se doit de posséder certaine invariance aux transformations euclidiennes, ce qui implique que le résultat final de l'amincissement doit se rapprocher le plus possible de l'axe médian euclidien, défini par les centres des boules euclidiennes maximales incluses dans la forme.

Or, en dépit d'expressions logiques distinctes, et malgré des différences d'aspect, principalement sur la régularité, du squelette obtenu, la géométrie d'un squelette par amincissement n'est globalement due qu'à un seul facteur : le mode de parallélisme. Ainsi dans la maille 2d carré, les squelettes par amincissement complètement parallèle sont fondés sur la distance de la 4-connexité car le retrait simultané de points dans les 4 directions cardinales s'assimile à une érosion conditionnelle par la boule élémentaire 4-connexe, tandis que les squelettes par amincissement semi-parallèle directionnel sont fondés sur la distance de la 8-connexité car le retrait simultané de points dans une seule direction cardinale s'assimile à une érosion conditionnelle par un élément structurant de deux points dans une direction cardinale, qui une fois composée avec l'érosion dans les 3 autres directions, équivaut à une érosion par la boule élémentaire 8-connexe [88].

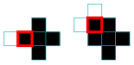
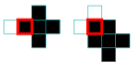
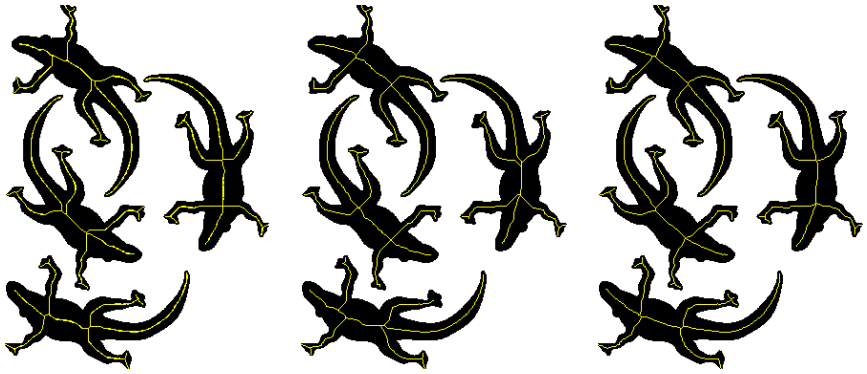
ALGORITHM	Removing condition	Non–removing condition	Parallelism	Topology preservation	Isotropy	1–pixel thickness	Boolean complexity	Support and size	P–simpleness
MBdir1–8			DIR	8	NO	YES	28 r	 8 (8)	YES
MBdir2–8			DIR	8	NO	YES	76 r	 8 (8)	YES
MBdir1–4			DIR	4	NO	YES	28 r	 7 (8)	YES
MBdir2–4			DIR	4	NO	YES	60 r	 7 (8)	YES
MBfp1–8 [Eckhardt et al. 93]			FP	8	YES	NO	18 p	 13	YES
MBfp2–8			FP	8	YES	NO	28 p	 21	YES
MBfp1–4 [Latecki et al. 95]			FP	4	NO	NO	16 p	 23	YES
MBfp2–4			FP	4	NO	NO	26 p	 38	NO

FIGURE 2.2 – La famille d'amincissement MB en 2d avec leurs propriétés topologiques, métriques, et combinatoires.

Dans le même esprit de généricité que pour la dimension, nous avons décliné différentes versions de l'algorithme MB dans les différents modes de parallélisme [87], créant une famille d'algorithmes d'amincissement avec différentes propriétés topologiques et métriques (voir Figure 2.2). Outre la flexibilité, l'avantage principal est de pouvoir appliquer des versions hybrides alternant itérations directionnelles et itérations complètement parallèles de façon à se rapprocher d'un axe médian euclidien en conservant les performances liées au parallélisme massif de l'amin-

cissement (voir Figure 2.3) .



MBfp2-8 MBdir2-8 MBhybride2-8

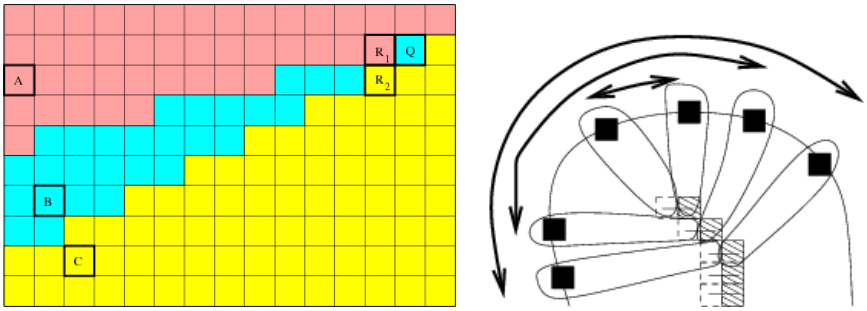
FIGURE 2.3 – La géométrie du squelette complètement parallèle (à droite) est fondée sur la distance de la 4-connexité, celle du squelette semi-parallèle (au milieu) sur la distance de la 8-connexité. Le squelette hybride (à droite) est plus invariant par rotation.

2.1.3 Squelettes multiéchelles par fonction de chocs

Au final les qualités métriques des squelettes par amincissement pure demeurent médiocres car le squelette MB hybride ne fait qu'approximer la distance euclidienne à l'aide de boules octogonales [88] qui gardent un biais pour certaines directions du plan. Un moyen simple de résoudre ce problème consiste à calculer une transformée en distance, fonction qui associe à chaque point de l'objet sa distance au complémentaire, et à contraindre l'algorithme d'amincissement à retirer les points dans l'ordre induit par la transformée en distance. Cette contrainte est préjudiciable au parallélisme mais peut s'implanter efficacement en séquentiel, par exemple avec des files d'attente [137].

Cependant il apparaît que le calcul de la transformée en distance induit une forte redondance avec l'algorithme d'amincissement. Nous avons donc étudié le calcul rapide d'un squelette connexe fondé sur le calcul de la transformée en distance euclidienne. Nous sommes partis du principe du squelette euclidien multiéchelles par fonction de chocs défini par Costa [32]. Le principe de base est simple et élégant : il consiste à définir en chaque point $\mathbf{z}$ du plan une fonction de choc définie comme le maximum de la distance géodésique le long du contour entre 2 points du contours à égale distance (euclidienne) de $\mathbf{z}$. Toutefois, la mise en œuvre discrète initiale, fondée sur le concept de dilatation exacte, était lourde, et de plus il n'existait pas de preuve de préservation de la topologie.

Recherchant une implantation rapide de la transformée euclidienne exacte, nous avons réalisé une première version des squelettes euclidiens fondée sur l'algorithme de Shih et Wu [124]. Cette version, rapide mais incorrecte, a permis une double constatation : (1) l'algorithme de Shih et Wu est faux, et (2) le squelette de Costa ne préserve pas toujours la topologie. L'explication, connue en fait depuis Danielsson [34], repose sur le fait que les zones d'influence d'un point (i.e. fournie par le diagramme de Voronoï) associées à la distance euclidienne peuvent ne pas être connexes dans la maille carrée (voir Fig. 2.4(1)). Le squelette par fonction de chocs en vraie distance euclidienne n'est donc pas toujours connexe, en revanche il l'est forcément lorsque les zones d'influence d'un point sont connexes (voir Fig. 2.4(2)), ce qui est vrai pour toute distance discrète associée à un algorithme de transformée par balayage récursif dans un voisinage 3×3.



(1) (2)

FIGURE 2.4 – (1) Le squelette par fonction de chocs fondée sur la vraie distance euclidienne ne préserve pas toujours la topologie car les zones d'influence (comme ici celle du point B) ne sont pas toujours connexes. (2) Si les zones d'influences sont connexes, la fonction de choc est croissante des extrémités au centre, et donc le squelette multiéchelle préserve la topologie.

Nous avons réalisé une version rapide du squelette multiéchelles connexe fondée sur l'algorithme de Leymarie et Levine [70]. Cette version qui n'est finalement qu'une implantation rapide des squelettes hiérarchiques de Voronoï [102], n'a pas été publiée, mais est détaillée dans un cours sur les applications des transformées en distance, et le code est disponible avec le logiciel de démonstration qui sert aussi de tutoriel sur les squelettes multiéchelles :

Logiciel : `http://www.ensta-paristech.fr/~manzaner/Softwares/multiscale_skeleton.tgz`

Cours : `http://www.ensta-paristech.fr/~manzaner/Cours/IAD/AM_Distance.pdf`

2.2 Représentations multiéchelles

Cette section traite des représentations multiéchelles, qui constituent l'un des principaux fils conducteurs de notre travail, dès nos premiers pas dans la recherche sous la direction de Jean-Michel Jolion [91]. Si nos préoccupations initiales étaient plutôt d'ordre combinatoire et architecturale, nous avons progressivement intégré l'espace d'échelles à la fois comme élément essentiel d'une représentation visuelle, et comme mécanisme «naturel» de passage du local au régional.

Beaucoup de variétés d'espaces d'échelles ont été abordées dans notre travail. Nous mettons ici l'accent sur quelques approches, les plus originales ou les plus importantes. Nous évoquons d'abord la représentation de forme binaire induite par les squelettes multiéchelles, puis nous mettons l'accent sur les dérivées multiéchelles, et une utilisation de ces dérivées dans une catégorie particulière de transformée de Hough pour la reconnaissance de formes paramétrées.

2.2.1 Représentations de formes multiéchelles

Les squelettes multiéchelles évoqués dans la section précédente induisent naturellement une représentation multiéchelles connexe d'une forme binaire, avec la reconstruction de la forme à partir de son squelette à une échelle donnée. Si on note Sk_X la fonction de choc associée à la forme X, le squelette d'échelle σ est simplement le seuil de niveau σ de la fonction de choc : $Sk_X^\sigma = \{\mathbf{z}; Sk_X(\mathbf{z}) \geq \sigma\}$. La reconstruction de X à l'échelle σ est définie par $R_X^\sigma = \bigcup_{\mathbf{z} \in Sk_X^\sigma} B(\mathbf{z}, F_X(\mathbf{z}))$, avec $B(\mathbf{z}, \rho)$ la boule de centre $\mathbf{z}$ et rayon ρ et $F_X(\mathbf{z}) = d(\mathbf{z}, X^c)$ la valeur de la transformée en distance au point $\mathbf{z}$.

On peut construire explicitement cette représentation multiéchelles en calculant la carte de reconstruction : $R_X(\mathbf{z}) = \max_{\mathbf{z}' \in Sk_X^1 / \mathbf{z} \in B(\mathbf{z}', F_X(\mathbf{z}'))} Sk_X(\mathbf{z}')$, de telle sorte que la reconstruction de X à l'échelle σ se déduise par seuillage : $R_X^\sigma = \{\mathbf{z}; R_X(\mathbf{z}) \geq \sigma\}$: voir Figure 2.5.

2.2.2 Dérivées multiéchelles

Les représentations multiéchelles issues de la modélisation ensembliste sont puissantes et variées, mais ne se prêtent pas bien à certains niveaux d'analyse qui sont fondamentaux pour la vision, comme le niveau différentiel (et dans une moindre mesure le niveau fréquentiel). Dans le modèle différentiel, l'image est considérée comme une fonction différentiable, et l'analyse est fondée sur une mesure des dérivées

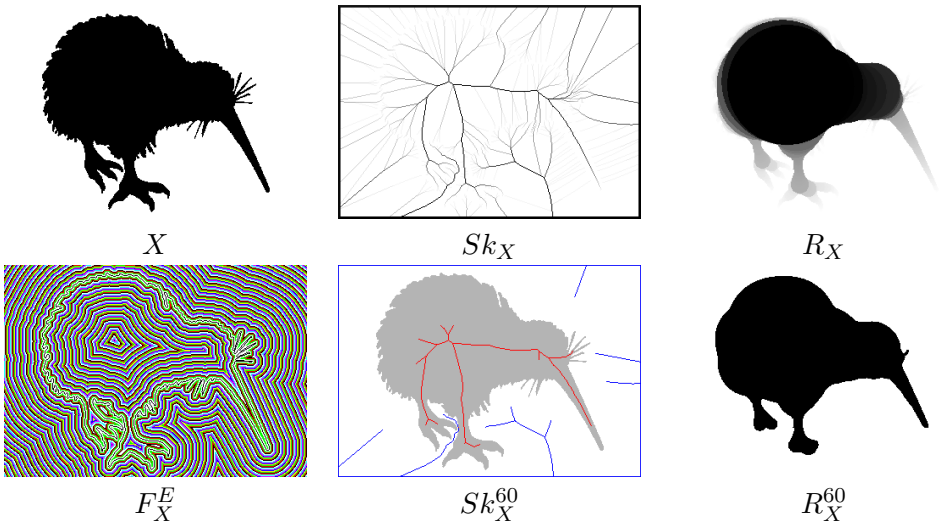


FIGURE 2.5 – Représentation multiéchelle associé au squelette connexe par fonction de choc en distance pseudo euclidienne.

partielles locales. C'est le cas pour beaucoup d'algorithmes de vision précoce tels que détection de contours, segmentation, réhaussement de contrastes, mais aussi filtrage, superrésolution ou désoccultation, car si la mesure des dérivées permet bien sûr d'estimer le contraste et de séparer des régions ou créer des frontières, elle permet aussi de régulariser et de regrouper, par la prédiction que fournit la formule de Taylor.

Le cadre formel des espaces d'échelles linéaires [72] est le mieux adapté dans la mesure où dans une image numérique, une dérivée n'a de sens qu'à une hypothèse de régularité près, qui correspond à l'échelle d'estimation, rendue explicite par la convolution avec la dérivée de gaussienne correspondante [71] : $I^{\sigma}_{x^i y^j} = I \star \frac{\partial^{i+j} G_\sigma}{\partial x^i \partial y^j}$ où G_σ est la fonction gaussienne 2d d'écart-type σ.

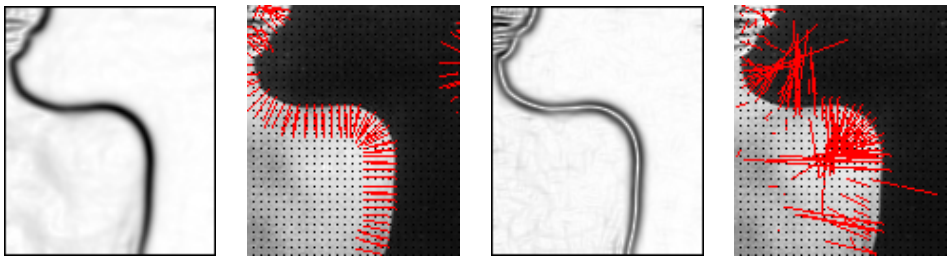
Outre leur intérêt immédiat pour les traitements de bas niveau, l'estimation des dérivées multiéchelles permet aussi une analyse directement liée à la géométrie différentielle : directions principales, courbures, etc, que nous avons appliquée dans des problèmes divers tels que la détection de mines ([47], Master D. Florins) ou l'imagerie vasculaire (Nouveau projet CMCU, Thèse A. Kerkeni). La catégorisation locale par géométrie différentielle est d'ailleurs utilisée pour la représentation d'objets texturés [29]. D'autre part, des arguments neuroscientifiques montrent l'importance pour la vision humaine de l'estimation des dérivées à plusieurs échelles jusqu'à l'ordre 2 [65]. Cela justifie notre utilisation intensive de ces dérivées dans plusieurs travaux récents qui sont présentés dans la suite du chapitre.

2.2.3 Transformées de Hough Denses

La transformée de Hough est l'une des techniques les plus anciennes en Vision par ordinateur [60], et un outil classique pour la détection de formes paramétrées [39, 8]. Classiquement elle est appliquée sur un ensemble de points formant l'espace image, selon l'une des approches duales : (i) la projection many-to-one, qui considère des n-uplets de l'espace image et sélectionne l'unique point correspondant dans l'espace n-dimensionnel de paramètres, et (ii) l'extrusion one-to-many, qui pour chaque point de l'espace image trace la surface de dimension n-1 correspondante dans l'espace des paramètres. Il est remarquable que l'intérêt pour ces techniques ne s'est jamais démenti et que de nombreuses variations sont encore proposées aujourd'hui [69, 43].

Mais généralement ces algorithmes conservent la forme originale au sens où la projection/extrusion (le vote) est appliquée de façon éparse sur des contours ou des points d'intérêt, en utilisant l'approche one-to-many (le plus souvent) ou many-to-one (généralement décimée). Il y a pourtant un intérêt fondamental à appliquer le vote de façon dense (i.e. tous les pixels votent), et ceci peut être réalisé en utilisant les dérivées multiéchelles. Bien que tous les pixels votent, cette méthode est plus rapide, d'une part parce qu'il n'y a pas de traitement préalable pour réduire le support du vote (si ce n'est le calcul des dérivées, mais celui-ci est généralement à la base du calcul des contours ou points d'intérêt), et d'autre part (et surtout) parce que - dans le cas des formes analytiques - on peut alors calculer une projection one-to-one de l'espace image à l'espace paramètres.

L'idée d'utiliser les dérivées locales pour accelérer la transformée de Hough n'est pas nouvelle [103, 122], mais ces techniques ont été employées sur des courbes, nécessitant donc une segmentation préalable. Plus récemment Valenti et Gevers [132] ont proposé un algorithme de localisation de la pupille fondé sur un mécanisme de vote à partir de l'estimation de la courbure, mais il réduisent aussi l'ensemble des pixels votant par segmentation préalable. Nous avons montré dans des travaux récents [86] qu'il est tout-à-fait pertinent d'appliquer la transformée de Hough directement dans l'image en niveaux de gris en utilisant les dérivées multiéchelles. Le vote est pondéré en fonction de l'intensité de la dérivée (norme du gradient à l'ordre 1, norme de Frobenius de la hessienne à l'ordre 2), elle-même normalisée selon l'ordre et l'échelle : voir Figure 2.6. Nous avons ainsi proposé une transformée de Hough dense one-to-one fondée sur les dérivées multiéchelles d'ordre 1 pour les droites, et d'ordre 2 pour les cercles. Nous avons également proposé une transformée de Hough généralisée dense indexée sur des dérivées quantifiées multiples.



(1) (2) (3) (4)

FIGURE 2.6 – Principe de la transformée one-to-one de Hough dense. (1) Poids des votes à l'ordre 1 (module du gradient), (2) Vote pour la normale (direction du gradient) dans la détection des droites. (3) Poids des votes à l'ordre 2 (norme de Frobenius de la matrice hessienne), (4) Vote pour la position du centre dans la détection de cercles (point dans la direction du gradient, dans le sens opposé au signe de la courbure et à une distance égale à l'inverse de la courbure). Pour une meilleure lisibilité, seul un vote sur 16 figure, et si son poids est supérieur à un certain seuil.

2.3 Espaces de Caractéristiques

Nos travaux sur les espaces de caractéristiques trouvent encore une fois leurs origines dans une volonté de rationalisation fonctionnelle d'un système de vision : quelles doivent être ses primitives fondamentales et le cœur de son architecture logicielle ? Notre approche consiste essentiellement à combiner des concepts existant au sein d'un modèle computationnel cohérent, qu'on cherche à rendre le plus universel possible à la fois sur un plan horizontal (le système doit supporter autant de traitements différents que possible) que sur un plan vertical (différents niveaux sémantiques doivent être abordés sans rupture du modèle).

Nous exploitons donc différentes idées issues de travaux sur la modélisation, la segmentation et la reconnaissance d'objets. Ainsi la décomposition par bancs de filtres est utilisée depuis longtemps pour extraire l'information visuelle pertinente en termes de direction, échelle, fréquence [49]. Nous l'avons utilisée dans divers travaux et études tels que la classification de surfaces navigables par modèles d'apparence ([74], Post-Doc Toby Low), l'imagerie biologique ([40], Thèse Gloria Díaz) ou la détection d'objets ([85], Projet ITEA2 SPY). Dans d'autres travaux plus fondamentaux, nous avons étudié l'adaptation automatique au contexte et/ou à la tâche, à partir de mesure ou de sélection de primitives, où les bancs de filtres jouaient un rôle important. C'est le cas de la Thèse de Renaud Barate [9], dont l'objectif était l'apprentissage automatique de fonction visuelle, modélisée à base d'un vocabulaire

de primitives mêlant traitements d'images, fonctions arithmétiques et logiques et mesures intégrales locales [11], et d'une grammaire pour combiner ces traitements, ainsi qu'une méthode d'optimisation fondée sur la programmation génétique [10]. Dans le cadre des travaux de Philippe Guermeur, nous nous sommes intéressés à la description visuelle de scène par analyse statistique d'un espace de descripteur local [55]. On peut aussi citer la Thèse de Christine Dubreu [36], qui bien que plus appliquée, a abordé certains concepts nouveaux, tels que les distances entre distributions sur des espaces de descripteurs locaux pour la poursuite de cibles [38].

La synthèse que nous présentons dans cette section est proche du modèle théorique de variété de Gabriel Peyré [106] : l'information image est projetée dans un espace de caractéristiques de plus grande dimension, et forme une variété. Beaucoup de problèmes inverses en vision précoce peuvent s'exprimer par une régularisation de cette variété suivi d'une rétro-projection de la variété transformée dans l'espace image. Les différents traitements présentés dans cette section (et dans le chapitre suivants) peuvent être considérés comme des instances de ce modèle, mais à l'inverse, notre système peut également être vu comme une extension du modèle de Peyré, dans la mesure où l'on cherche aussi à extraire des représentations de plus haut niveau à partir de la structure géométrique et topologique de la variété.

Notre travail repose aussi sur d'autres techniques telles que la quantification vectorielle des caractéristiques, qui est classique dans des modélisations de textures [117] ou d'objets [31, 44], et la structuration métriques des espaces multidimensionnels, avec les kd-trees [5]. L'espace de caractéristiques que nous avons utilisé est celui des dérivées multiéchelles (local jet), qui présente l'avantage de former un espace où la métrique euclidienne est directement liée à la similarité visuelle [65], et de fournir par combinaison une large variété d'invariants très utiles [121].

Notre système est donc composé des éléments suivants [83, 84], voir le schéma de la figure 2.7 :

- Projection dans l'espace du local jet, espace paramétrable quant à l'ordre, le nombre d'échelles, le repère cartésien ou local, et muni de différentes distances ou pseudo-distances.
- Quantification vectorielle pour réduire la taille de l'ensemble de représentation dans l'espace de caractéristiques et structuration du dictionnaire résultant en arbre de recherche. Ces deux éléments sont facultatifs.
- Rétro-projection d'un ensemble de vecteurs de local jet dans l'espace image.
- Analyse métrique et topologique de l'ensemble des caractéris-

tiques pour une représentation de haut niveau de l'information visuelle.

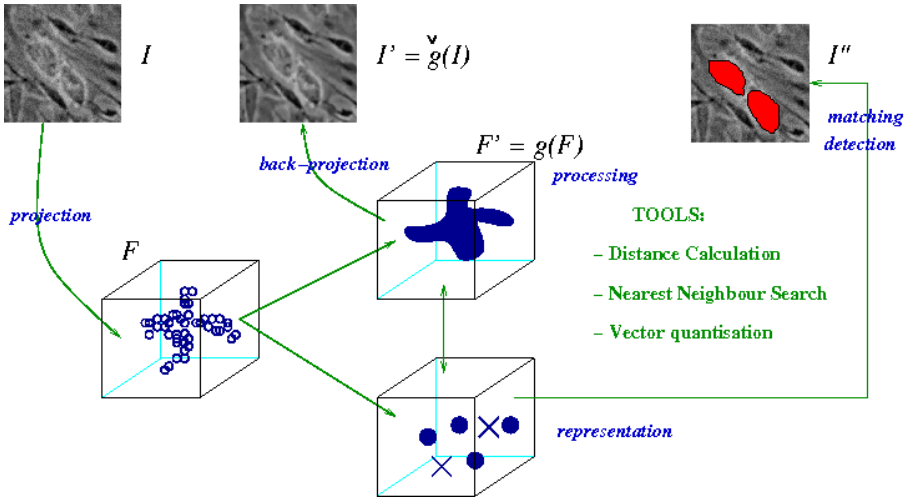


FIGURE 2.7 – Vue générale du modèle de représentation et de traitement fondé sur les espaces de caractéristiques.

Nous avons appliqué ce modèle dans trois traitements de bas niveau fondamentaux : (1) le débruitage par moyennes non locales, (2) l'estimation dense de flux optique, et (3) la détection d'objets mobiles par modélisation de fond statique. Les applications (2) et (3) seront présentées dans le chapitre suivant ; on résume ici la première : l'application d'un filtre linéaire de lissage dans l'espace des caractéristiques correspond à une implantation du filtre de débruitage non local dit NL-means, proposé par Buades *et al* [20] et considéré comme l'état de l'art en débruitage d'images. Il s'agit de remplacer la valeur de chaque pixel par une moyenne pondérée des autres pixels, où les poids ne dépendent pas de la distance au pixel considéré comme dans une convolution classique, mais de la ressemblance visuelle entre les pixels (d'où le caractère non local, des pixels ressemblants pouvant être éloignés dans l'espace image). Dans notre modèle le filtre NL-means s'exprime simplement en calculant les poids à partir de la distance dans l'espace du local jet. Si $\mathbf{u}$ et $\mathbf{v}$ sont deux vecteurs de local jet $\omega(\mathbf{u}, \mathbf{v})$ le poids relatif (symétrique) de $\mathbf{u}$ par rapport à $\mathbf{v}$ vaut $e^{-\frac{d_F(\mathbf{u},\mathbf{v})^2}{h^2}}$ où d_F est une distance dans l'espace du local jet, et h est un paramètre qui dépend du niveau de bruit. Deux variantes du Local-Jet-NL-Means peuvent être appliquées : (1) portée limitée : la moyenne est calculée dans un voisinage de chaque pixel l'espace image, et (2) portée infinie : la moyenne est calculée, pour chaque pixel, dans un voisinage limité du vecteur pro-

jeté dans l'espace des caractéristiques (voir Figure 2.8). En changeant la précision de représentation du local jet (et aussi la complexité) selon l'ordre et le nombre d'échelles, le Local-Jet-NL-Means établit une gradation progressive entre le filtrage tonal, ou bilatéral [129] et le NL-Means original.

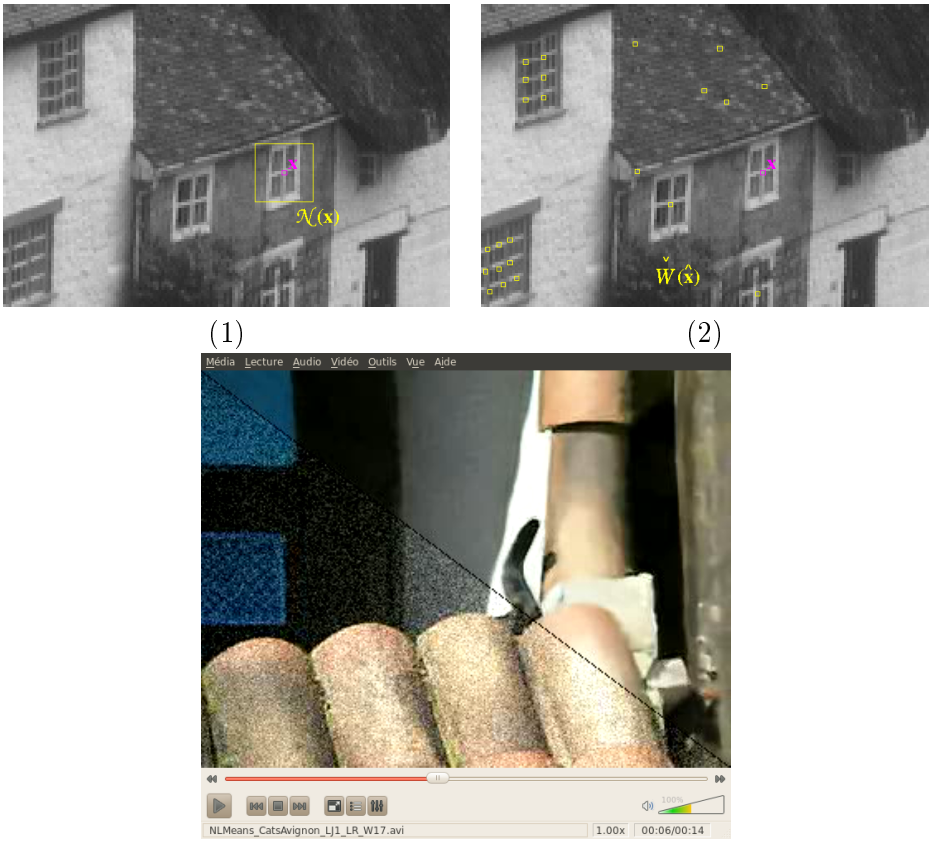
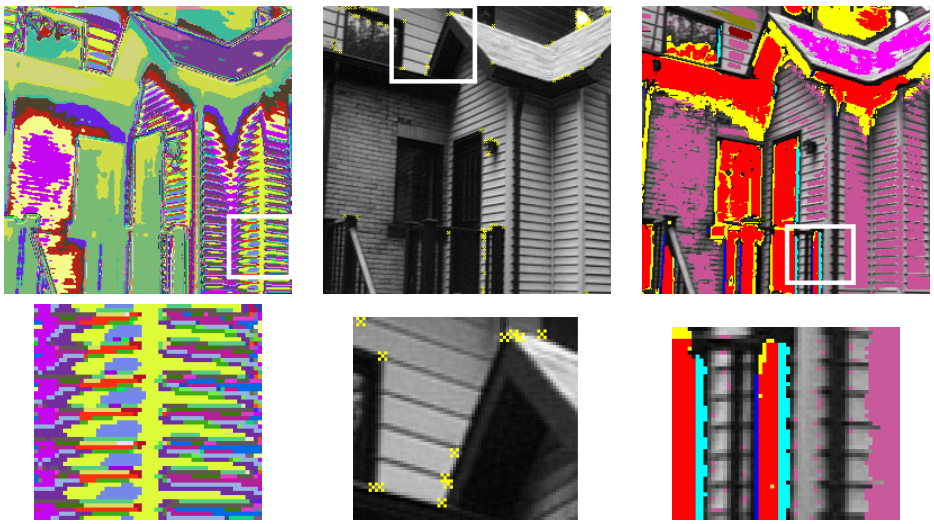


FIGURE 2.8 – Filtrage NL-Means avec calcul des poids en fonction de la distance dans l'espace du local jet. En haut, le support du calcul : (1) portée limitée : voisinage dans l'espace image. (2) portée infinie : rétroprojection dans l'espace image du voisinage dans l'espace du local jet. (3) Application de la version portée limitée en voisinage spatiotemporel, local jet couleur.

Dans la philosophie du modèle, l'espace de caractéristiques ne doit pas uniquement servir comme support de traitement de bas niveau, mais doit aussi fournir des représentations de plus haut niveau de par la structure même de l'ensemble. On peut d'abord remarquer que le dictionnaire lui-même fourni par la quantification du local jet est une base de description semblable aux textons ou aux sacs de mots visuels,

néanmoins dans notre modèle ce type de descripteur est intrinsèquement dense, ce qui facilite les statistiques d'ordre supérieur (cooccurence). De façon plus originale, l'analyse métrique et topologique de l'ensemble formé par l'information image fournit des descripteurs fondés sur une saillance statistique sans aucun *a priori* géométrique, et par opposition aux techniques connues de points d'intérêt [98, 75], ce principe fusionne la détection des structures saillantes et leur description. Nous proposons deux types de structures saillantes complémentaires purement fondés sur la géométrie de l'ensemble des caractéristiques : (1) les structures rares, qui sont la rétroprojection des points isolés dans l'espace des caractéristiques (points dont la distance moyenne aux k plus proches voisins est la plus grande), et (2) les structures dominantes, rétroprojection des modes de l'espace des caractéristiques, calculés par reconstruction géodésique des clusters (points dont la distance moyenne aux k plus proches voisins est la plus petite), selon une méthode inspirée par Burman et Polonik [21]. La figure 2.9 montre un exemple de ces différentes représentations.



(1) (2) (3)

FIGURE 2.9 – Représentations de niveau supérieur extraites de la structure métrique ou topologique de l'ensemble des caractéristiques. (1) Etiquettes issues de la quantification du local jet. (2) Saillance statistique : rétroprojection des points de local jet isolés. (3) Structures dominantes : rétroprojection des modes du local jet. La deuxième ligne représente un détail agrandi issu de la première.

Chapitre 3

Analyse du Mouvement

Le mouvement a toujours occupé une place majeure dans nos activités de recherche. On pourrait le justifier de façon purement pragmatiste par les projets co-financés qui nous ont permis de faire vivre notre recherche jusqu'ici : depuis le cadre applicatif de notre thèse CIFRE défini par Aérospatiale-Missiles, jusqu'au projet ITEA2 SPY actuel (CASSIDIAN), en passant par les PEA PATRICIA (MBDA), CALADIOM (Bertin Technologies), et la convention CIFRE avec CEDIP Infrared Systems pour la thèse de Christine Dubreu, tous ces projets ont en commun l'analyse d'objets et de scènes mobiles. Il y a cependant une explication plus profonde qui justifie également la place centrale dédiée au mouvement entre les chapitres «Représentations» et «Vision embarquée» :

- De façon assez évidente, un système embarqué est aussi un système mobile, capable de traiter l'information rapidement, c'est donc naturellement qu'on aborde le traitement d'un flux video dynamique, par opposition à une suite de traitements d'images statiques.
- De façon plus fondamentale, nous croyons que la continuité temporelle permet d'accéder à de nouvelles représentations de l'information visuelle, qui peuvent être plus riches, et qui sont souvent plus efficaces : une information spatiale fruste intégrée temporellement le long d'une trajectoire peut remplacer avantageusement un descripteur invariant complexe calculé de façon éparse.

La perception du mouvement joue un rôle majeur dans la vision biologique, y compris dans les systèmes visuels les plus sophistiqués, où des zones cérébrales se spécialisent dans la perception de certains mouvements. Pour la vision artificielle, on distingue habituellement trois tâches de bas niveau fondamentales liées à l'analyse du mouvement :

1. **Détection :** Il s'agit de déterminer dans chaque image, quels pixels appartiennent aux objets mobiles par rapport à la scène, laquelle est généralement fixe par rapport à la caméra. Le résultat est donc une image binaire par trame.
2. **Poursuite :** Il s'agit de localiser dans chaque image la position d'objets de taille plus ou moins étendue. Le résultat est pour chaque trame une localisation des objets plus ou moins précise : coordonnées du centre, rectangles englobants, silhouettes...
3. **Estimation :** Il s'agit d'estimer la vitesse apparente, i.e. la projection du vecteur vitesse de chaque point visible sur le plan image. Le résultat est, pour chaque trame, un champ de vecteur appelé flux optique, qui peut être dense ou épars.

Les trois premières sections présentent un bilan de nos contributions pour ces trois opérations de bas niveau ; la section 4 traite de la caractérisation de mouvement : il s'agit de modéliser un mouvement complexe, et d'extraire dans une vidéo des caractéristiques permettant de reconnaître une action ou une activité dont le support s'étend dans le temps et dans l'espace.

3.1 Détection

Si l'on peut considérer la détection de mouvement comme un problème relativement bien maîtrisé aujourd'hui, on ne saurait le qualifier de résolu, et les limitations des systèmes actuels laissent encore un large champ à la recherche :

- Variations globales de la scène : la détection ne doit pas être sensible à des changements - progressifs ou brutaux - qui affectent la scène fixe (conditions météo, changements d'illumination,...).
- Perturbations locales de la scène : la détection doit ignorer certains changements locaux naturels ou anodins (par ex. dus aux effets de la houle, du courant ou du vent sur une surface d'eau, un arbre, un drapeau...)
- Mouvements radiaux : la détection doit localiser avec précision un objet mobile quelque soit son mouvement, même lorsqu'il se déplace dans la direction de l'axe optique, ce qui occasionne peu de changement.

Cette section suit une logique chronologique : les premiers travaux sont fortement influencés par le parallélisme cellulaire des rétines, puis vient l'estimation Σ-Δ, fruit du projet CALADIOM, dont la conception vient aussi des rétines, mais qui a été implanté avec succès dans une large gamme d'architectures embarquées (voir Chap. 4). Enfin la der-

nière partie présente une autre application du système à base d'espaces de caractéristiques, présenté dans le chapitre précédent.

3.1.1 Les débuts

Les premiers travaux après la thèse sont fortement influencés par des contraintes de mémoire très fortes, qui limitent la représentation à quelques bits par pixel. Les algorithmes proposés, fondés sur une modélisation markovienne [18, 22], privilégient le calcul sur la mémoire. Dans [92] on peut trouver une forme extrême de ce déséquilibre : seule le changement temporel instantané est calculé (mais pas stocké) en chaque pixel, mais la puissance de calcul est telle qu'on peut appliquer à peu de frais un recuit simulé sur un champ de Markov avec des cliques spatiotemporelles et des étiquettes binaires.

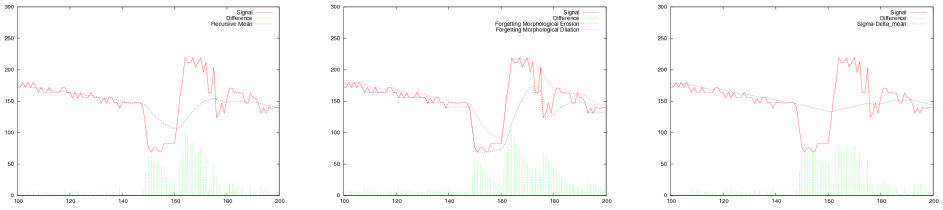
Dans la génération suivante de rétines programmables, la contrainte de pénurie mémoire se relaxe sensiblement, et on peut envisager de conserver en chaque pixel des grandeurs issues de statistiques temporelles. C'est le début du PEA CALADIOM, dont l'objectif est le développement de balises de renseignement à très longue autonomie utilisant une rétine programmable comme capteur d'alerte. C'est aussi la période de la thèse de Julien Richefeu [110], consacrée à l'analyse de mouvement sur systèmes à base de rétines programmables.

Lorsque le capteur est fixe, ce qui est le cas pour la plupart des systèmes de détection, il apparaît essentiel, pour pouvoir aborder les difficultés évoquées plus haut, d'estimer la distribution statistique des valeurs observées localement (en un pixel, ou un bloc de pixels) au cours du temps, de façon à pouvoir déterminer si la valeur observée à chaque instant est significative de cette distribution, ou si elle est plus probablement due à un objet mobile. Ce problème d'estimation de fond statique (Background estimation) a fait l'objet d'une abondante littérature dans les deux dernières décennies [68, 24, 107]. Certaines méthodes effectuent une analyse statistique à partir d'un historique de K valeurs passées du pixel et réalisent la classification de la valeur courante en Fond/Objet mobile sur des critères divers : prédiction linéaire [131], estimation de densité de probabilité [41, 99], ou analyse en composantes principales [104]. Toutefois le nombre K de valeurs d'historique doit en général être grand, rendant ces techniques difficilement applicables même sur les nouvelles rétines.

On s'est donc intéressé plus fortement aux techniques récursives, qui ne conservent en mémoire qu'un nombre réduit d'estimations statistiques mises à jour à chaque pas de temps. Les estimateurs les plus couramment utilisés sont la moyenne et la variance d'une distribution supposée gaussienne [141]. Il est possible d'ajouter des états dy-

namiques pour améliorer la prédiction dans le cas d'un fond variant rapidement [62]. Il est aussi possible d'étendre l'estimation à des distributions multimodales [125, 109], indispensables pour traiter les changements liés aux perturbations locales.

Nous avons étudié et comparé 3 opérateurs fondamentaux pour la détection de mouvement dans les systèmes à base de rétines [113], fondés sur des estimateurs M_t calculés récursivement à partir des valeurs I_t des pixels : (1) la moyenne récursive, (2) la morphologie oublieuse [112], et (3) l'estimation Σ-Δ [94]. Le premier est extrêmement classique, c'est le filtre linéaire couramment utilisé dans l'estimation des paramètres d'une gaussienne, et qui correspond à l'implantation récursive d'un filtre exponentiel : $M_t = M_{t-1} + \alpha(I_t - M_{t-1})$, avec $\alpha \in]0,1[$ dimensionné à une fréquence (Fig. 3.1(1)).



(1) (2) (3)

FIGURE 3.1 – Comparaison des 3 estimateurs récursifs sur une série temporelle (courbe rouge) correspondant à un pixel de fond avec passage d'objet. (1) Moyenne récursive (en bleu), (2) Morphologie oublieuse (érodé en bleu, dilaté en rose), (3) Moyenne Σ-Δ (en bleu). Le signal de différence est en vert.

Le deuxième opérateur est complètement nouveau, et correspond à l'estimation récursive des valeurs extrêmes lissées exponentiellement dans le temps : $m_t = \alpha I_t + (1-\alpha)\min(I_t, m_{t-1})$, et $M_t = \alpha I_t + (1-\alpha)\max(I_t, M_{t-1})$. Ces opérateurs dits respectivement d'érosion et dilatation temporelles oublieuses, permettent par différence d'estimer les amplitudes de variation sur une période de temps déterminée par $1/\alpha$ (Fig. 3.1(2)), ce qui est particulièrement adapté aux détections en dessous du seuil de discrétisation spatiotemporelle, i.e. objets petits ou mouvements lents [112]. La morphologie oublieuse peut aussi être appliquée dans le domaine spatial, créant ainsi des opérateurs hybrides de complexité indépendante du rayon de l'élément structurant (déterminé par $1/\alpha$). Ce concept a été utilisé dans l'opérateur de reconstruction hybride [93], qui constitue une version «soft» de la reconstruction géodésique.

Le troisième opérateur n'est pas vraiment nouveau, car il est fondé

sur la vénérable modulation Σ-Δ, une technique classique de conversion analogique numérique, à la base de nombreux algorithmes de discrétisation graphique [19], ou tonale [13]. Il semblerait qu'elle n'ait été employée qu'une fois avant nos travaux comme estimateur de fond statique, et justifiée comme une approximation du médian temporel [95]. L'estimateur est le suivant : $M_t = M_{t-1} + H(I_t - M_{t-1})$, où $H(x) = -1$ si $x < 0$ et $H(x) = +1$ si $x > 0$ (Fig. 3.1(3)). Outre son excellente adaptation au jeu d'instructions booléen de la rétine [81], nous avons montré que les performances de l'estimation Σ-Δ était généralement meilleure que celles de ses homologues dans le cas unimodal [80, 82].

3.1.2 Estimation Σ-Δ

Après l'implantation sur le capteur CALADIOM, nous nous sommes intéressés, d'une part à une justification statistique de l'approche, et d'autre part à une généralisation au cas multimodal. Nous nous sommes appuyés sur l'estimation récursive d'une moyenne à partir d'une série temporelle fondée sur un modèle multimodal de la distribution, telle que proposée par Stauffer et Grimson [125] et affinée par Power et Schoonees [109]. On peut généraliser les estimateurs récursifs d'ordre 1 évoqués plus haut en utilisant une fonction d'incrément $M_t = M_{t-1} + \delta_t(I_t)$. Selon Stauffer et Grimson, l'incrément δ_t ne devrait pas seulement dépendre de la différence entre la valeur courante I_t et l'estimateur précédent M_{t-1}, mais aussi de la probabilité d'occurence de la valeur I_t, selon la densité estimée f_t de la distribution à l'instant t, soit $\delta_t(I_t) \propto f_t(I_t) \times (I_t - M_{t-1})$. Selon ce modèle, l'estimation Σ-Δ est associée à une distribution unimodale de Zipf-Mandelbrot [145] : voir Figure 3.2.

Il est peu probable que la loi de Zipf corresponde à quelque réalité physique pour notre problème, mais le fait d'établir ce cadre a permis une meilleure compréhension des liens entre les modèles de distributions et les estimateurs récursifs. Il a aussi permis la conception d'un algorithme plus fin et plus complet, étant établi que dans le modèle zipfien, la valeur de l'incrément/décrément dépend de la variance du modèle. En pratique cependant, il est souhaitable que cette valeur soit toujours égal au bit de poids faible de la représentation. Nous avons donc proposé, par analogie à l'estimation gaussienne de Power et Schoonees, un algorithme d'estimation «zipfienne» [82] où la fréquence de mise à jour de la moyenne Σ-Δ est proportionnelle à la variance, calculée par estimation Σ-Δ des différences à la moyenne. Par la même analogie, on peut étendre l'estimation à une distribution multimodale, selon le même principe que [109]. Nous avons également montré [93] l'intérêt d'une estimation multiple au sens de la profondeur temporelle, en

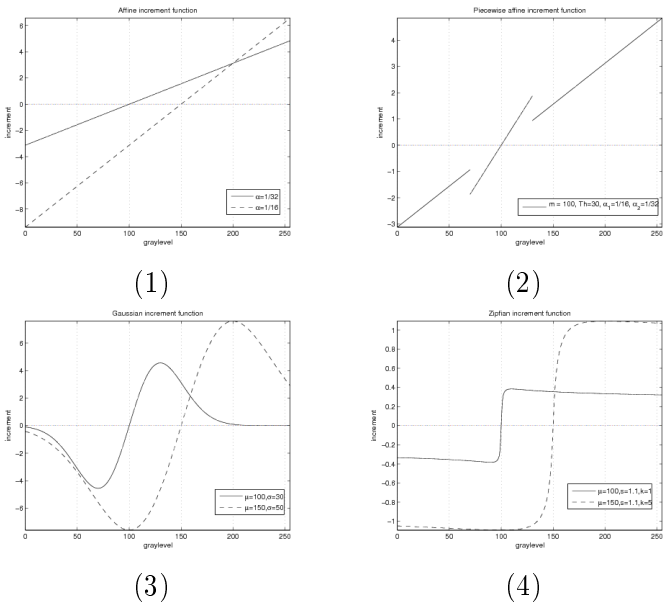


FIGURE 3.2 – Profils de la fonction d'incrément $\delta_t(I_t)$ selon plusieurs modèles et hypothèses. En abscisse : la valeur de niveau de gris observé, en ordonnée : la valeur de l'incrément correspondant. (1) Distribution uniforme : fonction d'incrément affine (moyenne récursive). On voit 2 exemples où les valeurs de M_t et de α sont différentes. (2) Distribution constante par morceaux : fonction d'incrément affine discontinue (moyenne récursive avec boucle de pertinence). (3) Distribution gaussienne : fonction d'incrément dérivée de gaussienne (estimation gaussienne). On voit 2 exemples où les valeurs des moyenne et variance sont différentes. (4) Distribution zipfienne : fonction d'incrément de forme Heaviside (estimation Σ-Δ). On voit 2 exemples où les valeurs des moyenne et variance sont différentes.

calculant pour chaque mode, la moyenne et la variance à différentes fréquences temporelles (par ex. à court, moyen et long termes).

3.1.3 Consensus dans l'espace des caractéristiques

Lorsque la distribution temporelle du fond devient très complexe, le modèle multimodal devient inefficace, et il est souvent plus rentable d'approcher la distribution par un moyen plus direct. C'est le principe des méthodes d'estimation de fond par échantillonnage et consensus [139, 12], qui consiste à conserver en mémoire un certain nombre de valeurs prises par chaque pixel, et à comparer la valeur courante à cet échantillon, pour décider si le pixel appartient au fond ou non.

Dans [84], une des applications du modèle de traitement dans l'es-

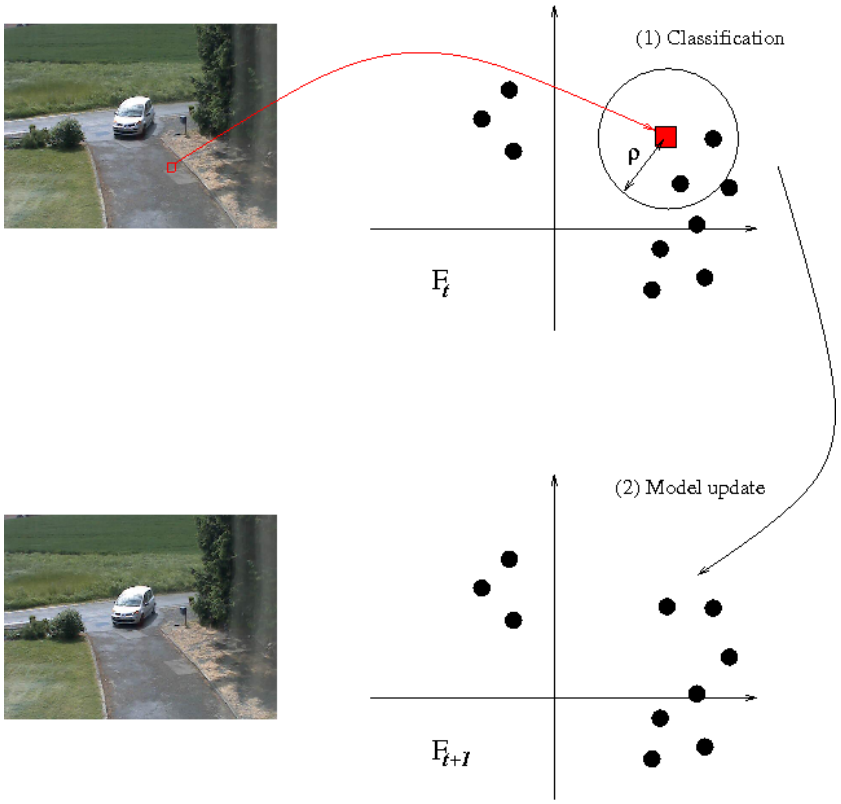


FIGURE 3.3 – Modélisation de fond statique par échantillonnage et consensus dans un dictionnaire de caractéristiques quantifiées.

pace des caractéristiques que nous proposons est une synthèse des méthodes d'échantillonnage et consensus, avec les méthodes d'estimation de fond par quantification vectorielle [64]. C'est en fait une simple adaptation de l'algorithme *ViBe* [12] aux espaces de caractéristiques. Pour cette application, l'espace des caractéristiques est quantifié, et chaque pixel est donc associé à un mot du dictionnaire, qu'on suppose déjà construit dans une phase d'initialisation, mais qui peut évoluer dans le temps. Le principe est le suivant (cf Figure 3.3) : Pour chaque pixel, on modélise l'activité temporelle par un ensemble de M prototypes qui sont des mots du dictionnaire correspondant à des valeurs passées prises par le pixel dans l'espace des caractéristiques (phase d'échantillonnage). La classification du pixel en Fond / Objet mobile est ensuite faite en choisissant un rayon $\rho > 0$ et en comptant le nombre de prototypes qui sont à une distance inférieure à ρ de la valeur actuelle du pixel, dans l'espace des caractéristiques (phase de consensus). Le mot correspondant à la valeur actuelle du pixel remplace ensuite l'un des prototypes existant (phase de mise à jour).

L'avantage d'utiliser un espace de caractéristiques est de représenter en chaque pixel une structure spatiale plus riche que la seule couleur, ce qui doit rendre la modélisation de fond par échantillonnage plus robuste. Dans le même temps, la quantification par un dictionnaire réduit fortement le coût en mémoire, puisque seul l'index du mot est codé pour chaque prototype, pas le vecteur caractéristique complet. Ce qu'on observe dans un fond statique typique est que la majorité des pixels ne sont représentés que par un ou deux mots différents parmi leurs M prototypes, tandis que quelques zones de fond complexe (perturbations locales), peuvent être représentées avec beaucoup plus de mots.

3.2 Poursuite

La poursuite d'objets (cibles) dans une vidéo est encore une problématique ancienne, sur laquelle les acteurs de la défense et de la sécurité - entre autres - travaillent depuis longtemps, sans être parvenus à une solution complètement satisfaisante du point de vue de la robustesse et de la variété des scénarios. Les principales difficultés de la poursuite sont les suivantes :

- Variations d'aspect de la cible, dues à : sa déformabilité, son changement d'échelle, de pose 3d, son ombrage, son occultation partielle ou totale, le bruit, le flou de bougé, etc.
- Complexité potentielle liée à la taille de l'espace de recherche dans lequel la cible doit être localisée.

Ces difficultés expliquent la place importante que prend souvent le filtrage prédictif dans un système de poursuite : l'appariement de descripteurs image par image suffit rarement, on doit estimer de façon temporellement cohérente un ensemble de paramètres décrivant l'état de la cible (en particulier cinématique), de façon à, d'une part, détecter les localisations absurdes dues aux erreurs d'appariement et, d'autre part, réduire la complexité de la recherche en fournissant une prédiction correcte de la position attendue. Et comme, contrairement à la détection, la caméra est généralement mobile, cette estimation / prédiction s'applique d'abord au mouvement du porteur de la caméra, où elle est généralement combinée avec une information odométrique.

Nous avons contribué au PEA PATRICIA, un système de pointage de cible assisté par traitement d'images pour les missiles antichars. Les difficultés principales étaient liées aux grands mouvements du porteur lors du délestage, et à l'occultation de la cible par les fumées ou par le missile lui-même. Nous avons développé un algorithme fondé sur la détection robuste de changement après recalage rigide du fond [78].

Un autre investissement significatif dans le domaine de la poursuite a été fait à l'occasion de la thèse de Christine Dubreu [36], en contrat CIFRE avec CEDIP Infrared Systems (aujourd'hui FLIR ATS), sur les systèmes de télésurveillance infrarouge héliportés. La première contribution est algorithmique : elle concerne l'extension des méthodes dites par centroïde aux espaces de caractéristiques. La poursuite par centroïde [4] est une méthode fruste mais très attrayante d'un point de vue calculatoire car elle supprime le parcours d'un espace de recherche, le remplaçant par un calcul de centre de gravité sur une carte de vraisemblance. Cette carte se calcule à partir de 2 histogrammes, l'un calculé à l'intérieur de la cible (foreground), l'autre à l'extérieur (background). La vraisemblance, obtenue pour chaque pixel, dépend du rapport entre les probabilités empiriques de sa valeur dans la cible d'une part, et à l'extérieur de la cible d'autre part. Si l'on calcule ces probabilités empiriques à partir du seul niveau de gris, la poursuite peut fonctionner dans des scénarios relativement simples : vidéos infrarouges avec cibles chaudes et fond peu texturé. En revanche elle crée trop de fausses alarmes dans les cas plus complexes, et fonctionne mal en visible. Une idée simple est de calculer les probabilités à partir des valeurs de plusieurs composantes de l'espaces des caractéristiques, afin de rendre la poursuite plus robuste, l'information en chaque point intégrant plusieurs échelles et plusieurs ordres de dérivation. Une amélioration des performances a pu être mise en évidence sur quelques tests préliminaires [36], mais l'idée reste à développer et à évaluer de façon plus poussée.

L'évaluation est justement le thème de la partie la plus académique du travail de Christine Dubreu, dont l'objectif était de répondre à une difficulté majeure liée à la conception des systèmes de poursuite : comment valider un système au regard de critères opérationnels divers, compte tenu de la diversité potentielle des scénarios ? L'évaluation est un problème coûteux et difficile, principalement à cause de l'acquisition d'une base de données de test, obtenue généralement soit par des campagnes d'acquisition réelle suivies d'une saisie plus ou moins manuelle de la vérité terrain, soit en utilisant des vidéos de synthèse produites par une simulation physique sophistiquée.

Or dans le cas de la poursuite (contrairement à la reconnaissance par exemple), la plupart des critères opérationnels (par ex. accélération du porteur, vitesse et taille de la cible, contraste cible / fond, etc) définissant le domaine de validité d'un algorithme peuvent être simulés très simplement en utilisant une synthèse non photoréaliste. Nous avons donc proposé [37] un nouveau protocole d'évaluation fondé sur la dérivation d'un ensemble minimal de paramètres formels (par

opposition aux paramètres physiques de la simulation classique), statiques ou dynamiques, à partir desquels on peut générer un ensemble de séquences non photoréalistes (voir Fig. 3.4), permettant d'évaluer un algorithme de poursuite donné en échantillonnant aussi finement que voulu l'espace des paramètres opérationnels. On a pu montrer [38] que ce protocole produisait des résultats d'évaluation proches de ceux obtenus sur des séquences réelles, et il a été utilisé pour la définition précise des domaines de validité de 2 algorithmes avant leur implantation dans le système en vol.

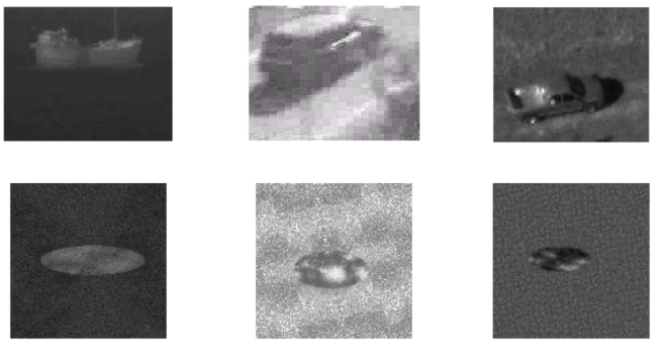


FIGURE 3.4 – Extraits de séquences non photoréalistes (en bas) obtenues par analyse et synthèse de textures [140] sur images réelles (en haut), plus d'autres paramètres formels, pour l'évaluation exhaustive des algorithmes de poursuite.

Outre l'intérêt en termes de simplification et de réduction du coût de la validation, cette évaluation potentiellement exhaustive permet d'envisager la conception de systèmes adaptatifs modulaires : au lieu de concevoir des algorithmes plus complexes et plus robustes, ces systèmes utiliseraient un ensemble de briques plus simples avec des domaines de validité différents, mais circonscrits de façon très précise. Si le système est capable d'estimer dans quelle configuration il se trouve dans l'espace des critères opérationnels, l'évaluation correspondante des briques, qui peut même être réalisée à la demande, fournit un mécanisme simple de sélection.

3.3 Estimation

L'estimation de flux optique est un problème idéal pour un chercheur car l'un des plus mal posés de la vision précoce : on ne peut localement estimer la vitesse apparente qu'en présence de frontières (problème de l'ouverture), mais si ces frontières séparent deux objets,

le flux n'est en général pas correctement défini (problème des discontinuités). Ces difficultés expliquent la diversité des algorithmes de calcul de flux optique, qu'on peut néanmoins regrouper en 4 catégories :

- **Appariement local :** Chaque pixel est apparié au pixel le plus ressemblant de l'image suivante, obtenu par recherche dans un domaine spatial restreint. Le champ de vitesse produit est par nature épars, l'appariement étant ambigu presque partout.
- **Différentiel local** [76] : La fonctionnelle quadratique d'appariement est approximée au premier ordre et le vecteur solution est calculé en résolvant en chaque point un système linéaire d'équations. Le champ de vitesse est également épars, le système d'équations n'étant bien conditionné que pour les points anguleux.
- **Différentiel global** [59] : L'image est approximée au premier ordre, et l'hypothèse de constance du niveau de gris dans le temps implique l'égalité entre le gradient temporel et le produit scalaire du gradient spatial avec le vecteur vitesse (équation de contrainte du flux optique). Le système est résolu de manière globale en ajoutant une condition de régularité spatiale du flux optique, et en minimisant une fonction quadratique combinant le terme lié à l'équation du flux optique, et un terme lié au gradient spatial du champ de vecteurs vitesse. Le champ produit est dense car la résolution itérative est telle que chaque pas de l'itération est appliqué partout avant de passer à l'itération suivante.
- **Fréquentiel :** On applique à l'image un ensemble de filtres spatiaux de fréquences et d'orientations diverses, et on mesure les différence de phase obtenues localement d'une image à l'autre [46], ou bien on convolue les reponses obtenues par une série de filtre temporels dérivateurs [2], et on déduit la vitesse apparente des composantes spatiotemporelles fournissant la réponse la plus élevée. Le champ produit est par nature dense, l'information corrélée en chaque point intégrant des données image sur une portée spatiale potentiellement grande.

Différentes méthodes de flux optique ont été utilisées comme briques dans le système d'évolution artificielle de la fonction d'évitement d'obstacles développé par Renaud Barate dans sa thèse [9, 11]. Ces briques pouvaient être associées à d'autres opérateurs locaux (par ex. mesure du temps avant collision) ou globaux (par ex. somme des normes ou des orientations dans une région rectangulaire), de telle sorte que certains algorithmes évolués dans des environnements suffisamment texturés pouvaient être fondés sur l'estimation de la profondeur relative par le mouvement, soit de façon locale pour détecter la présence d'un obstacle au sol, soit de façon globale pour se centrer dans un couloir

par équilibrage latéral de flux, à l'instar des abeilles [120].

Plus tard nous nous sommes intéressés à la conception même des algorithmes de flux optique, en cherchant, d'une part à obtenir un champ dense de la façon la plus directe possible en limitant au maximum la régularisation spatiale du champ de vecteurs, et d'autre part à obtenir une bonne stabilité temporelle du suivi, en sacrifiant le moins possible la densité du flux.

3.3.1 Un problème de plus proche voisin

Du point de vue conceptuel, l'estimation du flux optique est l'une des applications les plus immédiates du modèle de représentation et traitement par espaces de caractéristiques présenté au chapitre précédent [84]. Voir Figure 3.5 : chaque point de l'espace image est projeté dans l'espace des caractéristiques, et on recherche le plus proche voisin de ce vecteur dans l'ensemble des vecteurs caractéristiques associé à l'image précédente. La rétro-projection du plus proche voisin dans l'espace image fournit l'appariement.

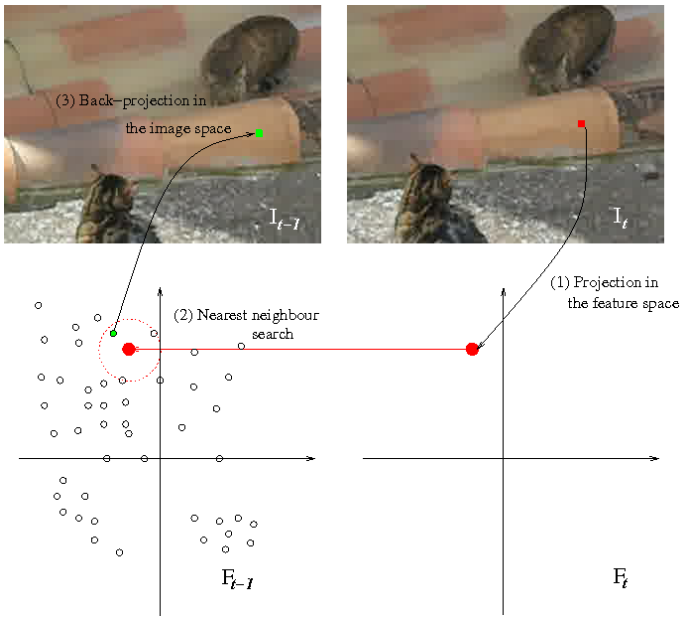
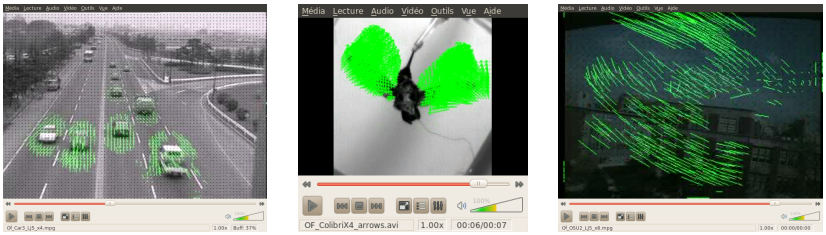


FIGURE 3.5 – Estimation dense de flux optique par calcul du plus proche voisin dans l'espace des caractéristiques.

A une seule échelle le résultat est rarement exploitable, mais en utilisant plusieurs échelles, cette méthode fournit une estimation dense du flux optique, sans aucune régularisation spatiale explicite. Le champ brut obtenu n'est pas très précis en termes de localisation, et ignore

les déplacements des structures trop petites, mais il est globalement cohérent et fournit un champ lisse grâce à la régularisation implicite que représente le calcul des dérivées multiéchelles (Fig. 3.6), ce en quoi il se rapproche plus des méthodes fréquentielles évoquées ci-dessus [2, 46] que des autres catégories.



(1) (2) (3)

FIGURE 3.6 – Exemples de flux denses calculés par plus proche voisin. (1) Scène statique, objets mobiles rigides. (2) Scène statique, objets mobiles déformables. (3) Scène mobile, mouvements abruptes.

Mais bien qu'élémentaire conceptuellement, cette technique est coûteuse à implanter : la recherche du plus proche voisin se fait dans des kd-trees de grande dimension à la fois en termes de dimensionalité de l'espace et de nombre de vecteurs, et le nombre de requêtes est très élevé. Dans le cadre des travaux de Matthieu Garrigues, une version temps réel de l'algorithme a été implantée sur processeur graphique (voir Chapitre suivant), en effectuant le calcul du plus proche voisin dans un voisinage spatiale limité.

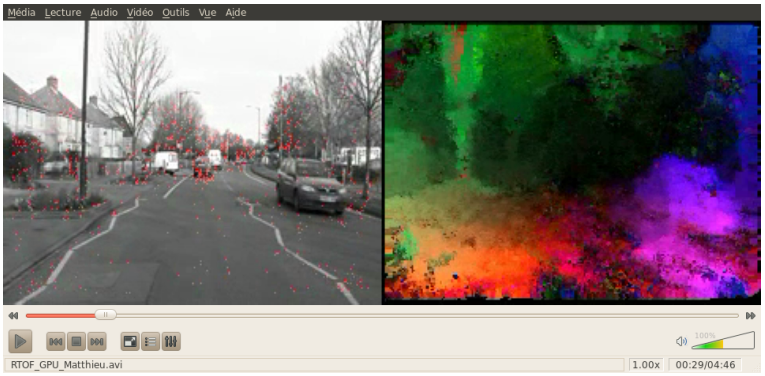


FIGURE 3.7 – Calcul sur GPU du flux optique dans l'espace des caractéristiques avec limitation spatiale de la recherche du plus proche voisin. A gauche, en rouge : points ayant un indice de confiance élevé (sélection *a posteriori*).

3.3.2 Poursuite semi-dense

La version temps réel du flux optique par plus proche voisin, bien qu'exploitable dans toute application temps réel où la densité du champ est essentielle (voir Fig. 3.7), a révélé certaines limitations de l'approche, en particulier liées à la portée limitée de la recherche, qui se traduit par un taux d'erreurs trop élevé dans les grandes zones homogènes. Il s'est avéré nécessaire de calculer un indice de confiance local calculé *a posteriori*, lié à la distance au plus proche voisin dans l'espace des caractéristiques. Or malgré le niveau de parallélisme élevé des processeurs graphiques, il est finalement apparu qu'un certain niveau de sélection *a priori* des points à apparier permettait un gain considérable, en ignorant une quantité significative de points sur lesquels les résultats de l'approche sans *a priori* n'étaient de toutes façons pas satisfaisants.

Nous nous sommes donc dirigés vers une approche différente, dans le but de calculer un flux optique qui soit à la fois cohérent spatialement, c'est-à-dire aussi dense que possible, et temporellement, c'est-à-dire assurant un suivi stable et continu dans le temps des points qui sont appariés. Cet objectif est lié à une exigence de généricité du projet ITEA2 SPY en cours, qui vise à développer des systèmes de télésurveillance mobile urbaine utilisant des caméras embarquées dans des véhicules de sécurité. Etant donné le nombre élevé de scénarios d'usage, et les contraintes d'embarquement du système de vision, il est vital de disposer de primitives de bas niveau qui soient les plus polyvalentes possibles. En l'occurence, la poursuite semi-dense conçue dans ce cadre doit permettre à la fois d'obtenir un flux optique suffisamment dense pour extraire de façon fiable les informations géométriques utiles à la stabilisation, l'estimation de profondeur, ou l'odométrie visuelle. Elle doit aussi permettre d'extraire les trajectoires de points de manière à pouvoir aborder la reconnaissance d'actions ou d'activités complexes.

La poursuite semi-dense de Matthieu Garrigues [51], est fondée sur une sélection la moins discriminante possible. Le principe est de n'éliminer que les points dont l'appariemment sera forcément ambigu. Partant de l'hypothèse que les points non appariables sont ceux au voisinage desquels, pour toutes les échelles de recherche, il existe une direction le long de laquelle le niveau de gris varie linéairement, la fonction de saillance calcule le maximum sur toutes les échelles de la déviation minimum par rapport à la linéarité dans toutes les directions. Ces mesures étant échantillonnées sur plusieurs cercles discrets concentriques, la fonction s'apparente à celle du détecteur de coin FAST [116], qui est d'ailleurs plus rapide mais qui fournit nettement moins de points. Les maxima locaux de la fonction de saillance forment les nouvelles particules candidates.

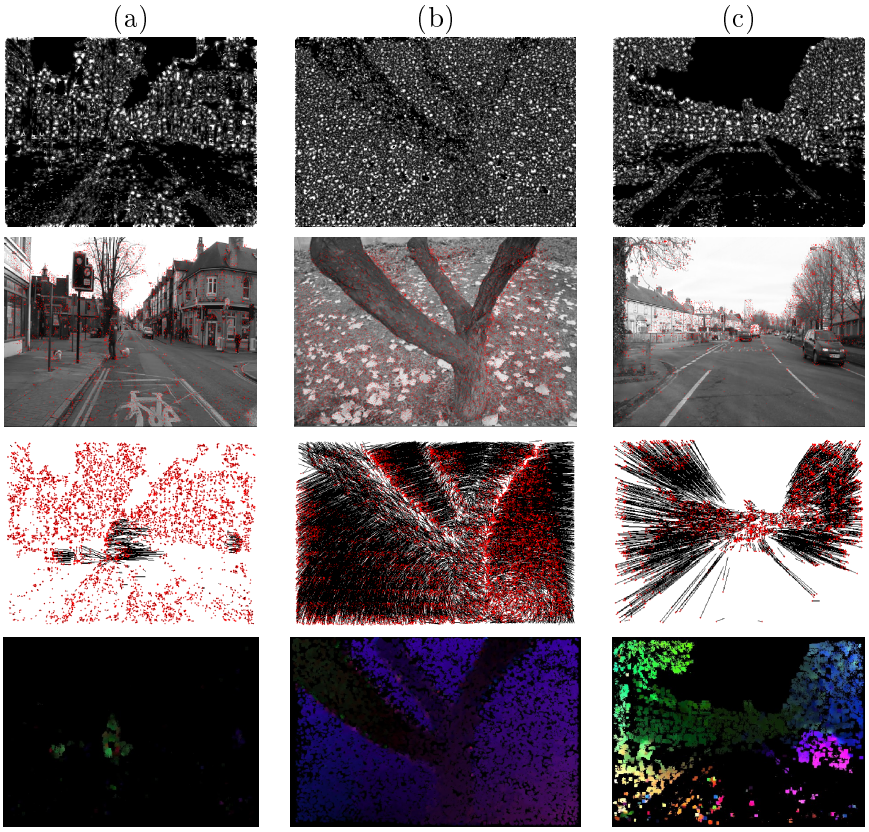


FIGURE 3.8 – Poursuite semi dense pour 3 types de mouvement : (a) Caméra statique, objets mobiles ; (b) Travelling circulaire, objets statiques ; (c) Zoom avant, objets mobiles. Ligne 1 : Fonction de saillance (sélection *a priori*). Ligne 2 : Trame courante avec les particules les plus âgées. Ligne 3 : Trajectoires des particules les plus âgées. Ligne 4 : Flux optique semi-dense reconstruit.

Dans une autre phase indépendante, les particules existantes sont localisées dans la trame courante. Chaque particule porte un certain nombre d'états relatifs à son apparence visuelle, sa vitesse, son âge. La localisation se fait par une phase d'appariement multiéchelles, dont la particularité est d'estimer l'accélération dominante due aux mouvements brusques du porteur, de façon à améliorer la prédiction sur la position des particules et augmenter leur durée de vie. Les particules sont éliminées si leur score d'appariement est trop faible ou si leur vitesse dévie trop par rapport à la vitesse des particules spatialement proches. Cet algorithme permet d'obtenir des trajectoires fiables comme les techniques classiques de poursuite [128], tout en permettant

une reconstruction semi-dense du flux optique grâce à un nombre de points significativement plus élevé : voir la figure 3.8. Ce travail est très proche dans ses objectifs du système *Particle Video* de Sand et Teller [119], mais dans notre cas le calcul est beaucoup plus rapide, principalement grâce au fait qu'il n'y a pas de régularisation spatiale explicite, laquelle nécessite dans [119], d'une part, une triangulation de Delaunay pour définir la topologie spatiale des particules éparses et, d'autre part, une optimisation globale itérative, qui limitent fortement la parallélisation. Au contraire, notre approche est pour l'essentiel massivement parallèle, la seule régularisation spatiale explicite étant l'élimination des particules déviantes, dont le calcul reste purement local.

3.4 Caractérisation

Nos travaux sur la caractérisation de mouvements complexes sont liés à une tâche de niveau sémantique sensiblement plus élevé que ce qui précède, la reconnaissance d'actions ou d'activités complexes. Généralement une action se réfère à un seul objet ou personne, et une activité à plusieurs objets en interaction, mais dans tous les cas, l'information image à traiter possède une certaine étendue à la fois dans l'espace et dans le temps. La reconnaissance d'actions est une thématique qui concentre beaucoup d'efforts de la recherche en vision depuis une dizaine d'années [108, 3], étant une tâche fondamentale à la base de nombreuses applications différentes : vidéosurveillance, indexation de documents vidéo, réalité augmentée, jeux vidéo, interaction humain-robot... Les principales difficultés sont :

- la grande variabilité liée à la taille, la pose, la direction, et l'apparence visuelle des protagonistes.
- la localisation temporelle, i.e. l'identification de la phase d'une action complexe de durée inconnue.

On peut distinguer les nombreuses méthodes proposées en trois catégories selon la nature du descripteur de mouvement :

1. **silhouette :** Le mouvement est interprété en fonction des variations de forme de la silhouette binaire de l'objet mobile obtenu par détection de mouvement [53].
2. **descripteur spatiotemporel :** Le mouvement est détecté et décrit comme une structure saillante spatiotemporelle obtenue souvent par généralisation d'un détecteur spatial aux volumes espace $\times$ temps [63].
3. **flux optique :** Le mouvement est décrit à partir des champs de vitesses apparents mesurés dans le volume espace $\times$ temps

considéré [144].

Nous nous intéressons aux problèmes de la modélisation d'action complexe, de la classification d'actions sur vidéos courtes, et de la reconnaissance d'actions dans un flux vidéo sans limite temporelle. Ces travaux récents sont liés à deux cadres clairement distincts sans être complètement indépendants : (1) la thèse de Fabio Martínez sur la caractérisation du mouvement biologique anormal, et (2) le projet SPY de vidéosurveillance urbaine mobile, dont certains scénarios demandent des capacités de reconnaissance d'actions complexes et d'activités, principalement humaines. La variété des types de mouvements qu'on peut rencontrer, à la fois dans les applications biologiques et en télésurveillance, nous ont conduits à rechercher des descripteurs qui soient le plus indépendants possible de l'apparence visuelle, et qui puissent également se généraliser à des contextes où la caméra est mobile, typique du projet SPY.

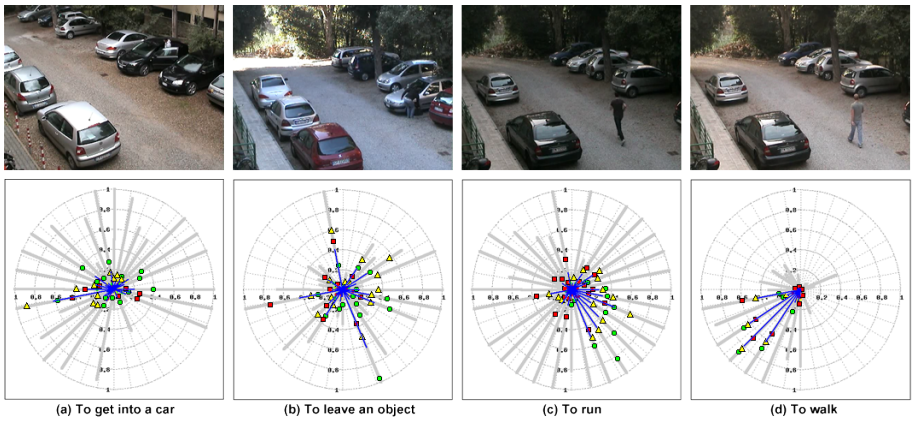


FIGURE 3.9 – Descripteurs d'actions par statistiques spatiotemporelles d'orientation des vitesses instantanées calculés à partir d'un flux optique dense, pour 4 actions tirées de la base ViSOR [7]. L'utilisation de différents supports temporels dans le descripteur permet par exemple de distinguer les actions périodiques des actions apériodiques.

Nous avons ainsi proposé [23] un descripteur fondé sur le flux optique : dans l'esprit du détecteur HOG [33], on construit pour chaque trame un histogramme des orientations du flux optique, où l'orientation est quantifiée sur 16 ou 32 valeurs, et les effectifs sont pondérés par la norme de la vitesse. Cette partie «instantanée» du descripteur est très proche du détecteur HOOF [25], mais sans la symétrie verticale. Le descripteur d'action est ensuite construit en calculant pour chaque orientation un certain nombre de statistiques temporelles : valeurs extrêmes, moyenne, variance... Ces statistiques sont calculées de

manière concurrente à plusieurs échelles temporelles, permettant d'intégrer les différentes phases d'une action complexe. Le descripteur a été évalué dans une tâche de classification sur séquences courtes sur les bases publiques Weizmann [53] et ViSOR [7].

Chapitre 4

Vision Haute Performance

Jusqu'au début des années 2000, la programmation parallèle avait un côté un peu ingrat : les performances obtenues sur une architecture exotique et coûteuse grâce à des innovations majeures et au prix de plusieurs années de travail, ne seraient-elle pas bientôt celles qu'un geek inculte tirerait de son CPU acheté en solde à Montgallet ? Mais depuis 10 ans, une tendance nouvelle a changé la donne [127] : la fréquence d'horloge a décroché de la loi de Moore. Si la capacité d'intégration des microprocesseurs va sans doute continuer de croître de façon exponentielle pendant quelques années encore, le nombre d'instructions élémentaires réalisées par seconde sur les registres d'une unité de calcul n'a guère augmenté depuis 2004. L'augmentation de la puissance de calcul des processeurs courants ne vient donc plus du fait qu'ils calculent la même chose plus vite, mais du fait que (1) de plus en plus d'unités travaillent en parallèle, et que (2) le flux de données entre unités de calcul et unités de stockage est amélioré par les hiérarchies de mémoire cache de plus en plus sophistiquées. La première conséquence est qu'on ne peut plus compter sur l'augmentation prochaine de la fréquence pour implanter un algorithme existant trop lourd. La seconde est qu'il faut prendre en compte dès la conception de l'algorithme le mode et le niveau de parallélisme, ainsi que les flux de données associés.

Les difficultés induites sont nombreuses, non seulement parce que certains algorithmes ne se parallélisent pas facilement, mais parce que chaque type de parallélisme, chaque niveau de découpage des tâches, chaque topologie de communication entre unités de calcul peut impliquer une remise en question profonde de l'algorithme. Quant à l'augmentation du nombre de hiérarchies mémoire, elle induit d'importantes non linéarités des performances vis-à-vis de nombreux paramètres, ce qui complique encore l'optimisation. Le développement d'une application tirant pleinement parti d'une plateforme standard est donc aujour-

d'hui plus difficile. *A fortiori*, l'implantation d'un algorithme sur une plateforme de calcul haute performance (HPC) oblige à se contraindre aux propriétés de chaque architecture.

Un parti pris fort de nos recherches est que ces contraintes ne doivent pas être considérées comme un carcan réducteur conduisant à des simplifications et à un appauvrissement algorithmique, mais au contraire comme une source d'inspiration pour repenser un algorithme en profondeur et peut-être aboutir à des idées nouvelles de portée plus générale. En ceci l'immersion dans l'univers un peu aride des rétines programmables a clairement été un bénéfice, non seulement par les algorithmes nouveaux qui en sont sortis [1], mais aussi à plus long terme, sur la manière d'aborder la Vision Haute Performance.

Ce chapitre comprend trois sections. La première est dédiée aux rétines programmables et à la minimisation logique. La seconde est centrée sur les aspects liées à la mémoire et à l'arithmétique sous contraintes, qui concernent à la fois les dernières générations de rétines et le parallélisme vectoriel. La troisième enfin résume nos expériences plus récentes avec diverses architectures sur étagères (COTS, pour Commercial off-the-shelf), et la confrontation de nos algorithmes avec différents modes de calcul.

4.1 Symphonie pour porte NAND

Les rétines programmables sont le fruit d'une volonté de réduire à l'extrême la consommation d'énergie d'un système de vision sans sacrifier la généricité. Ce sont à la fois des caméras CMOS et des machines massivement parallèles, qui traitent le flux vidéo directement dans le plan focal et ne transmettent pas d'images, mais des descripteurs évolués qui seront analysés par un processeur externe chargé des traitements de plus haut niveau. Outre le parallélisme massif, c'est la parcimonie extrême de la mémoire numérique (quelques bits par pixel / processeur) qui caractérise l'algorithmique des rétines programmables. C'est le concept des TCL [143] (Traitements Combinatoires Locaux) qui domine les premières réalisations [14], i.e. la combinaison d'opérateurs booléens locaux sur des images où le niveau de gris peut être codé par modulation spatiale de valeurs binaires (codage demi-teinte). Dans les générations suivantes [105], le concept NSIP (Near Sensor Image Processing) [48] permet de traiter en chaque pixel des niveaux de gris en modulant - dans le temps cette fois - les traitements booléens sur

1. Algorithmes dont l'intérêt a été reconnu au delà des rétines : les squelettes MB cumulent plus de 150 citations sur 5 articles depuis 1999, et l'estimation Σ-Δ plus de 100 citations sur 2 articles depuis 2004.

des ensembles de niveau de l'image pendant son acquisition.

L'architecture des systèmes à base de rétine programmable sur lesquels nous avons travaillé est représentée sur la Figure 4.1 : la rétine elle-même est une grille de processeurs à entrée optique interconnectée par une maille 4-connexe régulière. Les processeurs appliquent une suite d'instructions SIMD qui leur est envoyée par le contrôleur. Le processeur externe échange des données avec la rétine, détermine dynamiquement le programme envoyé par le contrôleur à la rétine, et se charge des calculs de plus haut niveau. Chaque processeur de la rétine se compose de : (1) un photorécepteur, (2) un convertisseur analogique-numérique, (3) de la mémoire numérique, et (4) une unité booléenne, qui peut lire deux bits de sa mémoire numérique ou de celle de ses voisins, réaliser une opération élémentaire (ex : NAND) et écrire le résultat sur un bit de sa mémoire. Il est important de comprendre que les quelques bits de la mémoire numérique sont utilisés à la fois comme mémoire de données (telle une RAM), comme mémoire de calcul (tels les registres d'une ALU), et comme mémoire de transit pour stocker un résultat intermédiaire utilisé plusieurs fois (tel un cache).

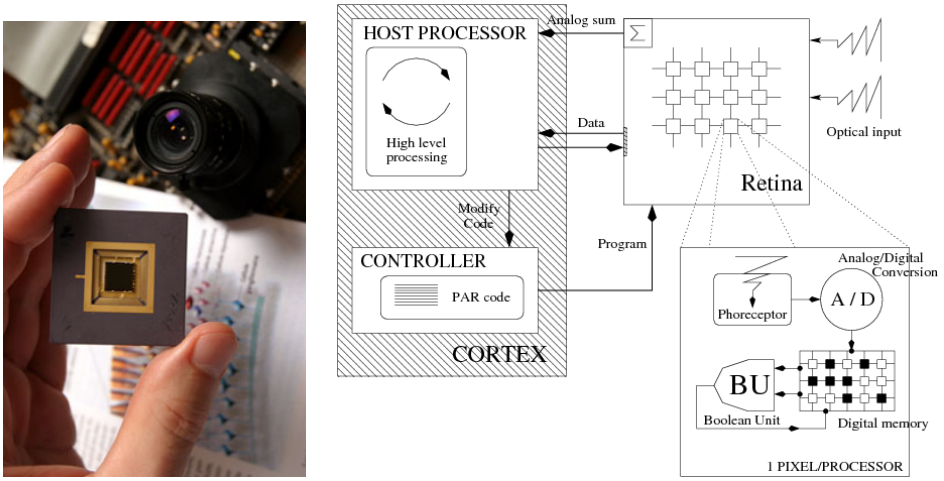


FIGURE 4.1 – La rétine programmable *Pvlsar 34*, et l'architecture d'un système Rétine-Cortex.

Une composante importante de l'algorithmique des rétines numériques est donc la minimisation logique : comment décomposer telle fonction booléenne de façon optimale étant donné le nombre de bits de mémoire à disposition ? Puisqu'une fonction se calcule en chaque pixel par une séquence d'opérations booléennes entre deux bits, l'objectif est donc de réduire au maximum le nombre de ces opérations, qui forment les nœuds de la représentation graphique d'un opérateur de rétine (voir

Figure 4.2). Un mécanisme fondamental de réduction est la décomposition spatiale : grâce au parallélisme, on peut calculer des fonctions de grand support par une composition de plusieurs fonctions de supports plus petits, ce qui réduit significativement le nombre d'opérations. Tous les algorithmes d'amincissement de la famille MB-2d [87] peuvent ainsi se calculer avec quelques bits de mémoire pour un nombre d'opérations élémentaires entre 16 et 60 par itération. Grâce à la décomposition spatiale qui correspond ici à la factorisation de quelques configurations géométriques (voir Figure 4.2), le nombre total d'opérations élémentaires peut être dans certains cas réduit jusqu'à devenir inférieur au nombre d'opérandes de la fonction booléenne résultante.

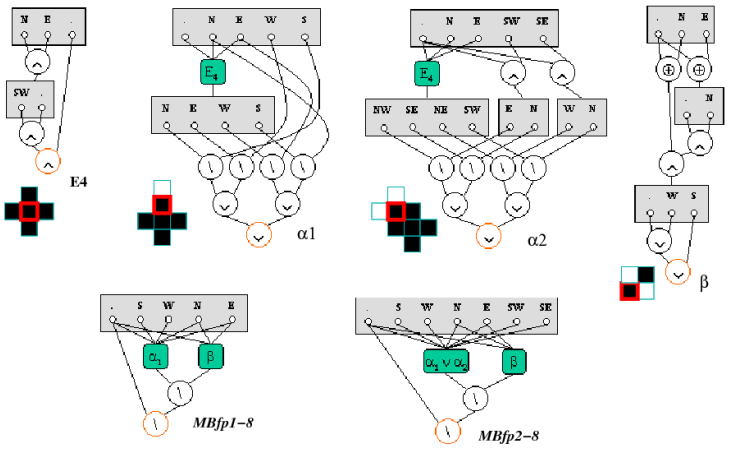


FIGURE 4.2 – Opérateurs rétiniens de squelettisation MB complètement parallèles en 8-connexité : on a mis en évidence la décomposition spatiale par factorisation de configurations géométriques.

Avec la possibilité de traiter les ensembles de niveau durant l'acquisition, les rétines numériques ont accédé au traitement en niveau de gris. Les applications les plus immédiates sont les opérations morphologiques de base, puisque c'est exactement sur ce principe ensembliste qu'est fondée la morphologie mathématique : opérations croissantes bien sûr (érosion, dilatation, filtres morphologiques...) mais aussi opérations résiduelles (gradients, laplaciens, top-hat...). On peut généraliser ce principe à des opérateurs plus sophistiqués, comme la fonction de saillance morphologique pour le calcul des points d'intérêt [111], égale au produit de la norme du gradient par la courbure de l'isophote, approximé en comptant le nombre de fois qu'un pixel appartient à une extrémité de l'endosquelette 8-connexe (courbure négative) ou de l'exosquelette 4-connexe (coubure positive), pour tous les ensembles de niveau (Thèse de Julien Richefeu).

Un élément important du traitement par ensembles de niveau est qu'on peut travailler avec une précision tonale beaucoup plus importante que le nombre de bits dont on dispose, en séquentialisant les traitements. Un exemple typique est le filtrage de dynamique que nous avons proposé pour le calcul de la ligne de partage des eaux (LPE) sur une rétine [79], qui consiste à fusionner les résultats de 2 LPE sur la même image vue selon 2 quantifications tonales (et donc 2 échantillonnages temporels des ensembles de niveau) en opposition de phase. Un autre exemple significatif est la détection de contours large dynamique réalisée en cumulant les contours de plusieurs images obtenues dans des plages d'acquisition différentes [101], pour éviter les phénomènes de saturation typiques des images à très grande dynamique d'illumination (Stage de Master de Paul Nadrag).

4.2 Et pour quelques bits de plus

Dans les dernières générations de rétines, la pénurie mémoire se relativise : on passe de 5 bits à 48 bits de mémoire par pixel. Le programmeur ému voit un nouveau monde s'ouvrir à lui : l'Arithmétique ! Même si le jeu d'instruction est toujours réduit peu ou prou à la porte NAND, la possibilité de stocker des données conduit à développer des routines arithmétiques bit-série (voir Figure 4.3) et une bibliothèque de fonctions optimisées pour l'arithmétique à virgule fixe [90]. C'est dans ce contexte que se fait l'implantation sur le circuit *Pvlsar 34* des différents algorithmes de détection de mouvement par estimation de fond statique évoqués au chapitre précédent [113].

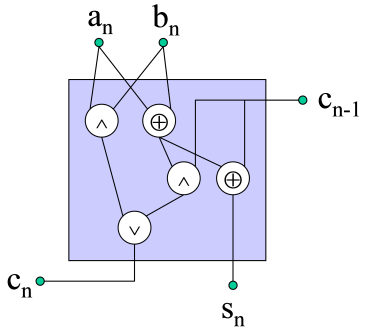


FIGURE 4.3 – Une routine bit-série typique : le full adder logiciel.

Parmi ces algorithmes, l'estimation Σ-Δ est celle qui réalise le meilleur compromis entre nombre d'instructions et parcimonie de la mémoire de données. En effet, elle ne comporte aucune approximation

liée à la troncature, ses seules opérations sont l'incrément / décrément élémentaire et la comparaison, qui s'appliquent en arithmétique de n'importe quelle taille, sans augmenter le besoin en précision. Même la version «zipfienne», qui modifie la valeur de l'incrément en fonction de la variance [82], garde exactement le même jeu d'instructions et la même complexité, puisque la mise à jour de la moyenne est seulement masquée en fonction des bits de la variance, de telle sorte que la valeur moyenne de l'incrément soit effectivement proportionnelle à la variance (voir Figure 4.4).

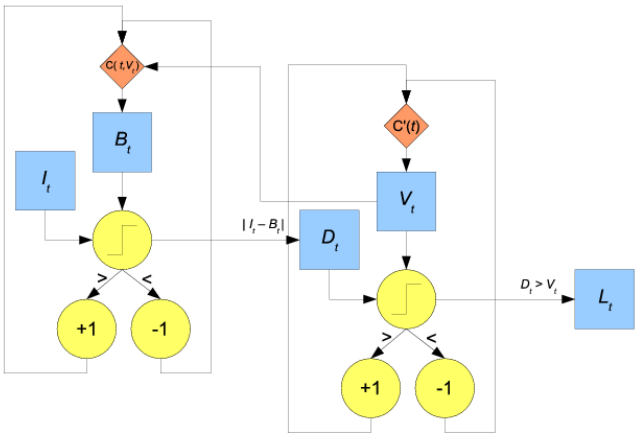


FIGURE 4.4 – Version zipfienne de l'estimation Σ-Δ : la complexité est transférée au contrôle (conditions des mises à jour symbolisées par les losanges rouges).

C'est donc l'estimation Σ-Δ qui a été implantée sur le prototype de CALADIOM construit par Bertin Technologies (voir Figure 4.5). La soustraction de fond temporelle est associée à une régularisation spatiale morphologique (filtre alterné séquentiel), et l'information extraite de la rétine est l'ensemble des coordonnées de boîtes englobantes des objets mobiles détectés. Ce calcul est fait par interaction entre la rétine et son séquenceur [2] : nous avons adapté un algorithme d'étiquetage de composantes connexes séquentiel en une passe [115] en l'optimisant à la lecture parallèle d'un bloc de pixels en sortie de rétine et au codage par plages (Stage de Master de Yahya Ould Isselmou, avec des contributions de Nicolas Burrus). Les coordonnées de boîtes englobantes sont ensuite transmises au processeur hôte [3], qui se charge de l'association de données et du pistage (Traitements réalisés par Bertin Technologies).

2. Processeur Nios, cœur IP synthétisable sur FPGA de la société *Altera*.
3. Processeur ARM, cœur hardware dans le circuit *Excalibur* d'*Altera*.

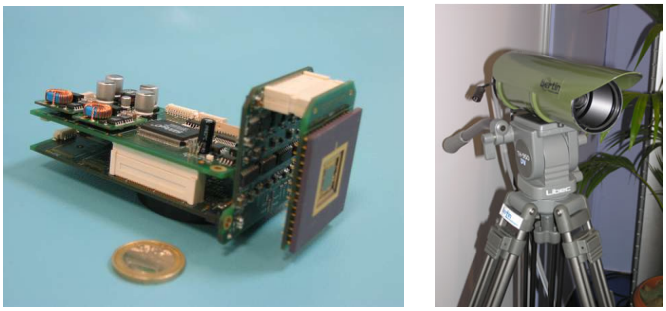


FIGURE 4.5 – Le capteur de réveil du CALADIOM nu et habillé, par Bertin Technologies.

Au delà des rétines, l'arithmétique fixe permet aussi de tirer le meilleur parti du parallélisme vectoriel dit SWAR (SIMD within a register [45]). Ce parallélisme interne au processeur, qui existe sur les processeurs grand public Intel, AMD ou PowerPC depuis le milieu des années 90, a été développé pour les besoins des applications multimédias. Il consiste en l'application d'un jeu d'instructions vectorielles appliquées sur des registres de grande taille segmentés, par exemple un registre de 128 bits peut représenter seize entiers de 8 bits, huit entiers de 16 bits, etc, de telle sorte qu'une seule opération sur un grand registre traite autant de données en parallèle (Voir un exemple sur la figure 4.6). Nous avons ainsi réalisé une implantation SWAR d'un détecteur de visage combinant morphologie et soustraction de fond statique [136] sur processeur G5 avec le jeu d'instructions *Altivec* [35] (Projet de Recherche d'Olivier Vermeulen). Nous avons également implanté la détection de mouvement par estimation zipfienne [82] sur processeur Intel avec les instructions vectorielles SSE2 [26].

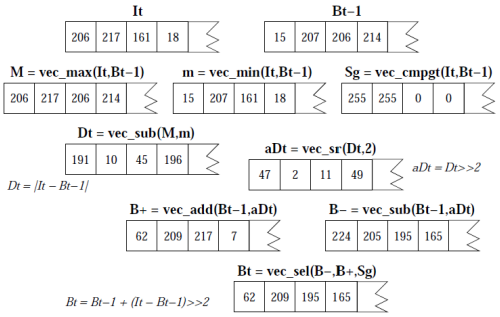


FIGURE 4.6 – Estimation vectorielle de fond par moyenne récursive.

4.3 Dans la jungle des COTS

Devoir prendre de la distance par rapport aux rétines pour s'investir plus dans d'autres architectures n'est pas sans douleur, comme un passage abrupt de l'ascèse à la débauche. Plus sérieusement, il n'est pas simple lorsqu'on a compté les portes logiques sur ses doigts d'admettre certaines incongruités dues à l'épaisseur des couches qui séparent le programmeur du silicium, comme le fait de passer des entiers en flottants pour accélérer des calculs (soupirs). Cette prise de distance s'est révélée nécessaire du fait de la multiplication des architectures adaptées à la vision haute performance, rivalisant favorablement avec les rétines sur plusieurs aspects. Les arguments les plus en défaveur des rétines actuelles sont (1) le manque de souplesse du parallélisme SIMD synchrone, qui cantonne l'utilisation des rétines aux traitements de bas niveau, et (2) la difficulté de programmation, due au jeu d'instruction minimal et à la limitation de la mémoire.

Nous avons été aidés dans cette démarche par le succès des algorithmes conçus pour la rétine, qui a conduit à les confronter à de multiples architectures. C'est le cas pour l'estimation Σ-Δ, en particulier grâce à la collaboration avec Lionel Lacassagne de l'IEF (Université Paris Sud). Outre le SWAR [82] déjà mentionné, l'implantation rétinienne a été comparée [66] avec les implantations sur PowerPC G4 et sur la Maille Associative d'Orsay [96]. Toujours à l'IEF, une implantation sur multi-cœurs avec OpenMP a été réalisée par Franck Bimbard, et comparée avec une autre implantation SWAR sur SSE2 [16]. A l'Université de Séville, Sergio Toral, Manuel Vargas et Federico Barrero ont implanté une adaptation de Σ-Δ sur une architecture RISC [130] (cœur ARM sur circuit Freescale IMX), et aussi sur FPGA [58], pour des applications véhiculaires embarquées. Abutaleb *et al* [1] ont implanté une version multimodale de Σ-Δ sur FPGA. Une version améliorée de Σ-Δ avec mécanismes d'adaptation globale, a été implantée par Verdant *et al* [135] sur un système de vision basse puissance à base de caméra CMOS pourvue d'unité de calcul analogique. Enfin, Ye *et al* [142] ont étudié et comparé des implantations de Σ-Δ sur FPGA et sur ASIC, réalisées par synthèse de haut niveau (HLS).

En fin de compte la diversification des plateformes et leur caractère de plus en plus hétérogène est une chance pour un algorithmicien qui peut expérimenter et inventer en multipliant les structures de données et les modèles algorithmiques. Nous nous sommes donc plus investis dans l'algorithmique pure, ce qui nous a permis de pratiquer une large gamme de modèles et outils : **Files d'attente :** [137] Squelettes, étiquetage en composantes connexes, reconstructions géodésiques... **Calculs**

récursifs : Calcul des dérivées multiéchelles [134], transformées en distance [70], opérateurs morphologiques [133], images intégrales [30], étiquetages en composantes connexes [115]... **Calculs pyramidaux :** [61] Représentations, poursuite semi-dense... **Méthodes cumulatives :** Transformées de Hough denses, poursuite semi-dense,... **Codage par plages :** Etiquetage en composantes connexes [67]... **Kd-trees :** Traitements locaux dans les espaces de caractéristiques, classification... **Listes dynamiques :** Modèles implicites de Formes...

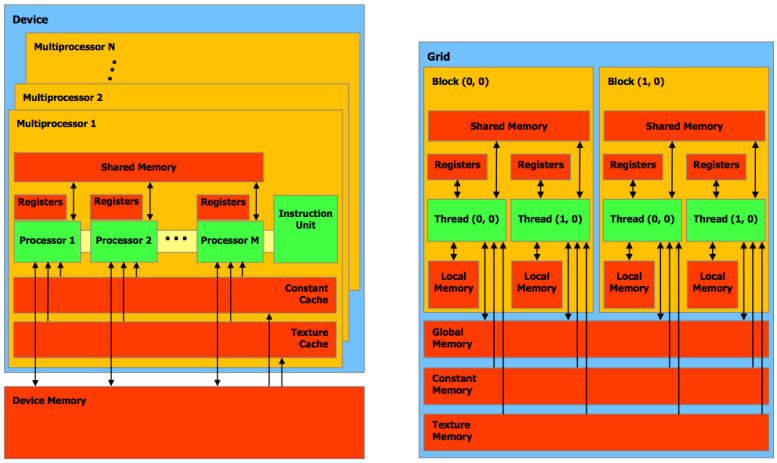


FIGURE 4.7 – L'architecture GPGPU CUDA, tiré de [27].

Une exploration des COTS pour la vision HPC ne peut ignorer les possibilités offertes aujourd'hui par les processeurs graphiques (GPU). S'il y a plus de 30 ans que les GPU déchargent les unités de calcul principales (CPU) des opérations massives liées à l'affichage, ce n'est qu'à partir des années 90 que leurs fonctionnalités s'enrichissent progressivement à mesure qu'ils prennent en charge des niveaux plus élevés du pipe-line graphique : rendu, projection, éclairage, géométrie 3d... Enfin, au début des années 2000, les GPU deviennent *programmables*. Pour autant, l'utilisation de cet immense potentiel de calcul en dehors des spécialistes du graphisme 3d devra attendre le développement d'un cadre d'abstraction suffisamment puissant. C'est le rôle de CUDA [27], qui permet aujourd'hui d'exploiter les GPU pour des tâches diverses (GPGPU). Le modèle de calcul générique d'un GPGPU CUDA (voir Figure 4.7) est organisé de façon hiérarchique. Un ensemble de multiprocesseurs (MP) exécutent des blocs de threads. Un thread est une tâche de granularité minimum. Dans un même MP, plusieurs threads d'un bloc sont exécutés en SIMD, avec une mémoire locale à chaque

thread (Local memory). Tous les threads d'un bloc sont exécutés dans le même MP, et partagent une mémoire locale au bloc (Shared memory). Enfin une mémoire globale au GPU peut échanger des données avec la RAM via le CPU. A ces différents niveaux de mémoire dont la vitesse d'accès est très variable s'ajoutent plusieurs niveaux de mémoire cache.

Les implantations sur GPGPU ont occupé une place importante dans les travaux de Matthieu Garrigues. Un premier résultat significatif a été l'implantation temps réel du flux optique dense dans l'espace des caractéristiques (cf Chapitre précédent), en optimisant le descripteur (vecteur de local jet) pour limiter sa longueur à 128 bits dans le but de restreindre la bande passante mémoire nécessaire aux traitements, et en réalisant une recherche du plus proche voisin limitée à un voisinage spatial. Les performances obtenues pour un flux dense sur une vidéo de taille 240x180 étaient de 33 ms par trame sur la carte graphique NVIDIA GeForce GTX 280.

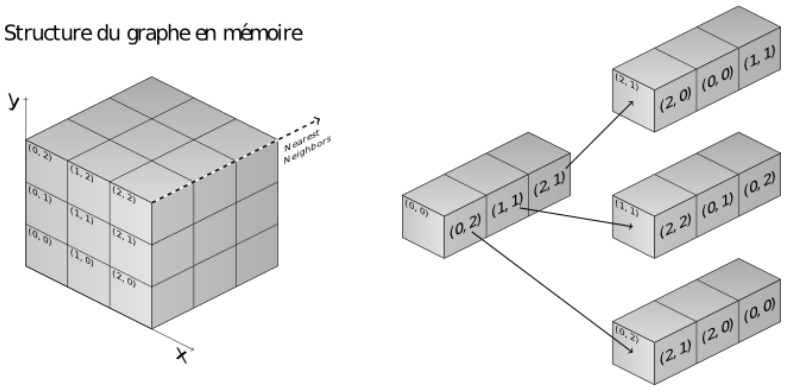


FIGURE 4.8 – Structure et principe de calcul des plus proches voisins (PPV) approximés sur le GPU (les valeurs indiquées correspondent aux coordonnées initiales des PPV temporaires).

Plus généralement, pour accélérer les traitements dans les espaces de caractéristiques, Matthieu Garrigues a expérimenté une alternative originale aux kd-trees pour calculer les plus proches voisins (approximés) en parallèle. Cette technique consiste à travailler sur un volume de données contenant pour chaque pixel les coordonnées de ses N plus proches voisins provisoires. Le principe itératif consiste à examiner pour chaque pixel les N^2 plus proches voisins des plus proches voisins, et de ne retenir que les N plus proches. La figure 4.8 illustre le principe du calcul. Cette technique, qui s'apparente à un algorithme de Dijkstra dans l'espace des caractéristiques, se prête à une parallélisation mas-

sive et permet de calculer des plus proches voisins à une distance dans l'espace image qui augmente exponentiellement avec le nombre d'itérations.

Pour le calcul temps réel du flux optique, tant que le nombre total de cœurs du GPU reste de plusieurs ordres inférieur au nombre de pixels, la limitation du calcul aux points (particules) les plus fiables représentera un gain en calcul considérable. C'est ce qui a motivé le développement de la poursuite semi-dense de Matthieu Garrigues présentée dans le chapitre précédent. Outre un certain niveau de sélection *a priori* des points par la fonction de saillance, d'autres optimisations déterminantes ont été réalisées : calcul multiéchelle pyramidale de la sélection des particules et des descripteurs associés, compression de la structure de donnée creuse codant la liste des particules en conservant la contiguïté en mémoire des particules proches spatialement [51]. Ces accélérations, qui ont permis d'atteindre une fréquence de calcul d'environ 200 images par secondes sur GPU 336-core (NIDIA GeForce 460 GTX) pour poursuivre 10 000 particules dans une vidéo 640x480, ne sont pas dans leur principe spécifiques au GPU : elles ont aussi permis d'atteindre une fréquence de 30 images par secondes sur CPU quadcore (Intel i5-2500k), et une implantation est actuellement en cours sur une plateforme à base de processeur ARM multi cœurs.

En conclusion de ce chapitre, nous aimerions souligner qu'une ambition de nos recherches est aussi d'exercer une influence sur les tendances architecturales des futures systèmes de vision. C'est l'esprit de l'étude comparative d'architectures très différentes faite en collaboration avec l'IEF [66], de l'architecture logicielle fondée sur les espaces de caractéristiques [84], ou de l'algorithme de médian grain fin distribué ([50], travaux de Matthieu Garrigues). Dans le même esprit, nous aimerions contribuer à la conception de rétines programmables de nouvelle génération, où les limitations au bas niveau des rétines actuelles seraient dépassées grâce à l'existence de primitives de calcul asynchrone [96]. Cette réflexion n'est pas nouvelle, et nous avions déjà perçu depuis longtemps l'intérêt fondamental [79] des connexions programmables associées à des mécanismes de propagation binaire. La thèse de Valentin Gies a montré qu'un certain nombre de primitives du calcul asynchrone pouvaient être implanté à un coût matériel très limité [52]. L'installation d'un arbre recouvrant ou le calcul de la somme sur un ensemble de pixels connexes sont deux exemples très importants. Toutefois, un vrai bilan algorithmique reste à faire qui contribuera à définir le jeu de primitives adapté au niveau d'analyse qu'on souhaite atteindre pour les prochains systèmes de vision sur puce.

Chapitre 5

Projet de Recherche

Le refus du choix entre polyvalence et efficacité est un parti pris primordial dans nos recherches. Le cadre universel de traitement qu'on cherche à obtenir est fondé sur des primitives fondamentales conçues et combinées dans un objectif de mutualisation et de minimisation des ressources de calcul. Réciproquement, l'implantation d'un algorithme sur une architecture embarquée n'est pas assimilée à une pure simplification, mais à une redéfinition des primitives visant à maximiser la généricité et l'évolutivité du système. Cette conception reste un élément fort de notre projet de recherche pour les années à venir. Le premier axe, détaillé dans la section 5.1 concerne l'élaboration d'un modèle unifié de représentation et traitement des vidéos qui soit en phase avec les architectures de calcul haute performance actuelles et à venir. La démarche scientifique envisagée est fondée sur deux éléments majeurs : (1) l'exploration algorithmique des modèles logiciels et matériels, et (2) l'exploitation de la continuité temporelle.

Plus un système est polyvalent, plus il doit intégrer de niveaux sémantiques différents dans l'information qu'il traite. Or le fossé sémantique de la perception artificielle est aussi un fossé calculatoire : les modèles, les algorithmes et les circuits qui les manipulent changent souvent radicalement quand on passe du bas vers le haut niveau. Dans notre volonté d'unification nous souhaitons remettre en question cette dichotomie en apportant plus de continuité dans le processus d'émergence de la signification. Le deuxième axe du projet de recherche, présenté dans la section 5.2 est donc lié à la volonté d'aller vers une analyse de plus haut niveau sémantique. Le mouvement et la géométrie 3d sont les éléments clefs des méthodes que nous souhaitons développer dans cette thématique.

5.1 Modèles, Architectures, Algorithmes

L'objectif global du premier axe est la proposition d'un cadre générique et évolutif, conçu comme un socle logiciel pour une large variété d'applications en vidéo embarquée. Nous allons poursuivre l'exploration des modèles, des algorithmes, des structures de données et des architectures existantes et émergentes, dans le but de dégager les primitives essentielles, et avec l'ambition d'influencer la conception logicielle et matérielle des systèmes de vision à venir.

Un exemple typique de cette démarche est le bilan algorithmique que nous envisageons dans le cadre de notre participation à la prochaine Convention DGA 2012-2015 «Calcul régional asynchrone pour vision embarquée faible consommation», dont l'objectif est de dégager les principes architecturaux essentiels à la conception de systèmes de vision sur puce aussi efficaces dans les traitements irréguliers (régionaux) que dans les traitements réguliers (locaux). Ce bilan est fondé sur la généralisation et l'abstraction de principes algorithmiques utilisés dans différentes applications où le niveau de traitement irrégulier régional est important : la segmentation, la poursuite, la détection d'objets. La démarche envisagée est d'extraire un jeu de primitives algorithmiques fondamentales à partir de l'étude comparée de différentes familles d'algorithmes couramment employés pour ces tâches : les fonctionnelles de Mumford-Shah [54], les graph cuts [123], les lignes de partage de eaux [28], et les forêts recouvrantes [42].

Dans la suite de nos travaux sur les espaces de caractéristiques [84], nous souhaitons développer de nouveaux types de descripteurs fondés sur une analyse géométrique de l'ensemble formé dans l'espace des caractéristiques (variété) par une catégorie visuelle : objet, pièce, contexte... Nous considérons ce type d'analyse, dont une version embryonaire a été présentée en fin du chapitre 2, comme un élément clef de la continuité calculatoire que nous recherchons pour l'augmentation du niveau sémantique.

Dans un registre très proche, nous souhaitons également exploiter la continuité temporelle pour construire de nouveaux descripteurs vidéos, en intégrant des informations d'apparence et de contexte le long de trajectoires obtenues par poursuite semi-dense [51]. Cette voie nous semble aussi un élément important dans la démarche d'optimisation globale des systèmes mobiles.

A plus long terme, nous souhaiterions aller plus loin dans cet axe en munissant le cadre générique que nous aurons construit de mécanismes d'adaptation automatique où la sélection des primitives et l'architecture même du système évolueraient en fonction de la tâche ou du

contexte. Ces idées de systèmes modulaires auto-adaptatifs ont déjà été partiellement élaborées dans nos travaux, en particulier dans les thèses de Renaud Barate [9] et de Christine Dubreu [36], et nous aimerions naturellement les mettre en œuvre dans un cadre plus universel.

Cette volonté d'unification, cette démarche d'exploration, et - avouons le - ce goût marqué pour l'éclectisme, ont un prix à payer : c'est le renoncement au statut de «spécialiste» pour toutes les notions du domaine qu'on tente de couvrir ; c'est un handicap certain dans la compétition au meilleur algorithme pour une application donnée, qui est une tendance forte (et tout à fait vertueuse à plusieurs égards) de l'évaluation des systèmes ; c'est enfin une certaine dilution de la visibilité scientifique. De ce point de vue, le succès des squelettes MB et de l'estimation Σ-Δ dans leurs communautés respectives (d'intersection vide est-il besoin de le préciser) sont des exemples à méditer : il pourra être judicieux de faire parfois certains compromis en sacrifiant temporairement à la monomanie pour soigner nos indicateurs...

5.2 Mouvement, Géométrie, Sémantique

Le deuxième axe est, on l'aura compris, intimement lié au premier : la volonté de généralisation des concepts et de rationalisation des primitives s'applique aussi aux différents niveaux conceptuels des systèmes, et le principe de continuité sémantique est au cœur de la démarche exposée dans la section précédente. Mais l'objectif d'aller vers un plus haut niveau sémantique dans nos recherches est aussi lié aux évolutions des enjeux et des contextes applicatifs de nos projets, vers le haut niveau : la compréhension de scènes et d'activités complexes.

Il s'agit d'abord du projet ITEA2 SPY - déjà en cours - sur les systèmes de vidéosurveillance mobile urbaine, où l'un de nos objectifs est la reconnaissance d'une activité suspicieuse ou dangereuse. Une partie des travaux de la thèse en cours de Fabio Martínez s'inscrit déjà dans ce cadre (voir Chapitre 3). Nous souhaitons aller plus loin en explorant une démarche différente, fondée sur la construction de descripteurs d'activités à partir des faisceaux de particules fournis par la poursuite semi dense (Post-Doc de Phuong Tranh Nguyen).

Toujours dans le projet SPY, un autre objectif majeur est de construire à partir de la vidéo capturée par le véhicule mobile un compte-rendu de la situation sous la forme d'un résumé à la fois géométrique et sémantique de la scène dynamique : (1) la géométrie 3d de la scène urbaine sera schématisée en plans principaux (maquette 3d) : routes, façades, obstacles, etc, par des techniques de reconstruction 3d par le mouvement (structure from motion, ou SfM), simplifiées grâce aux

hypothèses réduites de modèles planaires [17], et (2) des informations de niveaux sémantiques variables seront incrustées dans la maquette 3d : objets en mouvement, direction du focus, objets ou personnes reconnus, etc (Travaux de Matthieu Garrigues, et Post-Doc de Taha Ridene).

Cette combinaison des informations géométriques et sémantiques où le mouvement joue un rôle central sera aussi exploitée dans un autre volet de la Convention DGA 2012-2015 auquel nous allons participer : «Fusion de caméras couleur et profondeur pour l'analyse sémantique de scènes 3d». Ce projet vise à concevoir de nouvelles méthodes d'analyse sémantique de scènes combinant apparence visuelle et profondeur fournies par caméra RGB-D, adaptées à une implantation embarquée en environnements intérieur et extérieur urbain. Outre les problématiques liées à la fusion des représentations profondeur et apparence, et à la combinaison des techniques d'acquisition de la profondeur par RGB-D et par SfM, ce projet nous permettra d'expérimenter et de maîtriser la reconnaissance d'objets 3d [138, 6], qui pourrait, via leur interprétation dans le cadre SfM pur, constituer une des clefs des nouveaux descripteurs vidéos par intégration temporelle évoqués dans la section précédente.

Un avantage de la maquette 3d est de fournir un support explicite à la modélisation du contexte [100, 97]. La position du plan et son apparence visuelle pourront être exploitées en tant que fond (background) contextuel, dans l'esprit du modèle Things and Stuff de Heitz et Koller [56], pour améliorer la reconnaissance des objets. Nous envisageons d'introduire cette information sous la forme de paramètres contextuels dans une transformée de Hough dense généralisée (projet *Digiteo* avec les Mines (CAOR), déposé en 2012, classé en liste complémentaire).

Bibliographie

[1] M.M. Abutaleb, A. Hamdy, M.E. Abuelwafa, and E.M. Saad. FPGA-based object-extraction based on multimodal Σ-Δ background estimation. In *2nd Int. Conf. on Computer, Control and Communication (IC4'09)*, volume 2, pages 1–7. IEEE Computer Society, 2009.

[2] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2) :284–299, february 1985.

[3] J.K. Aggarwal and M.S Ryoo. Human activity analysis : A review. *ACM Computing Surveys*, 43 :1–43, 2011.

[4] John E. Albus, Lloyd J. Lewins, and Julie R. Schacht. Centroid tracking using a probability map for target segmentation. In M.K. Masten and L.A. Stockum, editors, *SPIE Conf. on Acquisition, Tracking, and Pointing XVI*, volume 4714, pages 175–185, Orlando - FL, april 2002.

[5] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. of the ACM*, 45(6) :891–923, 1998.

[6] B. B. Drost, M. Ulrich, N. Navab, and S Ilic. Model globally, match locally : Efficient and robust 3d object recognition. In *Computer Vision and Pattern Recognition (CVPR'10)*, pages 998–1005, 2010.

[7] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra. Effective codebooks for human action categorization. In *Proc. of ICCV International Workshop on Video-oriented Object and Event Classification (VOEC)*, Kyoto, Japan, September 2009. IEEE Computer Society.

[8] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2) :111–122, 1981.

[9] Renaud Barate. *Apprentissage de fonctions visuelles pour un robot mobile par programmation génétique*. PhD thesis, Université

Pierre et Marie Curie, Paris 6, Paris, France, 2008. Spécialité Informatique, Télécommunications et Electronique.

[10] Renaud Barate and Antoine Manzanera. Automatic design of vision-based obstacle avoidance controllers using genetic programming. In *8th International Conference on Artificial Evolution (EA'07)*, volume 4926, pages 25–36, Tours, France, oct. 2007. Lecture Notes in Computer Science - Springer Verlag.

[11] Renaud Barate and Antoine Manzanera. Evolution of visual controllers for obstacle avoidance in mobile robotics. *Evolutionary Intelligence*, 2(3) :85–102, 2009.

[12] O. Barnich and M. Van Droogenbroeck. ViBe : a powerful random technique to estimate the background in video sequences. In *Proc. ICASSP*, pages 945–948. IEEE, April 2009.

[13] T. Bernard. From sigma-delta modulation to digital halftoning of images. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 2805–2808, Toronto, Ontario, May 1991.

[14] Thierry M. Bernard, Bertrand Y. Zavidovique, and Francis Devos. A programmable Artificial Retina. *IEEE Journal of Solid-state Circuits*, 28-7 :789–798, 1993.

[15] Gilles Bertrand. On P-simple points. *Compte Rendus à l'Académie des Sciences*, 321-1 :1077–1084, 1995.

[16] Franck Bimbard. Improvement in motion-detection algorithms for real-time processing by using the OpenMP library and/or SIMD instructions. In *International Conference on Control, Communication and Computer (CCCT'2011)*, Orlando, FL, March 2011.

[17] Samia Bouchafa and Bertrand Zavidovique. c-velocity : A flow-cumulating uncalibrated approach for 3d plane detection. *Int. J. Comput. Vision*, 97(2) :148–166, April 2012.

[18] Patrick Bouthémy and Patrick Lalande. Recovery of moving object in an image sequence using local spatiotemporal contextual information. *Optical Engineering*, 32-6 :1205–1212, 1993.

[19] Jack E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1) :25–30, january 1965.

[20] A. Buades, B. Coll, and JM. Morel. A non-local algorithm for image denoising. In *Proc. CVPR*, volume 2, pages 60–65, 2005.

[21] Prabir Burman and Wolfgang Polonik. Multivariate mode hunting : Data analytic tools with measures of significance. *J. Multivar. Anal.*, 100(6) :1198–1218, 2009.

[22] Alice Caplier, Christophe Dumontier, Franck Luthon, and Pierre-Yves Coulon. Algorithme de détection de mouvement par modélisation markovienne. Mise en œuvre sur DSP. *Traitement du signal*, 13-2 :175–190, 1996.

[23] Fabio Martínez Carrillo, Antoine Manzanera, and Eduardo Romero Castro. A motion descriptor based on statistics of optical flow orientations for action classification in video-surveillance. In *Accepted to : Int. Conf. on Multimedia and Signal Processing (CMSP'12)*, Shanghai, China, december 2012.

[24] T. H. Chalidabhongse, K. Kim, D. Harwood, and L.S. Davis. A perturbation method for evaluating background subtraction algorithms. In *Proc. Joint IEEE Int. Work. on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Nice - France, 2003.

[25] R. Chaudhry, A. Ravichandran, G.D. Hager, and R. Vidal. Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition (CVPR'09)*, pages 1932–1939. IEEE, 2009.

[26] Collective. *Intel®C++ Compiler for Linux Systems - User's Guide*. Intel Corporation, 1996-2003. Document number : 253254-014.

[27] Collective. *CUDA Programming Guide*. NVIDIA, version 4.2 edition, April 2012. `http://developer.nvidia.com/cuda/nvidia-gpu-computing-documentation`.

[28] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts : minimum spanning forests and the drop of water principle. *EEE Trans. on Pattern Analysis and Machine Intelligence*, 31(8) :1362–1374, 2009.

[29] M. Crosier and L.D. Griffin. Using basic image features for texture classification. *Int. J. of Computer Vision*, 88(3) :447–460, 2010.

[30] Franklin C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Comput. Graph.*, 18(3) :207–212, January 1984.

[31] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision (ECCV'04)*, pages 1–22, 2004.

[32] L. Da Fontoura Costa. Robust skeletonization through exact Euclidean distance transform and its application to neuromorphometry. *Real-Time Imaging*, 6(6) :415–431, 2000.

[33] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[34] P.-E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14 :227–248, 1980.

[35] Keith Diefendorff, Pradeep K. Dubey, Ron Hochsprung, and Hunter Scales. Altivec extension to PowerPC accelerates media processing. *IEEE Micro*, 20(2) :85–95, March 2000.

[36] Christine Dubreu. *Algorithmique de traitement d'image des systèmes de surveillance infrarouges air-sol*. PhD thesis, Université de Bourgogne, Paris, France, 2009. Spécialité Instrumentation et Informatique de l'Image.

[37] Christine Dubreu, Antoine Manzanera, and Eric Bohain. Comprehensive evaluation of tracking systems by non-photorealistic simulation. In *Proceedings of SPIE - Acquisition, Tracking, Pointing, and Laser Systems Technologies XXI*, volume 6569, Orlando - FL, USA, may. 2007.

[38] Christine Dubreu, Antoine Manzanera, and Eric Bohain. Simulation of video sequences for an accurate evaluation of tracking algorithms on complex scenes. In *Proceedings of SPIE - Acquisition, Tracking, Pointing, and Laser Systems Technologies XXII*, volume 6971, Orlando - FL, USA, may. 2008.

[39] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Com. of the Association for Computing Machinery*, 15(1) :11–15, 1972.

[40] Gloria Mercedes Díaz and Antoine Manzanera. *Biomedical Image Analysis and Machine Learning Technologies : Applications and Techniques*, chapter VIII : Automatic Analysis of Microscopic Images in Hematological Cytology Applications. IGI Global, 2010.

[41] A. Elgammal, D. Harwood, and L. Davis. Non-parametric Model for Background Subtraction. In *Proc. IEEE ECCV*, Dublin - Ireland, 2000.

[42] A.X. Falcão, J. Stolfi, and R. de Alencar Lotufo. The image foresting transform : Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1) :19–29, january 2004.

[43] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87(3), 2010.

[44] D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2007.

[45] Randall J. Fisher, All J. Fisher, and Henry G. Dietz. Compiling for SIMD within a register. In *11th Annual Workshop on Languages and Compilers for Parallel Computing (LCPC'98)*, volume 1656 of *Lecture Notes in Computer Science*, pages 290–304. Springer Verlag, Chapel Hill, 1998.

[46] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1) :77–104, 1990.

[47] D. Florins and A. Manzanera. Detection of floating mines in infrared sequences by multiscale geometric filtering. In Broach J.T. and J. Holloway Jr, editors, *SPIE Conference in Defence Security and Sensing : Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*, volume 8357, Baltimore, MD, June 2012.

[48] R. Forchheimer and A. Astrøm. Near-sensor image processing : a new paradigm. *IEEE trans. on Image Processing*, 3 :736–746, 1994.

[49] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(9) :891–906, 1991.

[50] M. Garrigues and A. Manzanera. Exact and approximate median splitting on distributed memory machines. In *Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA 2012)*, Las Vegas, NV, july 2012.

[51] M. Garrigues and A. Manzanera. Real time semi-dense point tracking. In A.J.C. Campilho and M.S. Kamel, editors, *Int. Conf. on Image Analysis and Recognition (ICIAR 2012)*, volume 7324 of *Lecture Notes in Computer Science*, pages 245–252, Aveiro, Portugal, june 2012. Springer.

[52] V. Gies, T.M. Bernard, and A. Mérigot. Asynchronous regional computation capabilities for digital retinas. In *Computer Architecture for Machine Perception and Sensing (CAMPS'06)*, September 2006.

[53] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(12) :2247–2253, 2007.

[54] L. Grady and C. Alvino. The piecewise smooth Mumford-Shah functional on an arbitrary graph. *IEEE Trans. on Image Processing*, 18(11), november 2009.

[55] Philippe Guermeur and Antoine Manzanera. Image characterization from statistical reduction of local patterns. In *Progress in Pattern Recognition, Image Analysis and Applications (CIARP'09)*, volume 5856, pages 571–578, Guadalajara, Mexico, nov. 2009. Lecture Notes in Computer Science - Springer Verlag.

[56] Geremy Heitz and Daphne Koller. Learning spatial context : Using stuff to find things. In *European Conference on Computer Vision (ECCV'08)*, 2008.

[57] C.J. Hilditch. Linear skeletons from square cupboards. *Machine Intelligence*, 4 :403–420, 1969.

[58] J. Hiraiwa, E. Vargas, and S. Toral. An FPGA based embedded vision system for real-time motion segmentation. In *17th Int. Conf. on Systems, Signals and Image Processing (IWSSIP'10)*, pages 360–363, 2010.

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

[60] P.V.C. Hough. Machine analysis of bubble chamber pictures. In *Int. Conf. High Energy Accelerators and Instrumentation*, 1959.

[61] Jean-Michel Jolion and Azriel Rosenfeld. *A Pyramid Framework for Early Vision : Multiresolutional Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[62] K.-P. Karmann and A. von Brandt. *Time-Varying Image Processing and Moving Object Recognition*, chapter Moving Object Recognition Using an Adaptive Background Memory. Elsevier, 1990.

[63] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *International Conference on Computer Vision*, volume 1, pages 166 – 173, October 2005.

[64] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Background modeling and subtraction by codebook construction. In *Proc. ICIP*, volume 5, pages 3061–3064. IEEE, 2004.

[65] J.J. Koenderink and A.J. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55 :367–375, 1987.

[66] Lionel Lacassagne, Antoine Manzanera, Julien Denoulet, and Alain Mérigot. High performance motion detection : some trends

toward new embedded architectures for vision systems. *Journal of Real Time Image Processing*, 4(2) :127–146, 2009.

[67] Lionel Lacassagne and Bertrand Zavidovique. Light speed labeling : efficient connected component labeling on RISC architectures. *J. Real-Time Image Processing*, 6(2) :117–135, 2011.

[68] B.C. Lee and M. Hedley. Background estimation for video surveillance. In *Proc. IVCNZ'02*, pages 315–320, 2002.

[69] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[70] F. Leymarie and M. D. Levine. Fast raster scan distance propagation on the discrete rectangular lattice. *Computer Vision and Image Understanding*, 55(1), 1992.

[71] T. Lindeberg. Feature detection with automatic scale selection. *Int. J. of Computer Vision*, 30(2) :77–116, 1998.

[72] T. Lindeberg and B. ter Haar Romeny. *Geometry-Driven Diffusion in Computer Vision*, chapter Linear scale-space : I. Basic theory, II. Early visual operations, pages 1–77. Series in Mathematical Imaging and Vision. Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

[73] Christophe Lohou. *Etude d'algorithmes de squelettisation pour images 2D et 3D selon une approche topologie digitale ou topologie discrète*. PhD thesis, Université de Marne-la-Vallée, France, 2001.

[74] Tobias Low and Antoine Manzanera. Ground-plane classification for robot navigation. In *International Conference on Control, Automation, Robotics and Vision (ICARCV'10)*, Singapore, dec. 2010. IEEE Computer Society.

[75] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 60(2) :91–110, 2004.

[76] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA Image Understanding Workshop*, pages 121–130, 1981.

[77] Cherng Min Ma. On topology preservation in 3d thinning. *CVGIP : Image Understanding*, 59-3 :328–339, 1994.

[78] Antoine Manzanera. Algorithmique de traitement d'images pour la détection des cibles - système antichar ERYX. Technical Report 20005137, MBDA-F, jan. 2002.

[79] Antoine Manzanera. Morphological segmentation on the programmable retina : towards mixed synchronous/asynchronous al-

gorithms. In *6th International Symposium on Mathematical Morphology (ISMM'02)*, pages 25–37, Sydney - Australia, apr. 2002. CSIRO.

[80] Antoine Manzanera. Conception et validation des algorithmes de détection pour le capteur de réveil CALADIOM. Technical report, Bertin Technologie, dec. 2003.

[81] Antoine Manzanera. Implantation des algorithmes de détection sur rétine programmable pour le capteur de réveil CALADIOM. Technical report, Bertin Technologie, dec. 2004.

[82] Antoine Manzanera. Sigma-Delta background subtraction and the Zipf law. In *Progress in Pattern Recognition, Image Analysis and Applications (CIARP'07)*, volume 4756, pages 42–51, Viña del Mar-Valparaíso, Chile, nov. 2007. Lecture Notes in Computer Science - Springer Verlag.

[83] Antoine Manzanera. Image representation and processing through multiscale local jet features. Technical report, ENSTA/LEI, 2010.

[84] Antoine Manzanera. Local jet feature space framework for image processing and representation. In *International Conference on Signal Image Technology and Internet Based Systems (SITIS'11)*, pages 261–268, Dijon, France, dec. 2011. IEEE Computer Society.

[85] Antoine Manzanera. Object modelling, detection and localization in mobile video : a state-of-the-art. Technical report, ITEA2, 2011.

[86] Antoine Manzanera. Dense Hough transforms on gray level images using multi-scale derivatives (invited conference). In *The sixth International Workshop on Medical and Healthcare applications(AMINA'12)*, Mahdia, Tunisia, december 2012.

[87] Antoine Manzanera and Thierry M. Bernard. MB : a coherent collection of 2d parallel thinning algorithms. Technical Report LEI/AVA-02-002, ENSTA/LEI, 2002.

[88] Antoine Manzanera and Thierry M. Bernard. Metrical properties of a collection of 2d parallel thinning algorithms. In *International Workshop on Combinatorial Image Analysis (IWCIA'03)*, volume 12 of *Electronic Notes on Discrete Mathematics*, Palermo - Italy, may. 2003. Elsevier Science.

[89] Antoine Manzanera, Thierry M. Bernard, Françoise Prêteux, and Bernard Longuet. nd skeletons : a unified mathematical framework. *Electronic Imaging*, 11(1) :25–37, 2002.

[90] Antoine Manzanera and Nicolas Burrus. Arithmétique et logique des rétines programmables. Technical report, ENSTA/LEI, 2006.

[91] Antoine Manzanera and Jean-Michel Jolion. La pyramide irrégulière : un modèle pour la vision exploratoire. *Traitement du signal*, 12(2) :176–196, 1995.

[92] Antoine Manzanera, Françoise Prêteux, and Thierry M. Bernard. Markovian modeling on programmable retina. In *Conference on Controle Quality by Artificial Vision (QCAV'01)*, pages 232–237, Le Creusot - France, may. 2001. Cépaduès-Editions.

[93] Antoine Manzanera and Julien Richefeu. A new motion detection algorithm based on Sigma-Delta background estimation. *Pattern Recognition Letters*, 28(3) :320–328, 2007.

[94] Antoine Manzanera and Julien C. 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, Kolkata - India, dec. 2004.

[95] N.J.B. McFarlane and C.P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8 :187–193, 1995.

[96] Alain Mérigot. Associative nets : A graph-based parallel computing model. *IEEE Trans. Comput.*, 46(5) :558–571, May 1997.

[97] B. Micusik and J. Kosecka. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *ICCV Workshop on Video-Oriented Object and Event Classification*. IEEE, 2009.

[98] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int. J. of Computer Vision*, 60(1) :63–86, 2004.

[99] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Proc. IEEE CVPR*, 2004.

[100] K. Murphy, A. Torralba, D. Eaton, and W. Freeman. Object detection and localization using local and global features. In *Toward Category-Level Object Recognition*, pages 382–400. IEEE, 2006.

[101] Paul Nadrag, Antoine Manzanera, and Nicolas Burrus. Smart retina as a contour-based visual interface. In *Distributed Smart Cameras Workshop (DSC'06)*, Boulder - CO, USA, oct. 2006. ACM.

[102] R.L. Ogniewicz and O. Kübler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28(3) :343–359, March 1995.

[103] F. O'Gorman and B. Clowes. Finding picture edges through collinearity of feature points. *IEEE Trans. on Computers*, C-25(4) :449–456, April 1976.

[104] N.M. Oliver, B. Rosario, and A.P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Trans. on PAMI*, 2000.

[105] F. Paillet, D. Mercier, and T.M. Bernard. Second generation programmable artificial retina. In *Proc. IEEE ASIC/SOC Conf.*, pages 304–309, September 1999.

[106] Gabriel Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2) :249–260, 2009.

[107] M. Piccardi. Background subtraction techniques : a review. In *Proc. of IEEE SMC/ICSMC*, October 2004.

[108] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28, 2010.

[109] P.W. Power and J.A. Schoonees. Understanding background mixture models for foreground segmentation. In *Proc. IVCNZ'02*, pages 267–271, November 2002.

[110] Julien Richefeu. *Détection et analyse du mouvement sur système de vision à base de rétine numérique.* PhD thesis, Université Pierre et Marie Curie, Paris 6, Paris, France, 2006. Spécialité Informatique, Télécommunications et Electronique.

[111] Julien C. Richefeu and Antoine Manzanera. A morphological dominant points detection and its cellular implementation. In *International Symposium on Signal Processing and its Applications (ISSPA'03)*, volume 2, pages 181–184, Paris - France, jul. 2003. IEEE.

[112] Julien C. Richefeu and Antoine Manzanera. A new hybrid differential filter for motion detection. In *International Conference on Computer Vision and Graphics (ICCVG'04)*, Warsaw - Poland, sep 2004.

[113] Julien C. Richefeu and Antoine Manzanera. Détection de mouvement par capteur intelligent. In *ORASIS'05*, Clermont-Ferrand, France, may. 2005.

[114] Christian Ronse. Minimal test patterns for connectivity preservation in parallel thinning algorithms for binary digital images. *Discrete Applied Mathematics*, 21 :67–79, 1988.

[115] Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *J. ACM*, 13(4) :471–494, oct 1966.

[116] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV'06)*, volume 1, pages 430–443, may 2006.

[117] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. In *Proc. ICCV*, pages 1018–1024, Kerkyra, Greece, 1999.

[118] D. Rutovitz. Pattern Recognition. *J.R. Statist. Soc.*, 129 :504–530, 1966.

[119] P. Sand and S. Teller. Particle video : Long-range motion estimation using point trajectories. In *Computer Vision and Pattern Recognition (CVPR'06)*, pages 2195–2202, New York, june 2006.

[120] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation : learning from bees. In *Int. Conf. on Computer Vision and Pattern Recognition*, pages 434–439. IEEE Computer Society, 1993.

[121] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5) :530–534, 1997.

[122] S.D. Shapiro. Feature space transforms for curve detection. *Pattern Recognition*, 10(3) :129–143, 1978.

[123] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, august 2000.

[124] F.Y. Shih and Y-T Wu. Fast Euclidean distance transformation in two scans using a 3 x 3 neighborhood. *Computer Vision and Image Understanding*, 93(2) :195–205, 2004.

[125] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on PAMI*, 2000.

[126] R. Stefanelli and A. Rosenfeld. Some parallel thinning algorithms for digital pictures. *Journal of the A.C.M.*, 18 :255–264, 1971.

[127] Herb Sutter. The free lunch is over : A fundamental turn toward concurrency in software. *Dr. Dobb's Journal*, 30(3) :202–210, 2005.

[128] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, april 1991.

[129] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV 98)*, pages 839–846, Bombay, India, 1998. IEEE Computer Society.

[130] Sergio L. Toral, Manuel Vargas, and Federico Barrero. Embedded multimedia processors for road-traffic parameter estimation. *Computer*, 42 :61–68, 2009.

[131] K. Toyoma, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : Principles and Practice of Background Maintenance. In *Proc. IEEE ICCV*, pages 255–261, Kerkyra - Greece, 1999.

[132] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, Alaska, June 2008.

[133] M. van Herk. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels. *Pattern Recognition Letters*, 13 :517–521, 1992.

[134] LJ. van Vliet, IT. Young, and Verbeek PW. Recursive Gaussian derivative filters. In *Proc. ICPR*, volume 1, pages 509–514, 1998.

[135] Arnaud Verdant, Patrick Villard, Antoine Dupret, and Hervé Mathias. Three novell analog-domain algorithms for motion detection in video surveillance. *EURASIP Journal of Image and Video Processing*, 2011 :10 :1–10 :13, January 2011.

[136] Olivier Vermeulen, Antoine Manzanera, and Lionel Lacassagne. Ultra fast grey scale face detection using vector SIMD programming. In *3rd International Conference on Signal-Image Technology and Internet-based Systems (SITIS'07)*, Shanghai, China, dec. 2007. IEEE.

[137] Luc Vincent. *Algorithmes morphologiques à base de file d'attente et de lacets. Extension aux graphes.* PhD thesis, Ecole Nationale supérieure des Mines de Paris, May 1990.

[138] E. Wahl, U. Hillenbrand, and G. Hirzinger. Surflet-pair-relation histograms : a statistical 3d-shape representation for rapid classification. In *Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2003.

[139] H. Wang and D. Suter. A consensus-based method for tracking : Modelling background scenario and foreground appearance. *Pattern Recognition*, 40(3) :1091–1105, 2007.

[140] Li-Yi Wei. Texture synthesis from multiple sources. In *ACM SIGGRAPH 2003 Sketches & Applications*, SIGGRAPH '03, pages 1–1, New York, NY, USA, 2003. ACM.

[141] C.R. Wren, A. Azarbayejani, T. Darrell, and Alex Pentland. Pfinder : Real-time tracking of the human body. *IEEE Trans. on PAMI*, 1997.

[142] H. Ye, L. Lacassagne, D. Etiemble, Cabaret L., J. Falcou, A. Romero, and O. Florent. Impact of high-level transformers for high-level synthesis for motion detection algorithm. In *Conference on Design and Architectures for Signal and Image Processing (DASIP'2012)*, Karlsruhe, Germany, October 2012. ECSI.

[143] B. Zavidovique and G. Stamon. Bilevel processing of multilevel images. In *Pattern Recognition and Image Processing*, Dallas, TX, August 1981.

[144] G. Zhu, C. Xu, W. Gao, and Q. Huang. Action recognition in broadcast tennis video using optical flow and support vector machine. In *ECCV Workshop on HCI*, volume 3979 of *Lecture Notes in Computer Science*, pages 89–98. Springer, 2006.

[145] G.K Zipf. *Human behavior and the principle of least-effort.* Addison-Wesley, New-York, 1949.

Annexe A

Curriculum Vitae

Antoine Manzanera
Enseignant-Chercheur

Traitement d'Images
Vision par Ordinateur

ENSTA-ParisTech
32 Boulevard Victor
75739 Paris CEDEX 15

☎ +33 1 45 52 44 42
+ 33 6 28 20 99 46
Antoine.Manzanera@ensta.fr

1 Données biographiques et professionnelles

Poste occupé

Chercheur à l'ENSTA-ParisTech (Unité d'Informatique et Ingénierie des Systèmes) en Traitement d'images et Vision artificielle.

Enseignant à l'ENSTA-ParisTech (2^e et 3^e année du cycle Ingénieur) et aux Masters IAD et Imagerie de l'Université Pierre et Marie Curie (Paris 6).

Coordinateur de modules d'enseignement à l'ENSTA-ParisTech (2^e et 3^e années du cycle Ingénieur).

Correspondant Amérique Latine pour les Relations Internationales, responsable de l'admission sur titres des étudiants Brésiliens pour ENSTA-ParisTech.

Diplômes

2000 : **Doctorat** ès Sciences - spécialité Signal et Image
École Nationale Supérieure des Télécommunications - Paris
1993 : **DEA** d'Informatique Fondamentale
Université Claude Bernard - Lyon 1
1991 : **Licence** de Mathématiques
Université Claude Bernard - Lyon 1

Expérience professionnelle

2001- : **Enseignant-Chercheur** à l'Unité d'Informatique et d'Ingénierie des Systèmes de l'ENSTA-ParisTech
1997-00 : **Ingénieur Doctorant CIFRE** chez Aérospatiale-Matra Missiles à Chatillon (Hauts-de-Seine)
1994-96 : **Professeur** au Lycée (Mathématiques, Physique) et Collège (SVT, Technologie) Marcel Pagnol d'Asunción - Paraguay
1993 : **Enseignant vacataire** : Prépas scientifiques (Informatique) et B.E.P. Comptabilité (Maths financières) - Lyon

Formation par la recherche

Doctorat

Titre de la thèse : «Vision artificielle rétinienne»
Établissement : Télécom-ParisTech
Spécialité : Signal et Image
Soutenance : le 31 Août 2000
Financement : Convention CIFRE ANRT - Aérospatiale Missiles
Laboratoire d'accueil : Centre Technique d'Arcueil (DGA/DCE) - Laboratoire Géographie Imagerie Perception
Jury : Patrick GARDA (Président) - Gilles BERTRAND et Michel SCHMITT (rapporteurs) - Thierry BERNARD (Co-directeur de Thèse), Bernard LONGUET, Françoise PRÊTEUX (Co-directrice de Thèse), et Yves SOREL

DEA

Intitulé : Informatique Fondamentale
Établissement : ENS - Lyon
Année d'obtention : 1993
Responsable : Yves ROBERT
Parcours : Image et parallélisme
Stage de recherche : «Organisation rétinienne de données»
Directeur de stage : Jean-Michel JOLION - LISPI / UCB-Lyon 1

2 Activités professionnelles

Recherche

L'Unité d'Informatique et d'Ingénierie des Systèmes (UIIS) d'ENSTA-ParisTech regroupe les compétences de l'Ecole dans les Technologies de l'Information et de la Communication, et comprend les trois pôles «Sûreté des systèmes», «Ingénierie Système», et «Robotique et Vision».

Au sein de ce troisième pôle, mon thème de recherche «Algorithmique de Vision», se situe à l'interface des Systèmes embarqués et de la Robotique. Ma recherche vise à l'élaboration de nouveaux modèles et algorithmes de traitement d'images, avec des perspectives de temps réel et de minimisation de ressources.

Mes contributions scientifiques se sont développées dans divers domaines, notamment :

- **Vision précoce :**
  - Représentations visuelles
  - Extraction de caractéristiques
  - Filtrage et amélioration
- **Géométrie discrète :**
  - Squelettes multi-dimensionnels
  - Squelettes multi-échelles
  - Segmentation morphologique
- **Analyse du mouvement :**
  - Détection d'objet mobile
  - Flux optique et poursuite
  - Descripteurs d'activité
- **Calcul parallèle :**
  - Rétines programmables
  - SIMD vectoriel
  - XCore et GPGPU

Enseignement

Enseignements propres réguliers

Responsable du cours *Morphologie Mathématique* (ENSTA 2^e année, 21h dont 8h de TP)

Responsable du cours *Traitement d'Images et Vision* (ENSTA 3^e année, 42h dont 15h de TP)

Responsable de l'UE *Traitement et Reconnaissance d'Images* (M2 IAD UPMC, 21h)

Participant à l'UE *Représentations Discrètes et Morphologie Mathématique* (Resp. Isabelle Bloch, M2 Imagerie UPMC, env. 10h)

Participant à l'UE *Techniques du Traitement d'Images* (Resp. Florence Tupin, M2 Imagerie UPMC, env. 4h)

Participant à l'UE *Conférence Image* (Resp. Florence Tupin, M2 Imagerie UPMC, env. 3h)

Coordination des enseignements

Créateur et Coordinateur des Modules électifs *Imagerie* et *Perception visuelle* ENSTA 2^e année)

Coordinateur de la Filière *Multimédia et Communication* (avec Alain Sibille), **Créateur et responsable** des Modules *Information multimédia* et *Image* (ENSTA 3^e année, jusqu'en 2010)

Créateur et responsable des Modules *Perception et Interaction* et *Interaction du Véhicule avec l'Environnement* (ENSTA 3^e année, depuis 2010)

Coordinateur de la Semaine Européenne Athens *Imagerie médicale* (prof. Jean-Marie Rocchisani, Univ. Paris XIII)

Organisateur de la Semaine de Milieu *Multimedia* (visite d'entreprises et de laboratoires, jusqu'en 2006)

Autres activités liées à l'enseignement

Auteur du logiciel *Inti* dédié à l'apprentissage du Traitement d'images : Expérimentation et programmation en Morphologie Mathématique, Filtrage, Contours, Segmentation, Traitement dans le domaine fréquentiel (cf `http://www.ensta.fr/~manzaner/Support_Cours.html`)

Tutorat des élèves ENSTA : Choix de filières, de stages, aménagement de cursus (environ 6 élèves par an depuis 2001)

Recueil et Promotion des stages de recherche et stages ingénieurs pour les étudiants de l'ENSTA et des Masters IAD et IMA de Paris 6. (cf `http://www.ensta.fr/~manzaner/Stages_2012/`)

Jury de soutenance de PFE, Master, et Projet de Recherche (environ 10 élèves par an depuis 2002)

Jury d'admission sur titre à ENSTA-ParisTech : élèves français (2002-2005), puis brésiliens (depuis 2005)

Relations internationales

Langues Anglais, Espagnol et Portugais parlés couramment

depuis 2005 Membre du jury ParisTech Brésil, action mutualisée ParisTech de recrutement des élèves brésiliens en admission sur titre : Interface avec les Universités partenaires, sélection des dossiers de candidatures, mission annuelle de recrutement à Rio de Janeiro, São Paulo, Campinas, et Porto Alegre

depuis 2005 Correspondant *Amérique Latine* pour les relations internationales d'ENSTA-ParisTech

depuis 2006 Séjours en tant que professeur invité pour des écoles d'été, masters internationaux ou conférences : Roumanie (2006), Tunisie (2008 et 2010), Colombie (2008 et 2009), Mexique (2012)

depuis 2008 Convention de recherche avec l'Université Nationale de Colombie (UNAL-Bogotá). Co-encadrement d'étudiants en thèse de Master et Thèse de Doctorat en co-tutelle.

depuis 2012 Participation à un projet de Recherche Franco-Tunisien CMCU

Contrats de recherche

2001-2003 Programme d'Etude Amont PATRICIA (pilote DGA/SPMT, MO MBDA-F) : «Poursuite assisté par traitement d'images pour les missiles anti-char»

2003-2006 Programme d'Etude Amont CALADIOM (pilote DGA/SPART, MO Bertin Technologie) : «Détection d'objets mobiles pour capteur à longue autonomie par calculs sur rétine programmable»

2006-2009 Convention DGA/MRIS : «Apprentissage visuel pour la robotique»

2009-2012 Convention DGA/MRIS : «Système multisensoriel pour la navigation sémantique en robotique»

2011-2013 Projet EUREKA/ITEA2 SPY : (financement MEFI/DGCIS, pilote CASSIDIAN) «Surveillance imProved sYstem : Intelligent situation awareness»

2012-2014 Projet Franco-Tunisien CMCU : «Méthodologies d'optimisation algorithmique et architecturale de systèmes dédiés à des applications médicales»

2012-2015 Convention DGA/MRIS : «Fusion de caméra couleur et profondeur pour l'analyse sémantique de scène 3D»

2012-2015 Convention DGA/MRIS : «Calcul régional asynchrone pour vision embarquée faible consommation»

Relecture - Expertise - Animation scientifique

Relecteur régulier pour les journaux suivants : *Pattern Recognition Letters*, Journal of *Real-Time Image Processing*, Computer Methods in Biomechanics and Biomedical Engineering. Relectures occasionnelles : Journal of *Mathematical Imaging and Vision*, Journal of *Electronic Imaging*, EURASIP Journal on *Applied Signal Processing*, International Journal of *Image and Graphics*, IEEE Transactions on Broadcasting, IET-Intelligent Transportation Systems, IEEE Transactions on Systems Man and Cybernetics (part B),...

Membre du comité de lecture pour les conférences suivantes : *Advanced Concepts for Intelligent Vision Systems* (ACIVS), *Traitement et Analyse de l'Information : Méthodes et Applications* (TAIMA) *Chilean Workshop on Pattern Recognition* (CWPR), *Applications Médicales de l'Informatique : Nouvelles Approches* (AMINA), *International Seminar on Medical Information Processing and Analysis* (SIPAIM). Relectures occasionnelles : EuroGraphics, Graphic Interface, VECoS, MWSCAS, ICARCV, ICDSC, ITSC,...

Expert pour l'ANR : 8 projets expertisés depuis 2008. Expertises occasionnelles : IFREMER (1 Post-doc), Fondation EADS (1 projet), Institut Mines-Télécom (1 Thèse),...

Membre (SEE) du club scientifique «Systèmes Optroniques pour l'Observation et la Surveillance» (SOOS, club commun SEE/SFO) : Organisation de 2 journées scientifiques par an, réflexion sur la formation en optronique

Attributions diverses

Examinateur en Informatique pour le Concours Commun Agronomique et Vétérinaire, et concepteur d'exercices de l'épreuve orale (AgroParisTech, depuis 2007)

Concepteur de la partie *Informatique* du test écrit pour l'admission sur titres aux écoles de ParisTech (depuis 2007)

Membre du Conseil d'Administration de l'ENSTA (2004-2007), Membre suppléantdu Comité d'Hygiène et de Sécurité de l'ENSTA (depuis 2007).

Encadrement et jurys extérieurs

Encadrement

Encadrement de la Thèse de Julien Richefeu (UPMC, Dir. Jean Louchet) : «Détection et analyse du mouvement sur système de vision à base de rétine numérique» : Soutenue en Décembre 2006

Encadrement de la Thèse de Renaud Barate (UPMC, Dir. Jean Louchet) : «Apprentissage de fonctions visuelles pour un robot mobile par programmation génétique» : Soutenue en Novembre 2008

Encadrement de la Thèse de Christine Dubreu (U. de Bourgogne, Dir. Michel Paindavoine) : «Algorithmique de traitement d'images des systèmes de surveillance infra-rouge air-sol» : Soutenue en Juillet 2009

Encadrement de la Thèse de Fabio Martínez Carrillo (Co-tutelle UNAL-Bogotá, Dir. Eduardo Romero Castro) : «Analyse des mouvements humains anormaux» : Débutée en 2010

Encadrement du Post-Doc de Toby Low : «Modèles d'apparence pour la navigation autonome d'un robot mobile» : 2009-2010

Encadrement du Post-Doc de Taha Ridene : «Rapport visuel de situation par résumé géométrique de scène mobile» : 2012

Encadrement du contrat d'Ingénieur de Recherche de Matthieu Garrigues : «Tracking semi-dense temps réel pour la vidéosurveillance mobile embarquée» : 2011-2013

Encadrement du Post-Doc de Thanh Phuong Nguyen : «Modélisation et Reconnaissance d'activités pour la Vidéo-surveillance» : 2012-2013

Encadrement d'une vingtaine de stage de recherche : Master 2, projets de fin d'étude, projets en laboratoire.

Jurys de Thèse extérieurs

Examinateur de la Thèse de Christophe Lohou «Contribution à l'analyse topologique des images : étude d'algorithmes de squelettisation pour images 2D et 3D, selon une approche topologie digitale ou topologie discrète» (U. de Marne-la-Vallée), Dec 2001

Examinateur de la Thèse de Raphaël Sasportas «Etude d'architectures spécifiques aux applications d'analyse d'image par morphologie mathématique» (Ecoles des Mines de Paris), Oct. 2002

Rapporteur de la Thèse d'Olivier Barnich "Motion detection and human recognition in video sequences" (U. de Liège), Sept. 2010

Examinateur de la Thèse de Gloria Mercedes Díaz "Semantic Information Extraction from Microscopy Medical Images" (UNAL - Bogotá), Dec. 2010

Examinateur de la Thèse de Ramzi Mahmoudi «Stratégie de parallélisation commune des opérateurs à base de transformation topologique sur des machines parallèles à mémoire partagée » (U. de Marne-la-Vallée), Dec 2011

3 Publications

Journaux ou chapitres d'ouvrage

[1] Lionel Lacassagne, Antoine Manzanera, Julien Denoulet, and Alain Mérigot. High performance motion detection : some trends toward new embedded architectures for vision systems. *Journal of Real Time Image Processing*, 4(2) :127–146, 2009.

[2] Renaud Barate and Antoine Manzanera. Evolution of visual controllers for obstacle avoidance in mobile robotics. *Evolutionary Intelligence*, 2(3) :85–102, 2009.

[3] Gloria Mercedes Díaz and Antoine Manzanera. *Biomedical Image Analysis and Machine Learning Technologies : Applications and Techniques*, chapter Chapter VIII : Automatic Analysis of Microscopic Images in Hematological Cytology Applications. IGI Global, 2009.

[4] Antoine Manzanera and Julien Richefeu. A new motion detection algorithm based on sigma-delta background estimation. *Pattern Recognition Letters*, 28(3) :320–328, 2007.

[5] Antoine Manzanera, Thierry M. Bernard, Françoise Prêteux, and Bernard Longuet. nd skeletons : a unified mathematical framework. *Electronic Imaging*, 11(1) :25–37, 2002.

[6] Antoine Manzanera and Jean-Michel Jolion. La pyramide irrégulière : un modèle pour la vision exploratoire. *Traitement du signal*, 12(2) :176–196, 1995.

Conférences avec comités de lecture

[7] Antoine Manzanera. Dense hough transforms on gray level images using multi-scale derivatives (invited conference). In *The sixth International Workshop on Medical and Healthcare applications(AMINA'12)*, Mahdia, Tunisia, december 2012.

[8] Fabio Martínez Carrillo, Antoine Manzanera, and Eduardo Romero Castro. A motion descriptor based on statistics of optical flow orientations for action classification in video-surveillance. *Accepted to : Int. Conf. on Multimedia and Signal Processing (CMSP'12)*, Shanghai, China, december 2012.

[9] Fabio Martínez Carrillo, Antoine Manzanera, and Eduardo Romero Castro. Analysing the hovering flight of the hummingbird using statistics of the optical flow field. *Accepted to : ICPR Workshop on Visual observation and analysis of animal and insect behavior (VAIB 2012)*, Tsukuba, Japan, november 2012.

[10] M. Garrigues and A. Manzanera. Real time semi-dense point tracking. In A.J.C. Campilho and M.S. Kamel, editors, *Int. Conf. on Image Analysis and Recognition (ICIAR 2012)*, volume 7324 of *Lecture Notes in Computer Science*, pages 245–252, Aveiro, Portugal, june 2012. Springer.

[11] M. Garrigues and A. Manzanera. Exact and approximate median splitting on distributed memory machines. In *Int. Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA 2012)*, Las Vegas, NV, july 2012.

[12] D. Florins and A. Manzanera. Detection of floating mines in infrared sequences by multiscale geometric filtering. In Broach J.T. and J. Holloway Jr, editors, *SPIE Conference in Defence Security and Sensing : Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVII*, volume 8357, Baltimore, MD, June 2012.

[13] Antoine Manzanera. Local jet feature space framework for image processing and representation. In *International Conference on Signal Image Technology and Internet Based Systems (SITIS'11)*, pages 261–268, Dijon, France, dec. 2011. IEEE Computer Society.

[14] Fabio Martínez Carrillo, Antoine Manzanera, Cristina Santa Marta, and Eduardo Romero Castro. Characterization of motion cardiac patterns in magnetic resonance cine. In *International Conference on Image Information Processing (ICIIP'11)*, pages 1–5, Himachal Pradesh, India, Nov. 2011. IEEE Computer Society.

[15] Tobias Low and Antoine Manzanera. Ground-plane classification for robot navigation. In *International Conference on Control, Automation, Robotics and Vision (ICARCV'10)*, Singapore, dec. 2010. IEEE Computer Society.

[16] Lorenza Henao, Antoine Manzanera, and Eduardo Romero. Extracción y seguimiento de los miembros inferiores sin marcadores. In *VI Seminario Internacional de Procesamiento y Análisis de Imágenes Médicas (SIPAIM'10)*, Bogotá, Colombia, dec. 2010. Universidad Nacional de Colombia.

[17] Antoine Manzanera. Local jet based similarity for nl-means filtering. In *International Conference on Pattern Recognition (ICPR'10)*, volume 0, pages 2668–2671, Istambul, Turkey, aug. 2010. IEEE Computer Society.

[18] Lionel Lacassagne, Antoine Manzanera, and Antoine Dupret. Motion detection : Fast and robust algorithms for embedded systems. In *IEEE International Conference on Image Processing (ICIP'09)*, Cairo, Egypt, nov. 2009. IEEE.

[19] Philippe Guermeur and Antoine Manzanera. Image characterization from statistical reduction of local patterns. In *Progress in Pattern Recognition, Image Analysis and Applications (CIARP'09)*, volume 5856, pages 571–578, Guadalajara, Mexico, nov. 2009. Lecture Notes in Computer Science - Springer Verlag.

[20] Renaud Barate and Antoine Manzanera. Learning vision algorithms for real mobile robots with genetic programming. In *ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems (LAB-RS'08)*, Edinburgh, UK, aug. 2008. IEEE.

[21] Renaud Barate and Antoine Manzanera. Generalization performance of vision based controllers for mobile robots evolved with genetic programming. In *Genetic and Evolutionary Computation Conference (GECCO'08)*, Atlanta, GA, jul. 2008. ACM Press.

[22] Renaud Barate and Antoine Manzanera. Evolving vision controllers with a two-phase genetic programming system using imitation. In *10th International Conference on the Simulation of Adaptive Behavior (SAB'08)*, volume 5040, pages 73–82, Osaka, Japan, jul. 2008. Lecture Notes in Artificial Intelligence - Springer Verlag.

[23] Christine Dubreu, Antoine Manzanera, and Eric Bohain. Simulation of video sequences for an accurate evaluation of tracking algorithms on complex scenes. In *Proceedings of SPIE - Acquisition, Tracking, Pointing, and Laser Systems Technologies XXII*, volume 6971, Orlando - FL, USA, may. 2008.

[24] Renaud Barate and Antoine Manzanera. Automatic design of vision-based obstacle avoidance controllers using genetic programming. In *8th International Conference on Artificial Evolution (EA'07)*, volume 4926, pages 25–36, Tours, France, oct. 2007. Lecture Notes in Computer Science - Springer Verlag.

[25] Olivier Vermeulen, Antoine Manzanera, and Lionel Lacassagne. Ultra fast grey scale face detection using vector simd programming. In *3rd International Conference on Signal-Image Technology and Internet-based Systems (SITIS'07)*, Shanghai, China, dec. 2007. IEEE.

[26] Antoine Manzanera. Sigma-delta background subtraction and the zipf law. In *Progress in Pattern Recognition, Image Analysis and Applications (CIARP'07)*, volume 4756, pages 42–51, Viña del Mar-Valparaíso, Chile, nov. 2007. Lecture Notes in Computer Science - Springer Verlag.

[27] Philippe Guermeur, Petr Dokladal, Eva Dokladalova, and Antoine Manzanera. Fpga lab sessions in a general-purpose image processing course. In *2nd International Workshop on Reconfigurable Computing Education (RCE'07)*, Porto Alegre, Brasil, may. 2007.

[28] Christine Dubreu, Antoine Manzanera, and Eric Bohain. Comprehensive evaluation of tracking systems by non-photorealistic simulation. In *Proceedings of SPIE - Acquisition, Tracking, Pointing, and Laser Systems Technologies XXI*, volume 6569, Orlando - FL, USA, may. 2007.

[29] Taha Ridene and Antoine Manzanera. Mécanismes d'attention visuelle sur rétine programmable. In *Traitement et Analyse de l'Information : Méthodes et Applications (TAIMA'07)*, pages 301–306, Hammamet, Tunisia, may. 2007.

[30] Paul Nadrag, Antoine Manzanera, and Nicolas Burrus. Smart retina as a contour-based visual interface. In *Distributed Smart Cameras Workshop (DSC'06)*, Boulder - CO, USA, oct. 2006. ACM.

[31] Julien C. Richefeu and Antoine Manzanera. Détection de mouvement par capteur intelligent. In *ORASIS'05*, Clermont-Ferrand, France, may. 2005.

[32] Julien C. Richefeu and Antoine Manzanera. A new hybrid differential filter for motion detection. In *International Conference on Computer Vision and Graphics (ICCVG'04)*, Warsaw - Poland, sep 2004.

[33] Antoine Manzanera and Julien C. 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, Kolkata - India, dec. 2004.

[34] Antoine Manzanera and Thierry M. Bernard. Metrical properties of a collection of 2d parallel thinning algorithms. In *International Workshop on Combinatorial Image Analysis (IWCIA'03)*, volume 12 of *Electronic Notes on Discrete Mathematics*, Palermo - Italy, may. 2003. Elsevier Science.

[35] Julien C. Richefeu and Antoine Manzanera. A morphological dominant points detection and its cellular implementation. In *International Symposium on Signal Processing and its Applications (ISSPA'03)*, volume 2, pages 181–184, Paris - France, jul. 2003. IEEE.

[36] Antoine Manzanera. Morphological segmentation on the programmable retina : towards mixed synchronous/asynchronous algorithms. In *6th International Symposium on Mathematical Morphology (ISMM'02)*, pages 25–37, Sydney - Australia, apr. 2002. CSIRO.

[37] Antoine Manzanera, Françoise Prêteux, and Thierry M. Bernard. Markovian modeling on programmable retina. In *Conference on Controle Quality by Artificial Vision (QCAV'01)*, pages 232–237, Le Creusot - France, may. 2001. Cépaduès-Editions.

[38] Antoine Manzanera, Thierry M. Bernard, Françoise Prêteux, and Bernard Longuet. Ultra fast skeleton based on an isotropic fully parallel algorithm. In *8th Discrete Geometry for Computer Imagery (DGCI'99)*, volume 1568, pages 313–324, Marne-La-Vallée - France, mar. 1999. Lecture Notes in Computer Science - Springer Verlag.

[39] Antoine Manzanera, Thierry M. Bernard, Françoise Prêteux, and Bernard Longuet. A unified mathematical framework for a compact and fully parallel n-d skeletonization procedure. In *Vision Geometry VIII (VG'99)*, volume 3811, pages 57–68, Denver - CO, jul. 1999. SPIE.

[40] Antoine Manzanera, Thierry M. Bernard, Françoise Prêteux, and Bernard Longuet. Medial faces from a concise 3d thinning algorithm. In *International Conference on Computer Vision (ICCV'99)*, pages 337–343, Kerkyra - Greece, sep. 1999. IEEE.

[41] Thierry M. Bernard and Antoine Manzanera. Improved low complexity fully parallel thinning algorithm. In *International Conference on Image Analysis and Processing (ICIAP'99)*, pages 215–220, Venice - Italy, sep. 1999. IEEE.

[42] Antoine Manzanera and Jean-Michel Jolion. Pyramide irrégulière. In *9ème congrès AFCET/RFIA (RFIA'94)*, pages 221–229, Paris, France, jan. 1994.

[43] Antoine Manzanera and Jean-Michel Jolion. Tesselation hiérarchique irrégulière. In Christian Ronse, editor, *Colloque Géométrie Discrète en Imagerie (DGCI'93)*, pages 98–107, Strasbourg, France, sep. 1993.

Thèses encadrées

[44] Christine Dubreu. *Algorithmique de traitement d'image des systèmes de surveillance infrarouges air-sol.* PhD thesis, Université de Bourgogne, Paris, France, 2009. Spécialité Instrumentation et Informatique de l'Image.

[45] Renaud Barate. *Apprentissage de fonctions visuelles pour un robot mobile par programmation génétique.* PhD thesis, Université Pierre et Marie Curie, Paris 6, Paris, France, 2008. Spécialité Informatique, Télécommunications et Electronique.

[46] Julien Richefeu. *Détection et analyse du mouvement sur système de vision à base de rétine numérique.* PhD thesis, Université Pierre et Marie Curie, Paris 6, Paris, France, 2006. Spécialité Informatique, Télécommunications et Electronique.

[47] Antoine Manzanera. *Vision Artificielle Rétinienne.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2000. Spécialité Signal et Image.

Rapports de recherche ou contractuels

[48] Antoine Manzanera. Object modelling, detection and localization in mobile video : a state-of-the-art. Technical report, ITEA2, 2011.

[49] Antoine Manzanera. Image representation and processing through multiscale local jet features. Technical report, ENSTA/LEI, 2010.

[50] David Filliat, Antoine Manzanera, Thierry M. Bernard, and Ph. Guermeur. Apprentissage visuel pour la robotique. Technical report, Convention DGA-MRIS 2006-2009, dec. 2008.

[51] Antoine Manzanera and Nicolas Burrus. Programmation SIMD de la rétine programmable : Opérations arithmétiques et logiques. Rapport Technique ENSTA, 2006.

[52] Antoine Manzanera. Implantation des algorithmes de détection sur rétine programmable pour le capteur de réveil caladiom. Technical report, Bertin Technologie, dec. 2004.

[53] Antoine Manzanera. Conception et validation des algorithmes de détection pour le capteur de réveil caladiom. Technical report, Bertin Technologie, dec. 2003.

[54] Antoine Manzanera. On the properties of the (4,8)-median axis. Technical Report LEI/AVA-02-001, ENSTA/LEI, 2002.

[55] Antoine Manzanera and Thierry M. Bernard. Mb : a coherent collection of 2d parallel thinning algorithms. Technical Report LEI/AVA-02-002, ENSTA/LEI, 2002.

[56] Antoine Manzanera. Algorithmique de traitement d'images pour la détection des cibles - système antichar eryx. Technical Report 20005137, MBDA-F, jan. 2002.

[57] Antoine Manzanera. Architecture logicielle de traitement d'images pour la détection des cibles. Technical Report 20005137, MBDA-F, jan. 2002.

Polycopiés de Cours

[58] Antoine Manzanera. Morphologie Mathématique Support du cours ENSTA 2e année, 2009.

[59] Antoine Manzanera. Traitement d'images et Vision artificielle - Chapitre 1 : Les images numériques Polycopié de cours ENSTA 3e année, 2008.

[60] Antoine Manzanera. Traitement d'images et Vision artificielle - Chapitre 2 : Filtrage et Restauration Polycopié de cours ENSTA 3e année, 2008.

[61] Antoine Manzanera. Traitement d'images et Vision artificielle - Chapitre 3 : Les espaces d'échelle Polycopié de cours ENSTA 3e année, 2008.

Cours et Tutoriaux invités

[62] Antoine Manzanera. Multiscale differential Geometry and Dense Hough transforms : a unified framework for line, circle and object detection in videos. Invited conference - CONAIS'12, Villahermosa, September 2012

[63] Antoine Manzanera. Colour imaging from perception to processing Invited tutorial - CONAIS'12, Villahermosa, September 2012

[64] Antoine Manzanera. Visual Feature Spaces for Image Representation and Processing : The Multiscale Local Jet. Invited conference - AMINA'10, Monastir, November 2010

[65] Antoine Manzanera. Human motion analysis : tools, algorithms and applications. Tutoriel invité - *Latin American Conference on Networked and Electronic Media*, Bogotá. Août 2009

[66] Antoine Manzanera and Jean Serra. Morphologie Mathématique pour l'analyse d'images : Concepts et Algorithmes. Cours invité - Ecole d'été *Sciences et Technologies de l'information et de la Communication* Université de Sousse. Juillet 2008

[67] Antoine Manzanera. Distancias discretas : algoritmos y aplicaciones. Cours invité - Master *Ingeniería Biomédica* Universidad Nacional de Colombia, Bogotá. Juin 2008

[68] Antoine Manzanera. Análisis del movimiento : detección, estimación, seguimiento. Cours invité - Master *Ingeniería Biomédica* Universidad Nacional de Colombia, Bogotá. Juin 2008

[69] Antoine Manzanera. Indexación de imágenes y vídeos. Cours invité - Master *Ingeniería Biomédica* Universidad Nacional de Colombia, Bogotá. Juin 2008

[70] Antoine Manzanera. De la détection du changement à l'analyse du mouvement : applications en télésurveillance et navigation autonome. Cours invité - Ecole de Printemps *Traitement et Analyse des Signaux Multidimensionnels* Université Politehnica de Bucarest. Mai 2006

Annexe B

Sélection de publications

Cette annexe contient une sélection de publications permettant une présentation approfondie de certains aspects présentés précédemment. La numérotation suit l'ordre dans lequel les travaux correspondants sont évoqués dans le mémoire.

1. Antoine Manzanera and Thierry M. Bernard. "Metrical properties of a collection of 2d parallel thinning algorithms". In *International Workshop on Combinatorial Image Analysis (IWCIA'03)*, volume 12 of *Electronic Notes on Discrete Mathematics*, Palermo - Italy, may. 2003. Elsevier Science.
2. Antoine Manzanera. "Dense Hough transforms on gray level images using multi-scale derivatives" (invited conference). In *The sixth International Workshop on Medical and Healthcare applications (AMINA'12)*, Mahdia, Tunisia, december 2012.
3. Antoine Manzanera. "Local jet feature space framework for image processing and representation". In *International Conference on Signal Image Technology and Internet Based Systems (SITIS'11)*, pages 261–268, Dijon, France, dec. 2011. IEEE Computer Society.
4. Antoine Manzanera and Julien Richefeu. "A new motion detection algorithm based on sigma-delta background estimation". *Pattern Recognition Letters*, 28(3) :320–328, 2007.
5. Antoine Manzanera. "Sigma-delta background subtraction and the Zipf law". In *Progress in Pattern Recognition, Image Analysis and Applications (CIARP'07)*, volume 4756, pages 42–51, Viña del Mar-Valparaíso, Chile, nov. 2007. Lecture Notes in Computer Science - Springer Verlag.
6. Renaud Barate and Antoine Manzanera. "Evolution of visual controllers for obstacle avoidance in mobile robotics". *Evolutionary Intelligence*, 2(3) :85–102, 2009.

7. Lionel Lacassagne, Antoine Manzanera, Julien Denoulet, and Alain Mérigot. "High performance motion detection : some trends toward new embedded architectures for vision systems". *Journal of Real Time Image Processing*, 4(2) :127–146, 2009.

Metrical properties of a collection of 2D parallel thinning algorithms

Antoine Manzanera *

Ecole Nat. Sup. de Techniques Avancées, 32 Bd Victor, 75015 Paris - FRANCE

Thierry M.Bernard

Ecole Nat. Sup. de Techniques Avancées, 32 Bd Victor, 75015 Paris - FRANCE

Abstract

This paper is dedicated to the study of metrical properties of a collection of 2D thinning algorithms that we have proposed. Here, we characterize their underlying metrics and use it to reduce the classical metrical biases that affect thinning algorithms in the square grid. We show that some algorithms from the collection lead to skeletons based on a particular geometry, corresponding to the (4,8)-median axis, which is a new shape descriptor, featuring nice robustness and conditioning properties.

Key words: digital geometry, parallel thinning algorithm, (4,8)-median axis

1 Introduction

We have recently proposed a family of 2-dimensional thinning algorithms to compute the skeleton of binary discrete images. We systematically constructed them from the discretization of the evolution equation of a monotonous propagating front, under topology preservation constraints, in the different connectivity models of the square grid, and for different parallelization schemes. Logic minimization was an important issue in their genesis. The eight algorithms, referred to as MB, are summarized on Figure 1, with a comparison of their different properties. The name of every algorithm is given according to

* corresponding author
Email address: manzaner@ensta.fr (Antoine Manzanera).
URL: http://www.ensta.fr/~manzaner (Antoine Manzanera).

the three binary labels: (1) *-fp* or *-dir* for fully parallel or directional, (2) -1 or -2 depending on the number of directions of propagation (4 or 8 respectively), and (3) -4 or -8 depending on the topology. The Boolean definition of every algorithm is presented on Figure 1: the principle of the algorithms is to delete iteratively all pixels matching the removing condition, provided that they do not match the non removing condition. For the fully parallel algorithms, all patterns are to be considered with their $\pi/2$ rotated versions. For the MBfpx-4 algorithms (x = 1 or 2), a special convention is used: the white pixels with a red dot (resp. black pixels with a green square) are the black pixels of the original image matching (resp. not matching) the removing condition. The derivation, proof, and details of implementation of each one of these algorithms can be found in [8]. Note that MBfp1-8 is equivalent to the algorithm proposed in [3], and MBfp1-4 is equivalent to the algorithm proposed in [6]. Figure 1 synthesizes some combinatorial (e.g. Boolean complexity and support) and topological (e.g. P-simpleness [1]) properties that are not addressed in this paper, but detailed in [8].

The aim of this paper is to study the metrical properties of these algorithms.

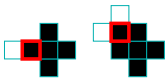
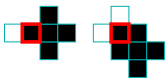
ALGORITHM	Removing condition	Non–removing condition	Parallelism	Topology preservation	Isotropy	1–pixel thickness	Boolean complexity	Support and size	P–simpleness
MBdir1–8			DIR	8	NO	YES	28 r	8 (8)	YES
MBdir2–8			DIR	8	NO	YES	76 r	 8 (8)	YES
MBdir1–4			DIR	4	NO	YES	28 r	 7 (8)	YES
MBdir2–4			DIR	4	NO	YES	60 r	 7 (8)	YES
MBfp1–8 [Eckhardt et al. 93]			FP	8	YES	NO	18 ρ	 13	YES
MBfp2–8			FP	8	YES	NO	28 ρ	 21	YES
MBfp1–4 [Latecki et al. 95]			FP	4	NO	NO	16 ρ	 23	YES
MBfp2–4			FP	4	NO	NO	26 ρ	 38	NO

Fig. 1. The MB family of parallel thinning algorithms.

In Section 2, we show that the geometry of the MB skeletons can be formally characterized by the type of median axis that they each contain. In particular, the (4,8)-median axis is defined as the mixed case of a generic median axis including the classical morphological skeletons for the two canonical distances of the square grid. We show that the (4,8)-median axis is the locus of the cen-

ters of the maximal elements from a collection of sets called (4,8)-fuzzy balls, which are formally defined. In Section 3, we show how the different underlying metrics of the algorithms lead to different behaviors with respect to rotation invariance and noise immunity, and discuss the issue of approximating Euclidean skeletons with thinning algorithms in the square grid.
Conclusions and perspective work are presented in Section 4.

2 (K-P)-median axis and metrical properties of MB

As shape descriptor, it is obvious that the metrical properties of a skeleton are very important ; the initial shape must be recovered at least approximately from its weighted skeleton, slight variations on the contour should not lead to significant changes in the skeleton, and the skeleton must be fairly invariant to arbitrary rotations or scalings. Nevertheless, these issues are rather poorly addressed by the thinning approaches. There are other skeletonization methods designed in Euclidean frameworks, either continuous [9] or discrete [4], that address explicitly the metrical issues, at the price of a representation change, implying a higher computational cost and a loss of regularity.

We wish to give in this paper a formal description of the metrical behavior of the proposed thinning algorithms, whose results can be seen on Figure 2. The geometry of the skeletons are based on different median axis, depending on the type of parallelism and the directions of deletion. We now recall the formalism needed to introduce our generic median axis:
The discrete plane is mapped to the *square grid* $\mathbb{Z}^2$, a (binary) *image* X is a subset of $\mathbb{Z}^2$. A *pixel* x is an element of $\mathbb{Z}^2$. The two canonical discrete distances of the square grid are respectively the *4-distance* d_4, and the *8-distance* d_8. If $x = (x_1, x_2)$ and $y = (y_1, y_2)$, then $d_4(x, y) = |x_1 - y_1| + |x_2 - y_2|$ and $d_8(x, y) = \max(|x_1 - y_1|, |x_2 - y_2|)$.
The (K, P)-*median axis* of image X (K and P are equal to 4 or 8 and $K \leq P$) is defined as:

$$S_K^P(X) = \bigcup\{x \in X; (y \in X \text{ and } d_P(x, y) = 1) \Rightarrow d_K(x, X^c) \geq d_K(y, X^c)\}$$

So the (K, P)-median axis is the set of the maxima of distance d_K, in the P-neighborhood. Depending on the values of K and P, this leads to three different median axes, shown on row 1 of Figure 2. We are going to show that these different median axes determine the metrical properties of the different algorithms.

As every iteration (resp. four successive sub-iterations) of MBfp (resp. MBdir) examines the 4-contour (resp. the 8-contour), the geometry of the resulting skeleton is based on distance d_4 (resp. d_8), as it can be seen on Figure 2. In the case of the MBfp 8-connected algorithms, the isotropy allows even to prove

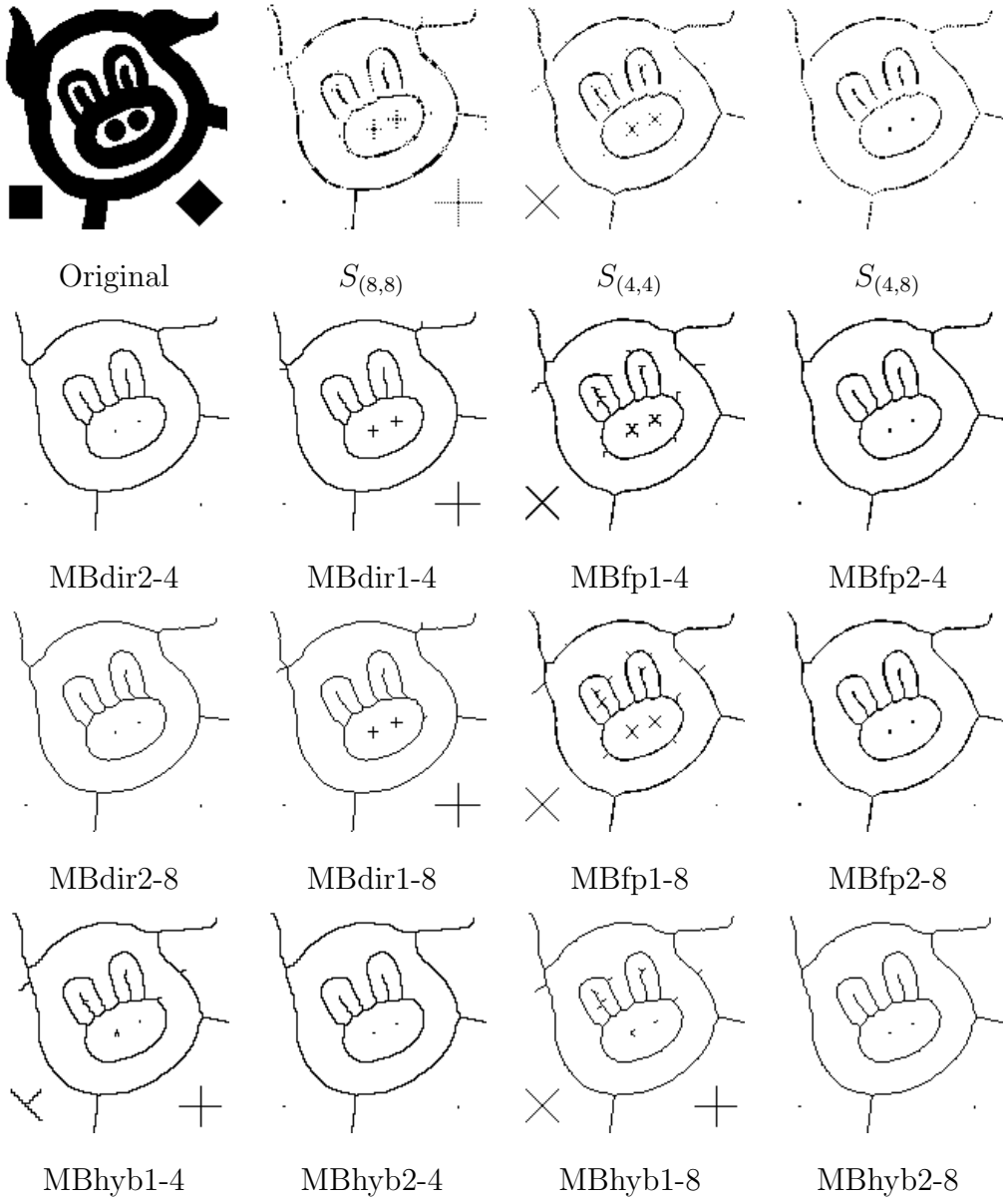


Fig. 2. Metrical properties of the MB skeletons.

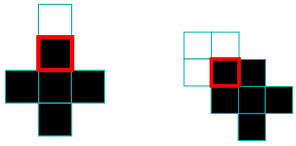


Fig. 3. Deletion patterns of the MBfp- skeletons.

formally that, if the pixels are examined in the order induced by the distance d_4 to the border, then:

(1) $S_4^4(X) \subset$ MBfp1-8(X) and (2) $S_4^8(X) \subset$ MBfp2-8(X).

Indeed, if a pixel p matches pattern (α_1) (resp. (α_1) or (α_2)) shown on Figure 3 (or one of their $\pi/2$ rotated versions), then it is 4−adjacent (resp. 8−adjacent) to a 4-interior point q, such that $d_4(q, X^c) = d_4(p, X^c)+1$, so $p \notin S_4^4(X)$ (resp. $p \notin S_4^8(X)$). These properties are verified for most images, because except in pathological cases (some examples can be seen in [7]), the MBfp thinning respect the order induced by distance d_4. It follows that the geometry of the each MB-skeleton is determined by the geometry of the corresponding (K, P)-*median axis*. We are now going to characterize the geometry of the $(4, 8)$-*median axis*, thanks to the notion of $(4, 8)$-*fuzzy balls*, that we define further:
For $K = 4$ or 8, the *K-ball* of center x and radius n is defined as $B_K(x, n) = \{z \in \mathbb{Z}^2, d_K(x, z) \leq n\}$. A ball $B_K(x, n)$ is said to be *maximal* in the image X if $\forall (y, n') \in \mathbb{Z}^2 \times \mathbb{N}, B_K(x, n) \subset B_K(y, n') \subset X \Rightarrow (x, n) = (y, n')$.
For $K = P$, it is well known that the (K, K)-median axis corresponds to the union of the centers of maximal K-balls [5]. We are going to prove that the $(4, 8)$-median axis corresponds to the union of the centers of maximal $(4, 8)$-*fuzzy balls* (see Figure 4), which are recursively defined as follows:

(1) A $(4, 8)$-fuzzy ball of radius 1 and center x $B_{(4,8)}(x, 1)$ is any set verifying: $B_4(x, 1) \subset B_{(4,8)}(x, 1) \subset B_8(x, 1)$.
(2) A $(4, 8)$-fuzzy ball of radius n+1 and center x is a set such that there exists F_n^x, a $(4, 8)$-fuzzy ball of center x and radius n such that: $B_{(4,8)}(x, n + 1) = \bigcup_{y \in F_n^x} B_{K_y}(y, 1)$, where K_y is 4 or 8, depending on y.

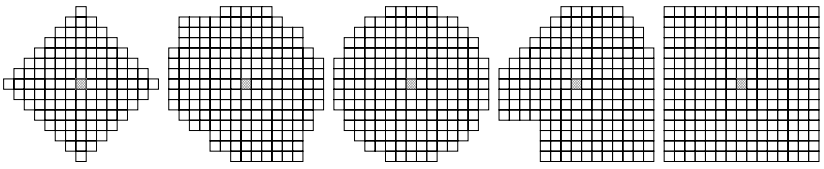


Fig. 4. Some $(4, 8)$-fuzzy balls of radius 7. The extremal cases of $(4, 8)$-fuzzy balls are respectively the 4-ball (on the left), and the 8-ball (on the right).

To prove the identity between the $(4, 8)$-median axis and the locus of the centers of maximal $(4, 8)$-fuzzy balls, we need to use the morphological erosion and dilation:
Let $b \in \mathbb{Z}^2$. The *translated* of X by b is the set $X_b = \{x + b; x \in X\}$.
Let $B \subset \mathbb{Z}^2$. The *morphological dilation* of X by B is defined as:

$$X \oplus B = \bigcup_{b \in B} X_{-b} = \{z \in \mathbb{Z}^2; B_z \cap X \neq \emptyset\}$$

The *morphological erosion* of X by B is defined as:

$$X \ominus B = \bigcap_{b \in B} X_{-b} = \{z \in \mathbb{Z}^2; B_z \subset X\}$$

We also need to prove the following lemma:

Lemma 1 *If $B_{(4,8)}(x,n)$ is a $(4,8)$-fuzzy ball of radius n and center x, and if $y \in B_8(x,1)$, then $[B_{(4,8)}(x,n) \cup B_4(y,n+1)]$ is a $(4,8)$-fuzzy ball of radius $n+1$ and center y.*

Preliminary remark: it is clear, by the definition of fuzzy balls, that if F_n^x is a $(4,8)$-fuzzy ball of center x and radius n, then any set S verifying:

$$F_n^x \oplus B_4(0,1) \subset S \subset F_n^x \oplus B_8(0,1) \tag{1}$$

is a $(4,8)$-fuzzy ball of center x and radius $n+1$. Now we prove the lemma by induction on n. If $n = 0$, $B_{(4,8)}(x,0) = \{x\}$. If $y \in B_8(x,1)$, $B_4(y,1) \subset \{x\} \cup B_4(y,1) \subset B_8(y,1)$, so $[B_{(4,8)}(x,0) \cup B_4(y,1)]$ is a $(4,8)$-fuzzy ball of radius 1 and center y.
Now suppose the lemma true for radii less than or equal to $(n-1)$. Let $B_{(4,8)}(x,n)$ be a $(4,8)$-fuzzy ball of radius n and center x. By definition, there exists F_{n-1}^x, a $(4,8)$-fuzzy ball of center x and radius $(n-1)$ such that:

$$B_{(4,8)}(x,n) = \bigcup_{z \in F_{n-1}^x} B_{K_z}(y,1) \tag{2}$$

and

$$F_{n-1}^x \oplus B_4(0,1) \subset B_{(4,8)}(x,n) \subset F_{n-1}^x \oplus B_8(0,1) \tag{3}$$

Let $y \in B_8(x,1)$. By induction hypothesis, $G_n^y = F_{n-1}^x \cup B_4(y,n)$ is a $(4,8)$-fuzzy ball of radius n and center y.

$$\begin{aligned} G_n^y \oplus B_4(0,1) &= (F_{n-1}^x \oplus B_4(0,1)) \cup (B_4(y,n) \oplus B_4(0,1)) &(4)\\ &= (F_{n-1}^x \oplus B_4(0,1)) \cup B_4(y,n+1) &(5)\end{aligned}$$

So, from (3), we get:

$$G_n^y \oplus B_4(0,1) \subset [B_{(4,8)}(x,n) \cup B_4(y,n+1)] \tag{6}$$

On the other hand, we have:

$$G_n^y \oplus B_8(0,1) = (F_{n-1}^x \oplus B_8(0,1)) \cup (B_4(y,n) \oplus B_8(0,1)) \tag{7}$$

and as

$$B_4(y, n+1) \subset (B_4(y, n) \oplus B_8(0, 1)) \tag{8}$$

from (3), we get:

$$[B_{(4,8)}(x, n) \cup B_4(y, n+1)] \subset G_n^y \oplus B_8(0, 1) \tag{9}$$

Finally, as G_n^y is a $(4, 8)$-fuzzy ball of radius n and center y, we conclude thanks to (6) and (9) that $[B_{(4,8)}(x, n) \cup B_4(y, n+1)]$ is a $(4, 8)$-fuzzy ball of radius $(n+1)$ and center y.

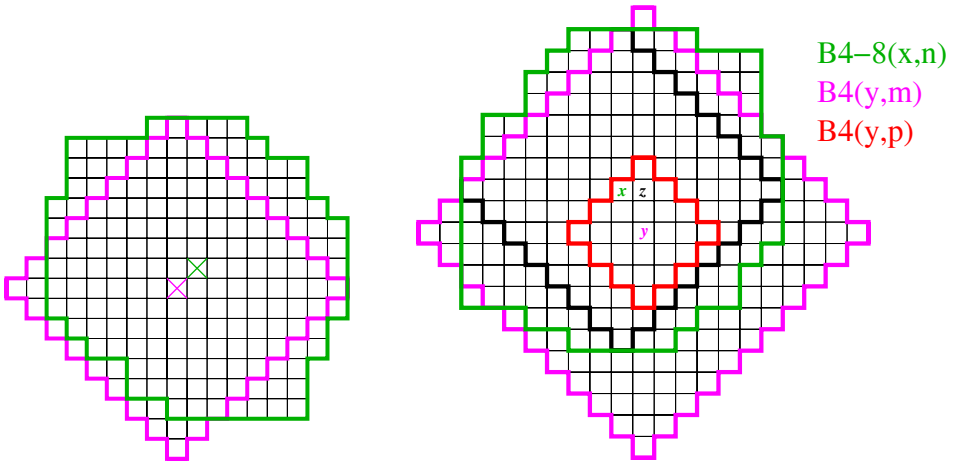
□

Theorem 1 *$S_{(4,8)}(X)$ is the locus of the centers of maximal $(4, 8)$-fuzzy balls in X.*

(1) Right inclusion. Let x be the center of a $(4, 8)$-fuzzy balls $B_{(4,8)}(x, n)$ that is maximal in X. Now suppose that there exists $y \in (B_8(x, 1) \cap X)$ such that $d_4(y, X^c) > d_4(x, X^c)$. Then we must have $B_4(y, n+1) \subset X$. And so:

$$[B_{(4,8)}(x, n) \cup B_4(y, n+1)] \subset X \tag{10}$$

But from lemma 1, $[B_{(4,8)}(x, n) \cup B_4(y, n+1)]$ is a $(4, 8)$-fuzzy ball of radius $n+1$ (see Figure 5(1)), which is in contradiction with the maximality of $B_{(4,8)}(x, n)$.



(1) (2)

Fig. 5. (1) The center of a maximal $(4, 8)$-fuzzy ball is an element of $S_{(4,8)}(X)$. (2) An element of $S_{(4,8)}(X)$ is center of a maximal $(4, 8)$-fuzzy ball.

(2) Left inclusion. Let $x \in S_{(4,8)}(X)$. Let $B_{(4,8)}(x,n)$ be the biggest $(4,8)$-fuzzy ball of center x contained in X. We are going to prove that $B_{(4,8)}(x,n)$ is maximal in X.
Suppose that there exists a $(4,8)$-fuzzy ball F_m^y of center y and radius m such that $(y,m) \neq (x,n)$ and $B_{(4,8)}(x,n) \subset F_m^y \subset X$. We have:

$$B_4(y,m) \subset F_m^y \tag{11}$$
$$B_4(x,n) \subset B_{(4,8)}(x,n) \tag{12}$$

The erosion of the ball $B_4(y,m)$ by $B_4(0,n)$ is a ball $B_4(y,p)$ containing x. But $x \notin B_4(y,p) \ominus B_4(0,1)$, otherwise it would mean that $B_4(x,n+1) \subset F_m^y \subset X$, and then $B_{(4,8)}(x,n) \cup B_4(x,n+1)$ would be a $(4,8)$-fuzzy ball of center x and radius $(n+1)$ (see lemma 1) contained in X, which is in contradiction with the fact that $B_{(4,8)}(x,n)$ is the biggest $(4,8)$-fuzzy ball of center x contained in X.
So there must exist $z \in B_4(x,1)$ (see Figure 5(2)) such that $z \in B_4(y,p) \ominus B_4(0,1)$. Then $B_4(z,n+1) \subset F_m^y \subset X$, and so $d_4(z,X^c) > d_4(x,X^c)$. As $z \in B_4(x,1)$, we get $x \notin S_{(4,8)}(X)$, which is in contradiction with our hypothesis.

□

We have now identified the relation between the mixed median axis $S_{(4,8)}(X)$ and a particular class of sets, the $(4,8)$-fuzzy balls. These balls are a new shape description tool, which interest lies in the robustness of morphological or connected skeletons defined in the square grid, as we shall illustrate in the following section.

3 Consequences on the geometrical behavior of MB

The fact that the MB-2 algorithms do not distinguish different $(4,8)$-fuzzy balls (Figure 6(1)), make them more robust with respect to noise (Figure 6(2)), and rotation (Figure 6(3)). Obviously, the outcome is that only partial re-constructibility is possible, unlike MBfp1-8, that allows exact re-constructibility (Figure 6(4), re-constructibility is performed over the skeleton weighted with the distance to the border, with d_4 balls for MBfp1-8, and octagonal balls as a median choice of $(4,8)$-fuzzy balls for MBfp2-8).
Nevertheless, the better behavior of the MB-2 algorithms with respect to rotation is limited to the fact that fewer branches are generated (Figure 6(3)), but if the number of branches is fairly stable for MB-2 on the different rotated versions, their relative positions can vary significantly: see for example Figure 7, and the changes in the hierarchy of the branches. This bias is due to the different errors of the underlying distances (d_4 or d_8 for the fully parallel or

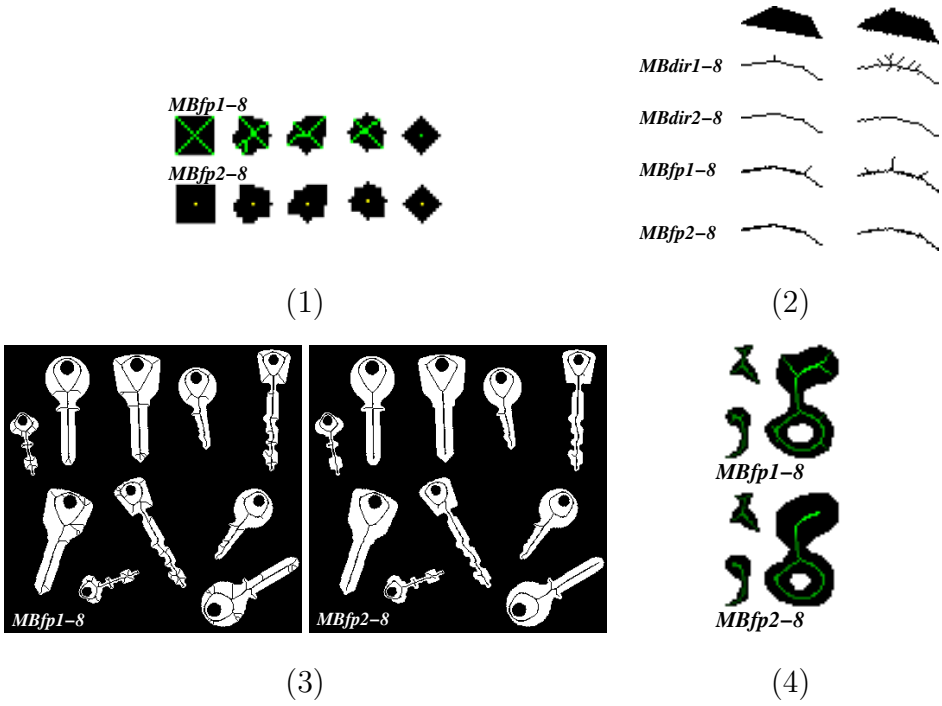


Fig. 6. (1) Some $(4,8)$-fuzzy balls of radius 7 (2) noise immunity (3) rotation invariance (4) re-constructibility.

directional algorithms, respectively) with respect to the Euclidean distance, depending on the angles of the objects.

A first basic idea to get a "more Euclidean" geometry for the skeleton is to

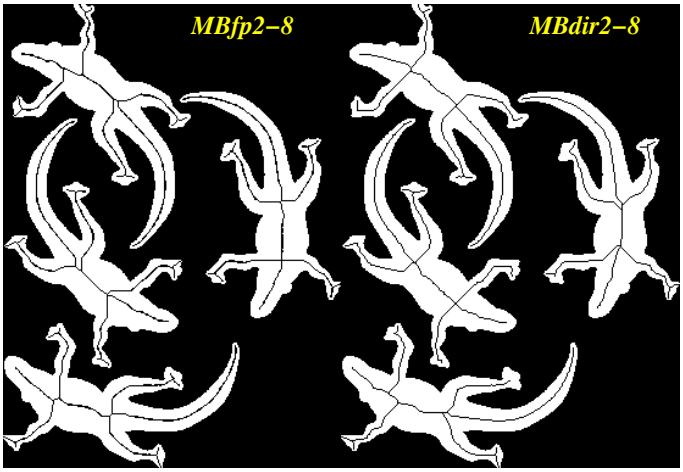


Fig. 7. Limitations of the rotation invariance shown on the image *lizard* at scale 2: the radius of the biggest d_4 (resp. d_8) ball is $\rho = 27$ (resp. $r = 19$).

alternate fully parallel and directional iterations. For example the MB *hybrid* algorithms are defined by Σ-Δ modulation of directional or fully parallel iterations as follows: starting from $S_0 = 0$, at iteration $n > 0$, if $|S_{n-1}+\sqrt{2}/2-n| < |S_{n-1}+\sqrt{2}-n|$, do $S_n = S_{n-1}+\sqrt{2}/2$ and perform one fully parallel iteration, else do $S_n = S_{n-1}+\sqrt{2}$ and perform four directional iterations. By

construction, it turns out that the underlying metrics of the MB hybrid thinning algorithms is generated by the octagonal discrete balls minimizing the maximal error with respect to the Euclidean distance (see an example of such "optimal" octagonal ball on Figure 8). MBhyb1 and MBhyb2 are shown on the last row of Figure 2. This method leads to a significant improvement (compare images of Figure 7 with image at scale 2 of Figure 8), but the bias with respect to Euclidean distance keeps increasing with scale.

Another interesting property of the fuzzy metrics of the MB-2 algorithms

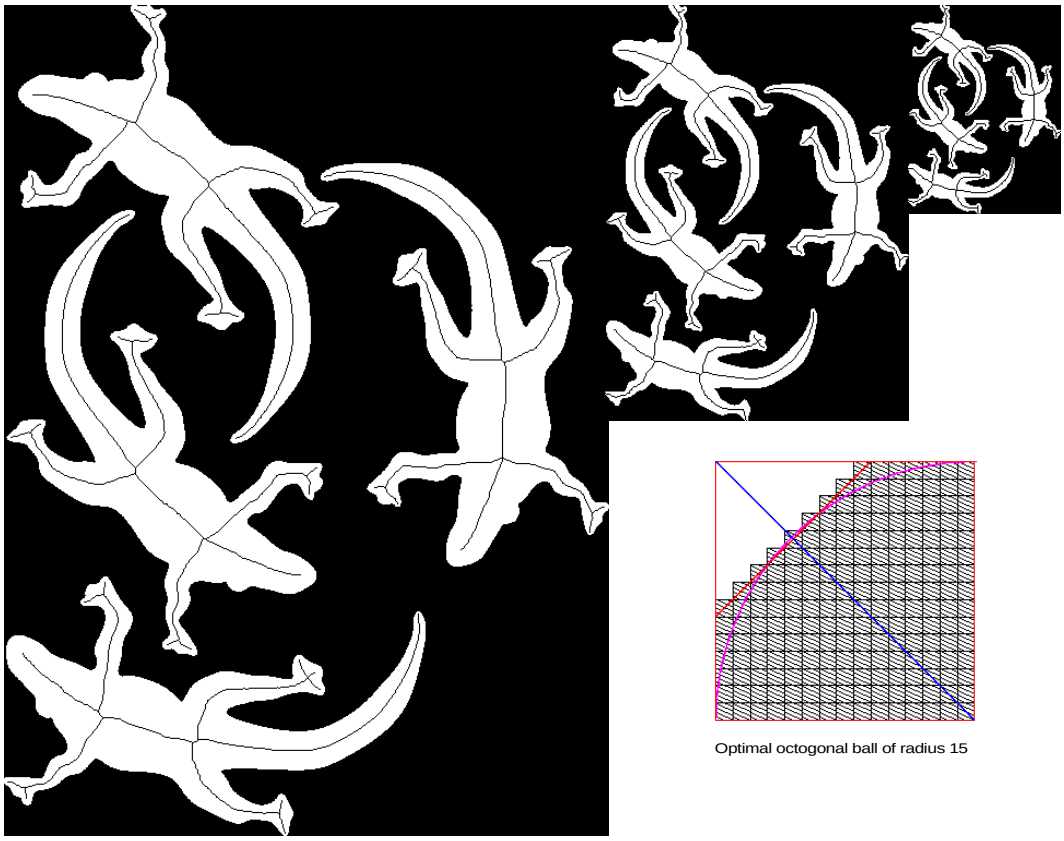


Fig. 8. The MB hybrid skeleton shown on image *lizard* at scales 1, 2 and 3: the radius of the biggest optimal octagonal ball is respectively 41, 21 and 11.

is their ability to be conditioned by a Euclidean or pseudo-Euclidean preprocessed distance. For example, in Figure 9, a chamfer distance transform of support 5 [2] is computed, and then the thinning algorithms are applied by imposing that the pixels deleted at the same iteration are at the same distance to the border. The difference of behaviors between MB-1 and MB-2 shown on Figure 9, and the good invariance to rotation that shows MB2 in that case is explained by the fact that *Euclidean discrete balls are* $(4, 8)$*-fuzzy balls*. So in the case of images where the maximal fuzzy balls are Euclidean, a quasi Euclidean skeleton can be obtained by conditioning MB-2 by the corresponding distance.

The limitation of conditioning lies in the fact that *the set of all* $(4, 8)$*-fuzzy*

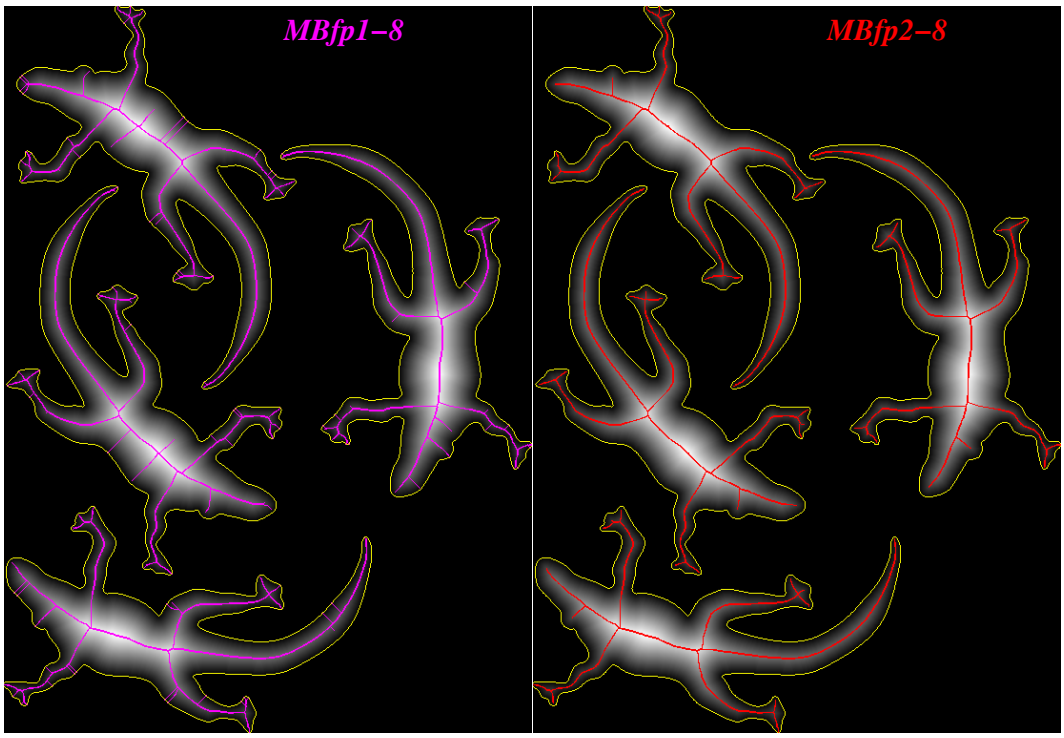


Fig. 9. Conditioning the MB thinning algorithms by a distance transform.

balls is not stable by arbitrary rotation, and this lead to important biases in case of big non Euclidean $(4, 8)$-fuzzy balls (big 4-balls or 8-balls, typically), so for shapes with large perpendicular straight contours.

4 Conclusion

We have shown in this paper the consistency in terms of metrical properties of the family of thinning algorithms we have recently proposed. The geometrical behavior, the advantages and limitations of every algorithm have been clearly identified.
We have shown in [7] that the 8-connected MBfp algorithms could be naturally expressed in the n-dimensional cubic grid, and we have proved the validity of the corresponding algorithms for $n = 3$. The same extension can be done for the other algorithms, but their validity remains to prove. This will be the subject of future work, in order to get hopefully a unified and cleanly justified thinning methodology for n-dimensional binary images.

References

[1] G. Bertrand. On P-simple points. *Comptes Rendus à l'Académie des Sciences*, 321-1:1077–1084, 1995.

[2] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, vol. 34, 344–371, 1986.

[3] U. Eckhardt and G. Maderlechner. Invariant thinning. *International Journal of Pattern Recognition and Artificial Intelligence*, 7-5:1115–1144, 1993.

[4] A.X. Falcão, L. da Fontoura Costa and B.S. da Cunha. Multiscale skeletons by image foresting transform and its application to neuromorphometry. *Pattern recognition*, vol. 35, 1571–1582, 2002.

[5] C. Lantuéjoul. La squelettisation et son application aux mesures topologiques des mosaïques polycristallines. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 1978.

[6] L.J. Latecki, U. Eckhardt, and A. Rosenfeld. Well-composed sets. *Computer Vision and Image Understanding*, 61-1:70–83, 1995.

[7] A. Manzanera, T.M. Bernard, F. Prêteux, and B. Longuet. nD skeletonization: a unified mathematical framework. *Journal of Electronic Imaging*, vol. 11(1), 25–37, 2002.

[8] A. Manzanera and T.M. Bernard. MB: A coherent collection of 2D parallel thinning algorithms. *ENSTA/LEI Technical report* LEI/AVA-02-002, 2002. *available at:* http://www.ensta.fr/~manzaner/publis.html/

[9] K. Siddiqi, S. Bouix, A. Tannenbaum and S.W. Zucker. Hamilton-Jacobi skeletons. *International Journal on Computer Vision*, vol. 48(3), 215–231, Kluwer Academic pub., 2002.

Dense Hough transforms on gray level images using multi-scale derivatives

Antoine Manzanera

ENSTA-ParisTech,
Electronics and Computer Science Laboratory
32 Boulevard Victor, 75739 PARIS CEDEX 15
http://www.ensta-paristech.fr/~manzaner/

Abstract. The Hough transform for detecting parameterised shapes in images is still today mostly applied on binary images of contours or connected sets, which implies pre-processing of the images that may be costly and fragile. However the simple estimation of the spatial derivatives provides in every pixel the local geometry that can be used for dense voting processes, directly applied on the gray scale image. For lines and circles, the local information even allows to perform a direct 1-to-1 projection from the image to the parameter space, which greatly accelerates the accumulation process. In this paper we advocate the use of direct detection on gray scale images by combining Hough transform and multi-scale derivatives. We present the algorithms and discuss their results in the case of analytical shapes for order one (lines), and two (circles), and then we present the generalised Hough transform based on quantised derivatives for detecting arbitrary (non-analytical) shapes.

1 Introduction

Since its introduction in one of the first applications of computer vision [1], the Hough transform has rapidly turned into a classical tool for detecting parameterised shapes in images [2, 3]. In its original form, it is applied on binary images of contours, which implies pre-processing of the images. Then, the transform on binary images is performed using either of the two classical dual approaches: (i) the many-to-1 projection, which picks n-tuples of points from the binary image and select the unique corresponding points in the n-dimensional parameter space, and (ii) the 1-to-many back-projection, which, for every point of the binary image, draws the corresponding (n-1)-manifold in the parameter space.

Remarkably, the interest for Hough transforms remained strong and many variations have been proposed until recently [4, 5]. But it is also remarkable that the proposed algorithms generally follow the original form in the sense that the projection -or voting process- is performed sparsely on contour portions or salient points, using 1-to-many projection (most often), or (generally decimated) many-to-1 projection. We believe that there is a fundamental interest in performing a dense projection, i.e. allowing every pixel to vote, and this can be done directly on the gray level image by estimating the (multi-scale) spatial derivatives. Although

more pixels are voting, these methods must be much faster, first because there is no pre-processing (other than the computation of the derivatives), and second because -in the case of parameterised shapes- one can perform a direct 1-to-1 projection from the image to the parameter space.

The idea of using the local derivatives to accelerate the Hough transform is not new, it has been proposed for lines by O'Gorman and Clowes [6] and for differentiable curves by Shapiro [7]. But those approaches were still used on curves and, to our knowledge, have not been densely applied on gray level images, probably because at the time they were proposed, the techniques for estimating the derivatives on discrete 2d functions, based on finite differences, were not considered precise enough. More recently, Valenti and Gevers [8] have proposed an efficient eye centre location algorithm based on a voting scheme using the scale space curvature estimation. However, they still reduced the voting pixels to a thin contour previously calculated.

In this paper we advocate the use of dense Hough transform directly on the gray level signal using multi-scale derivatives, and weighting the votes by the strength of the derivative (gradient magnitude and Frobenius norm of the Hessian matrix typically). We present the complete algorithms and discuss their results in the case of analytical shapes for order one (lines), and two (circles), and we present the generalised Hough transform based on quantised derivatives for detecting arbitrary (non-analytical) shapes.

2 Analytical shapes

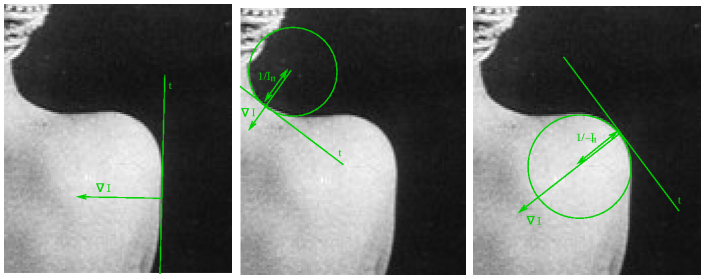
According to the scale space framework [9], the spatial derivatives are estimated in a digital image I relatively to a certain scale σ which represents the level of regularity, explicitly enforced by Gaussian smoothing:

$$I^{\sigma}_{x^i y^j} = I \star \frac{\partial^{i+j} G_\sigma}{\partial x^i \partial y^j} \ , \tag{1}$$

where $\star$ is the convolution, and G_σ the 2d Gaussian function of standard deviation σ. When working at a given scale, we will omit the σ superscript, and denote $\{I_x, I_y, I_{xx}, I_{xy}, I_{yy}\}$ the first and second order derivatives.

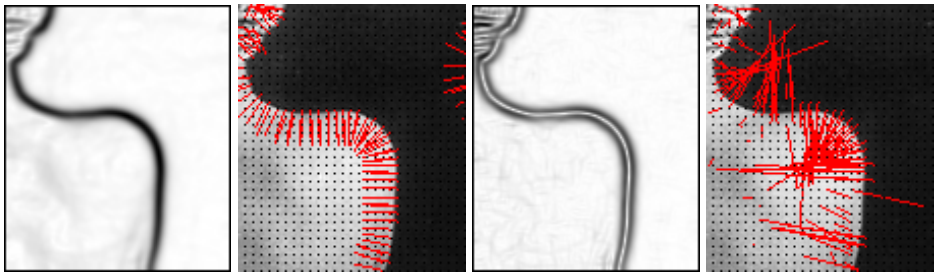
2.1 First order: lines

If $\nabla I = (I_x, I_y)$ is the estimated gradient vector, the value of the first derivative along any direction represented by unit vector $\mathbf{v}$ can also be estimated as $\mathbf{v}^T.\nabla I$. Thus the derivative along the direction orthogonal to the gradient is zero (isophote direction $\mathbf{t}$), and so if there is a line at this location, its orientation must the same as $\mathbf{t}$ (See Figure 1(a)). Now, to evaluate the significance of the location with respect to the presence of line, it is natural to use the strength of the first derivative, i.e. the norm of the gradient $||\nabla I|| = \sqrt{I_x^2 + I_y^2}$ (See Figure 2(a)), which must be normalised according to the scale σ [10], if this



(a) (b) (c)

Fig. 1. Direct detection from the spatial derivatives: Line from the gradient (a), and circle from the isophote positive (b) or negative (c) curvature.



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

Fig. 2. Voting weight at order 1: the norm of the gradient (a). Estimating the gradient direction (b), for pixels with weight over 10.0. Voting weight at order 2: the Frobenius norm of the Hessian matrix (c). Estimating the position of centre of osculating circle (d), for pixels with weight over 1.0. The scale estimation here is $\sigma = 2.0$.

evaluation is done at different scales. The multi-scale voting weight at order 1 is then $\sigma||\nabla I||$. Following the classical (θ, ρ) parameterisation, where ρ is the distance between the line and the origin, and θ is the angle made by the normal to the line with the x axis, the complete algorithm is shown on table 1.

Figure 3 shows the output of the transform, compared to the Many-to-1 transform computed on a contour of the same image. The computation time of the 1-to-1 transform is in fact smaller than the computation time of the contour image (which involves non local maxima suppression and hysteresis thresholding), while the complexity of the many-to-1 transform is one order of magnitude larger for each voting points, because every vote draws a sine curve in the parameter space. The 1-to-1 transform is naturally sparser, because, even if the number of voting pixels is significantly greater (all the pixels vote), only a few of them have significant vote, and more importantly, every pixel vote into one single point. The sparsity, which can be a difficulty in the detection of the local maxima used to select the best lines, can be moderated by interpolating the vote

Table 1. 1-to-1 line Hough transform based on multi-scale gradient.

```
Gamma = function Hough_Lines (Image I)
    forall scale σ ∈ {σ_1, ..., σ_n}
        forall pixel (p_x, p_y) ∈ {0, w_I} × {0, h_I}
            ∇I ← (I_x^σ(p), I_y^σ(p))
            if ||∇I|| > 0:
                d ← p_x I_x + p_y I_y
                ρ ← |d| / ||∇I||
                if (I_x I_y < 0) and (I_y d > 0)
                    θ = π + arctan(I_y / I_x)
                else
                    θ = arctan(I_y / I_x)
                endif
                Gamma(ρ, θ) ← Gamma(ρ, θ) + σ||∇I||
            endif
        endfor
    endfor
end
```

over neighbouring cells according to the quantisation or by explicitly smoothing the transform. But more interestingly, the concurrent use of multiple scales is a benefit for the detection, as illustrated on Figure 4: the finer scales improve the localisation of the main peaks while the coarser scales reduce the influence of the spurious structures. For the results shown on Figure 3, a non-local-maxima deletion was performed in the parameter space, and an exclusion distance of $(\pm\frac{2\pi}{100}, \pm 8)$ was applied to find the best (θ, ρ).

2.2 Second order: circles

If $H_I = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$ is the estimated Hessian matrix, the value of the second derivative along any couples of direction represented by unit vectors $\mathbf{u}$ and $\mathbf{v}$ can be estimated as $\mathbf{u}^T H_I \mathbf{v}$. When $\mathbf{u} = \mathbf{v} = \mathbf{t}$, with $\mathbf{t}$ in the isophote direction, we get:

$$I_{tt} = \frac{I_{xx}I_y^2 - 2I_{xy}I_xI_y + I_{yy}I_x^2}{||\nabla I||^3} \tag{2}$$

This is the second derivative in the direction of the isophote, that is, the estimation of the curvature, which represents the inverse radius of the osculating circle to the isophote curve. Then, if there is a circle at location $(\mathbf{p}_x, \mathbf{p}_y)$ with gradient ∇I and curvature $I_{tt} \neq 0$, the radius of this circle must be $r = \frac{1}{|I_{tt}|}$ and its centre must be $(\mathbf{p}_x, \mathbf{p}_y) - \frac{\nabla I}{I_{tt}||\nabla I||}$ (See Figure 1 (b) and (c)). Again, we can evaluate the significance of the location with respect to the presence of circle by using the strength of the second derivative, i.e. the Frobenius norm of the Hessian matrix $||H_I||_F = \sqrt{I_{xx}^2 + 2I_{xy}^2 + I_{yy}^2}$ (See Figure 2 (c) and (d)), which must be normalised by σ^2 if using different scales. See Table 2 for the complete algorithm.

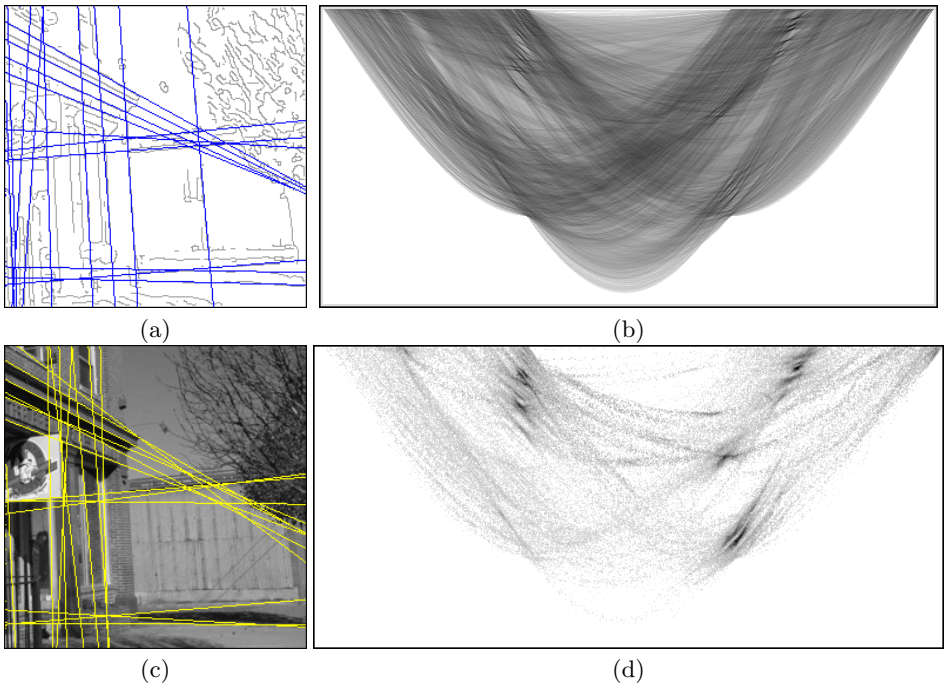
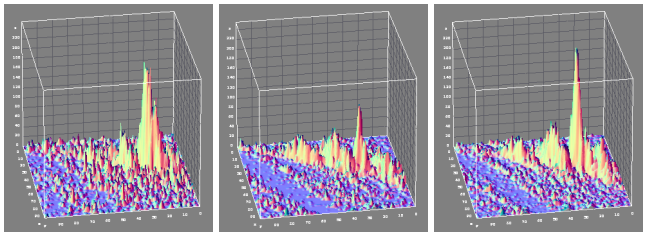


Fig. 3. Top: Many-to-1 line Hough transform (b) computed on the contour image (a): Canny algorithm with $\sigma = 1.5$, hysteresis threshold with $t_1 = 1.0$ and $t_2 = 6.0$. The 20 best lines are overlaid in blue. Bottom: 1-to-1 line Hough transform (d) computed on the grey level image (c) using multi-scale gradient ($\sigma \in \{1.0, 2.0, 4.0\}$).The 20 best lines are overlaid in yellow.

Figure 5 shows results of this algorithm, compared to the many-to-1 Hough transform performed on the contour of the same image. The computation time of the 1-to-1 transform is still smaller than the computation time of the contour image, whereas the computation complexity of the many-to-one transform is 2 order of magnitude more per voting points because every point draws a surface (cone) in the 3d parameter space. The sparsity of the 1-to-1 transform is still more visible than for the lines, because of a larger parameter space. The detection of maxima is then more challenging. For the results shown in Figure 5, we have applied a 3d recursive exponential smoothing filter ($\gamma = 2.0$) in the Hough space, followed by a non-local-maxima deletion and an exclusion distance of $(\pm 1, \pm 3, \pm 3)$ for the detection of the best $(r, \mathbf{c}_x, \mathbf{c}_y)$.

3 Non-analytical shapes

The direct calculation of the Hough transform on the gray scale image using multi-scale derivatives can also be used for the generalised Hough transform to



(a) (b) (c)

Fig. 4. Influence of the multi-scale: Topographic close-up around a maximum of the 1-to-1 line Hough transform, using 1 single fine scale (a): $\sigma = 1.0$, 1 single coarse scale (b): $\sigma = 4.0$ and 3 scales (c): $\sigma \in \{1.0, 2.0, 4.0\}$.

Table 2. 1-to-1 circle Hough transform based on multi-scale gradient and curvature.

```
Gamma = function Hough_Circles (Image I)
    forall scale σ ∈ {σ_1, ..., σ_n}
        forall pixel (p_x, p_y) ∈ {0, w_I} × {0, h_I}
            ∇I ← (I_x^σ(p), I_y^σ(p))
            H_I = ( I_xx^σ(p)  I_xy^σ(p) )
                  ( I_xy^σ(p)  I_yy^σ(p) )
            if ||H_I||_F > 0:
                κ ← I_xx I_y^2 − 2 I_xy I_x I_y + I_yy I_x^2
                r ← ||∇I||^3 / κ
                (c_x, c_y) = (p_x, p_y) − ∇I||∇I||^2 / κ
                Gamma(r, c_x, c_y) ← Gamma(r, c_x, c_y) + σ^2 ||H_I||_F
            endif
        endfor
    endfor
end
```

detect arbitrary objects. In the classical approach [3], the arbitrary shape is a closed contour indexed by the local orientation. Since then, many variations on implicit shape models have been proposed. For example Leibe *et al* [4] use a collection of interest points instead of a contour, and index every point using visual codebook obtained by clustering.

In voting based representation, it is clearly important to have a significant number of voting points. In this sense, we believe that the use of the whole image instead of a collection of contour or interest points is a benefit. Table 3 describes the algorithm used to create the representation (the R-table) of a shape template T which is simply a gray scale image. Every pixel of the template is indexed by a contrast invariant derivative of order 1: the argument of the gradient, and of order 2: the curvature. These derivatives are quantised to limit the size of the R-table, and the curvature is bounded to $[-1, 1]$ (the high curvatures are merged). Every new entry in the R-table add a new element to the list corresponding to the calculated index, which contains the relative coordinates of the voting point

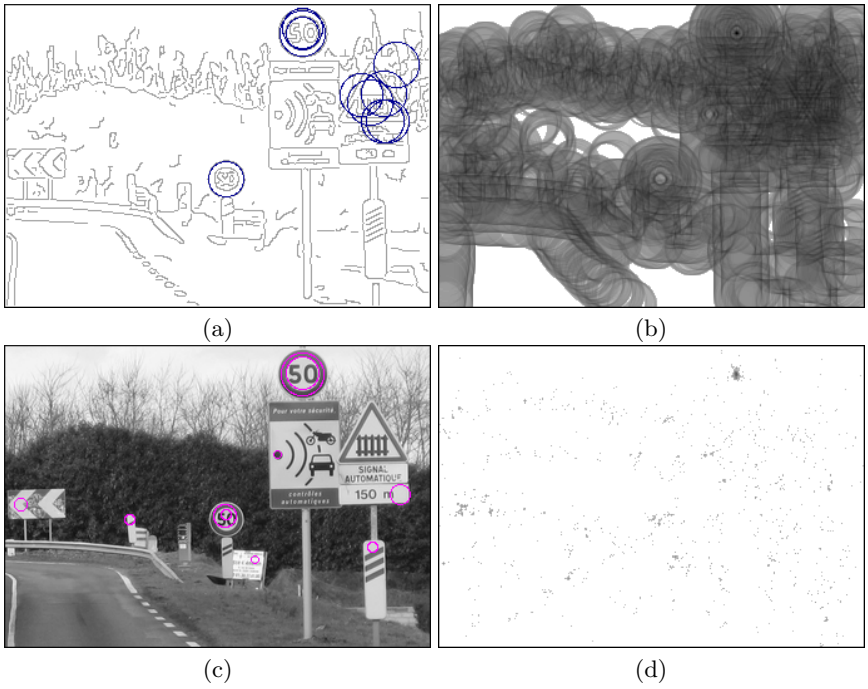


Fig. 5. Top: One plane ($\rho = 19$) from the many-to-1 circle Hough transform (b) computed on the contour image (a): same parameters as Figure 3. The 7 best circles are overlaid in blue. Bottom: The same plane from the 1-to-1 line Hough transform (d) computed on the grey level image (c) using multi-scale curvature ($\sigma \in \{1.0, 2.0, 4.0\}$). The 7 best circles are overlaid in magenta.

with respect to the centre of the template, and also, in conformity with the previous section, the voting weight of the point, corresponding to the magnitude of the gradient or to the Frobenius norm of the Hessian matrix according to the order of the index. Likewise, several scales of estimation can be used, at the cost of multiplying the number of R-tables. Obviously, the size of the R-tables can be reduced by eliminating the entries whose weight is considered too small.

The off-line calculated R-Table is then used for online detection using the generalised Hough transform shown in Table 4, which is basically the same as the classical algorithm, except that every pixel vote according to its orientation and curvature indexes, and its votes are weighted as indicated in the R-table.

One example of construction of the prototype is shown on Figure 6. In this example, one single image template is used, and 4 R-Tables are constructed, corresponding to 2 orders and 2 scales of estimation. The corresponding labels (gradient orientation and curvature) are quantised to 30 values which form the number of indexes of the R-tables. The calculation of the general Hough transform is shown on Figure 7 on a composite image of side viewed cars (All images

Table 3. R-Tables at order 2 calculated on a gray scale template T.

```
RT = function Create_R-Table (Template T, scale σ)
    forall pixel (p_x, p_y) ∈ {0, w_T} × {0, h_T}
        ∇I ← (I_x^σ(p), I_y^σ(p))
        H_I = ( I_xx^σ(p)  I_xy^σ(p) )
              ( I_xy^σ(p)  I_yy^σ(p) )
        if ||∇I|| > 0
            α = arctan I_y / I_x
            ξ ← (I_xx I_y^2 − 2 I_xy I_x I_y + I_yy I_x^2) / ||∇I||^3
            if ξ < −1 then ξ ← −1
            else if ξ > 1 then ξ ← 1
            RT(1, σ, α) ← RT(1, σ, α) ∪ (w_T/2 − p_x, h_T/2 − p_y, ||∇I||)
            RT(2, σ, ξ) ← RT(2, σ, ξ) ∪ (w_T/2 − p_x, h_T/2 − p_y, ||H_I||_F)
        endif
    endfor
end
```

are taken from the image database of UIUC for car detection [11]). For selecting the best detections, an exclusion distance corresponding to the quarter of the template sizes was used. What can be seen from this experiments is that the general Hough transform calculated from gray scale derivatives keeps the good properties of the transform on contours: invariance to contrast changes, robustness to occlusions, while being faster to compute and less sensitive to poor contrast in the detection, because every pixel is voting according to the significance of its counterpart in the template.

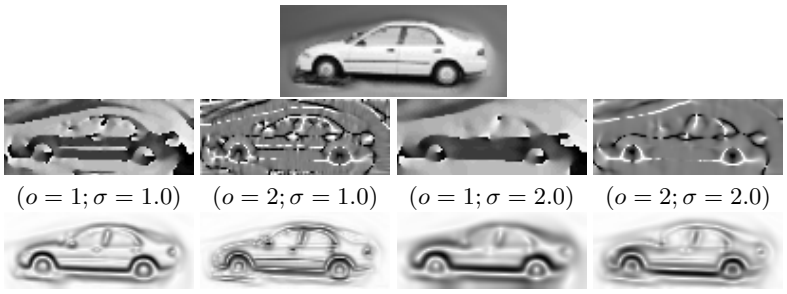


Fig. 6. Object template (top). Labels used as indexes of the R-tables (middle line), at order 1 and 2, for two scales, and their associated weights (bottom line).

Table 4. Generalised Hough transform on gray scale image at order 2.

```
Gamma = function Hough_General (Image I,R–Table RT)
    forall scale σ ∈ {σ_1, ..., σ_n}
        forall pixel (p_x, p_y) ∈ {0, w_I} × {0, h_I}
            ∇I ← (I_x^σ(p), I_y^σ(p))
            H_I = ( I_xx^σ(p)  I_xy^σ(p) )
                  ( I_xy^σ(p)  I_yy^σ(p) )
            if ||∇I|| > 0
                α = arctan I_y / I_x
                ξ ← (I_xx I_y^2 − 2 I_xy I_x I_y + I_yy I_x^2) / ||∇I||^3
                if ξ < −1 then ξ ← −1
                else if ξ > 1 then ξ ← 1
                forall (δ_x, δ_y, ω) ∈ RT(1, σ, α)
                    Gamma(p_x + δ_x, p_y + δ_y) ← Gamma(p_x + δ_x, p_y + δ_y) + ω
                endfor
                forall (δ_x, δ_y, ω) ∈ RT(2, σ, ξ)
                    Gamma(p_x + δ_x, p_y + δ_y) ← Gamma(p_x + δ_x, p_y + δ_y) + ω
                endfor
            endif
        endfor
    endfor
end
```

4 Concluding remarks

Our purpose in this paper was to convince that the dense Hough transform on gray level images using multi-scale derivatives is interesting both in terms of robustness (thanks to a higher number of weighted votes), and computational efficiency (thanks to lighter pre-processing and 1-to-1 vote). It seems hard to design a systematic evaluation to decide more objectively when dense derivatives should be preferred to sparse contours, because it hardly relies on the quality of the contour. Naturally, it can be said that contours will work better for finding curves made of discontinuous structures (e.g. finding alignments in plant fields). In the general case, a more thorough validation is needed to evaluate the influence of the chosen scales and weights. We are also planning to design a more general framework to apply the dense Hough transform for object detection based on multiple derivatives.

References

1. Hough, P.: Machine analysis of bubble chamber pictures. In: Int. Conf. High Energy Accelerators and Instrumentation. (1959)
2. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Com. of the Association for Computing Machinery **15** (1972) 11–15
3. Ballard, D.H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition **13** (1981) 111–122
4. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision. (2004)

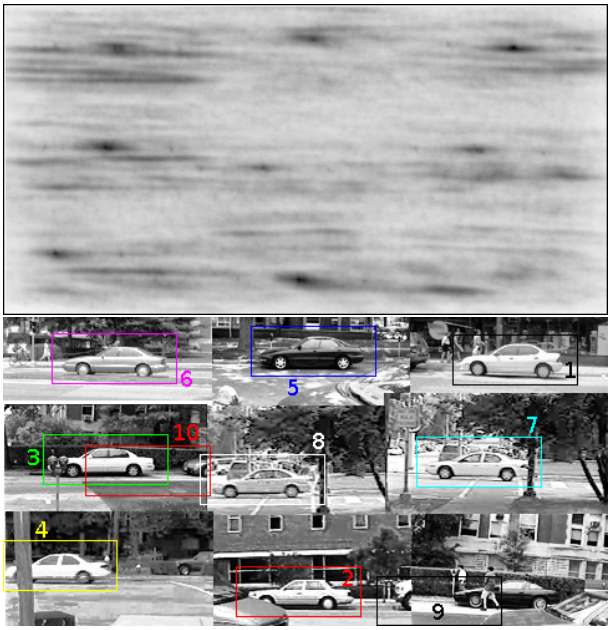


Fig. 7. General Hough transform calculated on a composite image of side viewed cars, using the R-tables obtained from the template of Figure 6. The 10 best detections are shown as overlaid rectangles, numbered by order of detection.

5. Ferrari, V., Jurie, F., Schmid, C.: From images to shape models for object detection. International Journal of Computer Vision **87** (2010)
6. O'Gorman, F., Clowes, B.: Finding picture edges through collinearity of feature points. IEEE Trans. on Computers **C-25** (1976) 449–456
7. Shapiro, S.: Feature space transforms for curve detection. Pattern Recognition **10** (1978) 129–143
8. Valenti, R., Gevers, T.: Accurate eye center location and tracking using isophote curvature. In: Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Anchorage, Alaska (2008)
9. Florack, L., Ter Haar Romeny, B., Viergever, M., Koenderink, J.: The Gaussian scale-space paradigm and the multiscale local jet. Int. J. of Computer Vision **18** (1996) 61–75
10. Lindeberg, T.: Feature detection with automatic scale selection. Int. J. of Computer Vision **30** (1998) 77–116
11. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. IEEE Trans. on Pattern Analysis and Machine Intelligence **26** (2004) 1475–1490

2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems

Local Jet Feature Space Framework for Image Processing and Representation

Antoine Manzanera
Laboratoire d'Electronique et Informatique
ENSTA-ParisTech, Paris, France
http://www.ensta-paristech.fr/~manzaner

***Abstract*—We present a unified framework for processing and representing images using a feature space related to local similarity. The visual data is represented by the versatile multiscale local jet feature space, possibly reduced by vector quantisation and/or represented by data structures enabling efficient nearest neighbours search (*e.g.* kd-trees). We demonstrate the interest of the local jet feature space processing through three fundamental low level tasks: noise reduction, motion estimation and background modelling/subtraction. We also show the potential of the framework in terms of higher level visual representation (*e.g.* recognition/retrieval).**

***Keywords*-vision framework; multiscale local jets; similarity space; nearest neighbours; optical flow; non local means; background modelling;**

I. Introduction

Many problems in image processing and vision relate to visual similarity. Since the earliest processes of denoising or perceptual grouping, to the higher level tasks of object recognition, measuring the resemblance, or matching two objects according to the visual appearance are fundamental functions. In the traditional space $\times$ time representations of video sequences, the canonical distance is not related to visual similarity, which induces a major computational drawback. Indeed, similar objects from the video data are expected to interact in the processing, and then should be contiguous in the representation. These general remarks do not only apply to the current data, image or recent frames history, but also to the global visual knowledge that the vision system is constructing during its operating lifetime.

The purpose of this work is to design a global, generic and computationally tractable framework for the representation and the processing of the visual data, based on: (1) the projection of the space $\times$ time image data within a transformed space whose metrics correspond to visual similarity, (2) a set of functions operating in the transformed and/or the image domain, for extracting relevant information from the video, updating the transformed domain structure accordingly, and/or modifying the video in the image domain according to some specific task (filtering, detecting, predicting), and (3) dedicated data structures for making such framework computationally feasible, in terms of memory and processing time. In our philosophy, such unified framework should be usable for the whole vision process, from the lowest level of regularisation and enhancement to the levels of higher semantics related to recognition and understanding. The framework should also be compliant with real-time video processing, which implies both dynamical and efficient construction of the visual representation.

The inspiring and related works are presented in Section II. In our work the preferred similarity space is made of the collection of spatial derivatives estimated at different scales (the local jet). This feature space and its data structure are presented in Section III. The following sections present the applications of the framework for different low level visual processing tasks: non-local means image denoising (Sec. IV), optical flow estimation (Sec. V) and background subtraction based motion detection (Sec. VI). Section VII presents some visual models that can be extracted from the feature space data structure, for higher level representations.

II. Related works

Our work is related to Peyré's manifold model [1]. In this theoretical framework, the image data is projected within a higher dimensional feature space, forming a manifold. Many inverse problems in low level computer vision can be expressed by regularising this manifold and then back-projecting the transformed manifold within the image space. In this sense, the different low level algorithms proposed in this paper can be seen as instances of the manifold model. Conversely, our work is also an extension of this model, with the aim of extracting higher level representations from the manifold structure.

Our framework exploits many ideas from previous works on textured objects modelling, segmentation and recognition. Filter banks have been used for a long time as a way to extract meaningful local information on direction, scale, and frequency [2]. Quantising such information is also a commonplace in textons [3] or bag of features [4] approaches. Compared with those methods, one fundamental property of our framework is that the feature is intrinsically dense in the image space, making the corresponding information available at any location. Another particularity of our work is that reducing the information support is done by finding the isolated or clustered points in the feature space, thus avoiding the common separation between detection and description of the salient structures [5], [6].

The importance of the local jet in image representation has been identified a few decades ago. Koenderink and

978-0-7695-4635-3/11 $26.00 © 2011 IEEE
DOI 10.1109/SITIS.2011.49

Van Doorn [7] pointed out the fundamental role of the first three orders of derivatives in the human visual system. They also noticed that some Euclidean distance on the local jet vectors could be used to approximate the sum of squared differences between image patches. To our knowledge this has not been really used in the literature, maybe because the approximation is crude for complicated patches. But, as we will see later, distances based on the local jet are actually significant to distinguish similar pixels.

Anyway, the local jet has been much used for the construction of invariants, particularly in image retrieval [8]. It has also been used more recently for the classification of pixels according to their local geometry, see for example [9]. As shown later, such classification can be exploited to reduce the dimension of the local jet descriptor.

III. Multiscale Local Jet Feature Space

A. Similarity space

Using the partial derivatives to measure the local similarity is a natural choice [10] since the local behaviour of any differentiable function f can be predicted from its derivatives (Taylor expansion at order r):

$$f(\mathbf{x}+\mathbf{c}) = \sum_{k=0}^{r}\sum_{i=0}^{k}\binom{k}{i} c_1^{k-i} c_2^{i} \frac{\partial^k f}{\partial \mathbf{x}_1^{k-i}\partial \mathbf{x}_2^{i}}(\mathbf{x}) + o(||\mathbf{c}||^r) \ , \tag{1}$$

with $\mathbf{x}_1$ and $\mathbf{x}_2$ a basis of $\mathbb{R}^2$, in which the components of the residual $\mathbf{c}$ are $c_1 = \mathbf{c}\cdot\mathbf{x}_1$ and $c_2 = \mathbf{c}\cdot\mathbf{x}_2$. To simplify we denote $f_{ij} = \frac{\partial^{i+j} f}{\partial \mathbf{x}_1^i \partial \mathbf{x}_2^j}$. The relevance of the local jet as a description vector is confirmed by the first singular (or eigen) vectors that arise in SVD or PCA based decomposition of natural image patches, that resemble the first derivatives of a 2d Gaussian function (see [11]). In digital images, the derivative only makes sense up to a level of regularity corresponding to the scale of estimation [12]:

$$f_{ij}^{\sigma} = f \star \frac{\partial^{i+j} G_\sigma}{\partial \mathbf{x}_1^i \partial \mathbf{x}_2^j} \ , \tag{2}$$

where G_σ is the 2d Gaussian function of standard deviation σ. The multiscale local jet is then the collection $\{f_{ij}^{\sigma}; i+j \leq r, \sigma \in S\}$, where r is the order of derivation, $S = \{\sigma_1, \ldots \sigma_q\}$ the selected scales. Figure 1 illustrates the induced representation for a few points taken from a natural image, at one scale $\sigma = 1.0$. The image is split into 15×15 patches, and the reconstruction is performed by Taylor expansion on patches of the same size, using only the local jet computed at the patch centre.

Our representation does not use patches but a multiscale local jet vector in every pixel, with normalised components combining the scale normalisation from scale space theory [12], and the number of $(i+j)$-order derivatives:

$$F_{ij}^{\sigma} = \frac{\sigma^{i+j}}{i+j+1} f_{ij}^{\sigma} \ . \tag{3}$$

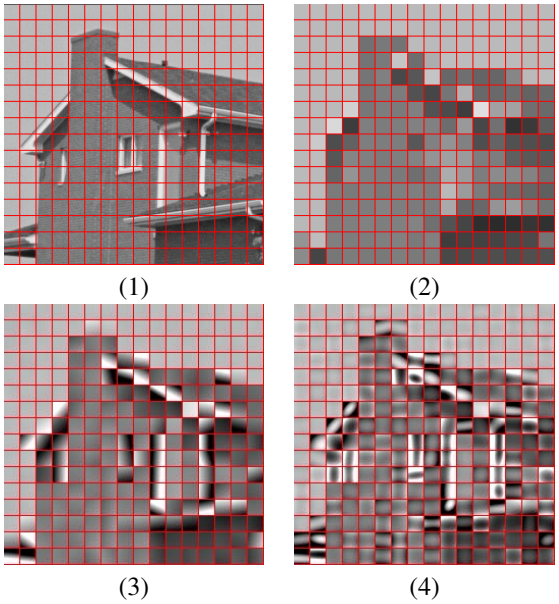


Figure 1. Local jet based representation at one scale ($\sigma = 1.0$): (1) Original patches (2) Order 0 (1d feature) (3) Order 1 (3d), (4) Order 2 (6d)

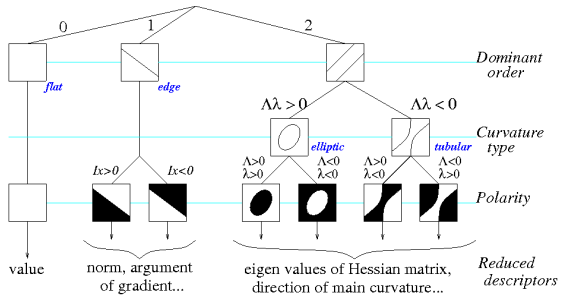


Figure 2. Local jet descriptor reduction by pixel categorisation: 4 categories at order 2 (7 categories if the polarity is considered).

If required, rotation invariant derivatives can be obtained by expressing them within the local basis of coordinates of the gradient and isophote components. The local jet also provides contrast invariant measures, *e.g.* the direction of the gradient/isophote at order 1, or the direction of main curvatures (eigen vectors of the Hessian matrix) at order 2. Finally, following [9], the local behaviour of every pixel can be categorised at every scale according to the dominant order of derivation: flat zone for order 0, straight contours for order 1, and elliptic or tubular curvatures for order 2 (according to the signs of Λ and λ, the eigen values of the Hessian matrix). Once categorised, the dimensions of the local jet descriptor can be reduced to significant derivatives (see Figure 2).

B. Metrics

To measure similarity, we typically consider three types of distance in the applications. Let F be the full local jet vector

$F = (F^{\sigma}_{ij})_{i+j\leq r,\sigma\in S}$, we denote $\hat{\mathbf{x}} = F(\mathbf{x})$ the feature vector associated to pixel $\mathbf{x}$. Let $F^{\sigma} = (F^{\sigma}_{ij})_{i+j\leq r}$ be the local jet at scale σ, and $||.||$ the Euclidean norm. First, the *single scale* distance, checking whether $\mathbf{x}$ at scale σ_1 is similar to $\mathbf{y}$ at scale σ_2:

$$d_F^{(\sigma_1,\sigma_2)}(\hat{\mathbf{x}},\hat{\mathbf{y}}) = ||F^{\sigma_1}(\mathbf{x}) - F^{\sigma_2}(\mathbf{y})|| \ . \tag{4}$$

Second, the *pan-scalic* distance, checking whether $\mathbf{x}$ and $\mathbf{y}$ are similar for all the scales $\sigma \in S$:

$$D_F^S(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \max_{\sigma\in S} d_F^{(\sigma,\sigma)}(\hat{\mathbf{x}},\hat{\mathbf{y}}) \ . \tag{5}$$

However for representation purposes (next subsection) the use of Euclidean distance $||\hat{\mathbf{x}} - \hat{\mathbf{y}}||$ can be more convenient.

Third, the *trans-scalic* pseudo-distance, checking whether there exists a couple of scales for which $\mathbf{x}$ and $\mathbf{y}$ are similar:

$$\delta_F^S(\hat{\mathbf{x}},\hat{\mathbf{y}}) = \min_{(\sigma_n,\sigma_m)\in S^2} \max_{(\sigma_{n+p},\sigma_{m+p})\in S^2} d_F^{(\sigma_{n+p},\sigma_{m+p})}(\hat{\mathbf{x}},\hat{\mathbf{y}}) \ . \tag{6}$$

The interest of using local jet based distances is not to approximate classical (e.g. sum of squared differences between patches) similarity distances, but to provide more flexible similarity space and measures (see Figure 3 showing different rotation and scale invariance properties), while reducing significantly the dimension of descriptors.

C. Data structures

The first step of the representation then consists in projecting the image data into the chosen similarity space. For every pixel, a feature vector is computed, and the collection of features is kept in adequate data structure for further processing. If the feature space dimensionality is low, the data structure may be a simple array, whose coordinates are indexed by each component of the feature space, which must then be quantised properly. The data structure is then a hash table whose hash function is the quotient of the quantisation. As the memory cost of such structure grows exponentially with the dimension, other solutions must be used for higher dimension, like the classical kd-tree [13], which is optimal in terms of memory occupation.

The kd-tree is a useful tool for performing nearest neighbours (NN) search in the feature space. It will be extensively used in the following to perform efficiently operations based on visual similarity, that are intrinsically non local in the image space. However, there are many operations where NN search will be needed very intensively (*e.g.* for every pixel / feature vector). In that case, the computational cost will remain too important for real-time video. Two important optimisations are employed: (1) Approximate Nearest Neighbour (ANN) search techniques [14] that reduces both worst case and average search complexity, and (2) Quantisation of the feature space, that reduces the size of the kd-tree. We have used in our experiments the ANN library developed by Arya and Mount [14], and a simple approximation of the K-means clustering method for vector quantisation. Depending on the used metrics, one kd-tree per scale or one single kd-tree has to be calculated. The figure 4 illustrates in dimension 2 the projection in the feature space and the construction of the kd-tree, without and with quantisation.

Figure 3. Similarity maps based on 2-order local jet metrics, for 3 different pixels: (1), (2) and (3), and 6 different distances: Single (same) scale: (a) canonical components (CC), (b) rotation invariant components (RIC), Pan-scalic (4 scales): (c) CC, (d) RIC, and Trans-scalic (4 scales): (e) CC, (f) RIC. For each pixel $\mathbf{x_0}$, the similarity map is defined as $M^{d_F}_{\mathbf{x_0}}(\mathbf{x}) = \Phi(d_F(\hat{\mathbf{x}}_0,\hat{\mathbf{x}}))$, with d_F the local jet distance, and $\Phi(z) = 1 - e^{-\frac{z^2}{C^2}}$, with $C = 10$. (Painting by Lowell Herrero)

D. Useful notations

Let $\mathbf{x}$ be a pixel from the image space. We denote $\hat{\mathbf{x}}_f$ the projection of $\mathbf{x}$ in the feature space of image f. Let $\mathcal{F}_f$ be the set of features of image f. If $\mathbf{u}$ is a feature vector, let $\nu_k^{\mathcal{F}_f}(\mathbf{u})$ be its k-th nearest neighbour in the feature space of f. We denote $\mathcal{F}_f^{-1}(\mathbf{u})$ the set of pixels which are assigned to the codeword $\mathbf{u}$ in the quantised feature space (codebook) of f. If there is no quantisation, the notation remains valid

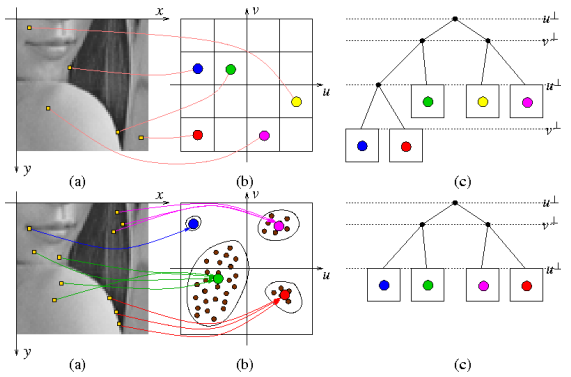


Figure 4. (a) Image data (b) projected into the feature space, and (c) collected into a kd-tree structure, without (top) and with (bottom) vector quantisation.

as $\mathcal{F}_f^{-1}(\mathbf{u}) = \{\mathbf{x}\}$ such that $\hat{\mathbf{x}}_f = \mathbf{u}$.

IV. Non-local means video filtering

The non-local (NL) means filter, originally proposed by Buades *et al* [15] is a powerful image denoising technique, in which every pixel value is replaced by a weighted average of the other pixels, the weights depending on pixel similarity, not on pixel distance in the image space (hence the "non local" property). In our framework, the NL-means is simply expressed by calculating the weights using a distance in the feature space. Let $\mathbf{u}$ and $\mathbf{v}$ be two feature vectors. $\omega(\mathbf{u}, \mathbf{v})$ the relative (symmetric) weight of $\mathbf{u}$ with respect to $\mathbf{v}$, is defined as follows:

$$\omega(\mathbf{u}, \mathbf{v}) = e^{-\frac{d_F(\mathbf{u},\mathbf{v})^2}{h^2}} , \tag{7}$$

where h is a decay parameter, related to the amount of noise to be removed. Now two variants of the NL means can be considered:

1) Limited range (LR) method

$$f_{LR}^{NL}(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} f(\mathbf{y})\omega(\hat{\mathbf{x}}_f, \hat{\mathbf{y}}_f)}{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \omega(\hat{\mathbf{x}}_f, \hat{\mathbf{y}}_f)} \tag{8}$$

2) Unlimited range (UR) method

$$f_{UR}^{NL}(\mathbf{x}) = \frac{\sum_{\mathbf{u}\in\mathcal{W}(\hat{\mathbf{x}}_f)} \check{f}(\mathbf{u})\omega(\hat{\mathbf{x}}_f, \mathbf{u})}{\sum_{\mathbf{u}\in\mathcal{W}(\hat{\mathbf{x}}_f)} \omega(\hat{\mathbf{x}}_f, \mathbf{u})} \tag{9}$$

where $\mathcal{N}(\mathbf{x})$ (resp. $\mathcal{W}(\mathbf{v})$) is a neighbourhood of $\mathbf{x}$ (resp. $\mathbf{v}$), corresponding to the k nearest neighbours of $\mathbf{x}$ (resp. $\mathbf{v}$) in the image (resp. feature) space. See figure 5.

$\check{f}(\mathbf{u})$ is defined as:

$$\check{f}(\mathbf{u}) = \frac{1}{|\mathcal{F}_f^{-1}(\mathbf{u})|} \sum_{\mathbf{x}\in\mathcal{F}_f^{-1}(\mathbf{u})} f(\mathbf{x}) , \tag{10}$$

i.e. the average value of f on the pixels corresponding to feature $\mathbf{u}$ (recursively calculated during the quantisation).

Figure 5. Limited (a) *vs* Unlimited (b) range approaches in the computation of the NL-means. $\mathcal{N}(\mathbf{x})$: nearest neighbours of $\mathbf{x}$ in the image space. $\check{\mathcal{W}}(\hat{\mathbf{x}})$: nearest neighbours of $\hat{\mathbf{x}}$ in the feature space back-projected in the image space.

In our experiments, the decay parameter h is automatically adjusted, using a fast estimation of the noise variance (see [16] for more details).

Figure 6. NL-means filtering in the local jet feature space. (1) LR ($\mathcal{N}(\mathbf{x})$: 17×17 neighbourhood of $\mathbf{x}$ in the image domain). (2) UR, exact search ($\mathcal{W}(\hat{\mathbf{x}}_f)$: 30 NN in the local jet domain), (3) UR, approximate search ($\varepsilon = 10.0$), (4) UR, approx. search, with quantised feature space (1938 words in the dictionary).

It can be said that the local jet based NL-means, by changing the order of derivation and number of scales, form a *continuum* between tone space (or bilateral) filtering and patch based NL-means. However, even at one single scale, the order 2 local jet based NL-means results are very close of patch based ones. See figure 6 for some results on the same noisy image (only the top half diagonal is processed). It is somewhat surprising that the denoising quality looks better for the LR (Fig. 6(1)) than for the UR (Fig. 6(2)). But on the one hand, the edge and corner pixels are more affected by the UR methods, the relative weights of their

neighbours being much higher in the feature than in the image space. One the other hand, for large noisy homogeneous regions, the UR method is able to find patterns that tends to exaggerate the texturing of these regions. Using kd-trees, the UR method is generally faster than the LR one since the cardinality of $\mathcal{W}(\hat{\mathbf{x}}_f)$ is usually much smaller than for $\mathcal{N}(\mathbf{x})$. Furthermore, using approximate search (Fig. 6(3)), and quantising the local jet space (Fig. 6(4)) significantly lowers the computation time, while partially compensating the drawbacks of the UR method evoked above, but more quantitative evaluation is needed.

V. Optical flow estimation

The apparent motion, or optical flow estimation turns out to be - from a conceptual point of view at least - one of the most straightforward applications of the feature space based similarity. At frame t, for image f_t, and for every pixel $\mathbf{x}$, we compute $\mathbf{u}(f_{t-1}, f_t, \mathbf{x})$, the nearest neighbour of the feature vector associated to $\mathbf{x}$, in the feature space of f_{t-1}:

$$\mathbf{u}(f_{t-1}, f_t, \mathbf{x}) = \arg \min_{\mathbf{v} \in \mathcal{F}_{f_{t-1}}} d_F(\hat{\mathbf{x}}_{f_t}, \mathbf{v}) = \nu_1^{\mathcal{F}_{f_{t-1}}}(\hat{\mathbf{x}}_{f_t}) \ . \tag{11}$$

Then we can compute $\mathbf{y}(f_{t-1}, f_t, \mathbf{x})$, the pixel from f_{t-1} which is the most similar to $\mathbf{x}$ from f_t:

$$\mathbf{y}(f_{t-1}, f_t, \mathbf{x}) = \arg \min_{\mathbf{z} \in \mathcal{F}^{-1}_{f_{t-1}}(\mathbf{u}(f_{t-1}, f_t, \mathbf{x}))} d_I(\mathbf{x}, \mathbf{z}) \ , \tag{12}$$

with d_I the distance in the image space. Without quantisation, this is simply the pixel corresponding to feature $\mathbf{u}$ in f_{t-1}, otherwise it is the pixel from the set of pixels associated to codeword $\mathbf{u}$ which is the closest from $\mathbf{x}$ in the image space: see Figure 7.

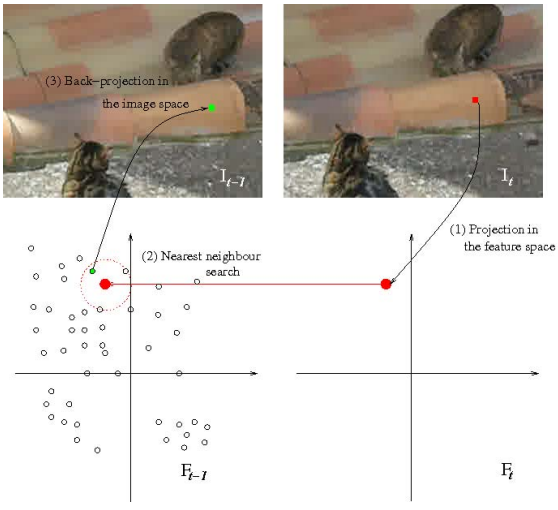


Figure 7. Optical Flow estimation by Nearest Neighbour Search in the Local Jet Feature Space.

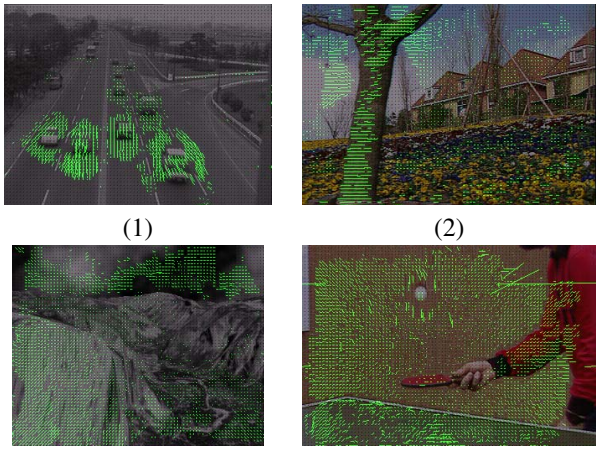


Figure 8. Optical flow fields estimated by NN search in the feature space. (1) Stationary camera, (2) Horizontal travelling, (3) Forward zooming, (4) Backward zooming and moving objects.

Finally, the velocity vector is computed as the difference:

$$\mathbf{c}(f_{t-1}, f_t, \mathbf{x}) = \mathbf{x} - \mathbf{y}(f_{t-1}, f_t, \mathbf{x}) \ . \tag{13}$$

At one single scale the result is hardly usable, but using several scales, the method provides a dense estimation without explicit regularisation of the vector field, that allows a fair estimation of the motion at a global level: See Figure 8 for examples taken from classical test sequences. Here the number of NN is 1, the local jet is at order 2, and 5 different scales, without quantisation.

VI. Background subtraction

Background modelling and subtraction is a popular approach of motion detection. It consists in calculating locally (say for every pixel or block), a set of temporal statistics measures of the background, and comparing every new value with those measures, to decide whether this value is typical or not. This problem is challenging in many cases where the background is not completely static. The precision of modelling has strong influence on the computational cost, in terms of memory and time. One good trade-off is obtained by the sample and consensus methods [17], [18], which consist in keeping in memory a limited set of sampled values, and then comparing the current value to those samples, to decide whether the pixel is background or not. Vector quantisation has also been used for background modelling [19] in colour/brightness space. The algorithm we propose here is a combination of sample/consensus and vector quantisation in the local jet feature space.

In this application, we use for the whole sequence one single codebook $\mathcal{F}_f$ of quantised features, but that may evolve over time. The principle is the following: In every pixel $\mathbf{x}$, the temporal activity is modelled by a set of M prototypes $\Pi(\mathbf{x}) = \{\mathbf{m}_j(f, \mathbf{x})\}_{j \in \{1,M\}} \subset \mathcal{F}_f$, that represent

a sample of its past values in the feature space, and M is a temporal depth parameter.

Let ρ be a positive number; τ an integer such that $1 < \tau < M$; let $\mathcal{B}^{d_F}(\mathbf{u}, r)$ be the ball of centre $\mathbf{u}$ and radius r for the distance d_F. The foreground label $e(f, t, \mathbf{x})$, indicating whether $\mathbf{x}$ in f_t belongs to a moving object or not is calculated as follows:

$$e(f,t,\mathbf{x}) = 1 \text{ if } |\Pi(\mathbf{x}) \cap \mathcal{B}^{d_F}(\hat{\mathbf{x}}_{f_t}, \rho)| < \tau \ , \quad (14)$$

$$= 0 \text{ otherwise.} \quad (15)$$

Then a pixel whose feature vector is at a distance smaller than ρ for less than τ of its M prototypes is considered foreground, elsewhere it is classified as background. The advantage of using a complex feature space instead of the mere colour is that we are able to capture more sophisticated image structure and then make the background modelling more robust. On the other hand, the vector quantisation dramatically reduces the memory cost, because only the index of the word from the codebook is used instead of a high dimensional vector. It is typically observed that a large majority of pixels only have one or two different indexes within their M background prototypes, whereas some more complicated background pixels (*e.g.* waving trees) can have much more indexes.

Our practical implementation for coding and updating the prototypes is a simple adaptation of the state-of-the-art *ViBe* algorithm [18]: The pixel prototypes are represented by a list of codebook indexes and weights (frequencies) such that the sum of weights is M. At time t the index of $\hat{\mathbf{x}}_{f_t}$ replaces one of the prototypes randomly selected, by decrementing the weight of one prototype, then incrementing the weight of another one or creating a new prototype index (See Figure 9).

For the creation of the codebook, we use, as in the NL-mean case a basic incremental version of the K-means algorithm for real-time video purposes. It is worth mentioning that the codebook does not need to be updated for every frame, nor everywhere, for example, it can be updated every 5 frames for the foreground pixels, and every 100 frames for the whole image. See Figure 10 for an example of foreground labelling in an outdoor colour sequence, using a 2 order, 1 scale, and 3 colour local jet feature space (*i.e.* 18D vector features), with a codebook of 3,000 words, a temporal depth $M = 20$, distance threshold $\rho = 0.08 d_{\max}$, and consensus threshold $\tau = M/2$. Note that, unlike [18], no spatial diffusion is performed, and the update is not strictly conservative, *i.e.* the update is made every 4 frames for background pixels, and every 16 frames for foreground pixels.

VII. Image and object characteristics

In our framework, the feature space should be used not only for image processing, but also for extracting relevant visual representation, usable at a higher level. The first

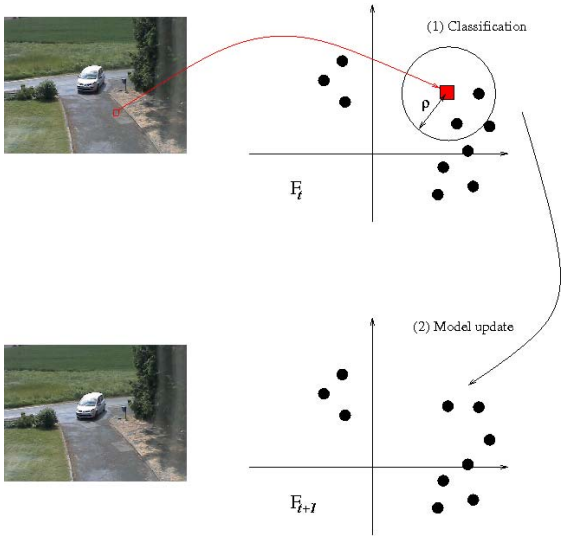


Figure 9. Adaptation of the ViBe algorithm to the local jet feature dictionary. Top, classification step: The pixel $\mathbf{x}$ is classified foreground if the number of prototypes at distance less than ρ from $\hat{\mathbf{x}}_{f_t}$ is inferior to a certain threshold. Bottom, update step: $\hat{\mathbf{x}}_{f_t}$ replaces one of the prototypes, randomly selected.

Figure 10. Background subtraction based on sample and consensus using a codebook of colour local jet features.

descriptor we can consider is the quantised local jet itself (parented to the classical texton approaches), whose statistics provide information on the visual appearance of objects (like in the classical bag of features methods). The histogram, or weight vector of the codebook is computed recursively during the quantisation, or the updating of the codebook. Figure 11 shows an example of local jet quantisation back-projected in the image space. The detail image (right) illustrates one advantage of the dense representation, with the possible use in terms of higher order statistics (*i.e.* co-occurrence) of visual words from the codebook.

The nearest neighbour framework also provides an interesting new conception of salient points. Whereas the classical characterisation of interest points is purely geometrical

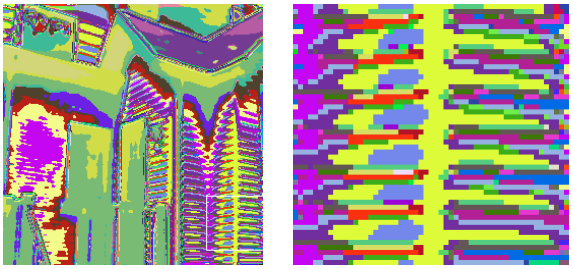


Figure 11. Quantisation of the local jet space (506 vectors). The right image is a detail of the white rectangle in the left image.

and relatively independent of the image content, the NN feature based salience is entirely statistical and content-dependent: The salient points correspond to the isolated points in the feature space. This has been done before in the space of patches by Kervrann and Boulanger [20]. More formally, the rarest pixels are defined as:

$$R_1^{\mathcal{F}_f} = \mathcal{F}_f^{-1}(\arg\max_{\mathbf{u}\in\mathcal{F}_f} \frac{1}{m}\sum_{k=1}^{m} d_F(\mathbf{u}, \nu_k^{\mathcal{F}_f}(\mathbf{u}))\ . \qquad (16)$$

The rarest pixels are those assigned to the word with maximal average distance to its m nearest neighbours. Without quantisation, there is only one such pixel. The second rarest pixels $R_2^{\mathcal{F}_f}$ are defined similarly by excluding the word with maximal distance and so on. The only parameter m merely acts as a filtering value and is of moderate practical importance. Figure 12 shows examples of NN based salient points in a single scale local jet feature space. The difference with the geometric approach is clearly visible on the left image.

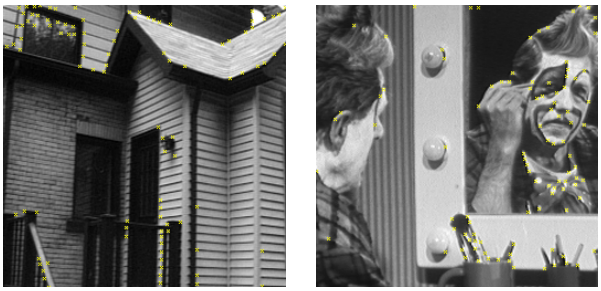


Figure 12. Salient points (isolated points in the feature space back-projected in the image): 100 rarest pixels ($m = 10$, Local jet of order 2, one single scale $\sigma = 1.5$, no quantisation); a minimal exclusion distance of 5 pixels is used to avoid clustering of the salient pixels.

Finally we propose another descriptor whose purpose is to provide an intermediate representation between the global codebook histogram and local salient point. It is based on the statistical modes of the feature space. Mode selection in multidimensional data is a difficult problem which has received relatively few attention. We use an adaptation of the method proposed by Burman and Polonik [21], implemented through the framework of geodesic reconstruction in the feature space.

Suppose defined a topology in the feature space, and let the centre of the main cluster $\kappa_1^{\mathcal{F}_f}$ be defined as the feature vector with minimal average distance to its m NN. The main cluster $K_1^{\mathcal{F}_f}$ is then defined as the connected component of $\mathcal{F}_f$ that contains $\kappa_1^{\mathcal{F}_f}$, or equivalently the geodesic reconstruction of $\kappa_1^{\mathcal{F}_f}$ within $\mathcal{F}_f$. The second main cluster $K_2^{\mathcal{F}_f}$ is defined the same way on $\mathcal{F}_f \setminus K_1^{\mathcal{F}_f}$, and so on. Now we get a topology which dynamically adapts to the data by using a distance threshold defined as the geometric mean between $\mu_m^{\mathcal{F}_f}$ and $\tau_m^{\mathcal{F}_f}$, respectively the average and minimal mean distance of a feature vector to its m NN. Then two feature vector $\mathbf{u}$ and $\mathbf{v}$ are connected if and only if: $d_F(\mathbf{u},\mathbf{v}) < \sqrt{\mu_m^{\mathcal{F}_f}\tau_m^{\mathcal{F}_f}}$. Figure 13 shows the result for 2 images. The modes appear as a complementary information of singularities (Figure 12). They represent homogeneous zone, simple regular textures, or long straight contours.

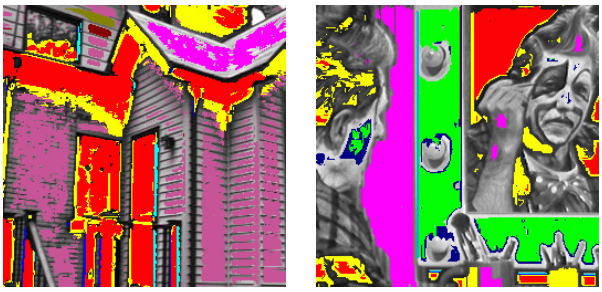


Figure 13. 12 first modes of the local jet representation : clusters of the feature space back-projected in the image space. ($m = 20$, local Jet of order 2, with 2 scales $\{\sigma_1 = 1.0, \sigma_2 = 2.0\}$, no quantisation).

VIII. Conclusion and Discussion

We have proposed a unified framework to address: (1) a large variety of video processing applications with the same formalism, by projection, distance based calculation in the feature space, and back-projection in the image space, and (2) a higher level visual characterization obtained by searching significant structures in the feature space (singularities and modes).

Regarding the choice of the feature space, the same framework can probably be used with other features, like wavelets, steerable or Gabor filters. However, the local jet space is easier to justify because of the Taylor expansion. It is also one of the most general because it implicitly contains many other features.

We have shown the relevance of the approach for several low level vision tasks. This representation also naturally provides image reduction and description tools that can be used at a higher processing level. We particularly think to object modelling and recognition, which is part of our ongoing work.

The presented work also contains more specific contributions, that we recall hereunder:

- The definition of distances in the local jet space, which, although proposed earlier, had not been used in practice to our knowledge.
- The local jet based NL-Mean filters, which can be seen as a continuum between tone space filtering and patch based NL-Means by increasing the order of derivation of the local jet.
- The optical flow solution as a nearest neighbour search in a similarity space.
- The singularities (isolated points) of the feature space, as way to fuse the detection and the characterization of interest points, classically addressed independently.
- The mode detection in the feature space, as a complementary information to salient (singular) features.

Because the aim of this work is to find a vision framework as universal as possible, we do not expect every application to compete with state-of-the-art dedicated algorithms. Experimental results were shown in this paper to convince that the framework makes sense, but obviously further evaluation is needed in every single case. The same applies for some parameters which were chosen either according to similar algorithm from the literature or empirically.

Some of the proposed algorithms, for example local jet based NL-means and background subtraction based on sample and consensus in the local jet space are particularly efficient and can be easily adapted to real-time. However, the computational cost remains an issue for different implementations: the optical flow by nearest neighbour search in the local jet space and the computation of the mode of the local jet distribution are two important examples. We are then investigating new ways to compute the nearest neighbours in the feature space using parallel implementations.

REFERENCES

[1] G. Peyré, "Manifold models for signals and images," *Computer Vision and Image Understanding*, vol. 113, no. 2, pp. 249–260, 2009.

[2] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[3] Y. Rubner and C. Tomasi, "Texture-based image retrieval without segmentation," in *Proc. ICCV*, Kerkyra, Greece, 1999, pp. 1018–1024.

[4] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision (ECCV'04)*, 2004, pp. 1–22.

[5] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[6] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] J. Koenderink and A. Van Doorn, "Representation of local geometry in the visual system," *Biological Cybernetics*, vol. 55, pp. 367–375, 1987.

[8] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–534, 1997.

[9] M. Crosier and L. Griffin, "Using basic image features for texture classification," *Int. J. of Computer Vision*, vol. 88, no. 3, pp. 447–460, 2010.

[10] L. Florack, B. Ter Haar Romeny, M. Viergever, and J. Koenderink, "The Gaussian scale-space paradigm and the multiscale local jet," *Int. J. of Computer Vision*, vol. 18, no. 1, pp. 61–75, January 1996.

[11] J. Orchard, M. Ebrahimi, and A. Wong, "Efficient non-local means denoising using the SVD," in *Proc. ICIP*, 2008, pp. 1732–1735.

[12] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. of Computer Vision*, vol. 30, no. 2, pp. 77–116, 1998.

[13] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Com. ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[14] D. Mount and S. Arya, "ANN: A library for approximate nearest neighbor searching," in *CGC Workshop on Computational Geometry*, 1997, http://www.cs.umd.edu/~mount/ANN/.

[15] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *Proc. CVPR*, vol. 2, 2005, pp. 60–65.

[16] A. Manzanera, "Local jet based similarity for NL-means filtering," in *Proc. ICPR*, Istambul, Turkey, 2010, pp. 2668–2671.

[17] H. Wang and D. Suter, "A consensus-based method for tracking: Modelling background scenario and foreground appearance," *Pattern Recognition*, vol. 40, no. 3, pp. 1091–1105, 2007.

[18] O. Barnich and M. Van Droogenbroeck, "ViBe: a powerful random technique to estimate the background in video sequences," in *Proc. ICASSP*. IEEE, April 2009, pp. 945–948.

[19] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. ICIP*, vol. 5. IEEE, 2004, pp. 3061–3064.

[20] C. Kervrann and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image denoising and representation," *Int. J. of Computer Vision*, vol. 79, no. 1, pp. 45–69, August 2008.

[21] P. Burman and W. Polonik, "Multivariate mode hunting: Data analytic tools with measures of significance," *J. Multivar. Anal.*, vol. 100, no. 6, pp. 1198–1218, 2009.

Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 320–328
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A new motion detection algorithm based on Σ–Δ background estimation

Antoine Manzanera *, Julien C. Richefeu

Ecole Nationale Supérieure de Techniques Avancées (ENSTA), Laboratoire d'Electronique de Informatique (LEI), 32 Boulevard Victor, Room No. 381, F-75739, Paris Cedex 15, France

Available online 5 June 2006

Abstract

Motion detection using a stationary camera can be done by estimating the static scene (background). In that purpose, we propose a new method based on a simple recursive non linear operator, the Σ–Δ filter. Used along with a spatiotemporal regularization algorithm, it allows robust, computationally efficient and accurate motion detection. To deal with complex scenes containing a wide range of motion models with very different time constants, we propose a generalization of the basic model to multiple Σ–Δ estimation.

Keywords: Motion detection; Background estimation; Recursive filtering

1. Introduction

The detection of moving objects in an image sequence is a very important low-level task for many computer vision applications, such as video surveillance, traffic monitoring or sign language recognition. When the camera is stationary, a class of methods usually employed is *background subtraction*. The principle of these methods is to build a model of the static scene (i.e., without moving objects) called *background*, and then to compare every frame of the sequence to this background in order to discriminate the regions of unusual motion, called *foreground* (the moving objects).

Many algorithms have been developed for background subtraction: recent reviews and evaluations can be found in (Lee and Hedley, 2002; Chalidabhongse et al., 2003; Cheung and Kamath, 2004; Piccardi, 2004). In this paper, we are more specifically interested in video surveillance systems with long autonomy. The difficulty in devising background subtraction algorithms in such context lies in the respect of several constraints:

- The system must keep working without human interaction for a long time, and then take into account gradual or sudden changes such as illumination variation or new static objects settling in the scene. This means that the background must be *temporally adaptive*.
- The system must be able to discard irrelevant motion such as waving bushes or flowing water. It should also be robust to slight oscillations of the camera. This means that there must be a *local* estimation for the *confidence* in the background value.
- The system must be real-time, compact and low-power, so the algorithms must not use much resource, in terms of computing power and memory.

The two first conditions imply that statistical measures on the temporal activity must be locally available in every pixel, and constantly updated. This excludes any basic approach like using a single model such as the previous frame or a temporal average for the background, and global thresholding for decision.

Some background estimation methods are based on the analysis of the histogram of the values taken by each pixel within a fixed number K of past frames. The mean, the median or the mode of the histogram can be chosen to set the background value, and the foreground can be

* Corresponding author. Fax: +33 1 45 52 83 27.
E-mail addresses: antoine.manzanera@ensta.fr (A. Manzanera), julien.richefeu@ensta.fr (J.C. Richefeu).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.04.007

discriminated by comparing the difference between the current frame and the background with the histogram variance. More sophisticated techniques are also based on the K past frames history: linear prediction (Toyoma et al., 1999), kernel density estimation (Elgammal et al., 2000; Mittal and Paragios, 2004), or principal component analysis (Oliver et al., 2000). These methods require a great amount of memory, since K needs to be large (usually more than 50) for robustness purposes. So they are not compatible with our third condition.

Much more attractive for our requirements are the *recursive* methods, that do not keep in memory a histogram for each pixel, but rather a fixed number of estimates computed recursively. These estimates can be the mean and variance of a Gaussian distribution (Wren et al., 1997), different states of the background (e.g., its values and temporal derivatives) estimated by predictive filter (Karmann and von Brandt, 1990), or recursive estimation of the extremal values (Richefeu and Manzanera, 2004). But it is difficult to get robust estimates of the background with linear recursive framework, unless a multi-modal distribution (e.g., multiple Gaussian (Stauffer and Grimson, 2000; Power and Schoonees, 2002)) is explicitly used, which is done at the price of an increasing complexity and memory requirement. Furthermore, these methods rely on parameters such as the learning rates used in the recursive linear filters, setting the relative weights of the background states and the new observations, whose tuning can be tricky, which makes difficult the fulfillment of the first condition stated above.

A recursive approximation of the temporal median was proposed in (McFarlane and Schofield, 1995) to compute the background. The interest of this method lies in the robustness provided by the non linearity compared to the linear recursive average, and in the very low computational cost. In this article, we investigate some nice properties of this operator, introducing the notion of *Σ–Δ estimation*, and using it to obtain a locally adaptive motion detection.

In Section 2, we present the basic Σ–Δ estimation method. The Σ–Δ filter is presented and used to compute two orders of temporal statistics for each pixel of the sequence providing a pixel-level decision framework. Then, in Section 3, we exploit the spatial correlation in these data using new hybrid linear/morphological operators, and use higher level processing to enhance and regularize the detection solution. Some results are presented, illustrating the robustness and accuracy of the method in the case of simple background (i.e., one single time-varying mode). For more complex scenes, we propose in Section 4 a generalization of the algorithm to multiple background estimation. Finally, conclusions and future works are presented in Section 5.

2. Σ–Δ estimation

Our first background estimate, whose computation is shown on Table 1(1), is the same as (McFarlane and Schofield, 1995), where I_t is the input sequence, and M_t the estimated background value. The sign function sgn is defined as $\mathrm{sgn}(a) = -1$ if $a < 0$, $\mathrm{sgn}(a) = 1$ if $a > 0$, and $\mathrm{sgn}(a) = 0$ if $a = 0$. So, at every frame, the estimate is simply incremented by one if it is smaller than the sample, or decremented by one if it is greater than the sample. If I_t is a discrete random signal, the ratio between the number of indexes $\tau < t$ such that $I_\tau < M_t$, and the number of indexes $\tau < t$ such that $I_\tau > M_t$ converges in mean to 1. So M_t is an approximation of the median of I_t. But this filter has other interesting properties, relative to the change detection in time-varying signals. Indeed, we interpret this background estimation as the simulation of a digital conversion of a time-varying analog signal using Σ–Δ modulation (A/D conversion using only comparison and elementary increment/decrement, hence the name Σ–Δ filter).

As the precision of the Σ–Δ modulation is limited to signals with absolute time-derivative less than unity, the modulation error is proportional to the variation rate of the signal, corresponding here to a motion likelihood measure of the pixels. We then use the absolute difference between I_t and M_t as our first differential estimate: the difference Δ_t (Table 1(2)).

Unlike (McFarlane and Schofield, 1995), we also use this filter to compute the time-variance of the pixels, representing their motion activity measure, used to decide whether the pixel is more likely "moving" or "stationary". Then, V_t (Table 1(3)) used in our method has the dimension of a temporal standard deviation. It is computed as

Table 1
The Σ–Δ background estimation

(1)
Initialization
for each pixel x:
$M_0(x) = I_0(x)$
For each frame t
for each pixel x:
$M_t(x) = M_{t-1}(x) + \mathrm{sgn}(I_t(x) - M_{t-1}(x))$

(2)
For each frame t
for each pixel x:
$\Delta_t(x) = |M_t(x) - I_t(x)|$

(3)
Initialization
for each pixel x:
$V_0(x) = \Delta_0(x)$
For each frame t
for each pixel x such that $\Delta_t(x) \neq 0$:
$V_t(x) = V_{t-1}(x) + \mathrm{sgn}(N \times \Delta_t(x) - V_{t-1}(x))$

(4)
For each frame t
for each pixel x:
if $\Delta_t(x) < V_t(x)$
then $D_t(x) = 0$
else $D_t(x) = 1$

(1) Computation of the Σ–Δ mean. (2) Computation of the difference between the image and the Σ–Δ mean (motion likelihood measure). (3) Computation of the Σ–Δ variance defined as the Σ–Δ mean of N times the non-zero differences. (4) Computation of the motion label by comparison between the difference and the variance.

a Σ–Δ filter of the difference sequence Δ_t. This provides a measure of *temporal activity* of the pixels. As we are interested in pixels whose variation rate is significantly over its temporal activity, we apply the Σ–Δ filter to the sequence of N times the non-zero differences.

Finally, the pixel-level detection is simply performed by comparing Δ_t and V_t (Table 1(4)).

Fig. 1 displays an example of the evolution over time of the different values computed as above, for three particular pixels extracted from a country scene for a 500 frames sequence. The solid line represents the input image I_t. The dashed line corresponds to the Σ–Δ mean M_t. The impulses represents the difference Δ_t. Finally, the dotted line is the Σ–Δ variance V_t (using $N = 4$). The detection label D_t is not represented explicitly, but corresponds to the Boolean indicator of the condition "an impulse is over the dotted line".

The pixel used in Fig. 1(1) is a pixel in a still zone, with flat temporal activity, such as a remote area of the static background (in our example, a sky lightly covered with slowly moving clouds). For such pixels, the high frequency variation corresponds to temporal noise due to the acquisition and digitization processes. The low frequency variations are due to illumination changes or slow motion of low contrast objects.

The pixel used in Fig. 1(2) is a pixel in a motion area, such as tracks or corridors (in our example, a country road with 2 vehicles passing away). In that case, the moving objects give rise to sharp changes that are not taken into account by the Σ–Δ mean, and then the difference signal shows a peak. Such peaks are discriminated thanks to the comparison with the Σ–Δ variance.

The pixel used in Fig. 1(3) is a pixel in a clutter area, i.e., a zone of physical changes due to intrinsic nature of the scene rather than moving objects. Examples of such areas are: trees moving with the wind or river (in our example, high grass in the foreground of the scene). In that case, the difference signal shows a repetition of peaks, and if these peaks are close enough from each other with respect to the delay induced by Σ–Δ modulation, then they will be taken into account in the Σ–Δ variance, in such a way that the difference will remain less that the variance.

The fundamental features of the Σ–Δ estimation, i.e., its non linearity and median convergence property come from the fact that the statistics M_t is always updated with a constant increment ± 1, not depending upon the difference between the current sample and the current mean $(I_t - M_{t-1})$. If the increment depended linearly on the difference, we would get $M_t = M_{t-1} + \alpha(I_t - M_{t-1}) = \alpha I_t + (1 - \alpha)M_{t-1}$, that is, the classical recursive exponential filter (or moving average) used, typically, in the recursive Gaussian Fitting methods (Wren et al., 1997; Stauffer and Grimson, 2000; Power and Schoonees, 2002). In its simplest form, α is a real constant in the interval $]0,1[$. Fig. 2(1) and (2) displays comparison between the Σ–Δ mean and the moving average, on a temporal sequence corresponding to the passage of a moving object. Note the difference between the unity slope of the Σ–Δ mean (Fig. 2(1)) and the exponential decrease of the moving average (Fig. 2(2)). In the more robust form of the Gaussian fitting

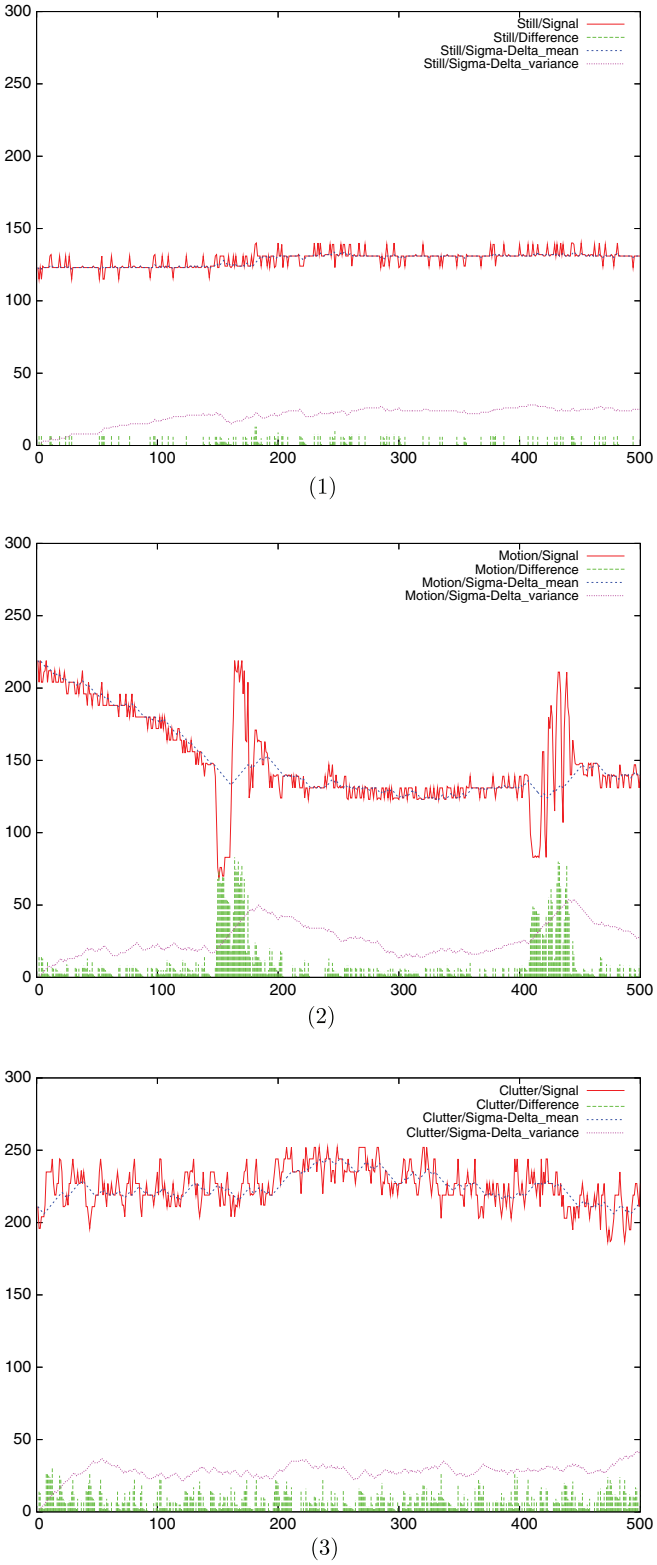


Fig. 1. Temporal variation of a pixel value and the corresponding Σ–Δ estimation, for pixels taken in three different areas (1) still area, (2) motion area, (3) clutter area.

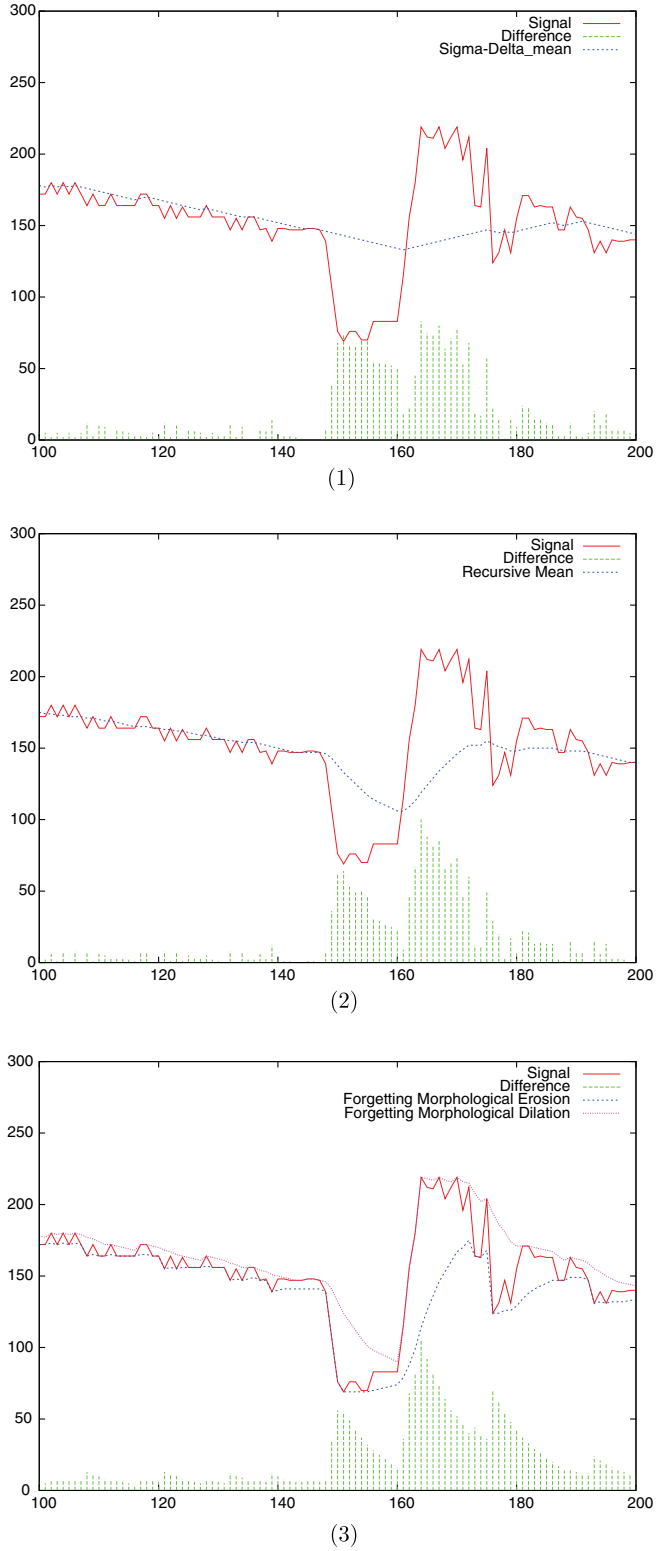


Fig. 2. Comparison between (1) Σ–Δ estimation. (2) Moving average ($\alpha = 1/16$). (3) Forgetting morphological operators ($\alpha = 1/8$).

estimation, α is no more a constant but is calculated from the probability of the current sample, i.e., $\alpha(t) = \mathcal{N}^{M_t}_{V_t}(I_t)$, where $\mathcal{N}^{\mu}_{\sigma^2}(x)$ is the normal density function of mean μ and variance σ^2. The variance V_t is estimated by the moving average of the squared differences $(I_t - M_t)^2$. Computation of the square and of the Gaussian probability of the sample make these methods sensitive to the numerical approximations, whereas the Σ–Δ estimation only uses exact integer computations. However, a nice feature of the Gaussian fitting methods is their extension to the estimation of multimodal Gaussian distributions (Stauffer and Grimson, 2000; Power and Schoonees, 2002), that allows to deal with very complex scenes. In the same spirit, we shall present, in Section 4, an extension of the Σ–Δ estimation adapted to complex backgrounds.

Fig. 2(3) also shows the behavior of another simple recursive estimation method, based on the forgetting morphological operators (Richefeu and Manzanera, 2004), because they will be used later in our algorithm, under the form of their spatial counterparts (see Section 3). The forgetting temporal dilation (resp. erosion) is defined by $M_t = \alpha I_t + (1-\alpha)\max(I_t, M_{t-1})$ (resp. $m_t = \alpha I_t + (1-\alpha)\min(I_t, m_{t-1})$).

Fig. 3(1)–(4) displays the result of the algorithm of Table 1 for one frame of an urban traffic sequence. The four images represent respectively I_t, M_t, V_t and D_t. It can be

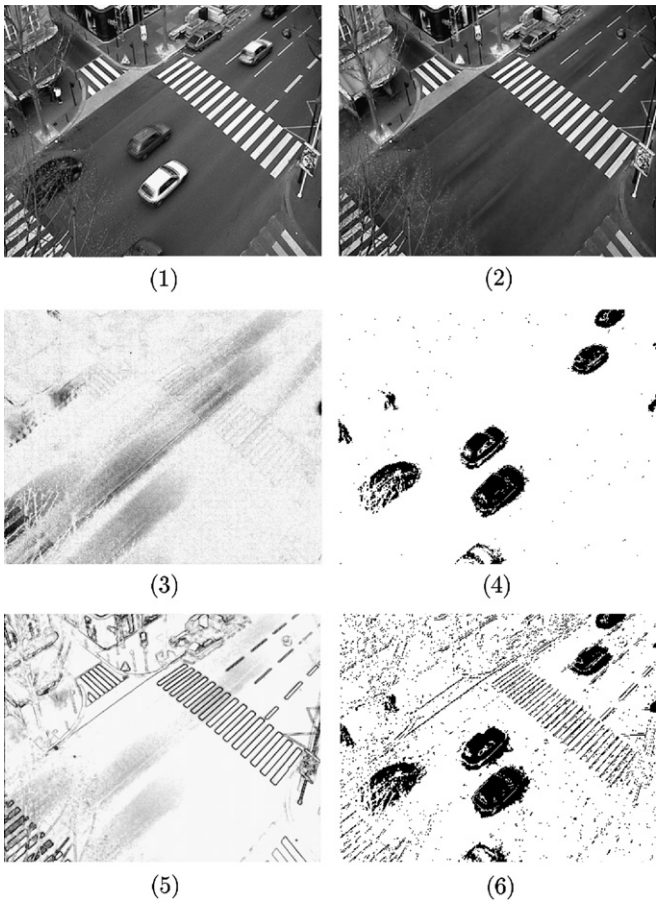


Fig. 3. Result of the Σ–Δ detection on a traffic sequence. (1) I_t, (2) M_t, (3) V_t (displayed with reverse video and normalized histogram), (4) D_t ($N = 2$), (5) V_t with simulated oscillations of the camera (displayed in reverse video and normalized histogram) and (6) D_t for the oscillating camera ($N = 2$).

seen that the discrimination of "moving" pixels corresponds to the detection of temporally *salient* pixels with respect to the temporal activity. This allows to discard irrelevant (clutter) motion, but also to be less sensitive to sensor oscillations, as it is shown in Fig. 3(5) and (6): A uniform random oscillation of ± 1 pixels has been simulated on the same sequence. In this case, M_t converges to an approximation of the spatiotemporal median, and then V_t (Fig. 3(5)) emphasizes the regions of high contrast, thus increasing the local threshold in these regions, and avoiding the detection of the whole scene contour in D_t (Fig. 3(6)).

The only visible parameter of this method is N, the number used in the computation of the variance V_t. The range value of N is small (between 1 and 4), and usually a power of 2 is chosen for optimization purposes. Furthermore, it can be automatically adjusted with respect to a noise estimation. Such estimation can be performed by counting the isolated pixels in the detection result D_t, under the (classical) hypothesis that such detected pixels are only due to noise.

In fact, there is another parameter which is less obvious; it is the frequency of update of the Σ–Δ statistics. This frequency has a dimension of number of gray levels per second. It is then clear that it has to be adapted to (1) the dynamics of the image (number of gray levels), (2) the acquisition frequency (frame rate). We usually use the same frequency as the frame rate for 25 Hz sequences of 8 bits images, but it can be lowered to adjust to the size and velocity of the observed objects in the application. In Section 4, we will present a sophistication of the method based on a multi-frequencies Σ–Δ estimation, in order to better discriminate the foreground in the case of a scene presenting a wide range of different motions.

As suggested by Lacassagne in (Denoulet et al., 2005), the robustness of the Σ–Δ estimation can be further improved by updating M_t only when $\Delta_t < V_t$. In this article, we make the choice to inhibit locally the update only after the spatial processing (see Section 3).

The Σ–Δ background estimation provides a simple and efficient method to detect the significantly changing pixels in a static scene, with respect to a time constant depending on the number of gray levels, and on the frame rate. Nevertheless it is a pure temporal processing, which can only provide pixel-level detection. In the following section, we present some spatial processing, to enhance and regularize the detection result.

3. Spatiotemporal processing

Recently, we have presented a Markovian modeling to perform a spatiotemporal regularization of the pixel-level Σ–Δ detection. It was an adaptation of the iterative algorithm presented in (Caplier et al., 1996 and Lacassagne et al., 1999), using the pixel-level detection D_t as initialization, and the Σ–Δ difference Δ_t and variance V_t, as a couple of observation fields used in the design of the energy. Details can be found in (Manzanera and Richefeu, 2004).

We present here another regularization strategy. The spatiotemporal processing that we propose in this section has a threefold purpose:

- eliminate the *non significant* pixels from the detection (noise, false detection), and enhance the segmentation of the moving objects;
- reduce the *ghost effect*, that produces false detection at the loci from where a moving object leaves after a long stay;
- reduce the *aperture effect*, that causes a poor detection for the objects whose projected motion is weak, e.g., radially moving objects.

3.1. Common edges hybrid reconstruction

The first part of the spatial processing is composed of gray level operations. The inputs are: (1) the original image I_t, and (2) the Σ–Δ difference image Δ_t. The purpose of this module is to eliminate the ghost objects in Δ_t by discriminating them within I_t. The actual operation we perform is the following one:

$$\Delta'_t = H\mathrm{Rec}_{\alpha}^{\Delta_t}(\mathrm{Min}(\|\nabla(I_t)\|, \|\nabla(\Delta_t)\|)) \quad (1)$$

Roughly speaking, this means the "reconstruction" (we shall make this word explicit later) within Δ_t, of the image of the minimum between the gradient module of Δ_t and the gradient module of I_t. The semantics of this formula is: "Δ'_t is made of the components of Δ_t that are also in I_t". Let us now detail the actual computation.

The gradient modules of Δ_t and I_t are computed by estimating the first derivative components with convolutions with Sobel masks, and then computing the Euclidean norm of the vector. We then compute the minimum image Min, defined for every pixel x by $\mathrm{Min}(I_1, I_2)(x) = \min(I_1(x), I_2(x))$. This acts like a logical conjunction, retaining only the edges that belong both to an object of Δ_t, and of I_t.

In order to recover the whole object in Δ_t, and not only its edges, we perform a "reconstruction" of the common edges $\mathrm{Min}(\|\nabla(I_t)\|, \|\nabla(\Delta_t)\|)$ within Δ_t.

The first idea should be to use the classical geodesic reconstruction $\mathrm{Rec}^Y(X)$, defined by the relaxation of the geodesic dilation $\delta_B^Y(X) = \mathrm{Min}(\delta_B(X), Y)$ (δ is the morphological dilation, B the elementary structuring element defining the topology, Y the reference image, X the marker image). In fact the geodesic reconstruction is not adapted, because the sole connection criterion is not robust enough, and in most cases, the object and its ghost are both reconstructed.

We rather use the *hybrid reconstruction* operator, based on the forgetting morphological operators that we introduced in (Richefeu and Manzanera, 2004). First we define the hybrid dilation as the spatial version of the forgetting dilation, computed by the causal sequence:

$$H\mathrm{Dil}_\alpha(I)^{(0)}(x,y) = \alpha I(x,y) + (1-\alpha)\max(I(x,y),\ H\mathrm{Dil}_\alpha(I)^{(0)}(x-1,y)) \quad (2)$$

followed by the anti-causal sequence:

$$H\mathrm{Dil}_\alpha(I)^{(1)}(x,y) = \alpha H\mathrm{Dil}_\alpha(I)^{(0)}(x,y) + (1-\alpha)\max(H\mathrm{Dil}_\alpha(I)^{(0)}(x,y),\ H\mathrm{Dil}_\alpha(I)^{(1)}(x+1,y)) \quad (3)$$

then the causal sequence vertically:

$$H\mathrm{Dil}_\alpha(I)^{(2)}(x,y) = \alpha H\mathrm{Dil}_\alpha(I)^{(1)}(x,y) + (1-\alpha)\max(H\mathrm{Dil}_\alpha(I)^{(1)}(x,y),\ H\mathrm{Dil}_\alpha(I)^{(2)}(x,y-1)) \quad (4)$$

and finally:

$$H\mathrm{Dil}_\alpha(I)(x,y) = \alpha H\mathrm{Dil}_\alpha(I)^{(2)}(x,y) + (1-\alpha)\max(H\mathrm{Dil}_\alpha(I)^{(2)}(x,y),\ H\mathrm{Dil}_\alpha(I)(x,y+1)) \quad (5)$$

Here $1/\alpha$ has the dimension of a spatial radius, and thus the parameter α replaces the structuring element.

The hybrid reconstruction is based on the same scheme, using the sequence:

$$H\mathrm{Rec}^J_\alpha(I)^{(0)}(x,y) = \min(J(x,y), \alpha I(x,y) + (1-\alpha)\max(I(x,y), H\mathrm{Rec}^J_\alpha(I)^{(0)}(x-1,y))) \quad (6)$$

and so on.

The behavior of the hybrid reconstruction can be seen in Fig. 4, where the objects (cars, pedestrian) move after leaving a halt 20 frames before. The advantage of the hybrid reconstruction in this application is to "forget" gradually (exponentially to be precise) the marker, which acts like a confidence function, instead of being strictly based on the connectivity.

It can be seen that the success of this common edges marking step depends on the contrasts of the background itself, (see Fig. 4(7)–(10)). But we have focused here on extreme cases, the objects being completely still and encrusted in the background at the beginning. Furthermore, as we will see at the end of this section, the relevance feedback will also reduce significantly the ghost effect.

3.2. Binary spatiotemporal morphology

After computing the hybrid reconstruction, we perform the adaptive thresholding on Δ'_t (instead of Δ_t in the purely temporal version):

$$\text{If} \quad \Delta'_t > V_t \quad \text{then} \quad D_t = 1 \quad \text{else} \quad D_t = 0 \quad (7)$$

We then eliminate the small connected components on D_t using the *opening by reconstruction*:

$$L_t^{(0)} = \mathrm{Rec}^{D_t}(\varepsilon_{B_\lambda}(D_t)) \quad (8)$$

where ε is the morphological erosion, and B_λ the structuring element, is a ball of radius λ.

Finally, we perform a *temporal confirmation* by computing another reconstruction:

$$L_t = \mathrm{Rec}^{L_t^{(0)}}(L_{t-1}^{(0)}) \quad (9)$$

L_t represents the final label, the result of the detection. Its semantics is: the objects "bigger than" λ that appear on 2 consecutive frames. It is illustrated in Fig. 5, using $\lambda = 1$, on a sequence showing two people walking.

3.3. Relevance feedback

As indicated in Section 2, the whole detection is made more robust if the background is not modified for the pixels of moving objects. We adopt this strategy by updating Σ–Δ mean $M_t(x)$ only where $L_t(x) = 0$. This implies a reordering of the algorithm shown in Table 1. The complete

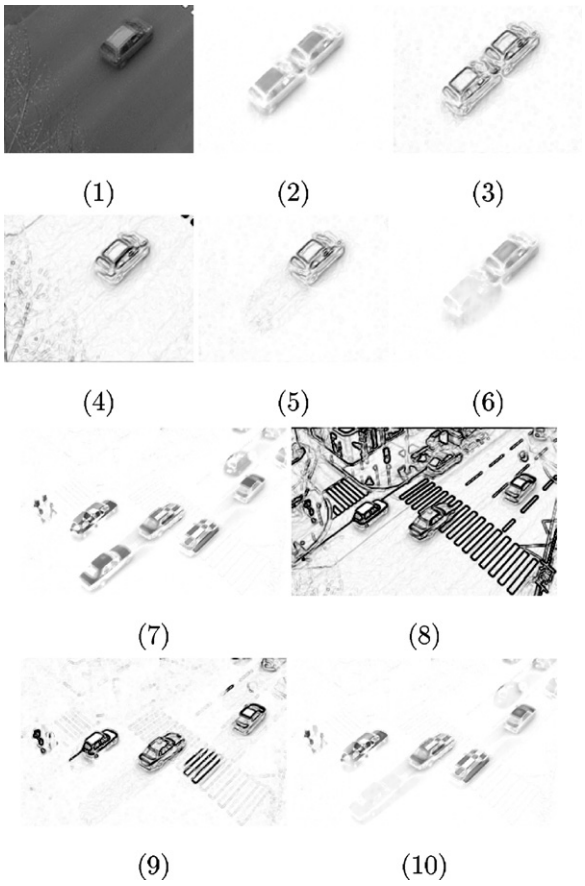


Fig. 4. Ghost busting by hybrid reconstruction of the common edges ($\alpha = 1/8$). First example (1 car): (1) I_t, (2) Δ_t, (3) $\|\nabla(\Delta_t)\|$, (4) $\|\nabla(I_t)\|$, (5) $\mathrm{Min}(\|\nabla(\Delta_t)\|, \|\nabla(I_t)\|)$, (6) Δ'_t. Second example (5 cars, 1 pedestrian): (7) Δ_t, (8) $\|\nabla(I_t)\|$, (9) $\mathrm{Min}(\|\nabla(\Delta_t)\|, \|\nabla(I_t)\|)$ (10) Δ'_t.

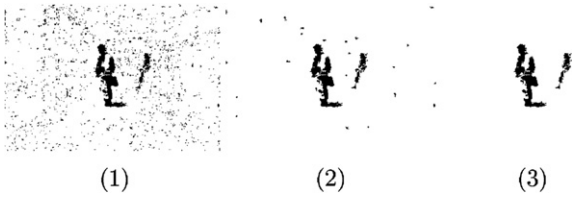


Fig. 5. Binary spatiotemporal morphology. (1) D_t, (2) $L_t^{(0)}$ and (3) L_t.

algorithm is then displayed in Table 2, discarding the pixel index x.

The relevance feedback effect can be seen in Fig. 6, showing a detail of a scene with two cars driving after a stop at a red light. This strategy improves the detection result by delaying the contribution of the moving objects to the background. This reduces significantly both the ghost effect and the aperture effect (radial movements).

As indicated in (Denoulet et al., 2005), it is safer, at this low level of processing, to apply the relevance feedback only to the mean M_t instead than the two estimates M_t and V_t, to avoid false detection objects to settle in the background. Relevance feedback can be used on both M_t and V_t if a high level of confidence is reached in the detected objects. This can be achieved by using further processing (e.g., kinematic filtering), that is beyond the scope of this article.

Table 2
The complete Σ–Δ detection algorithm with relevance feedback

Signed difference
$S_t = M_t - I_t$
Variance updating
if $|S_t| \neq 0$:
$V_t = V_{t-1} + \mathrm{sgn}(N \times |S_t| - V_{t-1})$
Common edges hybrid reconstruction
$\Delta'_t = H\mathrm{Rec}_{\alpha}^{|S_t|}(\mathrm{Min}(\|\nabla(I_t)\|, \|\nabla(|S_t|)\|))$
Temporal detection
if $\Delta'_t < V_t$
then $D_t = 0$
else $D_t = 1$
Spatiotemporal binary processing
$L_t^{(0)} = \mathrm{Rec}^{D_t}(\varepsilon_{B_x}(D_t))$
$L_t = \mathrm{Rec}^{L_t^{(0)}}(L_{t-1}^{(0)})$
Mean updating with relevance feedback
if $L_t = 0$:
$M_t = M_{t-1} + \mathrm{sgn}(S_t - M_{t-1})$

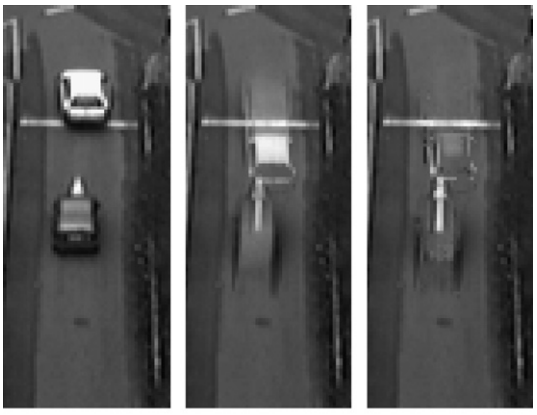


Fig. 6. Relevance feedback. (1) Original image, (2) Σ–Δ background and (3) background with relevance feedback.

Fig. 7. Result of the Σ–Δ detection with spatial processing on an indoor (Hall) sequence, shown at 3 frames: (a) $t = 60$, (b) $t = 130$, (c) $t = 200$. (1) Original I_t, (2) Σ–Δ mean M_t. (3) Detection label L_t.

Fig. 7 shows the results of the full algorithm on an indoor sequence (the "Hall" sequence). The results illustrate the effect of the spatial processing and relevance feedback on the reduction of the aperture effect: the two persons are well detected although they are moving radially with respect to the camera.

4. Multiple background Σ–Δ estimation

The spatiotemporal processing and relevance feedback, presented in the previous section, allows a visible enhancement of the robustness, in the case of slowing down, stopping or radially moving objects. Nevertheless, the Σ–Δ estimator is characterized by a time constant: its updating period, which has a dimension of number of gray levels per second. This induces a limitation of the basic approach in the adaptation capability to certain complex scenes, typically in the case of scenes permanently crossed by lots of objects of very different sizes and velocities. We propose in this section a generalization framework of the Σ–Δ estimation to multiple backgrounds.

The principle is to compute instead of one single background M_t, a set of K backgrounds $\{m_t^i\}_{1 \leqslant i \leqslant K}$. Each background m_t^i is characterized by its updating period α_i and by its phase ϕ_i. A set of K variances v_t^i is also computed as the Σ–Δ mean of the differences between I_t and m_t^i. The background/foreground decision is then made by comparing the sample to every background, which is attached a confidence value that is (1) proportional to α_i and (2) inversely proportional to v_t^i.

Table 3 shows an example of computation of multi-frequencies background using K different periods $\alpha_1 < \cdots < \alpha_K$. The phases are discarded in this example. The Σ–Δ means

Table 3
The multi-periods background Σ–Δ estimation

Multiple mean updating
For each frame t,
for each i, $i \in [0,K]$,
if t is a multiple of α_i:
$m_t^i = m_{t-1}^i + \mathrm{sgn}(m_t^{i-1} - m_{t-1}^i)$
Multiple variance updating
For each frame t,
for each i, $i \in [0,K]$,
$\delta_t^i = |I_t - m_t^i|$
if $\delta_t^i \neq 0$:
$v_t^i = v_{t-1}^i + \mathrm{sgn}(N \times \delta_t^i - v_{t-1}^i)$
Mean computing
For each frame t,
$$M_t = \frac{\sum_{i\in[0,K]} \frac{\alpha_i m_t^i}{v_t^i}}{\sum_{i\in[0,K]} \frac{\alpha_i}{v_t^i}}$$

m_t^i can be computed recursively: $m_t^i = m_{t-1}^i + \mathrm{sgn}(m_t^{i-1} - m_{t-1}^i)$, with the convention $m_t^0 = I_t$. In this example, we compute explicitly a "best confidence" background M_t, as shown in the last line of Table 3. The principle of the confidence coefficients attached to every background m_t^i is to give more weight to long-term backgrounds, in order to favor the most frequent value over long periods, and at the same time, the weights are lowered accordingly to the variance, in order to allow adaptation to a changing background.

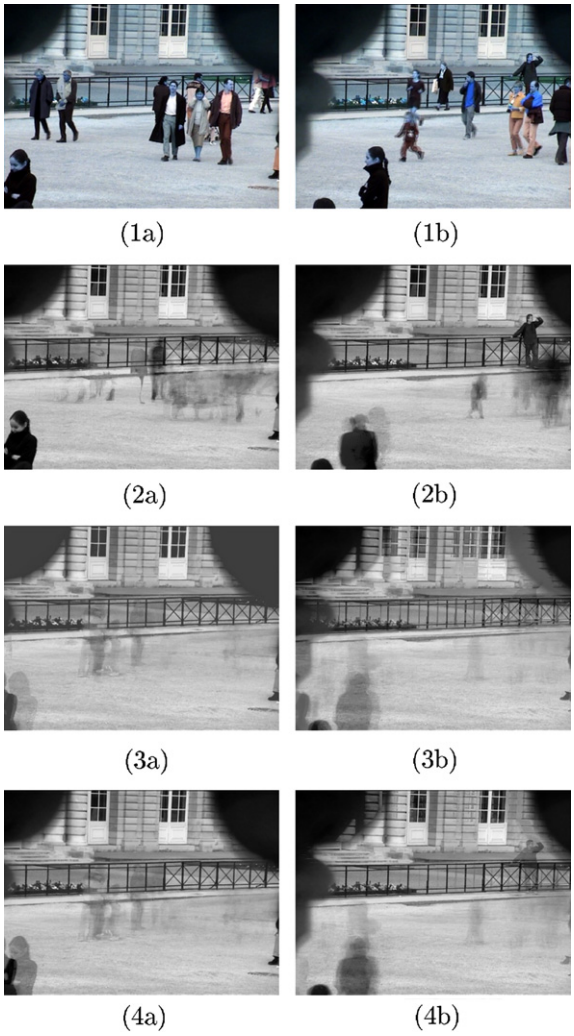


Fig. 8. Multi-periods background Σ–Δ estimation computed on a complex scene (Luxembourg sequence), shown at two instants (a) $t = 1136$, (b) $t = 2896$. (1) Original I_t. (2) Short-term background m_t^1 ($\alpha_1 = 1$). (3) Long-term background m_t^3 ($\alpha_3 = 16$). (4) Best confidence background M_t.

Fig. 8 shows an application of the multi-periods Σ–Δ estimation on a complex scene. This corresponds to a sequence taken in a very frequented part of the Jardin du Luxembourg. The scene is never empty, with lots of people stopping and remaining more or less still for different periods. For this sequence, we have taken $K = 3$, $\alpha_1 = 1$, $\alpha_2 = 8$, $\alpha_3 = 16$. This framework clearly enhances the robustness of the detection with respect to the range of motion models, while preserving a good adaptation capability to changing conditions. See for example in Fig. 8: at frame (a), best confidence is globally given to the long-term mean. At frame (b), the camera has been shifted just before, and so, in the upper part of the image, best confidence is given to the short-term mean, while in the lower part of the image, the confidence remains higher for the long-term, which prevents the two stopped people to appear in the final background.

5. Conclusions

We have presented a new algorithm allowing a robust and accurate detection of moving objects for a small cost in memory consumption and computational complexity. We have emphasized the nice properties of the Σ–Δ filter for the detection of salient features in time-varying signal, showing that the interest of such filter goes well beyond its temporal median convergence property.

We have proposed a new spatiotemporal regularization strategy, using an original hybrid reconstruction method and spatiotemporal binary morphology, to exploit the spatial correlation and increase the confidence of the Σ–Δ detection.

We have presented a generalization framework for multiple Σ–Δ estimation allowing to deal with complex scenes by combining Σ–Δ estimates with different frequencies or phases.

Because it only relies on pixel-wise or spatially limited interactions, the main part of this algorithm (everything excepted the common edge hybrid reconstruction module, which is a recursive scan algorithm) is suited to a *massively parallel* implementation. We have realized an implementation of the algorithm (temporal processing plus spatiotemporal morphology) on a *programmable artificial retina* (Komuro et al., 2003), which is a fine-grained parallel machine with optical input. The algorithm is indeed well adapted to the architecture, which consists in a mesh of tiny processors with limited memory and computation power. For a 200×200 retina array running at 25 MHz, using 8 bits per pixels: the computation performance is 2.25 ms per frame, of which only 0.75 ms for the sole

computation, and the rest for the acquisition. (Denoulet et al., 2005) have also made an implementation of the Σ–Δ background estimation associated with Markovian relaxation on the *Associative Mesh of Orsay* (Mérigot, 1997) which is a massively parallel asynchronous machine with programmable topology.

Acknowledgements

The authors wish to thank Lionel Lacassagne, who proposed improvements on the original algorithm, and provided most of the sequences used in this article.

References

Caplier, A., Dumontier, C., Luthon, F., Coulon, P., 1996. Mrf based motion detection algorithm image processing board implementation. Traitement du signal 13 (2), 177–190 (in French).

Chalidabhongse, T.H., Kim, K., Harwood, D., Davis, L., 2003. A perturbation method for evaluating background subtraction algorithms. In: Proc. Joint IEEE Int. Work. on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. Nice, France.

Cheung, S.-C., Kamath, C., 2004. Robust techniques for background subtraction in urban traffic video. In: Proc. SPIE Video Com and Image Proc. San Jose, CA.

Denoulet, J., Mostafaoui, G., Lacassagne, L., Mérigot, A., 2005. Implementing motion markov detection on general purpose processor and associative mesh. In Proc. CAMP'05.

Elgammal, A., Harwood, D., Davis, L., 2000. Non-parametric Model for Background Subtraction. In: Proc. IEEE ECCV. Dublin, Ireland.

Karmann, K.-P., von Brandt, A., 1990. Moving object recognition using an adaptive background memory. In: Time-Varying Image Processing and Moving Object Recognition, vol. 2, Elsevier.

Komuro, T., Ishii, I., Ishikawa, M., Yoshida, A., 2003. A digital vision chip specialized for high-speed target tracking. IEEE Trans. Electron Dev. 50 (1), 191–199.

Lacassagne, L., Milgram, M., Garda, P., 1999. Motion detection, labeling, data association and tracking in real-time on risc computer. In: Proc. IEEE ICIAP. pp. 520–525.

Lee, B., Hedley, M., 2002. Background estimation for video surveillance. In: Proc. IVCNZ'02. pp. 315–320.

Manzanera, A., Richefeu, J., Dec. 2004. A robust and computationally efficient motion detection algorithm based on Σ–Δ background estimation. In: Proc. ICVGIP'04. pp. 46–51.

McFarlane, N., Schofield, C., 1995. Segmentation and tracking of piglets in images. Mach. Vision Appl. 8, 187–193.

Mérigot, A., 1997. Associative nets model: a graph based parallel computing model. IEEE Trans. Comput. 46 (5), 558–571.

Mittal, A., Paragios, N., 2004. Motion-based background subtraction using adaptive kernel density estimation. In: Proc. IEEE CVPR.

Oliver, N., Rosario, B., Pentland, A., 2000. A bayesian computer vision system for modeling human interactions. IEEE Trans. PAMI 22 (8), 831–843.

Piccardi, M., Oct. 2004. Background subtraction techniques: a review. In: Proc. of IEEE SMC/ICSMC.

Power, P., Schoonees, J., Nov. 2002. Understanding background mixture models for foreground segmentation. In: Proc. IVCNZ'02. pp. 267–271.

Richefeu, J., Manzanera, A., Oct. 2004. A new hybrid differential filter for motion detection. In: Proc. ICCVG'04.

Stauffer, C., Grimson, E., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. PAMI 22 (8), 747–757.

Toyoma, K., Krumm, J., Brumitt, B., Meyers, B., 1999. Wallflower: Principles and Practice of Background Maintenance. In: Proc. IEEE ICCV. Kerkyra, Greece, pp. 255–261.

Wren, C., Azarbayejani, A., Darrell, T., Pentland, A., 1997. Pfinder: Real-time tracking of the human body. IEEE Trans. PAMI 19 (7), 780–785.

Σ-Δ Background Subtraction and the Zipf Law

Antoine Manzanera

ENSTA - Elec. and Comp. Sc. lab
32 Bd Victor, 75015 Paris, France
antoine.manzanera@ensta.fr
http://www.ensta.fr/~manzaner

Abstract. The Σ-Δ background estimation is a simple non linear method of background subtraction based on comparison and elementary increment/decrement. We propose here some elements of justification of this method with respect to statistical estimation, compared to other recursive methods: exponential smoothing, Gaussian estimation. We point out the relation between the Σ-Δ estimation and a probabilistic model: the Zipf law. A new algorithm is proposed for computing the background/foreground classification as the pixel-level part of a motion detection algorithm. Comparative results and computational advantages of the method are commented.

Keywords: Image processing, Motion detection, Background subtraction, Σ-Δ modulation, Vector data parallelism.

1 Introduction

Background subtraction is a very popular class of methods for detecting moving objects in a scene observed by a stationary camera [1] [2] [3] [4] [5]. In every pixel p of the image, the observed data is a time series $I_t(p)$, corresponding to the values taken by p in the video I, as a function of time t. As only temporal processing will be considered in this paper, the argument p will be discarded. The principle of background subtraction methods is to discriminate the pixels of moving objects (the foreground) from those of the static scene (the background), by detecting samples which are significantly deviating from the statistical behaviour of the time series. To do this, one needs to estimate the graylevel distribution with respect to time, i.e. $f_t(x) = P(I_t = x)$. As the conditions of the static scene are subject to changes (lighting conditions typically), f_t is not stationary, and must be constantly re-estimated. For the sake of computational efficiency, which is particularly critical for video processing, it is desirable that f_t should be represented by a small number of estimates which can be computed recursively. In this paper, we focus on recursive estimation of mono-modal distributions, which means that we assume that the time series corresponding to the different values possibly taken by the background along time, presents one single mode. This may not be a valid assumption for every pixel, but it does not affect the interest of the principle since the technique presented can be extended to multi-modal background estimation.

L. Rueda, D. Mery, and J. Kittler (Eds.): CIARP 2007, LNCS 4756, pp. 42–51, 2007.

The Σ-Δ background estimation [6] [7] [8] is a simple and powerful non linear background subtraction technique, which consists in incrementing (resp. decrementing) the current estimate by an constant value if it is smaller (resp greater) than the current sample. Our objective is to discuss the foundations of this method, with regards to statistical estimation. We show the relation between the Σ-Δ estimation and the probabilistic model of Zipf-Mandelbrot, and compare it with two other recursive methods: exponential smoothing and Gaussian estimation. Section 2 presents the general framework of recursive estimation. Section 3 presents the particular case of Σ-Δ estimation, and provides the full numerical algorithm to compute it in the mono-modal case. Section 4 shows some results and discuss the computational advantages of Σ-Δ background subtraction, proposing in particular a complete vector data parallel implementation adapted to the SIMD-within-register framework. Section 5 concludes and presents the possible extension of the primitive algorithm.

2 Recursive Estimation

If one should represent f_t by one single scalar estimate, one of the most natural would be the average M_t of the time series I_t. The naive recursive computation: $M_t = \frac{1}{t} I_t + \frac{t-1}{t} M_{t-1}$ can be used as initialisation for the small values of t, but is not numerically realisable for long series. So one common solution is to use a constant weight (or learning rate) $\alpha \in]0,1[$ for the current sample: $M_t = \alpha I_t + (1-\alpha) M_{t-1}$. This is sometimes referred to as running average, and corresponds to the recursive implementation of exponential smoothing.

One way of generalising this is to write the updating equation in an incremental form: $M_t = M_{t-1} + \delta_t(I_t)$, where δ_t is the increment function, depending on the current sample I_t. In the case of exponential smoothing, δ_t is the affine function $\alpha(I_t - M_{t-1})$ (Fig. 1(1)). This linear dependence is not satisfying, since, in most cases, a sample which is far from the average is out of the background model and should have little effect on the estimate updating. This problem can still be addressed in the exponential smoothing framework, by using two distinct constants α_1 and α_2 such that $\alpha_1 > \alpha_2$, and by defining $\delta_t(I_t) = \alpha_1(I_t - M_{t-1})$ if I_t is in the background model, and $\delta_t(I_t) = \alpha_2(I_t - M_{t-1})$ if I_t is foreground. This results in a discontinuous increment function δ_t, as shown in Fig. 1(2), where the decision background/foreground is done by simply thresholding the absolute difference: The pixel is foreground if $|I_t - M_{t-1}| > Th$. It appears however, that the discontinuity of δ_t makes the choice of Th critical.

To get a more continuous behaviour, we shall follow [9], who suggests that the weight α attached to the current sample I_t should depend on its probability $f_t(I_t)$. But, as noted by [10], the mere product $\delta_t(I_t) = f_t(I_t) \times \alpha(I_t - M_{t-1})$ suggested by [9] is not usable in practise because of increments too small in general to be numerically operative. A proper way to achieve this, if α_{max} is the maximal desired weight, and as M_{t-1} is the mode of the current distribution f_t, is to use:

$$\delta_t = \frac{\alpha_{max} f_t(I_t)}{f_t(M_{t-1})} \times (I_t - M_{t-1}). \tag{1}$$

If we use a Gaussian distribution as density model like in [9], we get the following increment function:

$$\delta_t = \alpha_{max} \times \exp(\frac{-(I_t - M_{t-1})^2}{2V_{t-1}}) \times (I_t - M_{t-1}). \quad (2)$$

The model needs the temporal variance V_t. In [9], it is computed as the Gaussian estimation of the series $(I_t - M_t)^2$. But this leads to a double auto-reference in the definitions of M_t and V_t, which is prejudicial to the adaptation capability of the algorithm. We recommend rather to compute V_t as the exponential smoothing of $(I_t - M_t)^2$, using a fixed learning rate α_V.

One of the interest of computing V_t is that it provides a natural criterion of decision background/foreground through the condition $|I_t - M_t| > N \times \sqrt{V_t}$, with N typically between 2 and 4. Note that the increment function (Fig. 1(3)) is very different from the previous ones, and has a derivative-of-Gaussian shape.

This Gaussian estimation provides some attractive features compared to the exponential smoothing: the update of the estimates depends on the probability of the current sample, and the increment values are globally higher when the background density is more dispersed. Nevertheless, it is less used than exponential smoothing because of the computational load much higher. Now, what does the increment function look like if we take the Zipf law as the probabilistic model ?

3 Zipfian Background Estimation

Originally the Zipf law is an empirical principle [11] at the crossroads of linguistic and information theory, stating that, in any sense-making text, the probability of occurrence of the n^{th} most frequent word is $1/n$ the probability of occurrence of the (first) most frequent word. So the Zipf distribution is a hyperbolic decreasing function. Recently, it has been used in several applications of image processing [12], in particular as a model for the distribution of local spatial features. We use it here as a model for (pixel-wise) temporal distribution.

Because of the divergence of the sum $1/n$, the Zipf density function includes a power factor: $1/n^s$, with $s > 1$. The general expression of the continuous symmetric Zipf-Mandelbrot distribution can be written:

$$Z_{(\mu,k,s)}(x) = \frac{(s-1)k^{s-1}}{2(|x-\mu|+k)^s}. \quad (3)$$

In this expression, the parameter μ represents the mode of the distribution, and k determines its dispersion. The remarkable property of Z, taken as the density model f_t of the time series I_t (and then, replacing eq. 3 in eq. 1), is the shape of the increment function δ_t (Fig. 1(3)), which is close to the Heaviside shaped function: $H_{(\mu,\kappa)}(x) = -\kappa$ if $x < \mu$, $+\kappa$ if $x > \mu$, with $\kappa = \alpha_{max}k^s$. Thus it is naturally related to the Σ-Δ modulation, classically used in Analog to Digital Conversion:

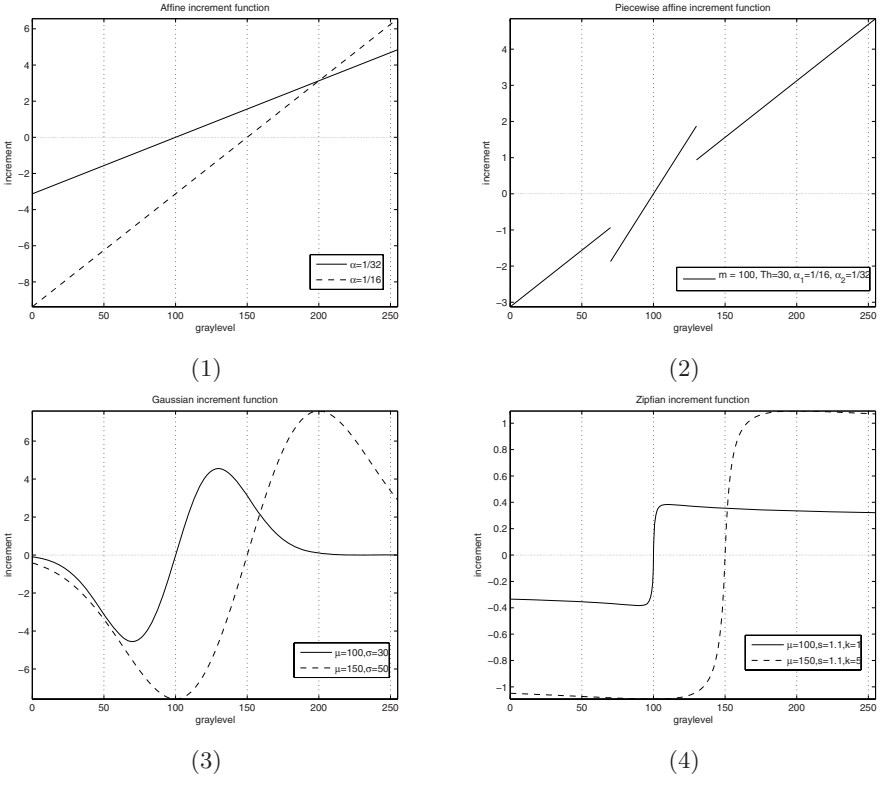


Fig. 1. The different increment functions δ_t associated to the different distribution models: (X axis: graylevel I_t, Y axis: increment value $\delta_t(I_t)$). (1) Exponential smoothing (plain: $\alpha = 1/32$; dashed: $\alpha = 1/16$) (1) Bi-level exponential smoothing ($m = 100$, $Th = 30$, $\alpha_1 = 1/16$, $\alpha_2 = 1/32$) (3) Gaussian laws, $\alpha_{max} = 1/4$ (plain: $\mu = 100$, $\sigma = 30$; dashed: $\mu = 150$, $\sigma = 50$) (4) Zipf laws, $\alpha_{max} = 1/4$ (plain: $\mu = 100$, $k = 1$, $s = 1.1$; dashed: $\mu = 150$, $k = 5$, $s = 1.1$).

For every time step Δt:
If $M_{t-\Delta t} > I_t$ then $M_t = M_{t-\Delta t} - \varepsilon$;
If $M_{t-\Delta t} < I_t$ then $M_t = M_{t-\Delta t} + \varepsilon$;

Here, the average increment per time unit is $\kappa = \frac{\varepsilon}{\Delta t}$. Digitally, the elementary increment ε is the least significant bit of the representation, i.e. 1 if the values are integer-coded. Adaptation to the dispersion of the model can then be done by tuning the updating period Δt: the greater the variance, the smaller Δt should be. The following algorithm reproduces such behaviour. The principle is to attach to every pixel, in addition to the mode estimator M_t, a dispersion estimator V_t. Suppose that $V_t \in]0, 2^m - 1[$, which means that it is coded on m bits:

For every frame t: {
rank = t % 2^m ; pow2 = 1 ;
do { pow2 = 2× pow2 ; } while((rank % pow2 == 0) && (pow2 < 2^m))

If $(V_{t-1} > \frac{2^m}{pow2})$ {
  If $M_{t-1} > I_t$ then $M_t = M_{t-1} - 1$;
  If $M_{t-1} < I_t$ then $M_t = M_{t-1} + 1$;
  }
$D_t = |I_t - M_t|$;
If $(t \ \% \ T_V == 0)$ {
  If $V_{t-1} > \max(V_{min}, N \times D_t)$ then $V_t = V_{t-1} - 1$;
  If $V_{t-1} < \min(V_{max}, N \times D_t)$ then $V_t = V_{t-1} + 1$;
  }
}

Here $x\%y$ is x modulo y. The purpose of the two first lines of the algorithm (which are computed once for all the pixels at every frame) is to find the greatest power of two (pow2) that divides the time index modulo 2^m (rank). Once this has been determined, it is used to compute the minimal value of V_{t-1} for which the Σ-Δ estimate M_t will be updated. Thus the (log-)period of update of M_t is inversely proportional to the (log-)dispersion: if $V_t > 2^{m-1}$, M_t will be updated every frame, if $2^{m-2} \leq V_t < 2^{m-1}$, M_t will be updated every 2 frames, and so on.

The dispersion factor V_t is computed here as the Σ-Δ estimation of the absolute differences D_t, amplified by a parameter N. Like in Gaussian estimation, we avoid double auto-reference by updating V_t at a fixed period T_V. V_t can be used as a foreground criterion directly: the sample I_t is considered foreground if $D_t > V_t$. V_{min} and V_{max} are simply used to control the overflows ; 2 and $2^m - 1$ are their typical values.

Note that the time constants, which represent the period response of the background estimation algorithm, are related here to the dynamics (the number of significant bits) of V_t, and to its updating period T_V. For the other methods, the time constants were associated to the inverse of the learning rates: $1/\alpha_i$ for the exponential smoothing and $1/\alpha_{max}$ and $1/\alpha_V$ for Gaussian estimation.

4 Results

Figure 2 shows the background estimation for all the time indexes, and one particular pixel. This is a difficult case for pixel-level motion detection: an outdoor scene where the background signal (high grass meadow with wind) is corrupted by the passage of two foreground objects. The Boolean condition "$D_t > V_t$" is used as foreground classification.

Figure 3 shows the result for all the pixels, at 4 different time indexes of the classical *Hall* sequence, in which two people are moving in radial motion, i.e. in the direction of the optical axis. This is a difficult case too, since the values in the centre of the moving objects do not change much (aperture problem). For reference, the last row of Figure 3 displays the hand drawn ground truth for the 4 frames.

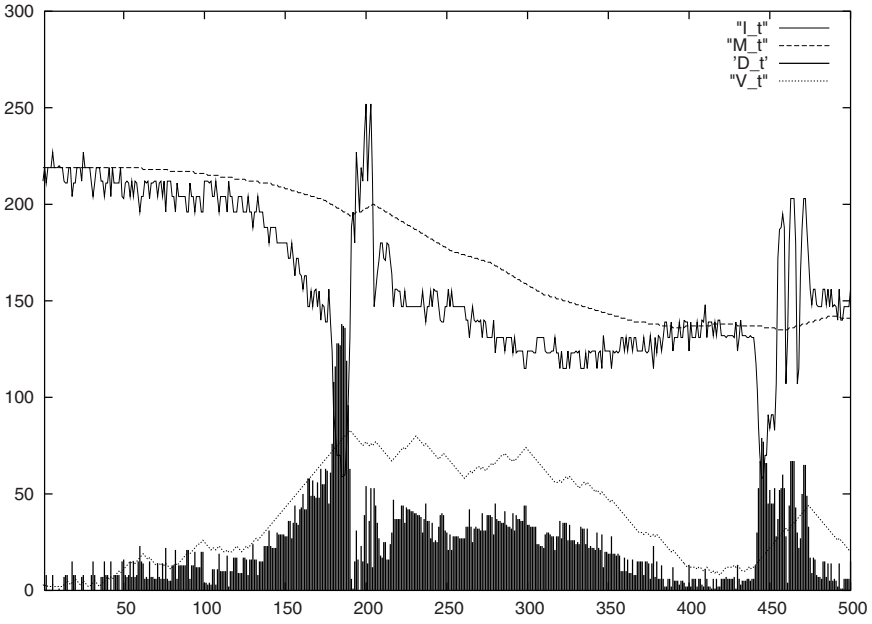


Fig. 2. Σ-Δ background estimation running on a given pixel. (X axis: time index, Y axis: graylevel). All values are 8-bit coded. Amplification factor N is 2. Variance updating period T_V is 1. Plain line: I_t, Dotted line: M_t, Impulses: D_t, Dashed line: V_t.

This ground truth is used for the quantitative evaluation (detection and false alarm rates are averaged on these 4 key frames) shown on Table 1, for different values of the amplification constant N, and of the updating period T_V. Those results are resumed on Figure 4, where the 9 Σ-Δ algorithms are compared with 6 different Gaussian algorithms. Note that these figures relate to pixel-level methods, and should not be interpreted in absolute, since a simple spatial regularisation appreciably improves the two measures, in all cases.

Table 1. (Detection, False alarm) rates for 9 Σ-Δ background subtraction algorithms. Measures are averaged on the 4 key frames of the *Hall* sequence.

	N=1	N=2	N=4
$T_V = 1$	(0.74,0.25)	(0.62,0.10)	(0.53,0.02)
$T_V = 8$	(0.91,0.38)	(0.87,0.23)	(0.85,0.12)
$T_V = 32$	(0.95,0.45)	(0.94,0.38)	(0.94,0.33)

The relevance and power of the Σ-Δ estimation, as a pixel-level temporal filter, is comparable to that of the Gaussian estimation, whereas its computational cost is even inferior to that of exponential smoothing. Indeed, the algorithm proposed in Section 3 is straightforward to compute in any fixed-point

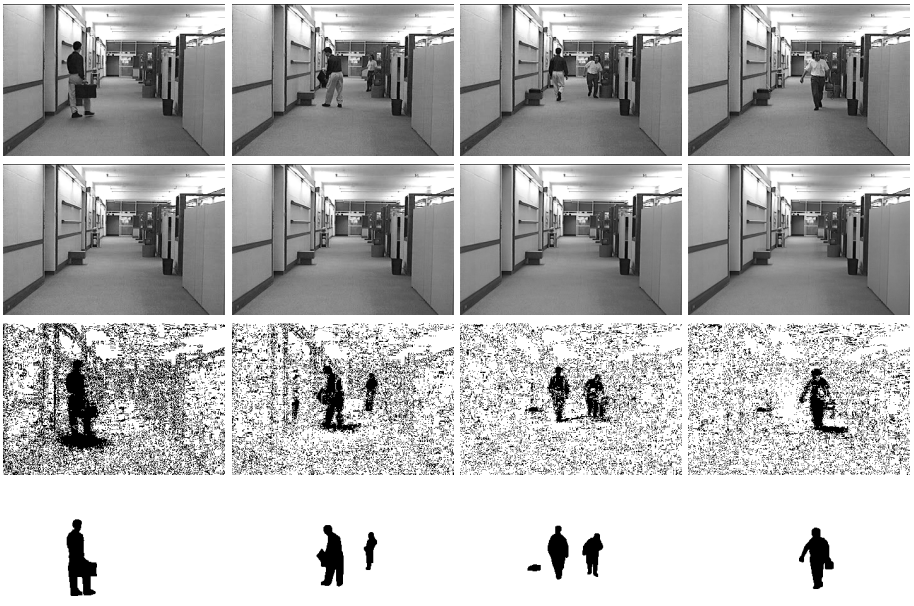


Fig. 3. Background subtraction shown at different frames of the *Hall* image sequence. Row 1: Original sequence, frames 38, 90, 170 and 250. Rows 2 and 3: Σ-Δ Background and foreground, with N=2, and $T_V = 8$. Row 4: (Fore)ground truth.

arithmetic, using an instruction set limited to: absolute difference, comparison, increment/decrement. Thus, it is well adapted to real-time implementation using dedicated or programmable hardware.

Another important computational property of Σ-Δ background subtraction, is that, once chosen the number of bits used to represent the estimates M_t and V_t, every computation can be made at full precision without increasing the data width. This allows in particular to make the most of the data parallelism provided by the SIMD-WAR (Single Instruction Multiple Data Within A Register) paradigm, which consists in concatenating many short operands in one single very long variable, and then applying scalar operations on the long variables. This implementation is available on most personal computers, using for example the SSE-2 (Streaming SIMD Extensions 2) instructions of the Intel®C++ compiler [13]. We provide hereunder the vectorised pseudo-code of the Σ-Δ background subtraction. Here, a 16-times acceleration is achieved by performing the operations on a 128-bit register made of 16 8-bit data.

```
vmin = 2; vmax = 255; logN = 1; Tv = 4;// Scalar constants definition
// Vector constants definition: creates 128-bit constants
// by concatenating 16 8-bit constants
VMIN = vector16_define(vmin);
VMAX = vector16_define(vmax);
// Sigma-Delta initializations
for(i=0; i<height; i++) {
```

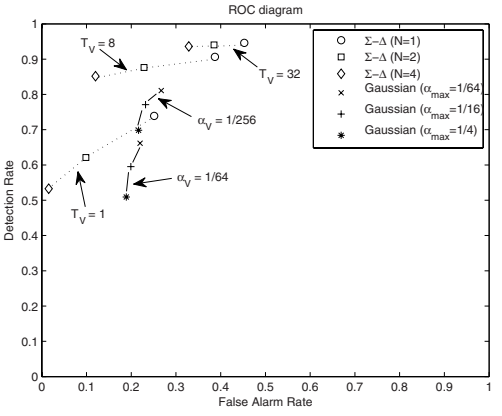


Fig. 4. Detection / false alarm rates diagram, for 9 Σ-Δ and 6 Gaussian background subtraction algorithms. Measures are averaged on the 4 key frames of the *Hall* sequence.

```
|  for(j=0; j<width/16; j++) {
|  |  I = I(0);// I(0): first image
|  |  M = I; // M(0) = I(0)
|  |  V = VMIN; // V(0) = vmin
|  }
}
for(t=1; t<=tstop; t+=1) {// Time loop*********************************
|  // Computation of the update threshold according to the time index
|  rank = (t%256); pow2 = 1; thres = 256;
|  do { pow2 = pow2*2; thres = thres/2;
|     } while (((rank%pow2)==0)&&(thres>1));
|  TH = vector16_define(thres);// vector variable
|  for(i=0; i<height; i++) {// Space loop------------------------------
|  |  for(j=0; j<width/16; j++) {
|  |  |  // (1) Update of Background M(t)
|  |  |  I = I(t);//loading I(t)
|  |  |  UPDATE = vector16_compare_greater(V,TH);// Comparison (>)
|  |  |  //if V(t-1)>th, update= FF (-1), else update = 0
|  |  |  C1 = vector16_compare_greater(I,M);
|  |  |  //if I(t)>M(t-1), c1= FF (-1), else c1 = 0
|  |  |  C2 = vector16_compare_less(I,M);// Comparison (<)
|  |  |  //if M(t-1)>I(t), c2= FF (-1), else c2 = 0
|  |  |  C1 = vector128_and(C1,UPDATE);// Bit-wise logical AND: Update is
|  |  |  C2 = vector128_and(C2,UPDATE);// effective only if V(t-1) > th
|  |  |  M = vector16_sub(M,C1);//M(t) = M(t-1) - c1
|  |  |  M = vector16_add(M,C2);//M(t) = M(t-1) + c2
|  |  |  // (2) Computation of absolute difference D(t)
|  |  |  MAX = vector16_max(I,M); // max(I(t),M(t))
|  |  |  MIN = vector16_min(I,M); // min(I(t),M(t))
|  |  |  D = vector16_sub(MAX,MIN); // d = |I(t) - M(t)|
|  |  |  // (3) Update of variance V(t): one over Tv frames
```

```
|  |  |  if (t % Tv == 0) {
|  |  |  |  ND = D;  // Difference amplification (Saturated addition)
|  |  |  |  for (k=1;k<=logN;k++) ND = vector16_add_sat(ND,ND);
|  |  |  |  BDEC = vector16_max(ND,VMIN);// Variance is bounded
|  |  |  |  BINC = vector16_min(ND,VMAX);// between Vmin and Vmax
|  |  |  |  C1 = vector16_compare_greater(V,BDEC);
|  |  |  |  //if V(t-1)>max(D(t),Vmin) c1= FF (-1), else c1 = 0
|  |  |  |  C2 = vector16_compare_less(V,BINC);
|  |  |  |  //if V(t-1)<min(D(t),Vmax) c2= FF (-1), else c2 = 0
|  |  |  |  V = vector16_add(V,C1);//V(t) = V(t-1) + c1
|  |  |  |  V = vector16_sub(V,C2);//V(t) = V(t-1) - c2
|  |  |  }
|  |  |  // (4) Computation of Foreground label L(t)
|  |  |  L = vector16_compare_greater(D,V);
|  |  |  //if D(t)>V(t) L(t)= FF, else L(t) = 0
|  |  }
|  }// end of space loop-----------------------------------------
}// end of time loop*********************************************
```

5 Conclusion and Extensions

We have proposed a justification of using the Σ-Δ estimation as a background subtraction method, based on the use of the Zipf law as a density model. We have proposed an algorithm implementing this method and allowing to adapt the background updating to the temporal dispersion. We have shown the computational advantages of the Σ-Δ estimation, illustrated by the vector SIMD implementation.

The limitations of this algorithm - used "as is" in a motion detection system - are inherent to its mono-modal nature: first, one single mode in the density model can be inefficient to discriminate moving objects over a complicated background, and second, one single dispersion estimate, related to one time constant, may not be sufficient for certain kind of motion such as remote objects with radial velocity w.r.t. the optical centre. Nevertheless the basic model can be enriched, either by using a multi-modal Zipfian distribution like it is done in [9] for Gaussian estimation, or by using several time magnitudes, as shown in [8].

References

1. Karmann, K.P., von Brandt, A.: Moving Object Recognition Using an Adaptive Background Memory. In: Time-Varying Image Processing and Moving Object Recognition, Elsevier, Amsterdam (1990)
2. Toyoma, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and Practice of Background Maintenance. In: Proc. IEEE ICCV, Kerkyra - Greece, pp. 255–261 (1999)
3. Elgammal, A., Harwood, D., Davis, L.: Non-parametric Model for Background Subtraction. In: Proc. IEEE ECCV, Dublin - Ireland (2000)

4. Piccardi, M.: Background subtraction techniques: a review. In: Proc. of IEEE SMC/ICSMC (October 2004)
5. Cheung, S.C., Kamath, C.: Robust techniques for background subtraction in urban traffic video. In: Proc. SPIE Video Com. and Image Proc. San Jose - CA (2004)
6. McFarlane, N., Schofield, C.: Segmentation and tracking of piglets in images. Machine Vision and Applications 8, 187–193 (1995)
7. Manzanera, A., Richefeu, J.: A robust and computationally efficient motion detection algorithm based on Σ-Δ background estimation. In: Proc. ICVGIP 2004, pp. 46–51 (December 2004)
8. Manzanera, A., Richefeu, J.: A new motion detection algorithm based on Σ-Δ background estimation. Pattern Recognition Letters 28, 320–328 (2007)
9. Stauffer, C., Grimson, E.: Learning patterns of activity using real-time tracking. IEEE Trans. on PAMI 22(8), 747–757 (2000)
10. Power, P., Schoonees, J.: Understanding background mixture models for foreground segmentation. In: Imaging and Vision Computing New Zealand, Auckland, NZ (2002)
11. Zipf, G.: Human behavior and the principle of least-effort. Addison-Wesley, New-York (1949)
12. Caron, Y., Makris, P., Vincent, N.: A method for detecting artificial objects in natural environments. In: Int. Conf. in Pattern Recognition, pp. 600–603 (2002)
13. Intel, C.: Intel®C++ Compiler for Linux Systems - User's Guide (1996-2003) Document number 253254-014

Evol. Intel. (2009) 2:85–102
DOI 10.1007/s12065-009-0021-4

RESEARCH PAPER

Evolution of visual controllers for obstacle avoidance in mobile robotics

Renaud Barate · Antoine Manzanera

Received: 18 February 2009 / Revised: 16 July 2009 / Accepted: 23 September 2009 / Published online: 23 October 2009

Abstract The purpose of this work is to automatically design vision algorithms for a mobile robot, adapted to its current visual context. In this paper we address the particular task of obstacle avoidance using monocular vision. Starting from a set of primitives composed of the different techniques found in the literature, we propose a generic structure to represent the algorithms, using standard resolution video sequences as an input, and velocity commands to control a wheel robot as an output. Grammar rules are then used to construct correct instances of algorithms, that are then evaluated using different protocols: evaluation of trajectories performed in a goal reaching task, or imitation of a hand-guided trajectory. A genetic program is applied to evolve populations of algorithms in order to optimize the performances of the controllers. The first results obtained in a simulated environment show that the evolution produces algorithms that can be easily interpreted and which are clearly adapted to the visual context. However, the resulting trajectories are often erratic, and the generalization capacities are poor. To improve the results, we propose to use a two-phase evolution combining imitation and goal reaching evaluations, and to add some constraints in the grammar rules to enforce a more generic behavior. The results obtained in simulation show that the evolved algorithms are more efficient and more generic. Finally, we apply the imitation based evolution on real sequences and test the evolved algorithms on a real robot. Though simplified by dropping the goal reaching constraint, the resulting algorithms behave well in a corridor centering task, and show certain generalization capacities.

Keywords Genetic programming · Computer vision · Mobile robotics · Obstacle avoidance

R. Barate · A. Manzanera (✉)
ENSTA, 32 Bd Victor, 75739 Paris Cedex 15, France
e-mail: Antoine.Manzanera@ensta.fr

R. Barate
e-mail: Renaud.Barate@ensta.fr

1 Introduction

The autonomy of a mobile robot refers to its capacity of interacting with a—possibly unknown—environment, with the lowest level of human supervision. This capacity obviously requires perception abilities. Amongst these, vision is certainly one of the cheapest, most informative, but most difficult to use.

Aside from perception, the other condition for acquiring autonomy is cognition: the robot makes the most of experiences gathered through learning or adaptation. An abundance of work can be found in the literature combining machine learning and vision in mobile robotics.

But it is notable that, generally, the cognitive dimension of vision itself is ignored: the visual processing is not the fruit of learning but some benchmarked, validated, and often sophisticated algorithm which produces features, and only some descriptors attached to these features take part in the learning process.

However, from a biological point of view, even the earliest stages of visual perception are the result of an adaptation process. For a mobile robot, using some simple and *ad hoc* visual procedure results in a loss of genericity, whereas using more sophisticated and robust visual feature induces a high computational load, both solutions being at the detriment of autonomy.

It is true that there are good combinatorial arguments for not integrating image processing in the adaptation loop: the

size and variability of data and algorithms lead to very high computational costs. Nevertheless, the present development level of High Performance Computing [15] allows us to envisage massive concurrent computation of large number of visual primitives on the same machine.

Our objective is to automatically design vision algorithms that are well suited to a given environment, and to a given task of a mobile robot. The task that we address in this paper is the obstacle avoidance problem, as it is one of the basics for the robot to acquire autonomy. Considering a large family of visual primitives, we propose a generic structure of visual obstacle avoidance algorithm, used to construct any instance of valid algorithms from those primitives. Evolutionary computation is then used, together with an evaluation protocol to automatically select the algorithms that are best suited in the current environment. Experiments are shown in a simulated environment, then on a real robot.

Section 2 recalls the related works, i.e. learning visual obstacle avoidance techniques, or evolutionary computer vision. Section 3 presents some state of the art visual obstacle avoidance techniques (Sect. 3.1) used to define the set of algorithmic primitives (vocabulary) chosen to construct the algorithms (Sect. 3.2). The rules (grammar) that construct valid obstacle avoidance algorithms are presented in (Sect. 3.3). Section 4 presents the different evaluation methods used to select the algorithms, first in the context of reaching a given target (Sect. 4.1), and then imitating an existing trajectory (Sect. 4.2). We then present the evolution process, which is implemented through genetic programming (Sect. 4.3). In Sect. 5, we discuss the experimental results obtained in simulation. The global evolution, and the behavior of the best evolved algorithms are analyzed (Sect. 5.1). We then perform a comparison of different strategies used to improve the evolution (Sect. 5.2). Finally we evaluate the generalization abilities of the evolved controllers (Sect. 5.3), i.e. their capacity to keep on avoiding obstacles when their positions change (the environmental visual appearance remaining the same). Section 6 presents the results on a real Pioneer 3 DX Robot. We detail the evolution principles in the real world and show the evolved controllers, first in their evolution environment (Sect. 6.1), and next in an unseen environment (Sect. 6.2). Finally, Sect. 7 draws the conclusion and discusses the perspective of this research.

2 Related works

Different machine learning methods have been proposed in the literature for obstacle avoidance using vision. Michels and Saxena [22, 29] used reinforcement learning to associate depth to texture patches. Low and Wyeth [18] used three-layer neural network to learn the robot heading command from the apparent motion (optical flow) vector field input. In these works, a depth map acquired by laser range sensor is used as ground truth for supervision purposes. In all cases, one visual primitive is used exclusively, neglecting any other source of information. Le Cun et al. [16] designed an obstacle avoidance learning system using a neural network whose input is a pair of stereo images and output is a steer angle. This system uses video and command sequences recorded by a hand guided robot as supervised learning input.

On the other hand, evolutionary techniques have already been used for robotic navigation and the design of visual obstacle avoidance controllers [28, 34] but in general vision is overly simplified. For instance, Marocco [20] used only a 5×5 pixels retina as visual input. Aside from obstacle avoidance, genetic programming has been proved to achieve human-competitive results in image processing systems, e.g. for the detection of interest points, as shown by Olague and his co-workers [27, 31, 32]. Cooperative coevolution methods (e.g. Parisian evolution) have also produced good results for obstacle detection [26] and 3D reconstruction, the latter used either for computing the 3D coordinates from a pair of images [25], or for optimizing the placement of the different cameras [5]. A recent tutorial on evolutionary computer vision was given by Cagnoni [2].

Ebner and Zell have used genetic programming to automatically design interest point detectors. From a set of basic image operations, they first tried to retrieve an existing operator (the Moravec detector), by minimizing the sum of squared differences between the desired and actual output images [6]. As a second approach, they used a task dependent fitness based on the quality of matches between two sets of interest points in a training sequence [8]. Furthermore, they were also interested in the obstacle avoidance problem and developed a monocular vision system to center a robot in a hallway that performed quite well [9]. However, this system did not use an automatically evolved algorithm, instead it used the original Moravec detector.

To our knowledge, only Martin [21] tried evolutionary techniques with monocular images for obstacle avoidance. The structure of his algorithm is based on the floor segmentation technique and the evaluation is done with a database of hand labeled real world images. The advantage of such an approach is that the evolved algorithms are more likely to work well with real images than those evolved with computer rendered images. Nevertheless, it introduces an important bias since the algorithms are only selected on their ability to label images in the database correctly and not on their ability to avoid obstacles.

3 Generic structure of visual controllers

3.1 Visual obstacle avoidance algorithms

In the literature, the obstacle avoidance algorithms using monocular vision can be clearly divided in two categories: motion-based and appearance-based strategies.

The first category, sometimes referred to as "monocular stereovision" uses the parallax principle, according to which, when the robot moves in purely translational motion, the apparent velocity of closer objects is greater. This geometric property can be used through the estimation—by image processing—of different types of measure:

- local flow divergence, which is based on the local apparent velocity, and which can provide the depth or the time before impact of the projected point, depending on the available odometry information [3, 24].
- global flow balance, which is a measure of the symmetry between the left and the right sides of the apparent motion field. This technique is inspired by the strategy of flying insects, which use it to center themselves when flying in a narrow environment. The insects naturally take advantage of their pair of lateral sensors, however, the technique has been used with success on an autonomous helicopter using a wide field single camera [23].

Within this category, the necessary algorithmic primitives are: computation of the apparent motion field (optical flow), spatio-temporal smoothing filters, and regional integral computations.

In the second category of visual obstacle avoidance algorithms, a local visual feature is associated with contextual information, which can be related to depth, orientation, or even to a segmentation of the projected scene in principal surfaces (floor, walls typically) [14, 17, 33]. The algorithmic primitives from this category are: multi-scale and multi-orientation derivative measures, image threshold and binarization functions, horizontal and vertical projection measures, and regional integral computations.

Our objective is now to construct a generic description model of a visual obstacle avoidance controller which can be used to automatically design a formally valid algorithm, in such a way that any of the approaches presented above could be derived as an instance of a valid algorithm.

3.2 Choice of the primitives

Generally speaking, a vision algorithm can be divided in three main parts: first, the algorithm will process the input image with a number of filters to highlight some features. These features are then extracted, i.e. represented by a small set of scalar values. Finally these values are used for a domain dependent task, in our case to generate motor commands to avoid obstacles. We designed the structure of our algorithms according to this general scheme. First, the filter chain consists of spatial and temporal filters, optical flow calculation and projection that will produce an image highlighting the desired features. We then compute the mean of the pixel values on several windows of this transformed image (feature extraction step). Finally those means are used to compute a single scalar value by a linear combination. We will use this scalar value to determine the presence of an obstacle and to generate a motor command to avoid it.

An algorithm is represented as a tree, the leaves being input data (video image or scalar constant value), the root being output command (generating linear and yaw speed), and the internal nodes being primitives (transformation steps). The program can use different types of data internally, i.e. scalar values, images, optical flow vector fields or motor commands. For each primitive, the input and output data types are fixed. Some primitives can internally store information from previous states, thus allowing temporal computations like the calculation of the optical flow.

Most of those primitives also use parameters along with the input data (for example, the standard deviation value for the Gaussian filter). The parameters are specific to each algorithm; they are randomly generated when the corresponding primitive is created by the genetic programming system described in Sect. 4.3 For each parameter, we define the distribution used to generate it (uniform or normal) along with the range of valid values. We also define the mean and standard deviation of the normal distribution where applicable. The parameters with their distributions are detailed in Table 1. Here is the list of all the primitives that can be used in the programs and the data types they manipulate:

– Spatial filters (*input: image, output: image*):

  - Gaussian filter: This low-pass filter has one parameter: the standard deviation of the Gaussian function.
  - Laplacian filter: This high-pass filter has no parameter, it is defined by the following 3 × 3 convolution kernel:

0	1	0
1	-4	1
0	1	0

Convolution kernel defining the Laplacian filter used in our system.

  - Threshold filter: The parameter defining this filter is the threshold value.
  - Gabor filter: This filter is used to detect patterns of given orientation and frequency. The parameters are: orientation, wavelength and bandwidth.

- Difference of Gaussians: The parameters are the standard deviations of each Gaussian function.
- Sobel filter: This filter is defined by its orientation. The 3×3 convolution kernels used for each orientation are:

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Convolution kernels defining the vertical (left) and horizontal (right) Sobel filters used in our system.

- Subsampling filter: This filter is defined by the size coefficient.

– Temporal filters (*input: image, output: image*):

- Pixel-to-pixel min, max, sum and difference of the last two frames. These filters have no parameters.
- Recursive mean operator. This filter is computed with the following formula at each pixel: $M_t = \alpha I_t + (1 - \alpha)M_{t-1}$ where I_t is the current pixel value and M_{t-1} is the filtered value at time $t - 1$. The parameter for this filter is the coefficient α.

– Optical flow (*input: image, output: vector field*):

- Horn and Schunck global regularization method [13].
- Lucas and Kanade local least squares calculation [19].
- Simple block matching method.

The rotation movement is first eliminated [9] by a transformation of the two images in order to facilitate further use of the optical flow.

– Projection (*input: vector field, output: image*):

- Projection on the horizontal or vertical axis.
- Euclidean norm computation.
- Manhattan norm computation.
- Time to contact calculation using the flow divergence.

– Windows integral computation (*input: image, output: scalar*): The method used for this transformation is:

1. A global coefficient α_0 is defined for the primitive.
2. Several windows are defined on the left half of the image with different positions and sizes. With each window is paired a second window defined by symmetry along the vertical axis. A coefficient α_i and an operator ($+$ or $-$) are defined for each pair.
3. The resulting scalar value R is a simple linear combination calculated with the following formula:

$$R = \alpha_0 + \sum_{i=1}^{n} \alpha_i \mu_i \qquad (1)$$

$$\mu_i = \mu_{Li} + \mu_{Ri} \text{ or } \mu_i = \mu_{Li} - \mu_{Ri}$$

where n is the number of windows and μ_{Li} and μ_{Ri} are the means of the pixel values over respectively the left and right window of pair i. The number of windows pairs, their positions, sizes, operator and coefficient along with the global coefficient are characteristic parts of the primitive and will be customized by the evolutionary process.

– Scalar operators (*input: scalar(s), output: scalar*):

- Addition, subtraction, multiplication and division operators.
- Temporal mean calculation. This is simply the mean value computed on the N last time steps.

– Command generation (*input: two scalars, output: command*): The motor command is represented by two scalar values: the requested linear and angular speeds.

3.3 Construction grammars

We use context-free grammars to represent and build these vision algorithms. We will describe here the sets of terminal and non-terminal symbols in these grammars, and the two different structures we use to design our obstacle avoidance algorithms. The build process itself will be detailed in Sect. 4.3.

The base set of non-terminal symbols is simply the list of primitives that we have just described in Sect. 3.2. The terminal symbols are the current video image and a scalar constant (the constant value is a random generated parameter, see Table 1). Depending on the kind of structure used to build the algorithm, these sets will be completed with a few other symbols.

The first controllers we have constructed can be represented with a single algorithmic tree. This tree is entirely built by the evolutionary process without any *a priori* in the structure of the algorithm. In this paper, we will call them *structure-free controllers*. In this case, we shall add two terminal symbols to the base set: the distance (in cm) and the direction (in degrees) of the target point. We shall also add one non-terminal symbol: an if-then-else test, which is a scalar operator with four inputs and one output. If the first input value is greater than the second, the output value is equal to the third input value, otherwise it is equal to the fourth input value. This allows the evolutionary process to create more complex and non-linear combinations between new scalar input variables and the scalars issued from the vision part of the algorithm. Figure 1 shows a controller that can be built with this structure. The algorithm is a single tree and all the primitives between the input data and the resulting motor command are created and parameterized by the evolution.

Table 1 Parameters of the primitives used in our system (units are shown in parenthesis where applicable)

Primitive and parameter	Distribution	Min	Max	Mean	Std
Scalar constant					
Value	Normal	–	–	0.0	50.0
Temporal regularization					
Number of values	Normal	1	100	1	3
Windows integral computation					
Global coefficient α_0	Uniform	−180.0	180.0	–	–
Window					
Position x_1 (rel. to image width)	Uniform	0.0	0.5	–	–
Position x_2 (rel. to image width)	Uniform	0.0	0.5	–	–
Position y_1 (rel. to image height)	Uniform	0.0	1.0	–	–
Position y_2 (rel. to image height)	Uniform	0.0	1.0	–	–
Coefficients α_i	Normal	–	–	0.0	3.0
Operator (+ or −)	Uniform	–	–	–	–
Gaussian filter					
Standard deviation (pixels)	Normal	0.0001	20.0	3.0	2.0
Threshold filter					
Threshold (gray level)	Uniform	0	255	–	–
Gabor filter					
Orientation (degrees)	Uniform	0.0	180.0	–	–
Wavelength (pixels)	Normal	2.1	20.0	5.0	2.0
Bandwidth	Normal	0.5	2.0	1.0	0.3
Difference of Gaussians					
Standard deviation 1 (pixels)	Normal	0.0001	20.0	3.0	2.0
Standard deviation 2 (pixels)	Normal	0.0001	20.0	3.0	2.0
Sobel filter					
Orientation (H or V)	Uniform	–	–	–	–
Subsampling filter					
Size coefficient	Uniform	0.01	1.0	–	–
Recursive mean					
Multiplying coefficient α	Uniform	0.01	1.0	–	–
Horn-Schunck optical flow					
Weight coefficient α^2	Normal	0.0	100.0	2.0	1.0
Lucas-Kanade optical flow					
Size of the window (pixels)	Uniform	1	15	–	–

The price to pay for the maximal genericity in these structure-free controllers is that there is no guarantee that the evolved controllers will use the input variables (target heading and distance in particular). Thus, it is quite unlikely that such a controller evolved in a given environment could develop a geometric abstraction of the target reaching task.

This is why we also designed a second kind of controllers with a more constrained structure, which facilitates the trade-off between avoiding obstacles and reaching the target point. We will call them *structure-restricted controllers* in the rest of this paper. First, a vision algorithm is used to detect nearby obstacles. If no obstacle is around, the robot will just go straight to the target point. If an obstacle is detected, another vision algorithm will be used to generate a command to avoid this obstacle. The parts that will be customized by the evolution are the two vision algorithms along with a few parameters like the speed of the robot when going straight to the goal. This global structure is represented on Fig. 2.

The first vision algorithm (obstacle detection) is in fact always a simple function chain, starting with the video image and computing a Boolean as output. We obtain this Boolean value by comparing the scalar produced by the feature extraction step with a scalar threshold. This result will indicate the presence or absence of a nearby obstacle.

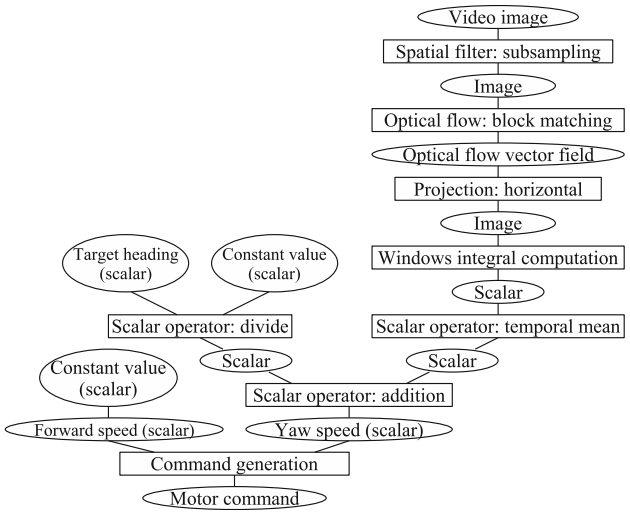


Fig. 1 Example of a structure-free controller for obstacle avoidance based on optical flow. *Rectangles* represent primitives and *ellipses* represent data

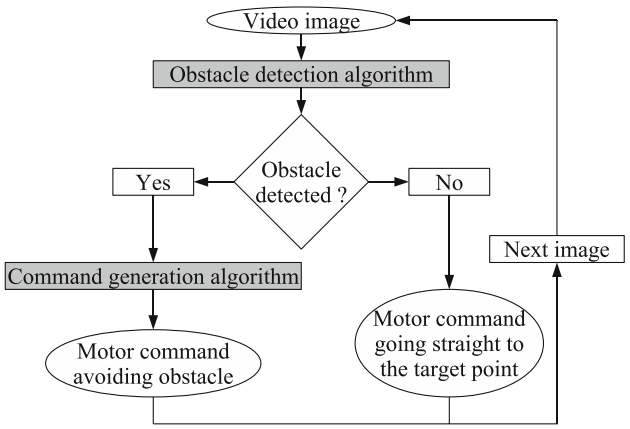


Fig. 2 Overview of the global fixed structure of the structure-restricted controllers. The algorithms in *light gray* will be customized by the evolution

Formally, this means that the grammar used to generate this algorithm will be slightly different. We add two non-terminal symbols to the base set: the Boolean function *not* and a threshold function with a scalar value as input, a Boolean value as output, and a scalar parameter (the threshold value). We also remove the terminal symbol *scalar constant*, the binary scalar operators and the command generation primitive.

The second vision algorithm (command generation) can use as an input either the original video image or the image filtered by the obstacle detection algorithm. This allows the reuse of interesting features between the two algorithms. The grammar used to generate this algorithm contains the base set of terminal and non-terminal symbols, completed with this *filtered image* terminal symbol. Figure 3 illustrates this restricted structure with an example of the two different algorithms.

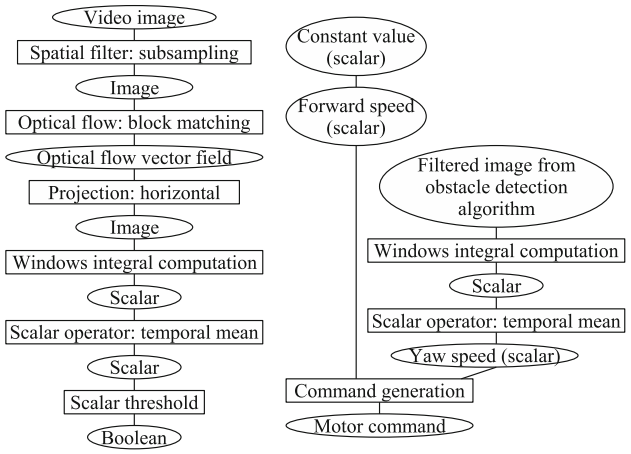


Fig. 3 Structure-restricted controller. *Left* an example of obstacle detection algorithm. *Right* an example of command generation algorithm. *Rectangles* represent primitives and *ellipses* represent data

4 Evaluation and evolution process

In this section we describe how the grammar presented above is used to automatically construct efficient algorithms. Starting from an initial population of randomly produced algorithms, we evaluate them to select the ones most adapted to the obstacle avoidance task, we then combine the best individuals to produce a new population, and reiterate the process to obtain a certain number of generations. Let us first present the different evaluation (fitness) functions we have used.

4.1 Evaluation 1: goal reaching versus contact

The principle of the first evaluation method is to measure the ability of the controllers to reach a given target whilst avoiding the obstacles. The protocol is as follows: a *course* is defined by the location of a starting point and a goal point in the environment. We place the robot at the starting point and let it move in the environment during 30 or 60 s (depending on the experiments) driven by the obstacle avoidance algorithm. Two scores are attributed to the algorithm depending on its performance: a goal-reaching score G rewards algorithms reaching or approaching the goal location, whereas contact score C rewards the individuals that did not hit obstacles on their way. Those scores are calculated with the following formulas:

$$G = \begin{cases} t_G \text{ if the goal is reached} \\ t_{\max} + d_{\min}/V \text{ otherwise} \end{cases} \qquad (2)$$
$$C = t_C$$

where t_G is the time needed to reach the goal in seconds, $t_{\max}$ is the maximum time in seconds (30 or 60 s), $d_{\min}$ is the minimum distance to the goal achieved in meters, V is a

constant of 0.1 m/s and t_C is the time spent near an obstacle (i.e. less than 18 cm, which forces the robot to keep some distance away from obstacles). The goal is hence to minimize those two scores G and C. For better generalization performances, we designed several courses with different starting points and goal locations (two or four courses depending on the experiments). Final scores are the means of the scores obtained on the different courses. The starting and goal points are fixed because we want to evaluate all algorithms on the same problem.

Due to the number of individuals to be evaluated and the necessity of keeping the conditions the same for all of them, this protocol cannot be implemented in a real environment. As such, we use a simulation environment in which the robot moves freely during each experiment. The simulation is based on the open-source robot simulator Gazebo (http://www.playerstage.sourceforge.net/index.php?src=gazebo). The simulator uses ODE physics engine (http://www.ode.org/) for the movement of the robot and collisions detection and OpenGL (http://www.opengl.org/) for the rendering of the camera images. The physics engine update rate is 50 Hz, while the camera update rate is 10 Hz. In all the experiments presented in this paper, the simulated camera produces 8-bits gray-value images of size 320 × 160 representing a field of view of approximately 100° × 60°. This large field of view reduces the dead angles and hence facilitates obstacle detection and avoidance. The simulation environment is a closed room of 36 m^2 area (6 m × 6 m) containing three bookshelves (Fig. 4). All the obstacles are immovable to prevent the robot from just pushing them instead of avoiding them.

4.2 Evaluation 2: imitation

In the obstacle avoidance problem, it is very difficult to manually design a mediocre controller but it is very easy to manually guide the robot toward the target point while avoiding obstacles. We can obtain a good example of an efficient behavior by recording the video sequence and command parameters while we guide the robot. With this evaluation method, we try to evolve algorithms that imitate this efficient example behavior.

Fig. 4 Snapshot of the simulation environment, containing three bookshelves in a 6 m × 6 m closed room

For the evaluation of the algorithms, we replay the recorded sequence, using it as input of the evaluated algorithm, and compare the command issued by this algorithm with the command recorded during the manual control of the robot. The goal is to minimize the difference between these two commands along the recorded sequence. Formally, we try to minimize two variables F and Y defined by the formulas:

$$F = \sqrt{\sum_{i=1}^{n} (f_{Ri} - f_{Ai})^2} \text{ and } Y = \sqrt{\sum_{i=1}^{n} (y_{Ri} - y_{Ai})^2} \qquad (3)$$

where f_{Ri} and y_{Ri} are the recorded forward and yaw speed commands for frame i, f_{Ai} and y_{Ai} are the forward and yaw speed commands from the tested algorithm for frame i and n is the number of frames in the video sequence.

In this case, we also perform a set of several courses using different recorded sequences, and compute the average of the scores obtained on the different sequences.

Compared with the previous evaluation, this method is somewhat less generic, as it forces the controller to follow a guided trajectory, which restricts the type of solution that an individual can provide. Nonetheless, the imitation strategy may favor efficient solutions from an energetic point of view, as the guided trajectories are deliberately the smoothest, safest and as direct as possible.

Furthermore, unlike the last method, the imitation strategy can be implemented in a real environment as easily as in a virtual environment: the recorded video and command sequence can be acquired using a real platform, such as the Pioneer 3 DX robot (Fig. 5).

4.3 Evolution through genetic programming

We use genetic programming to evolve vision algorithms with little *a priori* on their structure. As usual with evolutionary algorithms, the population is initially filled with randomly generated individuals. We use the grammar

Fig. 5 The Pioneer 3 DX robot with its Canon VC-C50i camera

Table 2 Grammar used in the genetic programming system for the creation and transformation of the command generation algorithm for structure-restricted controllers. The grammars used for the obstacle detection algorithm and for the structure-free controllers are very similar to this one

[1.0] START → COMMAND	[0.15] SPATIAL_FILTER → gaussian	[0.2] PROJECTION → horizontalProjection
[1.0] COMMAND → directMove(REAL,REAL)	[0.14] SPATIAL_FILTER → laplacian	[0.2] PROJECTION → verticalProjection
[0.15] REAL → scalarConstant	[0.14] SPATIAL_FILTER → threshold	[0.2] PROJECTION → euclideanNorm
[0.075] REAL → add(REAL,REAL)	[0.14] SPATIAL_FILTER → gabor	[0.2] PROJECTION → manhattanNorm
[0.075] REAL → subtract(REAL,REAL)	[0.14] SPATIAL_FILTER → diffOfGaussians	[0.2] PROJECTION → timeToContact
[0.05] REAL → multiply(REAL,REAL)	[0.14] SPATIAL_FILTER → sobel	
[0.05] REAL → divide(REAL,REAL)	[0.15] SPATIAL_FILTER → subsampling	
[0.1] REAL → temporalRegularize(REAL)	[0.2] TEMPORAL_FILTER → temporalMin	
[0.5] REAL → windowsIntegralCompute(IMAGE)	[0.2] TEMPORAL_FILTER → temporalMax	
[0.3] IMAGE → videoImage	[0.2] TEMPORAL_FILTER → temporalSum	
[0.3] IMAGE → previouslyFilteredImage	[0.2] TEMPORAL_FILTER → temporalDiff	
[0.25] IMAGE → SPATIAL_FILTER(IMAGE)	[0.2] TEMPORAL_FILTER → recursiveMean	
[0.1] IMAGE → PROJECTION(OPTICAL_FLOW)		
[0.05] IMAGE → TEMPORAL_FILTER(IMAGE)		
[0.33] OPTICAL_FLOW → hornSchunck(IMAGE)		
[0.33] OPTICAL_FLOW → lucasKanade(IMAGE)		
[0.34] OPTICAL_FLOW → blockMatching(IMAGE)		

based genetic programming system introduced by Whigham [35] to overcome the data typing problem. It also allows us to bias the search toward more promising primitives and to control the growth of the algorithmic tree.

In the same way that a grammar can be used to generate syntactically correct random sentences, a genetic programming grammar is used to generate valid algorithms. The grammar defines the primitives and data (the bricks of the algorithm) and the rules that describe how to combine them. The generation process consists in successively transforming each non-terminal node of the tree with one of the rules. This grammar is used for the initial generation of the algorithms and for the transformation operators. The crossover consists in swapping two subtrees issued from identical non-terminal nodes in two different individuals. The mutation consists in replacing a subtree by a newly generated one. Table 2 presents the grammar we used in our experiments with the structure-restricted controllers.

The numbers in brackets are the probability of selection for each rule. A major advantage of this system is that we can bias the search toward the usage of more promising primitives by setting a high probability for the rules that generate them. We can also control the size of the tree by setting small probabilities for the rules that are likely to cause an exponential growth (rules like REAL → add(REAL,REAL) for example).

As described previously, we wish to minimize two criteria (G and C for the first method, F and Y for the second one). There are different ways to use evolutionary algorithms to perform optimization on several and sometimes conflicting criteria. For the experiments described in this paper, we chose the widely used multi-objective evolutionary algorithm called NSGA-II. This algorithm is based on the Pareto dominance principle. Individuals are sorted by non-dominance rank, so that non-dominated individuals get a higher probability of being selected for breeding. This algorithm is elitist and a "crowding distance" is used to promote diversity among the individuals. More details can be found in the paper by Deb [4].

In order to prevent problems of premature convergence, we separate the population of algorithms in four islands, each containing 100 individuals. Those islands are connected with a ring topology; every tenth generation, five individuals selected with binary tournament will migrate to the neighbor island while five other individuals are received from the other neighbor island. The evolution lasts 100 generations, so each experiment represents 40,000 evaluations.

For the parameters of the evolution, we use a crossover rate of 0.8 and a probability of mutation of 0.01 for each non-terminal node. We use a classical binary tournament selection in all our experiments. Due to the length of the experiments, we did not conduct a thorough statistical analysis of the influence of those parameters, which were determined empirically.

5 Experiments in simulation

In this section we present and discuss the results obtained in simulation using the proposed system. In Sect. 5.1, we focus on the first results, obtained with the structure-free grammar, i.e. with the least *a priori*, and using the goal-reaching evaluation. In Sect. 5.2, we apply a two-phase evolution using the imitation strategy to guide the evolution, and compare the results with other strategies classically used to improve or speed up the evolution. In Sect. 5.3, we discuss the generalization performance and show the interest of using the restricted structure grammar.

5.1 Analysis of the evolved controllers

The objectives of our first experiments were to see what kind of controllers could be automatically designed with the minimal level of *a priori*, and without biasing the evolution with any subjective decision. As such, the evolution process in this section has been made with genetic programming using structure free grammar, and an evaluation based on

the objective performances of the algorithm, i.e. the goal-reaching evaluation. Three experiments have been conducted, using three different simulation environment: (1) A simple environment made of non-textured blocks, (2) A simple environment made of textured blocks and walls, and (3) A more realistic environment, made of a room with three bookshelves (analogous to Fig. 4).

The performance analysis of the successive generations can be done by plotting the contact vs. goal accession scores of the non dominated individuals of every generation, which correspond to the Pareto fronts. In the three types of environment, it can be observed that the performances increase rapidly during the first generations. The progression is much slower during the second half of the evolution, but is always globally significant, and the best algorithms are always better than the reference controller, that has been designed by hand for this specific environment. As an example, Fig. 6 shows the Pareto fronts for the textured blocks environment. In this case, the reference controller (represented here as the cross), is the algorithm based on balancing the average optical flow horizontal components on the left and right sides of the image. (The constructed algorithm corresponds to the tree shown on Fig. 1).

Another useful analysis to be made is to observe the results of the best individuals of the evolution, to see what type of behavior they have developed to avoid the obstacles. To do this, we first plot the actual trajectory that has been performed by the robot guided by the evolved algorithm, and then display what we can call the genotype of the individual, which corresponds to the constructed algorithm. Figure 7 shows as an example those data for one individual from the last Pareto front of the evolution realized on the textured blocks environment. More specifically, this individual corresponds to the point with the smallest contact score, i.e. the most careful behavior. If we look at the algorithmic tree, we can see that the forward speed

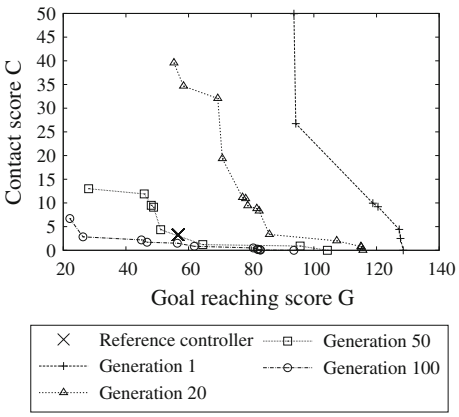


Fig. 6 Pareto fronts during the evolution process with the textured blocks environment

command is based on an optical flow computation, which allows to detect close frontal obstacles, and to generate a negative velocity command (the robot moves backward when a frontal obstacle is detected). The yaw speed command is based on a Gabor filter whose purpose is to move away from the lateral obstacles seen at a certain distance. The resulting trajectories show many backward motions, relatively poor results in goal reaching, but very few collisions.

Figure 8 displays the same observations for an individual evolved in the bookshelves environment. In this case, the forward speed relies on integral measures made after a Gaussian filter followed by a threshold. This corresponds to the detection of an obstacle, since the floor is globally lighter than the bookshelves or the walls. When the area covered by an obstacle is beyond a certain threshold, it generates a negative forward speed, corresponding to a backward motion. The yaw speed is simply provided by a linear function of the target direction, which allows the robot to maintain the global heading of the trajectory. As seen in the resulting trajectories, this algorithm is very efficient in the evolution environment: the target is always reached, and the trajectories are relatively rapid with very few contacts.

To summarise these first results, it can be said that our initial evolutionary system with objective evaluation measure has shown a certain level of efficiency since relevant adaptation behaviors were observed in the different environments. Furthermore, the progression of the Pareto curves proves that the best individuals of the last generations can favorably compete with hand-designed controllers in the evolution environment. Now, the main problems we have to address at this point are that: (1) the extreme variability of the best individuals from one experience to the other limits the usability and the generality of the evolved controllers, and (2) the trajectories obtained with the best individuals are often chaotic and not very efficient.

5.2 Comparison of different strategies

It is obvious that the size of the optimization space, corresponding to all the constructible algorithms is huge, and that only a tiny part of this space can be explored with the 40,000 evaluations. In order to limit the variability of the experiments and to get smoother trajectories, we have decided to use a 2-phase evolution using the imitation based evaluation, with the objective to guide the optimization process toward more promising regions of the controller space. Hence, in this section, the evolution is split into 2 phases: the overall number of evaluations remains the same, but, during the first 50 generations, the fitness functions correspond to formula 3, every candidate

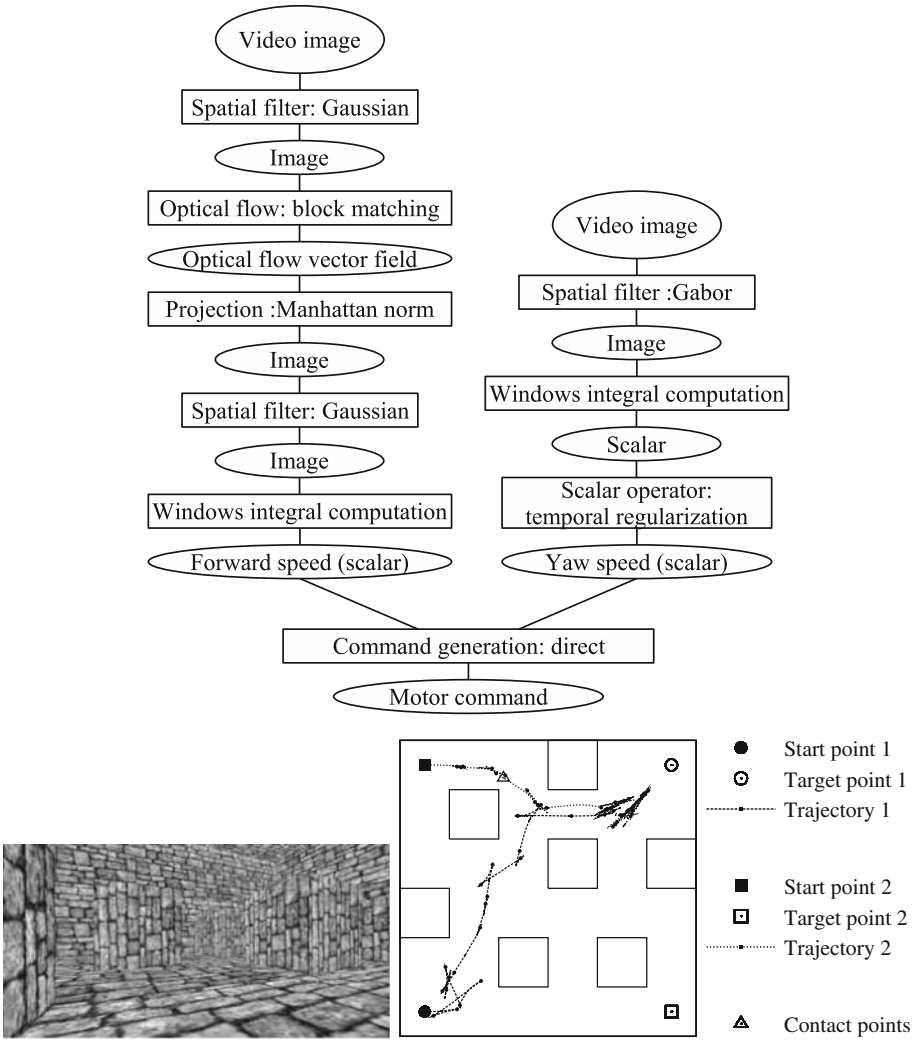


Fig. 7 Example of an algorithm evolved in the textured blocks environment

algorithm using as input the video sequence and the command sequence recorded by a hand guided robot. For the 50 last generations, we use the same goal accession *vs* contact fitness functions as in Sect. 5.1 (i.e. using formula 2).

In Fig. 9, we show the resulting trajectories using this method, compared with the one-phase evolution presented in the previous section, and with two other classical methods used to guide the evolution, that will be presented in further detail. To get an idea of the variability of the different systems, we have realized several experiments (every experiment corresponding to a set of 40,000 evaluations) for each type of evolution. In Fig. 9, we display for each system one individual from the worst experiment, and one individual from the best experiment. The ranking of two different experiments can generally be done in an objective way, as long as the Pareto curves of their last generation do not cross each other, which is often the case. The choice of one individual in the Pareto curve is more subjective, and was selected here by using the "visually best" trajectory.

The controllers obtained in one phase (Fig. 9a), have been presented in the previous section. The chaotic character of the trajectory is particularly visible in the individual issued from the worst experiment (bottom).

Figure 9b corresponds to incremental evolution, which is a classical method used to improve the evolution. The principle is to divide the evolution into several phases corresponding to different environments with increasing complexity [12]. In our case, we have divided the evolution in 3 phases, using the realistic synthesis environment with 1, 2 then 3 bookshelves. What we observe in this case is that the evolution immediately provides adapted individuals in the simplest environment, which is quite trivial. In the intermediate environment, it manages to improve the algorithms performance a little, but in the final environment it generally fails to improve the controller's behavior. The main problem is that in our case it is very difficult to design an evolving environment with increasing complexity. It seems that better results could be obtained by increasing the difficulty in a more gradual way, but this

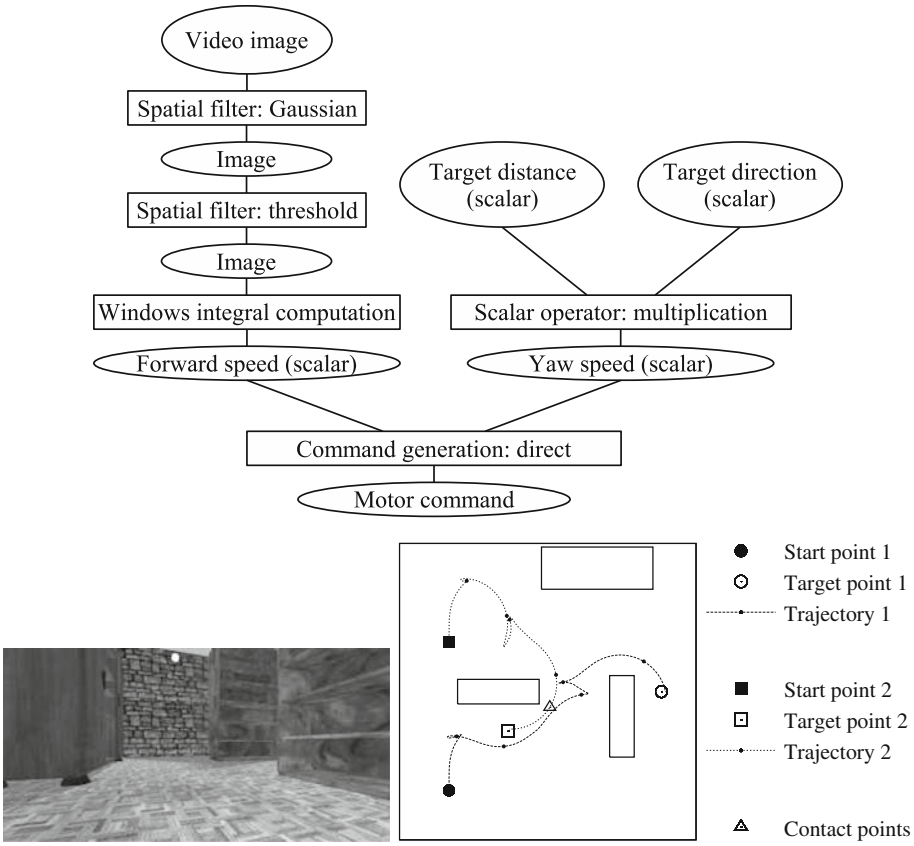


Fig. 8 Example of an algorithm evolved in the bookshelves environment

would require deeper modifications of the simulation protocol.

Another classical method to guide the evolution is seeding (Fig. 9c). Its principle is to introduce in the evolution individuals with acceptable performances. In our case, we simply added in the initial population the individual corresponding to the hand-design algorithm for the specific environment. What we observe in this case is that the evolution manages to improve the performance with respect to the seed, but that the structure (genotype) of the evolved individuals is always very close to that of the seed, which means that this approach seems to drastically limit the innovation within the algorithms.

Finally, the two-phase evolution (Fig. 9d) seems the most stable with respect to the different experiments (Note the little difference between the best and worst experiment). The evolved controllers are efficient and rapid in the evolution environment, and the resulting trajectories are much smoother than in other cases, which is clearly a benefit from an energetic point of view. Another important advantage of two-phase evolution with respect to the other strategies is that it is much easier to implement: recording a video sequence from a hand-guided robot is indeed straightforward, compared to conceiving a gradual complexity increase in the environment, or designing a visual controller by hand.

5.3 Generalization performances

In this section, we discuss the ability of the evolved controllers to generalize obstacle avoidance behavior, i.e. we create an environment whose visual appearance is the same as the evolution environment, but the geometry and location of the obstacles have been changed. Figure 10 shows the results for the different evolution strategies presented in the previous section.

We can see that the generalization performances are poor for all the strategies. The first explanation that can be given for this problem is over-learning: the best evolved individuals have not only learned to avoid obstacles from their appearance or apparent motion, but they also have learned the geometry and location of the different obstacles in the evolution environment. The first idea to develop a more position-independent controller was to increase the number of courses that should be performed by each individual in its evaluation process (Those courses are referred to as “learning courses” from now on). Changing the number of learning courses from two to four resulted in better generalization performances (see Fig. 11), but consequently the evolution time was multiplied by two.

In fact, a more fundamental problem in the structure of the controllers can explain the difficulty in generalizing the obstacle avoidance behavior. In the structure-free

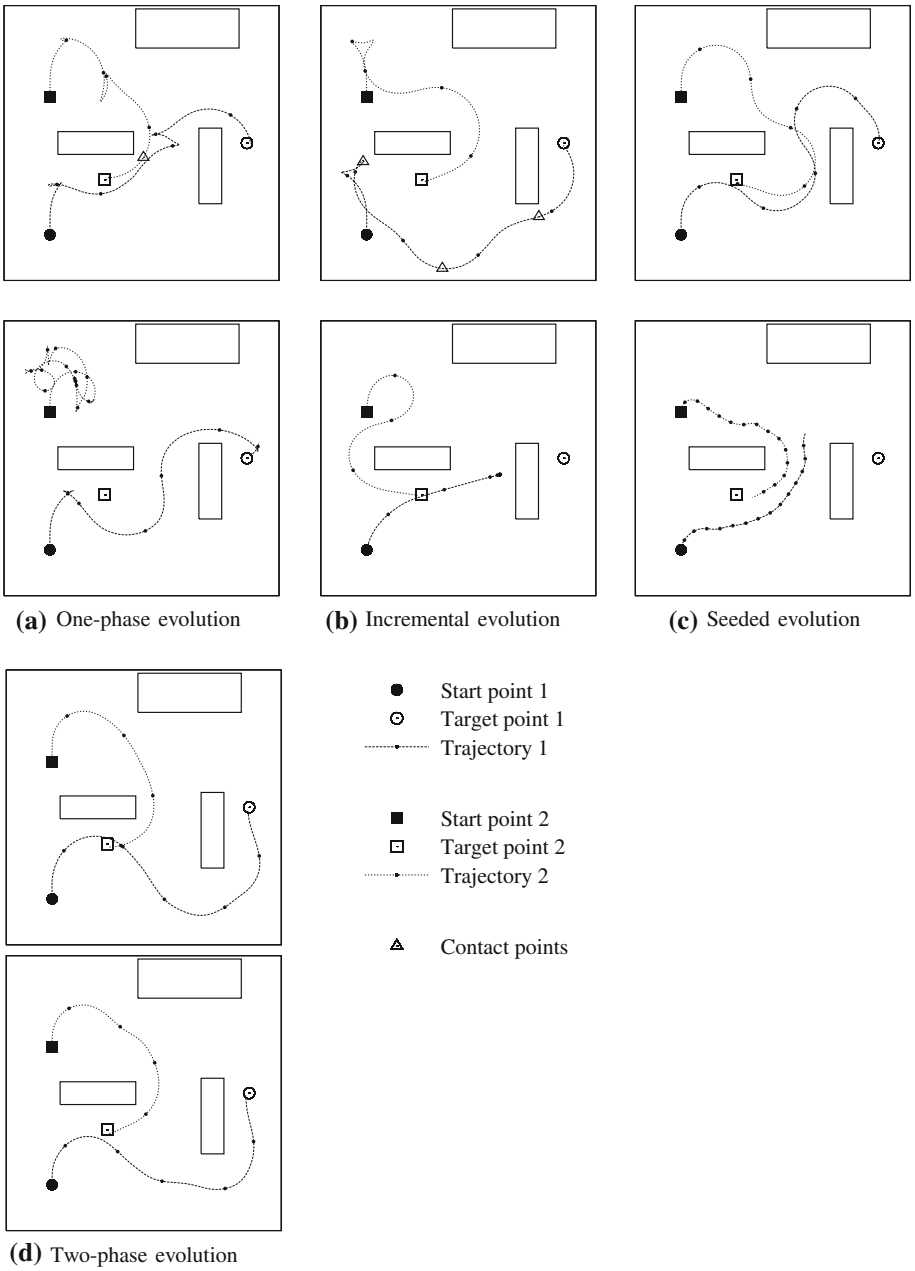


Fig. 9 Comparison of trajectories produced by controllers evolved with different kinds of evolution process. *Top* best experiment. *Bottom* worst experiment. **a** One-phase evolution, **b** incremental evolution, **c** seeded evolution, **d** two-phase evolution

grammar, i.e. without *a priori* in the structure of the controller, it is very difficult for the algorithms to automatically develop a trade-off between obstacle avoidance and target reaching, as the target heading information is just an input of the algorithm, and nothing guarantees a relevant use of this input. This is why we also developed a more restricted grammar, making a more explicit distinction in the controller structure between goal reaching and obstacle avoidance behaviors. The structure restricted grammar has been presented in detail in Sect. 3.3

Figure 12 shows the performances of an individual evolved with structure restricted grammar on four courses different from the learning courses. The generalization performance in this case is much better, due to explicit separation between target reaching and obstacle avoidance within the algorithms structure, thus making it easier for the emergence of a real obstacle detection and avoidance behavior. Another positive impact of the structure restricted grammar is that it usually simplifies the structure of the algorithms, thus lowering the evaluation time of an individual on one course. This allows for a greater number of courses to be evaluated, this further improving the generalization capacity.

Regarding the quality of the results, it is noticeable that the individuals such as the one shown Fig. 12, considered to be the "most evolved" algorithms obtained from

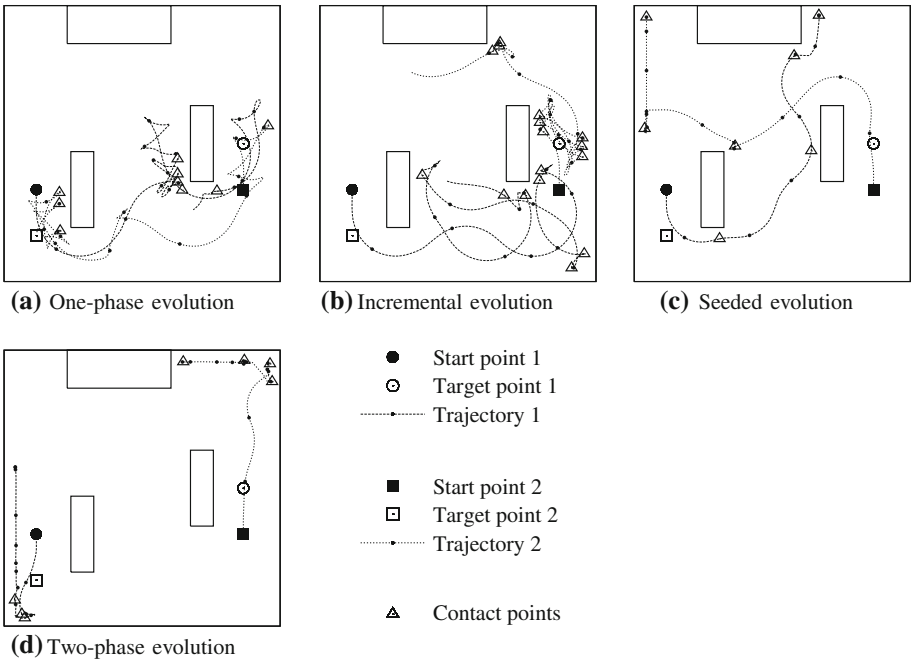


Fig. 10 Performance of the different controllers (designed with a free structure) in a test environment where the obstacles, and start and target points, have been moved

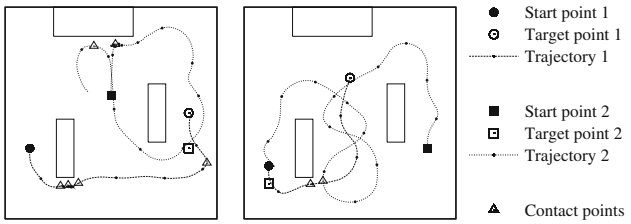


Fig. 11 Generalization performance of a 2-phase evolved controller using a free structure and four learning courses. The two figures show four trajectories obtained on test courses that are all different from the learning courses

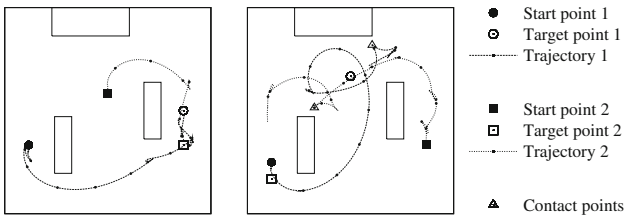


Fig. 12 Generalization performance of a 2-phase evolved controller using a restricted structure and four learning courses. The two figures show four trajectories obtained on test courses that are all different from the learning courses

simulation, may fail the goal reaching task or hit some obstacles. However, it should be pointed out that: (1) We do not consider in this work the goal reaching task as fundamental; it is rather a way to enforce a certain heading and keeping the robot from staying still or turning around. (2) The most generic algorithms are those which globally best perform on different environments, but they also make more errors than the less generic individuals on a specific environment.

An open problem regarding the generalization capacities of the algorithm, is how to select the best individuals in terms of generalization, and what is the best moment to stop the evolution process to avoid over-learning and favor the generalization abilities. Gagné [11] proposed an interesting solution to address this problem. The idea is to evaluate all the individuals from the Pareto front in a validation environment, which is different from the evolution environment. When the performances on the validation environment decrease, it means that we enter the phase of over-learning and the evolution is stopped. Naturally, this biases the evolution results with respect to the validation environment. As such, a third distinct environment (test data) should be used to test the generalization performance.

6 Experiments on the real robot

In this section, we show the results obtained on the real robot platform Pioneer 3DX. Using an off-line evolution method based on the imitation strategy, we present and discuss some results obtained for a corridor centering task, first around the evolution environment, and then in another unseen environment.

6.1 Evolving controllers in real environment

When evolving controllers in a real environment, it is clearly not feasible to perform the same evaluation protocol as in simulation, because that would imply repeating a huge number of experiences with exactly the same conditions.

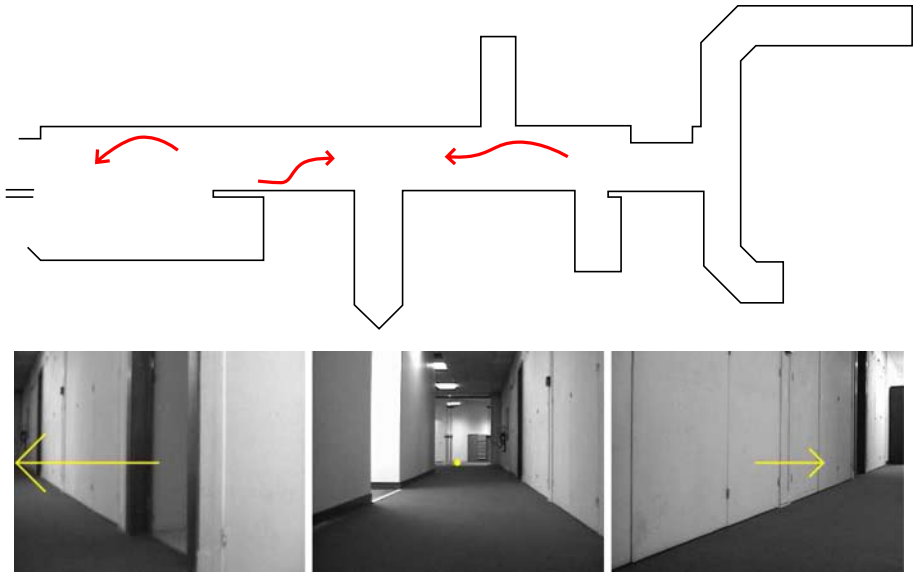


Fig. 13 *Top* examples of trajectories used for the learning base. *Bottom* examples of recorded images and commands

However, the imitation strategy presented above (Sect. 4.2) can be applied on synthesis or real sequences alike, therefore we have implemented an off-line evolutionary algorithm based on the imitation only, using video sequences recorded by the robot Pioneer 3DX guided by hand.

Our first experiments have shown that is was difficult to obtain acceptable obstacle avoidance performances using long sequences involving complex trajectories. On the contrary, very promising results have been obtained relatively quickly, using a large set (around 20) of very short (approx. 2 or 3 s) video sequences. In those sequences, the robot was placed in different positions along the corridor, with its optical axis forming an angle of around 30° with the wall. The robot was then guided manually in such a way that it moves away from the closest wall and centers itself in the corridor. The top of Fig. 13 shows three examples of learned trajectories (red arrows) in a corridor of our laboratory. At the bottom of the figure, three images extracted from these sequences are shown, with the corresponding angular speed command represented as the yellow arrow. The forward speed was approximately constant (30 cm/s) in all the sequences and is not represented in the figure.

The genetic algorithm is then run in one phase, using 100 generations and fitness functions equal to the mean values of Y and F functions of formulas 3 over the 20 sequences. As there is no starting and target points here, the expected behavior when plugging the evolved algorithm in the robot is not a precise displacement, but rather a wandering behavior, allowing to explore the environment with no pre-defined objective.

The progression of the Pareto fronts in the successive generations shows that learning the forward speed is straightforward, which is logical since the forward speed was almost constant in all the command sequences.

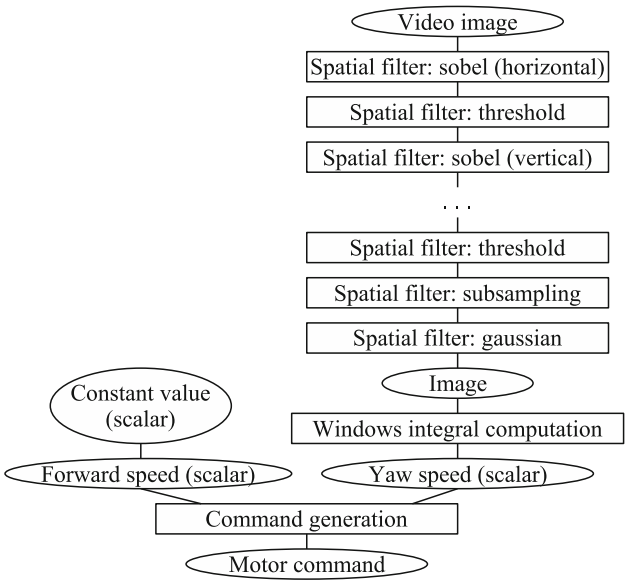


Fig. 14 Example of an evolved algorithm

Consequently, the system acts like mono-objective genetic algorithm, as shown by a significant progression in the yaw speed command error values Y.

Figure 14 shows an example of one of the best evolved algorithms using this method. The filter chain used to generate the angular speed is not shown completely, as it contains more than 29 operators. This complexity is partially due to bloating, but it is also a solution found by the evolution process to overcome a limitation of our system. This algorithm is mostly based on the use of Sobel filters, to detect the edge between the floor and the walls. As the command generation operator needs to extract a scalar from an image, and as the functions that produce scalars from images are based on combination of integral measures on left and right

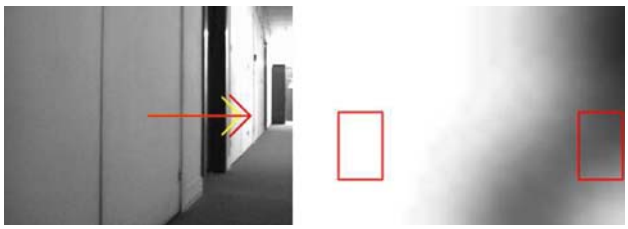


Fig. 15 *Left* resulting command from the evolved algorithm on an image from the learning base. *Right* the same image transformed by the filter chain

region in the images, the long operator sequence of the filter chain is mainly used to enlarge the edge in order to enhance the detection of the floor border. This could have been done more efficiently using a morphological dilatation filter (not in the available primitives), but the fact that the evolution found a way to compensate a limitation of the system is a good indication of its adaptation capabilities.

Figure 15 illustrates the method used by this evolved algorithm to compute the motor command. The filter chain highlights and enlarges the boundary between the floor and the wall, as well as the more contrasted zone at the end of the corridor. In the resulting image, the wall appears completely white and the boundary is darker. This difference is used by the windows integral computation operator to produce a command that drives the robot away from the wall: the resulting command depends on the difference between the mean pixel value of each red window (right image). On the left image, we display a red arrow corresponding to the command issued by the algorithm, and a yellow one which is the command that was recorded when the robot was manually guided.

6.2 Generalization performances

In order to test the robustness and generalization performance of these evolved controllers, we placed the robot at different positions in the corridor and allowed it to be driven by the evolved algorithm. The robot should move to the end of the corridor without hitting the walls. We placed the robot so that the direction it faces and the corridor make an angle of approximately 30°. In this position, the problem is possible to solve without being trivial. We made about ten tests with different starting positions. Each time, the robot managed to reach the end of the corridor except once where it turned into one of the openings in the wall. In one test, it even turned at the end of the corridor to go into the smaller corridor on the right of the map. Figure 16 shows

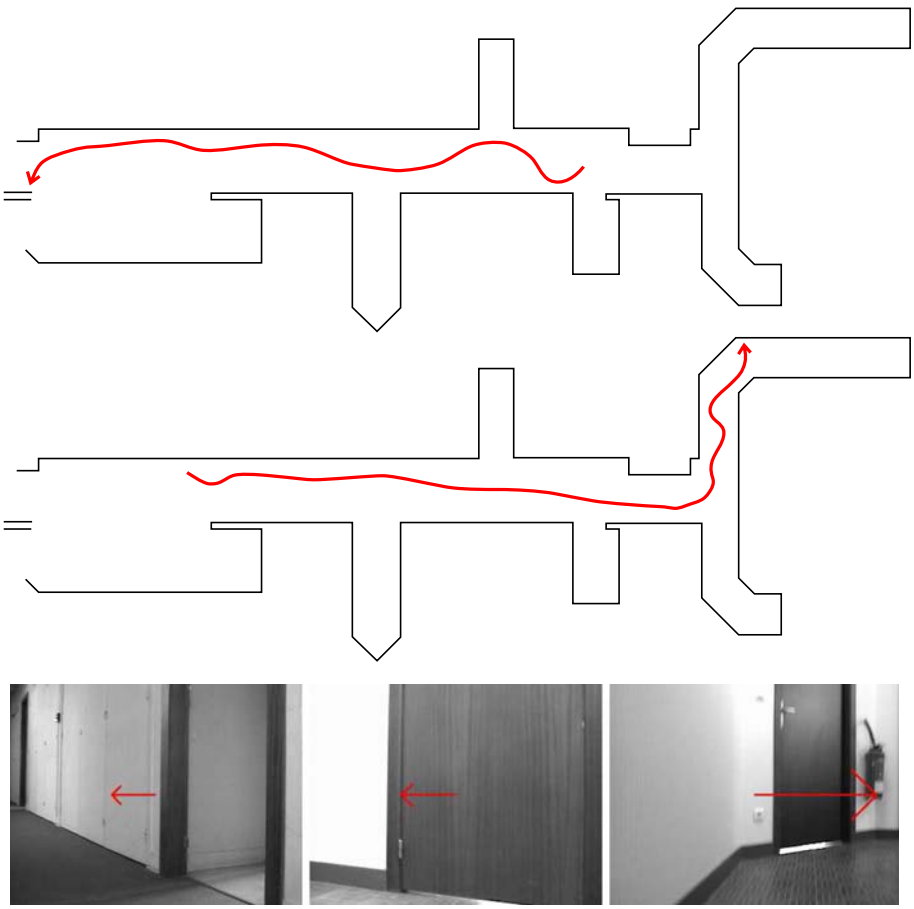


Fig. 16 *Top* trajectories followed by the robot when driven by an evolved algorithm. *Bottom* example images and commands issued by the algorithm in the generalization tests

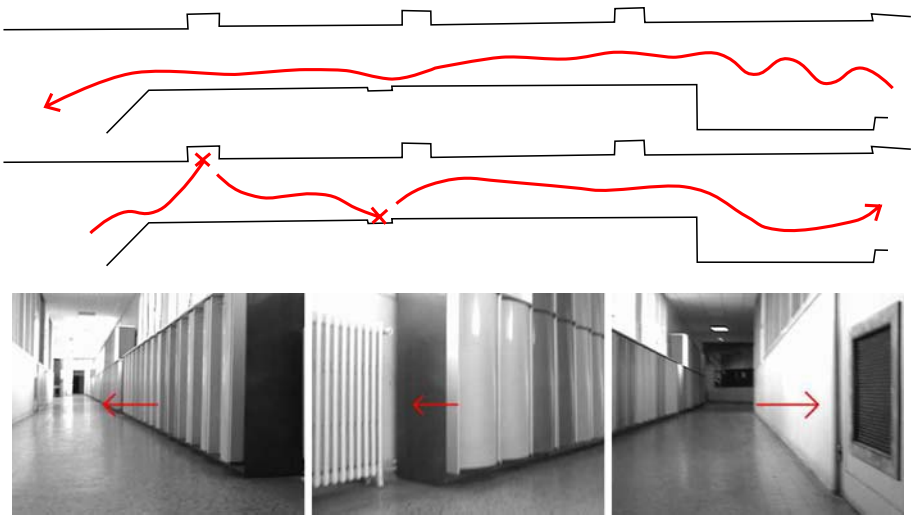


Fig. 17 *Top* trajectories followed by the robot driven by the evolved algorithm in another corridor. *Bottom* images and commands issued by the evolved algorithm in this other corridor

two trajectories, together with some sample images and the corresponding angular speed command.

We also tested this evolved algorithm in another corridor, visually different from the previous one. In one direction the robot reaches the end of the corridor without problem. On the return trip it failed against two obstacles as shown on Fig. 17. Nevertheless this result is encouraging since this corridor is very different from the one that was recorded in the learning base. The second and third images of Fig. 17 show the 2 obstacles that the algorithm failed to avoid.

7 Conclusion

We have presented in this article a genetic programming system to automatically design vision algorithms from a collection of primitives and a set of construction rules. The originality of our approach lies in the fact that the visual task is considered as a whole, with the lowest levels of processing also taking part of the learning process. From a computational point of view, the collection of primitives could be ideally identified with the instruction set(s) of the processor(s) used to compute the algorithm. We have made a more realistic choice from a combinatorial point of view, which consists in constructing the primitive collection from the earliest stages of visual perception (mostly, spatio-temporal filtering). Nonetheless, the choice of the primitives and their complexity level is an interesting open problem that should be addressed in the future.

Although we have concentrated here on the obstacle avoidance problem, we believe that the proposed system can be adapted to automatically design other artificial vision tasks. This can be easily done as long as an objective evaluation can be determined. Typically, such automatic design can be envisaged for: visual categorization, salient features detection, room recognition, etc.

The results we have obtained in simulation have shown that our system was able to provide interpretable controllers adapted to the visual environment. The 2-phase evolution has proven an efficient way to guide the evolution towards more promising solutions. Regarding the generalization capabilities, the restricted structure controllers behaved better than the free structured ones, but at the price of an important *a priori* in the structure of the algorithms. Finding a trade-off between goal accession and obstacle avoidance remains a difficult problem. In that sense, it is possible that the combination between two objectives with very different cognitive levels (obstacle avoidance and goal accession) constitutes a fundamental difficulty.

Unexpectedly, the final results obtained on the real robot were better than those obtained in simulation, possibly due to the removal of the goal accession constraint. The task to achieve was simpler (wandering in a corridor while avoiding the walls), but the results, particularly in generalization were much better. As it turns out, this type of imitation based learning is both easy to implement and promising in real environment, thus we will continue to investigate future solutions in a similar manner.

One limitation of our system is that it is subject to bloating. Several solutions have been proposed to limit bloating in genetic programming. One such example is the inclusion of program size as an independent criterion in a multi-objective evolutionary algorithm, which has been shown to produce efficient and small-sized programs (see [1] for instance). Such adaptation should be envisaged in the future.

Finally the most important perspective of our work is the adaptation to on-line learning. Presently, the proposed system only performs off-line evolution. To reach the global objective of our research, which is providing the mobile robots with more autonomy, we must also integrate a certain level of reactive adaptation. Ebner is currently

working on parallel on-line evaluation of vision algorithms using graphical processing units (GPUs) as a first step toward on-line adaptation [7]. Our method can also be adapted to on-line evolution in several ways. A first level of adaptation can be experimented while keeping the structure of the existing off-line evolution system: a collection of evolved algorithms can be plugged into the robot and a subsequent on-line learning is used to select and parameterize the algorithm which is best adapted to the current context. Another level would be to modify the structure of the algorithms, which is essentially bottom-up, in order to explicitly allow the implementation of top-down mechanisms. Typically, the size and position of the integration windows for the extraction primitive could vary according to the image content; the nature of the spatio-temporal filters could also be made variable according to the context. Such mechanisms are not excluded by our current system, but could be more explicitly taken into account in the structure of the controllers. In this purpose, active vision and attention mechanisms [20, 10, 30] provide frameworks that could help to improve the automatic design of the algorithms and hence will be investigated in our future works.

Acknowledgments This work was supported by the French Armaments Procurement Agency (DGA). The authors would like to thank the anonymous reviewers for their helpful comments, and Mr Toby Low for scientific and English proofreading.

References

1. Bleuler S, Brack M, Thiele L, Zitzler E (2001) Multiobjective genetic programming: reducing bloat using SPEA2. Evolutionary computation, 2001. In: Proceedings of the 2001 Congress on, vol 1, pp 536–543
2. Cagnoni S (2008) Evolutionary computer vision: a taxonomic tutorial. In: Eighth international conference on hybrid intelligent systems. Los Alamitos, CA, pp 1–6. IEEE Computer Society
3. Coombs D, Herman M, Hong TH, Nashman M (1998) Real-time obstacle avoidance using central flow divergence, and peripheral flow. IEEE Trans Rob Autom 14(1):49–59
4. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197
5. Dunn E, Olague G, Lutton E (2006) Parisian camera placement for vision metrology. Pattern Recognit Lett 27(11):1209–1219
6. Ebner M (1998) On the evolution of interest operators using genetic programming. In: Poli R, Langdon WB, Schoenauer M, Fogarty T, Banzhaf W (eds) Late breaking papers at EuroGP'98: the first European workshop on genetic programming. Paris, France, April 1998. The University of Birmingham, UK, pp 6–10
7. Ebner M (2009) A real-time evolutionary object recognition system. In: Genetic programming: proceedings of the 12th European conference EuroGP 2009, Tübingen, Germany, 2009. Springer, Berlin, pp 268–279
8. Ebner M, Zell A (1999) Evolving a task specific image operator. Evolutionary image analysis, signal processing and telecommunications: first European Workshop, EVOIASP, pp 74–89
9. Ebner M, Zell A (2000) Centering behavior with a mobile robot using monocular foveated vision. Rob Auton Syst 32(4):207–218
10. Floreano D, Kato T, Marocco D, Sauser E (2004) Coevolution of active vision and feature selection. Biol Cybern 90(3):218–228
11. Gagné C, Schoenauer M, Parizeau M, Tomassini M (2006) Genetic programming, validation sets, and parsimony pressure. In: Proceedings of EuroGP 2006, vol 3905 of lecture notes in computer science. Springer, pp 109–120
12. Gomez F, Miikkulainen R (1997) Incremental evolution of complex general behavior. Adapt Behav 5(3–4):317–342
13. Horn BKP, Schunck BG (1981) Determining optical flow. Artif Intell 17:185–203
14. Horswill I (1993) Polly: a vision-based artificial agent. In: Proceedings of the eleventh national conference on artificial intelligence (AAAI-93), pp 824–829
15. Lacassagne L, Manzanera A, Denoulet J, Mérigot A (2009) High performance motion detection: some trends toward new embedded architectures for vision systems. J Real-Time Image Process 4(2):127–146
16. Le Cun Y, Muller U, Ben J, Cosatto E, Flepp B (2006) Off-road obstacle avoidance through end-to-end learning. In: Proceedings of the conference on neural information processing systems, pp 739–746, 2006
17. Lorigo LM, Brooks RA, Grimson WEL (1997) Visually-guided obstacle avoidance in unstructured environments. In: Proceedings of the 1997 IEEE/RSJ international conference on intelligent robots and systems, vol 1, pp 373–379
18. Low T, Wyeth G (2007) Learning to avoid indoor obstacles from optical flow. In: Proceedings of the 2007 Australasian conference on robotics and automation. Brisbane, Australia, pp 1–10
19. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of DARPA image understanding Workshop, pp 121–130
20. Marocco D, Floreano D (2002) Active vision and feature selection in evolutionary behavioral systems. From Animals Animat 7:247–255
21. Martin MC (2006) Evolving visual sonar: depth from monocular images. Pattern Recognit Lett 27(11):1174–1180
22. Michels J, Saxena A, Ng AY (2005) High speed obstacle avoidance using monocular vision and reinforcement learning. In: Proceedings of the 22nd international conference on machine learning, pp 593–600
23. Muratet L, Doncieux S, Brière Y, Meyer J.-A (2005) A contribution to vision-based autonomous helicopter flight in urban environments. Rob Auto Syst 50(4):195–209
24. Nelson RC, Aloimonos J (1989) Obstacle avoidance using flow field divergence. IEEE Trans Pattern Anal Mach Intell 11(10):1102–1106
25. Olague G, Puente C (2006) Parisian evolution with honeybees for three-dimensional reconstruction. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, pp 191–198
26. Pauplin O, Louchet J, Lutton E, De La Fortelle A (2005) Evolutionary optimisation for obstacle detection and avoidance in mobile robotics. J Adv Comput Intell Intell Inform 9(6):622–629
27. Perez CB, Olague G (2009) Evolutionary learning of local descriptor operators for object recognition. In: Proceedings of the 11th annual conference on genetic and evolutionary computation, pp 1051–1058
28. Reynolds CW (1994) An evolved, vision-based model of obstacle avoidance behavior. Artificial Life III, pp 327–346
29. Saxena A, Chung SH, Ng AY (2008) 3-D depth reconstruction from a single still image. Int J Comput Vis 76(1):53–69
30. Suzuki M (2007) Enactive robot vision. Ph.D. thesis, École Polytechnique Fédérale de Lausanne (EPFL)

31. Trujillo L, Olague G (2006) Synthesis of interest point detectors through genetic programming. In: Proceedings of the 8th annual conference on genetic and evolutionary computation, pp 887–894
32. Trujillo L, Olague G (2008) Automated design of image operators that detect interest points. Evol Comput 16(4):483–507
33. Ulrich I, Nourbakhsh I (2000) Appearance-based obstacle detection with monocular color vision. In: Proceedings of AAAI conference, pp 866–871
34. Walker J, Garrett S, Wilson M (2003) Evolving controllers for real robots: a survey of the literature. Adapt Behav 11(3):179–203
35. Whigham PA (1995) Grammatically-based genetic programming. In: Proceedings of the workshop on genetic programming: from theory to real-world applications, pp 33–41

J Real-Time Image Proc (2009) 4:127–146
DOI 10.1007/s11554-008-0096-7

ORIGINAL RESEARCH PAPER

High performance motion detection: some trends toward new embedded architectures for vision systems

Lionel Lacassagne · Antoine Manzanera · Julien Denoulet · Alain Mérigot

Received: 30 June 2007 / Accepted: 2 September 2008 / Published online: 14 October 2008

Abstract The goal of this article is to compare some optimised implementations on current high performance platforms in order to highlight architectural trends in the field of embedded architectures and to get an estimation of what should be the components of a *next generation* vision system. We present some implementations of robust motion detection algorithms on three architectures: a general purpose RISC processor—the PowerPC G4—a parallel artificial retina dedicated to low level image processing—*Pvlsar34*—and the Associative Mesh, a specialized architecture based on associative net. To handle the different aspects and constraints of embedded systems, execution time and power consumption of these architectures are compared.

L. Lacassagne (✉) · A. Mérigot
Institut d'Electronique Fondamentale (IEF/AXIS),
Université Paris Sud, Orsay, France
e-mail: lionel.lacassagne@u-psud.fr;
lionel.lacassagne@ief.u-psud.fr

A. Mérigot
e-mail: alain.merigot@u-psud.fr

A. Manzanera
Laboratoire d'Electronique et d'Informatique,
Ecole Nationale Supérieure de Techniques avancées (ENSTA),
Paris, France
e-mail: antoine.manzanera@ensta.fr

J. Denoulet
Laboratoire des Instruments et Systèmes d'Ile de France
(LISIF/SYEL), Université Pierre et Marie Curie, Paris, France
e-mail: julien.denoulet@upmc.fr

Keywords Embedded system · Vision-SoC · RISC · SIMD · SWAR · Parallel architecture · Programmable artificial retina · Associative nets model · Vision · Image processing · Motion detection · High performance computing

1 Introduction

For more than 30 years, Moore's law had ruled the performance and the development of computers, speed and clock frequency were the races to win. This trend slowly drifted as the processing power of computers reached a seemingly stable value. Other constraints (static current consumption, leakage, less MIPS per gates and less MIPS per Watts) of current technology—90 and 65 nm—gave researchers an impulse to look for innovative directions to improve efficiency and performance of their architectures. Current challenge is to tackle power consumption to increase systems autonomy. Such technology, like IP core within embedded systems, make the processor frequency adaptable and lead to a finely optimised energy consumption. As image processing and computer vision are very CPU demanding, we focus on the impact of the architecture for a frequently used class of algorithms: the motion detection algorithms.

Three architectural paradigms are compared:

SIMD within a register (SWAR) the impact of the SIMD multimedia extension inside RISC processors, to enhance performance;
Programmable artificial retina one elementary processor per pixel for cellular massively parallel computation and low power consumption;
Associative net impact of reconfigurable graph/net between processors for local and global computations

and also the impact of asynchronous processors on power consumption.

We focus on the advantages and limitations of these architectures through a set of benchmarks. We also show how to modify the algorithms to take advantage of each architecture's specificities. We provide different performance indexes like speed, energy required and a *down-clocking* frequency to enforce real-time execution. Such indexes provide insight on future trends in computer architecture for embedded systems.

The paper is organized as follow. Section 2 introduces a set of motion detection algorithms: frame difference, Markovian relaxation, Sigma–Delta algorithm and post-processing morphological operators. Section 3 presents the three architectures: the PowerPC G4, *Pvlsar34* (a 200 × 200 Programmable Artificial Retina) and the Associative Mesh (an asynchronous net with SIMD functional units). This section also provides details about how algorithms are optimised in regard to the targeted architectures. Section 4 deals with benchmarking: benchmark of the different algorithms in term of speed and in term of power consumption. To conclude, a synthesis of two extensive benchmarks is provided.

2 Motion detection algorithms

As the number of places observed by cameras is constantly increasing, a natural trend is to eliminate the human interaction within the video monitoring systems and to design fully automatic video surveillance devices. Although the relative importance of the low level image processing may vary from one system to the other, the computational weight of the low level operators is generally high, because they involve a great amount of data. Thus, the ability of video surveillance systems to detect a relevant event (intrusion, riot, distress, etc.) is strongly related to the performance of some crucial image processing functions.

Such fundamental processing step is the motion detection, whose purpose is to partition the pixels of every frame of the image sequence into two classes: the *background*, corresponding to pixels belonging to the static scene (label 0) and the *foreground*, corresponding to pixels belonging to a moving object (label 1). A motion detection algorithm must discriminate the moving objects from the background as accurately as possible, without being too sensitive to the sizes and velocities of the objects, or to the changing conditions of the static scene. For long autonomy and discretion purposes, the system must not consume too much computational resources (energy and circuit area). The motion detection is usually the most computationally demanding function of a video surveillance system. How the algorithm is actually computed and on which architecture, then become crucial questions.

Three algorithm/architecture pairs will be considered here. In order to compare those very different architectures, we will consider different versions of motion detection algorithms with similar quality but relying on different computational models, some of them being more adapted to one architecture than the other.

The motion detection algorithm can be separated into two parts: time-differentiation and spatiotemporal regularization.

The purpose of the time-differentiation part is to provide, for every pixel x and every time index t: a measure of the temporal variation (the observation) is denoted as O_t and an initial value of the motion binary label is denoted as $\hat{E}_t$. The "frame difference" option is classical and fairly obvious: the temporal derivative is approximated by a difference between consecutive frames, whose absolute value is used as a single motion map (observation) $O_t(x) = |I_t(x) - I_{t-1}(x)|$ and the initial value of the motion label $\hat{E}_t$ is obtained by thresholding O_t. The "Sigma–Delta" option—detailed in Sect. 2.1—is a recent algorithm [31], based on nonlinear estimation of temporal statistics of every pixel.

The spatiotemporal regularization part aims at exploiting the correlations between neighboring pixels in the motion measures in order to improve the localization of the moving objects. Two main options are considered here (1) morphological filtering, detailed in Sect. 2.2 and (2) Markovian relaxation, detailed in Sect. 2.3.

So, the "Sigma–Delta" can be seen as a pre-processing step for the Markovian regularization or as the main algorithm when followed by a morphological post-processing.

2.1 Sigma–Delta estimation

The principle of the ΣΔ algorithm is to estimate two parameters M_t and V_t of the temporal signal I_t within every pixel using ΣΔ modulations. It is composed of four steps: (1) update the current background image M_t with a ΣΔ filter, (2) compute the frame difference between M_t and I_t, (3) update the time-variance image V_t from the difference O_t using a ΣΔ filter and (4) estimate the initial motion label $\hat{E}_t$ by comparing the current difference O_t and time-variance V_t.

for each pixel x:
if $M_t(x) < I_t(x), M_t(x) = M_{t-1}(x) + 1$
if $M_t(x) > I_t(x), M_t(x) = M_{t-1}(x) - 1$
otherwise $M_t(x) = M_{t-1}(x)$

step1: update M_t

for each pixel x:
$O_t(x) = |M_t(x) - I_t(x)|$

step2: compute O_t

for each pixel x such that $O_t(x) \neq 0$:
if $V_t(x) < N \times O_t(x), V_t(x) = V_{t-1}(x) + 1$
if $V_t(x) > N \times O_t(x), V_t(x) = V_{t-1}(x) - 1$
otherwise $V_t(x) = V_{t-1}(x)$

step3: update V_t

for each pixel x:
if $O_t(x) < V_t(x)$
then $\hat{E}_t = 0$
else $\hat{E}_t = 1$

step4: estimate $\hat{E}_t$

Apparently, the only parameter is the amplification factor N of the difference (typical values of N are in 2–4). The dimension of N is the number of standard deviation used in the initialization of the motion label. In fact, the updating frequency, which has the dimension of number of gray level per second, can also be adapted. This is a way of customizing the $\Sigma\Delta$ estimation to different kinds of motion and image noise [32].

2.2 Morphological filtering

2.2.1 Alternate sequential filters (ASF)

The first option of morphological filtering is to perform a sequence of dilatations and erosions using a set of structuring elements of increasing size, such as a sequence of discrete balls $(B_n)_n$, $B_n = \{z \in \mathbb{Z}^2; d(z, O) \leq n\}$, with O the origin of the discrete plane $\mathbb{Z}^2$ and d a discrete distance of $\mathbb{Z}^2$. Table 1 shows the definitions of such operators.

$\wedge$ and $\vee$, respectively, represent the logical AND and OR. By convention, Ξ_0 and Θ_0 both correspond to the identity function. In this option, the spatiotemporal regularization is performed by applying an alternated sequential filter of certain size to the output of the temporal detection. Typically, $E_t = \Xi_2(\hat{E}_t)$.

Table 1 Morphological operators

$\varepsilon_B(I)(x) = \wedge_{b\in B} I(x-b)$	$\gamma_B = \delta_B \circ \varepsilon_B$
$\delta_B(I)(x) = \vee_{b\in B} I(x+b)$	$\varphi_B = \varepsilon_B \circ \delta_B$
(a)	(b)
$\xi_B = \varphi_B \circ \gamma_B$	$\Xi_n = \xi_{B_n} \circ \Xi_{n-1}$
$\theta_B = \gamma_B \circ \varphi_B$	$\Theta_n = \theta_{B_n} \circ \Theta_{n-1}$
(c)	(d)

a Erosion and dilatation, *b* opening and closing, *c* alternate filters and *d* alternate sequential filters

2.2.2 Density operators

In a similar fashion, density operators are defined using a structuring element B, except that the binary response is based on counting-thresholding instead of AND−OR combinations :

$$\mathcal{D}_B(I)(x) = 1 \Longleftrightarrow |\{b; I(x-b) = 1\}| \geq \theta$$

where $|S|$ represents the cardinality of set S and θ a threshold representing a required density of 1s. In this case, the final label is computed using a density operator with a ball of radius n : $E_t = \mathcal{D}_{B_n}(\hat{E}_t)$. Typically n equals 1, 2 or 3 and usually $\theta = \lceil |B_n|/2 \rceil$ (majority voting).

2.2.3 Geodesic reconstruction

Defined from a binary reference image R, the geodesic reconstruction $\mathrm{Rec}^R(I)$ of image I within reference R is the relaxation of the geodesic dilatation of I within R: $\delta_B^R(I) = \delta_B(I) \wedge R$. Assuming that the structuring element B—basically a discrete ball of radius 1—is defining the topology, $\mathrm{Rec}^R(I)$ corresponds to the connected components of R having a non-empty intersection with I.

In this option, the final label E_t is computed as follows: small connected components elimination using an opening by reconstruction with a ball of radius n: $\tilde{E}_t = \mathrm{Rec}^{\hat{E}_t}(\gamma_{B_n}(\hat{E}_t))$, then temporal confirmation by computing another reconstruction: $E_t = \mathrm{Rec}^{\tilde{E}_t}(\tilde{E}_{t-1})$.

The final motion label E_t then corresponds to the objects (connected components) bigger than B_n that appear on two consecutive frames.

2.3 Markovian relaxation

Markov random field based algorithms (MRF) have asserted themselves in a lot of image processing areas for regularizing ill-posed problems. Albeit robust, their well-known drawback is their CPU consumption due to a large amount of computations, which led researchers to look for solution to speedup its execution time, using parallel machines or dedicated architectures [1, 2, 9, 21, 30].

We follow the MRF model introduced for motion detection purposes proposed by the LIS-Grenoble laboratory [7] and derived from the IRISA model [4, 28]. This model is based on the estimation of a binary (background/foreground) motion field e given an *observation field* o, by maximizing a Bayesian *maximum a posteriori* criterion, i.e. given a realization of the observation field $o = y$, finding the realization x of the motion label field e that maximizes the conditional probability $P(e = x/o = y)$. Assuming that e is an MRF linked

to o with a probabilistic relation, this corresponds to finding the motion field e that minimizes the global *energy* function defined over the set of pixels $\mathbb{S}$ as follows:

$$U = \sum_{s\in\mathbb{S}} [U_m(e(s)) + U_a(e(s), o(s))],$$

$$\text{with } U_m(e(s)) = \sum_{r\in\mathcal{V}(s)} V_e(e(s), e(r)),$$

$$\text{and } U_a(e(s), o(s)) = \frac{1}{2\sigma^2}[o(s) - \Psi(e(s))]^2.$$

$U_m(e(s))$ is called *model energy* and is designed to provide spatiotemporal regularity in the motion field. It is based on the Markovian modeling of e as a Gibbs field, where $\mathcal{V}$ is the set of neighbors of the pixel s and the potential functions $V_e(e(s),e(r))$:

$$V(e_s, e_r) = \begin{cases} -\beta_{sr} & \text{if } e_s = e_r \\ +\beta_{sr} & \text{if } e_s \neq e_r \end{cases}$$

The β_{sr} are positive constants whose values depend on the nature of the neighborhood. We use a uniform 10-connected spatiotemporal topology (see Fig. 1), with 3 different values $\beta_s = 20$ for the 8 spatial neighbors, $\beta_p = 10$ for the past neighbor and $\beta_f = 30$ for the future neighbor. Experimental tests demonstrate that these parameters do not have to be tuned according to the image sequence.

$U_a(e(s), o(s))$ is called *fitness energy* and is designed to ensure a certain level of attachment to the input data, i.e. the observation o. This term comes from the conditional probability of the observation field o, with respect to the motion field e, assuming that $o(s) = \Psi(e(s)) + n(0, \sigma^2)$, with $n(0, \sigma^2)$ a centered Gaussian noise of variance σ^2, $\Psi(e(s)) = 0$ if $e(s)$ has the background value and $\Psi(e(s)) = \alpha$ if $e(s)$ has the foreground value. The α parameter can be set to usual value 20, or updated on the fly, as the average value of the moving observations; σ^2, the variance of the moving observation, is computed for every frame.

The minimization of the global energy U is realized by the deterministic relaxation called iterated conditional mode (ICM): all the pixels are sequentially updated and each pixel s is given the label $e(s)$ corresponding to the smallest local energy $U_m(e(s)) + U_a(e(s), o(s))$. Usually, instead of a true relaxation, a limited number of scans is performed (typically 4). The advantage is that the computation time becomes independent of the data, in particular of the initial value of the motion field e.

But the drawback is that the quality of the final labeling is very dependent on that initial value, which must be close enough to the final solution. In our algorithm, we use the output of the $\Sigma\Delta$ temporal differentiation $\hat{E}_t$, which as proved a good choice of initial guess [31]. The observation field o corresponds to the difference map O_t.

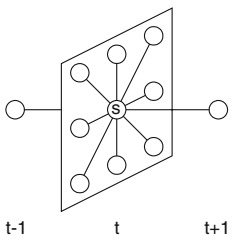


Fig. 1 Spatiotemporal topology

3 Architectures and their optimizations

In order to perform a fair comparison of these architectures, the algorithm must be optimised for each one. This section describes how the different algorithms are implemented on the three architectures, the impact of the architecture on the algorithm and how the algorithms' structure and the architectures themselves should be modified to obtain optimised implementation.

3.1 PowerPC

The powerPC used is a PPC 7447 running at 1 GHz. It has a 32 KB L1 cache, a 512 KB L2 cache and its power consumption is 10 W. Its specifications are detailed in Tables 5 and 6. From a functional point of view (Fig. 2), it has one Load/Store Unit, one ALU, one FPU and a superscalar SWAR unit: Altivec. Altivec is a multimedia instruction set extension which has been designed to efficiently accelerate image and signal processing [15] applications. Altivec is composed of four 128-bit SWAR units (following the Freescale vocabulary):

- Vector Permute Unit, which handles the instructions to rearrange data within SWAR registers (permutation, selection),
- Vector Simple Integer Unit, which handles all the fast and simple integer instructions,
- Vector Complex Integer Unit, which handles the slower and complex instruction like multiply, multiply-add,
- Vector Floating Point Unit, that handles all the SWAR floating-point instructions.

Main advantages of Altivec are:

- Each of the four vector units are pipelined,
- Two instructions from the four units can be issued per cycle without constraint on which unit is used.

To optimize a given code for a SWAR RISC processor, we have to address the following points:

- *bandwidth problem* by optimizing loads and data reuse, avoiding data reload and optimizing cache locality [36],

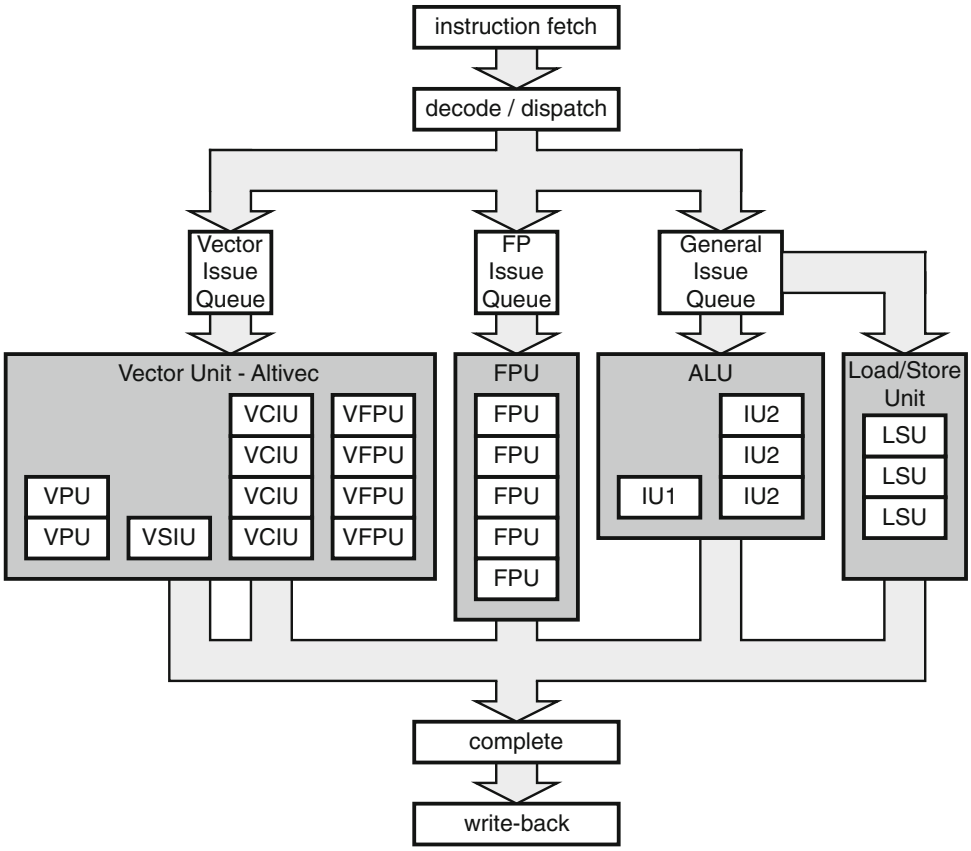


Fig. 2 PowerPC G4 pipeline

- *pipeline stalls due to un-predictable test instructions* by trying to remove tests from MRF energy computation,
- pipeline throughput with loop transformations.

In this section, we focus on SWAR optimization (also known as SIMDization) and algorithm transformations. Details about loop transformation techniques and above optimizations are given in [14].

To speedup the computation of the model energy U_m, we have to transform the equation of the potential function V, that comes from the Ising model (the spin associated with a particle). Usually spin-up and spin-down are coded with $+1$ or -1. In our case, rather than labeling the state of a site -1, $+1$ for *background* or *motion* pixel, we use the binary code 0, 1. Let p_1, s_1 and f_1 the number of sites, connected to e_s, with a value 1, in the past, present and future images ($p_1 \in \{0, 1\}$, $s_1 \in \{0,\ldots, 8\}$, $f_1 \in \{0, 1\}$). Then the energy model can be computed without any test or comparison:

$$u_{m1} = (8 - 2s_1)\beta_s + (1 - 2p_1)\beta_p + (1 - 2f_1)\beta_f,$$
$$u_{m0} = -u_{m1}$$

where u_{m1} is the energy associated to a central site at 1. The fitness energy can also be computed without taking into account the state of the site:

$$u_{a0} = \frac{1}{2\sigma^2}[o(s)]^2, \quad u_{a1} = \frac{1}{2\sigma^2}[o(s) - \alpha]^2$$

If $u_{m1} + u_{a1} < u_{m0} + u_{a0}$, the state is set to 1 otherwise it is set to 0. The change is performed whatever the previous state was. The same approach is used to remove tests from the $\Sigma\Delta$ algorithm which is actually very hard to optimize since only a few additions and comparisons are done compared to the amount of memory accesses, as described in [14]. Note that this test can be optimised by rewriting $2u_{m1} < u_{a0} - u_{a1}$:

$$u_{m1} < \delta u_a, \quad \delta u_a = \frac{u_{a0} - u_{a1}}{2} = \frac{\alpha(2o - \alpha)}{4\sigma^2}$$

3.1.1 Density and opening

We have implemented three kernel size for these operators: 3×3, for regular use, 5×5 and 7×7 to estimate the adequacy of the considered architectures to those well known kernel operators. The cardinal computation of the ball of diameter k, i.e. the summation of pixel value over the $k \times k$ kernel (Fig. 3) requires k^2 `LOAD`, 1 `STORE` and k^2-1 mathematical operations (typically `ADD` but could be `AND` or `OR` Boolean operator for erosion and dilatation). Taking into account that $k \times k$ kernels overlap from one

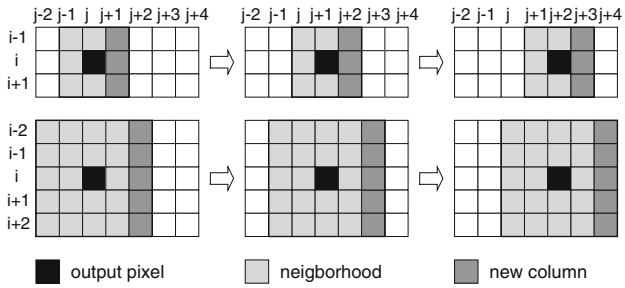


Fig. 3 Operators with neighborhood: overlapping for 3 × 3 and 5 × 5 kernels

iteration to another one, such summation can be optimised by splitting this summation into k columns summation. The cardinal is then the sum of these k columns. For the next iteration, only one new column should be computed and added to the previous one. The new complexity is k `LOAD`, 1 `STORE` and only $2(k-1)$ `ADD` (see Table 2).

For SWAR computation, the same optimizations can be applied except that 16 pixels are computed in parallel instead of only one SWAR results are given in Table 2 (for 16 pixels). SWAR implementation requires the construction of unaligned vector registers to compute the partial sums. This is quickly done thanks to the dedicated Altivec instruction `vec_sld` (Fig. 4). For example, given three

Table 2 Instructions per point, scalar and SWAR version

Instruction	Without split	With split
Scalar `LOAD`	k^2	k
Scalar `STORE`	1	1
Scalar mathematical *Op*	k^2-1	$2(k-1)$
SWAR `LOAD`	$3k$	k
SWAR `STORE`	1	1
SWAR mathematical *Op*	k^2-1	$2(k-1)$

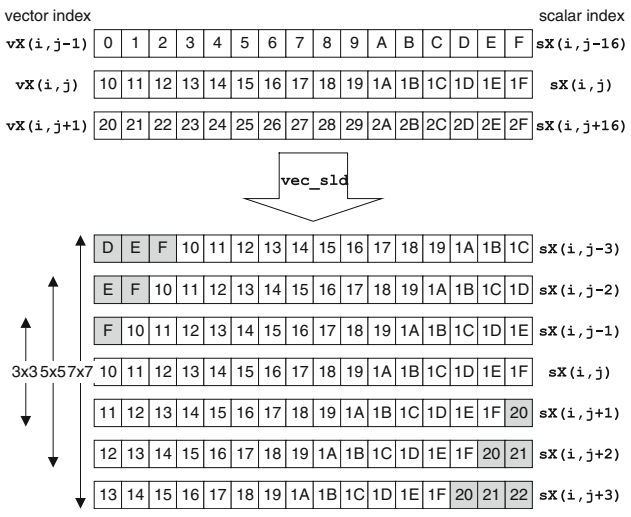


Fig. 4 SWAR convolution reuse

aligned vector registers (Fig. 4) $vX(i, j-1)$, $vX(i, j)$, $vX(i, j+1)$, the $k = 7$ unaligned vector registers sX are constructed and used to compute the sum of line i, (same computations are performed for lines $i-3$, $i-2$,…, $i+3$). The interesting fact is that, up to a value of $n = 16$, pixels $x(i, j-n)$ and $x(i, j+n)$ are in the left and right vector registers of $x(i, j)$ (Fig. 4). So, for kernels size up to $k \times k = 33 \times 33$, only $3 \times k$ SWAR `LOAD` are required.

3.2 Programmable artificial retina

The purpose of the programmable artificial retina (PAR) project is to develop *versatile*, *real-time*, *compact* and *low-power* vision systems. In the vision machines today, most of the resource consumption is due to the transfer of huge amounts of data throughout the different parts of the system. The data flow thus becomes the main source of cost in time, circuit area and/or energy. In PAR-based vision systems, the data transfers are limited to the minimum, by processing the information where it is acquired, i.e. within every pixel of the sensor and by performing an information reduction in the focal plane in order to extract only a few descriptors representing a very small data flow that can be processed by a low-power external processor.

The PAR concept originates from the neighborhood combinatorial processing (NCP) retinas [42] which were SIMD Boolean machines. The near sensor image processing (NSIP) concept [20] then allowed to process gray level images. Now, the deep sub-micron level of CMOS technology allows to put more and more powerful processing circuitry aside the photo receptors while preserving good acquisition performance and resolution [26, 40]. The circuit used in our work was designed by Bernard at ENSTA and fabricated using 0.35 µm technology: *Pvlsar34* is a 200 × 200 retina, with an elementary digital processor and 48 bits of memory within every pixel. The architecture of *Pvlsar34* is presented in Sect. 3.2.1, and the retinal algorithms in Sect. 3.2.2 Now, whereas this architecture has proved well adapted to low and medium level image processing [32, 35], the interest of asynchronism has been identified to enhance the processing power of the PARs by providing them with a higher (i.e. regional) level of computation [18, 23, 24]. This is discussed in Sect. 3.2.3.

3.2.1 Retina and cortex architecture

The detection algorithm presented in this paper was actually implemented on the architecture presented in Fig. 5. The PAR *Pvlsar34* is a CMOS sensor and a parallel machine at the same time. It is a grid of 200 × 200 pixels/processors connected by a regular 4-neighbors rectangular mesh. The processors execute synchronously, on their local data, a sequence of instructions sent by the controller,

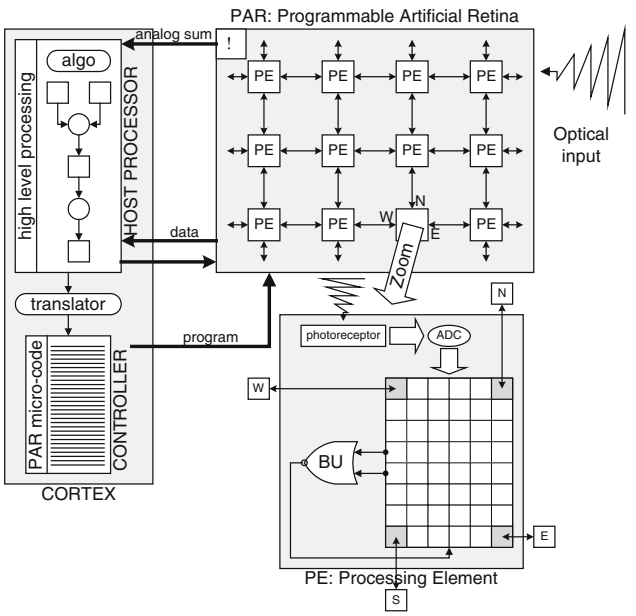


Fig. 5 Architecture of the system composed of the PAR and cortex, with focus on one elementary processor

which is the NIOS processor IP core of the Excalibur FPGA chip. The host processor or cortex is the ARM processor hardware core of the Excalibur. It can exchange data with the PAR, modify the program sent by the NIOS to the PAR and is in charge of the higher levels of computation (i.e. non-image processing) of the vision task.

Every pixel/processor of the PAR is composed of:

- one photo-receptor;
- one analog to digital converter;
- 48 bits of digital memory;
- one Boolean unit (BU), which can read some bits of the digital memory, compute a Boolean operation and write its output on one bit of the digital memory.

The actual instruction set of *Pvlsar34* is composed of only five instructions. If reg_1 and reg_2 are two binary registers of the digital memory:

one: The BU takes the logical value `1`.
rd(reg_1): The BU takes the logical value of register reg_1.
ror(reg_1, reg_2): The BU takes the logical value of the binary `OR` between the two binary registers reg_1 and reg_2.
wd(reg_1): The BU writes down its logical value on register reg_1.
wc(reg_1): The BU writes down the complementary of its logical value on register reg_1.

Boolean algebra shows that this instruction set is sufficient to compute any Boolean function. Now, for readability purposes, we shall use in the presentation of the primitives a generic Boolean instruction set, made of the instructions of the form $y = OP(x_1, x_2)$, where x_1, x_2, y are 3 bits (not necessarily distinct) of the digital memory and *OP* is any binary Boolean function (e.g. `AND`, `XOR`, `ADD` `NOT`, etc). Note that every instruction is computed in a massively parallel SIMD mode, the operators are then performed simultaneously on all pixels.

Every pixel of the PAR shares 1 bit of its memory with each one of its four closest neighbors, allowing spatial interactions and image translations. Regarding data extraction, there are two ways to output information from the PAR:

- by translating the image and reading the output on the edge of the grid, to get the exact content of one or more bit planes of the digital memory.
- by using the Analog Global Summer, which provides in constant time an approximate measure of the number of 1s in any bit plane of the digital memory.

Although simple, this last feature is important as it provides efficiently global measures that are very useful to get spatial statistics or to detect the convergence of relaxation algorithms.

3.2.2 *Cellular synchronous algorithms*

From the architecture presented above, it turns out that the retinal algorithmics at the present time is essentially a cellular SIMD parallelism. A retinal program is a sequence of binary Boolean instructions. All the pixels/processors perform the same instruction at the same time on their own data, part of which can be taken from one of their closest neighbors. The extreme level of granularity and the small amount of digital memory are the main characteristics of the retinal algorithmic. The algorithm designer is imposed a constant effort of logic minimization, in order to find the Boolean expression of its algorithm that minimizes the number of elementary instructions (related to the computation time) while fitting in the available memory (just like a hardware designer will make a circuit trying to minimize the critical paths and using the minimal amount of logical gates).

Naturally, the memory limitations also affect the data representation that can be used by the algorithm. In the case of motion processing which concerns this paper, this means that the memory used to represent the past history of every pixel must be rigorously controlled. Typically, we shall not keep histograms nor a large set of past values within every pixel, but rather a limited number of temporal statistics, computed recursively.

Despite these constraints, the retinal computation model offers some very attractive features. In particular, the fusion of acquisition and processing functions allows a close adaptation to the lighting conditions and to the scene dynamics. More precisely, the analog to digital conversion (ADC) performed at the output of the photo-receptor is done by a multiple reading which provides *N* binary images

(level sets). As the ADC itself is fully programmable, it is possible to perform a constant feedback from the local and global computations to the acquisition, thus providing sophisticated adaptation to lighting conditions.

Once the gray levels of the image are coded within every pixel of the PAR, the retinal program applies a sequence of arithmetic and logic operations that are completely written in software, at the bit level. We now present such program in the particular case of the motion detection.

The ΣΔ change detection algorithm relies on very simple primitives: comparison, difference and elementary increment and decrement. Furthermore, it is based on non-linear computations which does not involve neither truncation nor dynamics increasing. It is thus well adapted to the minimal instruction set and the small memory of the PAR elementary processors. The implementation on *Pvlsar34* was performed using the four primitives presented in Table 3. To avoid confusing notation, I_t is noted here X_t.

Table 3(3) represents the strict comparison primitive between X_t and M_t; e and f are the two bits of result, indicating whether $M_t < X_t$ and whether $X_t < M_t$, respectively. These indicators are used in the ΣΔ algorithm, to update the statistics, by decrementing e (Table 3(4)) and incrementing f (Table 3(5)). Table 3(1) shows the computation of the difference O coded on n bits $\{o_0,\ldots, o_{n-1}\}$, between the current mean M_t and the current sample X_t. At the end of the computation, O_t is coded in classical two-complement, with c the sign bit. For the second-order statistics (ΣΔ variance V_t), it is necessary to compute the absolute value of the difference O_t (Table 3(2)).

Table 3 The PAR algorithmic primitives used in the ΣΔ motion detection

```
c = 1;
for i=0 to (n-1) {
  a = m_i ⊕ x̄_i;
  o_i = a ⊕ c;
  a = a ∧ c;
  c = m_i ∧ x̄_i;
  c = c ∨ a;
}
```

(1) Signed difference

```
b = c ∧ o_0;
for i=1 to (n-1) {
  o_i = o_i ⊕ b;
  a = c ∧ o_i;
  b = b ∨ a;
}
```

(2) Absolute value

```
c = m̄_0 ∧ e;
m_0 = m_0 ⊕ e;
for i=1 to (n-1) {
  m_i = m_i ⊕ c;
  c = c ∧ m_i;
}
```

(4) Decrement

```
d = 0; e = 0; f = 0;
for i=(n-1) down to 0 {
  a = m_i ∧ x̄_i;
  b = x_i ∧ m̄_i;
  g = a ∧ d̄;
  e = e ∨ g;
  g = b ∧ d̄;
  f = f ∨ g;
  a = a ∨ b;
  d = d ∨ a;
}
```

(3) Strict comparison

```
c = m_0 ∧ f;
m_0 = m_0 ⊕ f;
for i=1 to (n-1) {
  m_i = m_i ⊕ c;
  c = c ∧ m̄_i;
}
```

(5) Increment

The above primitives allow to implement the whole temporal (pixel-wise) part of the algorithm. On *Pvlsar34*, it was completed by using binary morphology as spatial regularization. An alternate sequential filter was applied on the temporal output $E_t = \Xi_2(E_t)$ (see Sect. 2.2.1). So the only algorithmic primitives that are needed are the logical `OR` and the logical `ADD` between one pixel and its immediate neighbor, in each of the four directions. The filtered output E_t represents the final detection label and it is used as a binary mask to inhibit the update of the ΣΔ mean M_t.

The implementation of other spatial operators have been also optimised on the PAR taking care of its constraints (four-connectivity). The 2D filters are split into 1D filters (Fig. 6). There are, at least, two passes: one pass for the vertical operator and one pass for the horizontal operator. After each pass, results are stored into memory. If the operator is not idempotent (like `ADD` used for the density computation) ($k \times 1$) and ($1 \times k$) operators are not split into smaller operators (Fig. 6, top). But if the operator is idempotent (like `AND` and `OR` operators used for ASF), each ($k \times 1$) and ($1 \times k$) operator is split into a set of (3×1) and (1×3) operators (Fig. 6, bottom), with, at each time, a memory access. This decomposition reduces memory access to directly connected neighbors.

Thus, for $k > 3$, non-idempotent $k \times k$ operators are expensive to implement. There are two reasons: the first one is, the great amount of cycles dedicated to gather far pixels to the current PE and the second one is the cost of *serial-bit ALU* operations. For these reasons the density filter is much more slower than the erosion/dilatation filter (×4.5, ×5.9 and ×6.3 slower, respectively). As ASF are based on erosions and dilatations, their implementation remains efficient even if they have a great complexity, making them faster than density operator.

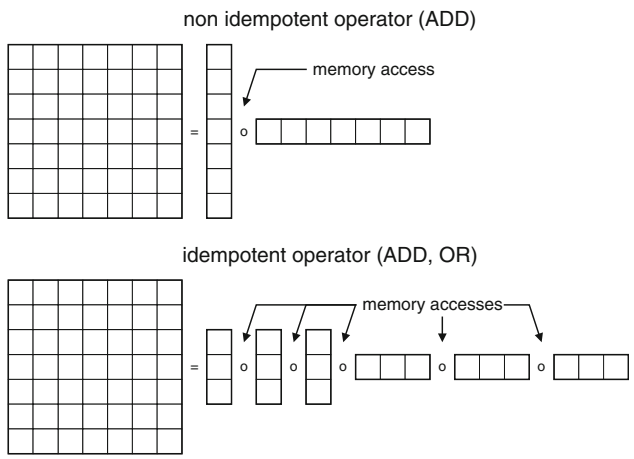


Fig. 6 2D Spatial filters optimization on retina

3.2.3 Hybrid algorithms

Although *Pvlsar34* can simply and quickly compute relaxation operators such as skeletons, or morphological connected operators, its efficiency in terms of useful computation is low for such irregular operators because of the expense due to the synchronous sequencing of the whole grid, that will only serve in some specific regions of the image. For that reason, a reduced set of asynchronous operators has recently been proposed by [24] to increase the computing level of the PARs. Thus, programmable connections, spanning tree constructions, OR and SUM asynchronous associations will be integrated in the next generations of PARs.

Such hybrid synchronous/asynchronous architecture will allow us to perform operations over a selected region (connected component) very efficient. This is the case of the geodesic reconstruction, which is useful for the motion detection algorithm (see Sect. 2.2). In the asynchronous model, the corresponding operation is computed like in the Associative Mesh (see Sect. 3.3): the reference set X is used as a binary mask to open/close the connections of the programmable mesh. Then an OR − association is computed on the marker set Y; the output is the result of the reconstruction $\mathrm{Rec}^X(Y)$.

3.3 Associative Mesh

The Associative Mesh [19] intends to exploit a massive data-parallelism, originating from a model based on network reconfigurability: the Associative Nets Model [33]. To allow efficient hardware optimizations for the large diversity of algorithms in image processing [34], the architecture is built from the observation of data-movements and data-structures encountered in this field. The Associative Mesh relies on a dynamic reconfigurability of its processors network and on an asynchronous electronic used to perform global operations and communication tasks. Reconfigurability and asynchronism offer solutions to adapt architectures to this context [5]. Several studies have shown that most techniques of image processing can be implemented using the Associative Mesh [3, 8, 16, 17] or architecture using some of the implementation techniques of the Mesh and the Associatives Nets concepts [22, 23].

3.3.1 Associative Nets Model theory

The Associative Nets Model is characterized by the application of associative operators on a locally reconfigurable, directed interconnection graph called *mgraph* implemented locally in each processor to enable its dynamic evolution in the course of an algorithm. *Mgraphs* can represent objects (Fig. 7) coded, processed or manipulated in image processing such as connected areas, edges, oriented trees, etc. It allows us to think not only in terms of point-to-point communication between processors but to apprehend information at a higher level.

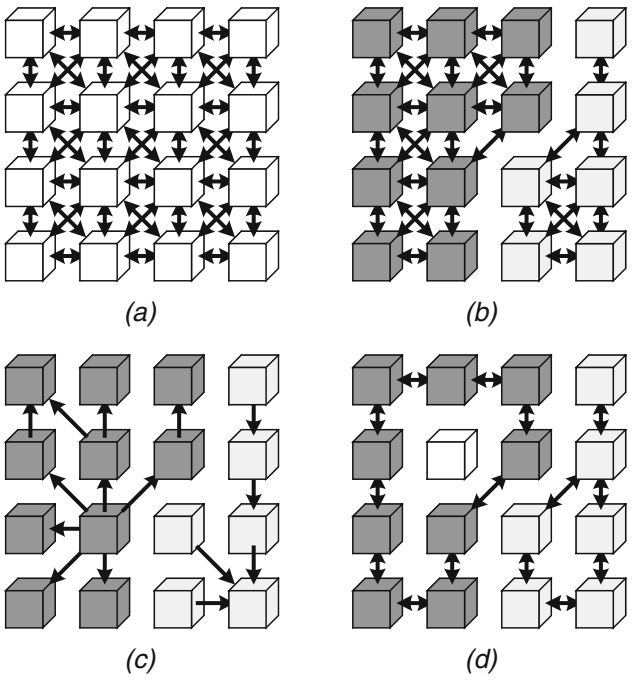


Fig. 7 Different mgraphs configurations: **a** full graph; **b** region graph; **c** oriented trees and **d** edges graph

Operations in the Associative Nets Model combine communication and computation aspects and are called 'associations'. They consist in a global application of an operator—such as logical operators, addition, minimum/maximum or spanning tree generation—on data spread over a connect set of the considered *mgraph*. As a basic example, this primitive can be used to asynchronously compute the area of a region by globally summing 1 per pixel on the mgraph connected components. It happens that most complex algorithms can be realized by iterating these primitive operations. Local associations are also allowed and are named *Step Associations*; the operator in this case is used to combine the local value of a processor with its nearest neighbors on the *mgraph*. Figure 8 presents an example of a global MAX − association.

3.3.2 Associative Mesh architecture

The Associative Mesh is a SIMD hardware transposition of the Associative Nets Model, featuring an 8-connected 2D mesh. Its originality comes from an asynchronous implementation of associations: the interconnection graph can be seen as an asynchronous path where data freely circulate from a processor to another, propagating local results to neighbors until global stability is reached.

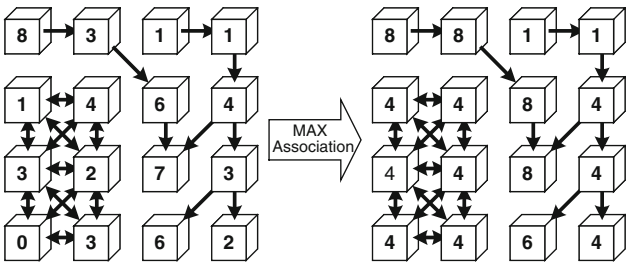


Fig. 8 MAX-association

Reconfigurability directly stems from the concept of mgraphs: each processor includes an 8-bit mgraph register, where each bit emulates the absence or presence of an incoming edge originating from a neighboring processor. The mgraph register is connected to the input of an AND-gate mask, which filters data emitted by the neighbors.

A Mesh processor is built around two distinct parts: an Associative Element (AE) which performs asynchronous associations and a Processing Element (PE) dedicated to internal operations and memory tasks, featuring an all-purpose memory bench, dedicated registers to save the local mgraph value, an independent scan-register for image input/output and an ALU to perform basic local operations (Fig. 9).

In order to save space, AEs have a 1-bit data-path. A n-bit association will then be performed as an iteration of 1-bit associations. Operators have been designed to ensure that data cross a minimum of logical layers to optimize the traversal time of each AE. As an example, in a simulation based on a 90-nm technology and a Mesh running at 500 MHz, 40 AEs can be crossed in one clock cycle during an OR − association. As a result, the basic global primitives of the model, associations, are performed in a very interesting computation time: simulations using the same technological parameter indicate that for a 512 × 512 image, OR − association on 8-bit data is performed in 60 ns and PLUS − associations in 200 ns [13]. Such a speed on

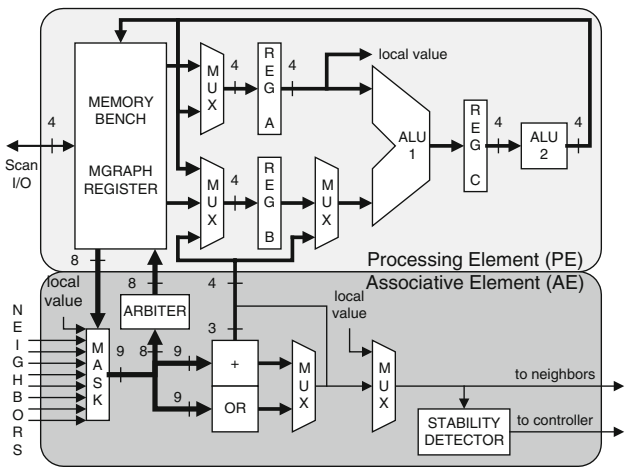


Fig. 9 Processor architecture

Table 4 List of SIMD instructions on the Associative Mesh

Affectation
Equal
Boolean instructions
OR, AND, XOR, NOT, Shift
Comparisons
$=, \neq, >, \geq, <, \leq$
Arithmetical operations
$+$, minus; , $\times, /$, modulo

global operations emphasizes the impact of the asynchronous network on the Mesh's performance. Available instructions are listed in Table 4.

3.3.3 Processor virtualization and SIMD

With current technologies, the architecture discussed above is not optimised with a SoC approach, meaning a complete image analysis machine inside one chip. We can improve the Mesh integration by changing the PEs granularity: we now assign a group of N pixels to each processing element (now called SIMD PE) and consider that we have N virtual PEs per physical SIMD PE (N is called degree of virtualization) [12]. To retain the benefits of asynchronism (very fast computation time, easy controllability), the AE structure is preserved in its original configuration. Thus, only the synchronous parts of the design are affected by the virtualization process. Figure 10 presents a virtualized PE dealing with 2 pixels.

This reorganization allows us to envision the architecture as the juxtaposition of an asynchronous communication network and a set of virtualized synchronous units, each managing N pixels. This new structure enables a significant area gain: we have shown that the design area is reduced by 20% if $N = 16$, 25% if $N = 1{,}024$. With $N = 1{,}024$, the hardware cost of a 256 × 256 Associative Mesh, including 64 SIMD PEs, each managing 32 × 32 pixels, is about 165 millions of transistors. However, virtualization induces an increase of computation time due to the serialization of local operations. Still, this increase can be limited by implementing a SIMD unit in each SIMD PE, so we can parallelize, up to a certain point, operations for pixels managed by the same SIMD PE and reduce computation times in significant proportions [13].

3.3.4 ΣΔ initialization

The ΣΔ initialization is entirely performed by the SIMD PEs. Parallel conditional statements like WHERE or ELSEWHERE implement the IF−THEN−ELSE instructions by performing a sequence of operations in each PE, according to the result of a local logic comparison.

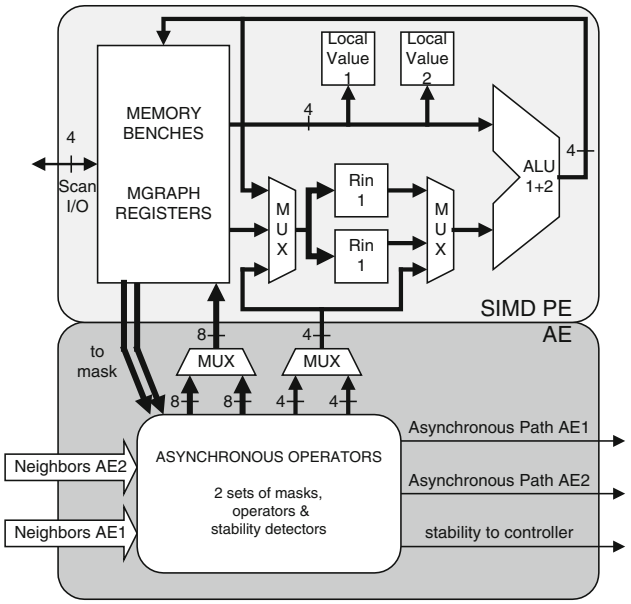


Fig. 10 Processor architecture with virtualization

3.3.5 Markov

The update strategy used is *image recursive* for full-parallel updates. The energies computation are held by the SIMD PEs while the AEs are used to compute p, the sum of spatial 8-connected sites, using a local PLUS − *association*. Conditional statements are used to collect sites label from E_{t-1} and E_{t+1}, compute V_p and V_f and also to set the final label to the site, depending on the total energy u. Note that the graph mask is configured by the set of the eight masks (the four principal directions: North, South, West, East and the four secondary directions North-West, North-East, South-West and South-East).

```
// Difference of Adequacy energy computation
ΔU_a = (α × (2 × o × α))/4 × σ^2;
// Energy due to potential V_p
WHERE(E_{t-1} == 1)   U_p = −β_p;
ELSEWHERE U_p = β_p; ENDWHERE;
// Energy due to potential V_f
WHERE(E_{t+1} == 1)   U_f = −β_f;
ELSEWHERE U_f = β_f; ENDWHERE;
// Energy due to potential V_s
// Graph configuration: fully 8-connected graph
Graph = mNW+mW+mSW+mS+mSE+mE+mNE+mN;
s=PLUS-ASSOCIATION(Graph),E_t);
U_s = (8 − 2s) × β_s;
// Model energy computation
U_{m1} = U_s + U_p + U_f;
// Pixel labeling
WHERE(U_{m1} < ΔU_a) Ê_t = 1;
ELSEWHERE Ê_t = 0; ENDWHERE;
```

Associative Mesh ICM version

3.3.6 Binary geodesic reconstruction

Reconstruction takes an efficient use of the Mesh's AE units: the geodesic mask is represented as a graph, where each object is a unique connected component. Pixels of the mask (set to 1) are linked together with the LINK − WITH − ONES mgraph creation primitive. The markers are then dilated up to the mask's limits by performing a global OR − association on the graph. The worst case is met when a unique object—shaped as a spiral—with a marker on one of its extremities fills the 200 × 200 image. Simulations based on a 90-nm technology reveal that for this extremely rare configuration, this operation on a Mesh running at 500 MHz will take 500 cycles. However, since data are 1-bit wide, it will only take a handful of cycles in most cases for the association to complete, thus providing a very interesting computation time.

```
// Creation of a graph representing the geodesic mask
// from the ImageMask binary image
LINK-WITH-ONES(GraphMask,ImageMask);
// Markers dilatation
Result=OR-ASSOCIATION(GraphMask,ImageMarker);
```

Binary geodesic reconstruction on Associative Mesh

3.3.7 Morphological opening

A dilatation on binary data, with a 3 × 3 structuring object, is simply achieved by a local OR − association. Operating with a 5 × 5 or 7 × 7 structuring object only requires an iteration of local associations. Erosion is computed in a similar way, this time with an AND − association. Therefore, a morphological opening will be implemented on the Mesh by computing 1, 2 or 3 local AND − association, followed by 1, 2 or 3 OR − association, depending on the size of the object.

3.3.8 Density operator

On a 3 × 3 window, each pixel's eight neighbors are summed in parallel by a local PLUS − association. The final threshold is performed in the SIMD PEs. To operate on larger windows, we must ensure that each pixel will be counted once and once only. Addition is not idempotent, so it is impossible to simply iterate local associations as we did with the morphological opening. In consequence, we have to divide a 5 × 5 or 7 × 7 window into 3 × 3 or smaller sub-windows (Fig. 11). A sub-total is then computed in each sub-window, using a local PLUS − association. Finally, each sub-total is sequentially propagated to the central node to be added in the final sum. A last threshold is then performed in the SIMD PEs. As Boolean associations

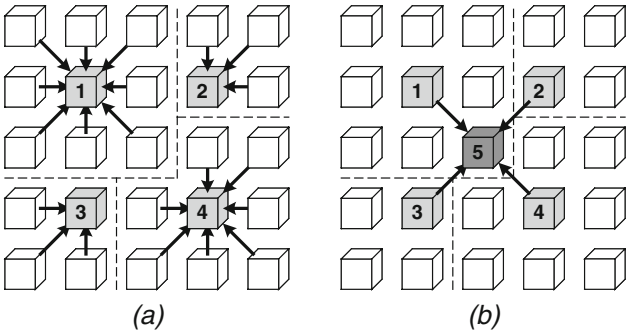


Fig. 11 **a** Sub-window split for a 5 × 5 density operation; **b** sub-totals propagation to the central node

like OR − association and AND − association are idempotent, they require less graph configurations and associative operations. For the opening, respectively, 2 and 3 for 5 × 5 and 7 × 7 associative operators and only 1 graph configuration versus 5 and 17 associative operators and as many as graphs configurations for non-idempotent operation like the addition for the density operator.

```
// Graph config and sub-total for the upper left 3x3 window (1)
// (for example, the mW value activates the link between
// the processor and its west neighbor)
Graph=mNW+mW+mSW+mS+mSE+mE+mNE+mN;
Sum1=OR-LOCAL-ASSOCIATION(Graph,Image);
// Graph config and sub-total for the upper right 2x2 window (2)
Graph=mE+mNE+mN;
Sum2=OR-LOCAL-ASSOCIATION(Graph,Image);
// Graph config and sub-total for the bottom left 2x2 window (3)
Graph=mW+mSW+mS;
Sum3=OR-LOCAL-ASSOCIATION(Graph,Image);
// Graph config & sub-total for the last sub-window (4)
Graph=mW+mSW+mS+mSE+mE+mNE+mN;
Sum4=OR-LOCAL-ASSOCIATION(Graph,Image);

// Propagation to the central node
Graph=mNW; // for sub-window (1)
Final=OR-LOCAL-ASSOCIATION(Graph,Sum1);
Graph=mNE; // for sub-window (2)
Final=Final+OR-LOCAL-ASSOCIATION(Graph,Sum2);
Graph=mSW; // for sub-window (3)
Final=Final+OR-LOCAL-ASSOCIATION(Graph,Sum3);
Graph=mSE; // for sub-window (4)
Final=Final+OR-LOCAL-ASSOCIATION(mSE,Sum4);

// Final Threshold
WHERE(Final>12) Pixel=1;
ELSEWHERE Pixel=0;
ENDWHERE;
```

5 × 5 density operation on Associative Mesh

3.4 Architectures specification summary

In order to compare the three architectures and to focus on their advantage and drawback, their specifications are summed up into two tables: the architectural specifications (RAM, amount of transistors and power consumption) and bandwidth specifications (access to internal data and external data).

Table 5 Architectures specifications

Architectures	Frequency	Internal RAM	Transistors	Watts
AM	500 MHz	2 MB	160 M	2 W
Retina	5 MHz	225 KB	4 M	100 mW
G4	1 GHz	32 KB + 512 KB	58 M	10 W

Table 5 provides the size of the internal RAM (size of the cache hierarchy on PowerPC G4 and size of the distributed memory on the Mesh and the retina) and an estimation of the power consumption. For the PowerPC G4, this is an average value, for the Mesh this is an estimation and for the retina, this is the measured value.

One very important point for comparison is the bandwidth of these architectures. As the Mesh and Retina are parallel architecture, we use the concept of *aggregate bandwidth* originating from high performance computing. The aggregate bandwidth is the sum of the bandwidth of all processors (Table 6). Then we consider the internal bandwidth as the bandwidth between the processor and its closest RAM (L1 cache for the PowerPC G4 and distributed internal RAM for Mesh and Retina) and the external bandwidth as the bandwidth of the external bus, connecting the processor, to the external RAM or to another processor. For the Mesh, this is the capability of the asynchronous network to transfer data from one AE to another AE. For the Retina this is the bandwidth to transfer data from one memory bank associated to one processor to one of its connected processors. The reason is that, for the retina, the bandwidth cannot be computed in the same way than for RISC processor or an associative network, where internal and external buses can be easily identified. Each elementary processor (PE) of the retina has 48 bits of memory and 4 bits are shared with the four neighbors. Internal bus bandwidth capacity is based on the number of cycles for a READ, i.e. 6 cycles. External bus capacity is the number of cycle to perform a copy from one of the four bits (6 cycles for the READ) to one of the 44 private bits (3 cycles for the WRITE). Internal bus bandwidth is to access private memory, external bus to access shared

Table 6 Bandwidths, per cycle and per second

Architecture	External bus	Internal bus
AM	64 B/c	1,024 B/c
AM	30 GB/s	476 GB/s
Retina	555 B/c	833 B/c
Retina	2.8 GB/s	4.1 GB/s
G4	1 B/c	16 B/c
G4	1 GB/s	16 GB/s

memory. Note that for the Mesh, the bandwidths are computed for an architecture of 256 × 256 AEs with a virtualization N of 1,024, i.e. 64 SIMD PEs.

We can notice and it is one of the main advantage of specialized architectures, that both Retina and Mesh can transfer much more data per cycle than a generalist RISC processor (×64 for internal and external buses). When considering bandwidth per second, the total aggregate bandwidth of the Mesh is close to the latest Cray vector processor performance [10] which has a peak bandwidth of 800 GB/s. Keeping in mind that most of the image processing algorithms are faced with *memory wall problem*, it is like if RISC still *wait for data* when the distributed buses of specialized machine can transfer data in time to feed processors.

4 Benchmarks

In order to compare the architectures, both from a qualitative and quantitative point of view, we used the frame rate and the cycle per point (cpp):

$$\text{cpp} = \frac{t \times F}{n^2}$$

where t is the execution time, F the processor frequency and n^2 the number of pixel to process, per processor. The cpp is an architectural metric to estimate the adequacy of an algorithm to an architecture [27]. For each architecture, we provide the cpp and the speedup for every operator (ΣΔ, ICM, morphological operator) and also for the whole algorithm as described in the first section. The algorithms have been implemented on a PowerPC G4 and PAR and have been simulated on the Associative Mesh with SystemC. For parallel architectures, the cpp expression is modified, depending on the number of pixels to be processed by a processor:

$$\text{cpp}_{\text{PAR}} = t \times F, \quad \text{cpp}_{\text{Mesh}} = \frac{t \times F}{n^2/N}$$

The cpp values have been calculated for 128 × 128, 256 × 256, 512 × 512 and 1,024 × 1,024 image size to analyze the cache behavior. We only provide the results for 256 × 256 image size to reduce the amount of results. For specialized parallel architecture like PAR or Mesh, the scalability is quite ideal so extensive results will not provide more information. For the PowerPC, more detailed results are provided to focus on the problem of cache misses.

4.1 Benchmark procedure

For the PowerPC, we used the following approach. As there is no clock cycle 64-bit counter, on powerPC under MacOS, we have used a micro-second counter based on the micro kernel MACH. As execution time is very short for small images, the measure is done on i iterations of the loop, to get a duration of ×1,000 the resolution of the timer. As this measure can be polluted by the OS, r runs are performed, and the minimum is selected.

For the PAR, a logic analyzer Agilent 1670 has been used. Acquisition, conversion and computation time are readable on the analyzer. The figures only take into account the computation time.

For the Mesh, algorithms have been implemented and simulated using a Mesh simulator based on a SystemC description of the architecture allowing a cycle-accurate evaluation. In order to achieve this feature, we need to evaluate the duration of an association. Besides technological or architectural issues, this duration depends on the initial value of the data and of the graph type used for the operation. For instance, an OR-based-association computation time is given by the longest distance between two logical 1s, Therefore, estimation can be performed by computing the number of processors walked through by data during the operation. To implement this process on the Mesh simulator, data circulating in the asynchronous network provide two informations, each going through a specific data path: on one hand, the local result of the association as a 1-bit value uses the standard architecture data path and, on the other hand, a counter representing the number of processors walked through so far by this data, which is incremented after going through a processor (Fig. 12). When the association terminates, data in the network with the highest counter value gives the duration of the association.

4.2 PowerPC G4 results

Four algorithms have been benchmarked: ICM, ΣΔ, density filters for 3 × 3 and 7 × 7 kernels and also Frame Difference (FD) algorithm. We added FD to get a reference in term of complexity and then in term of cpp, since no algorithm can be simpler than an absolute difference

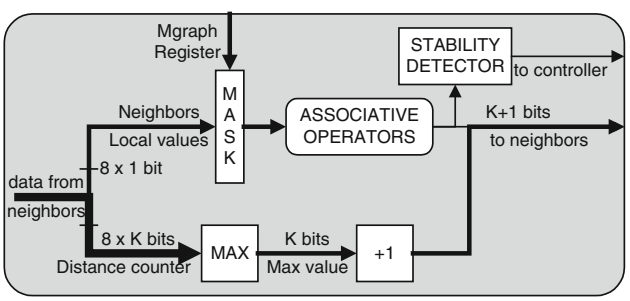


Fig. 12 Asynchronous data path on Associative Mesh simulator

followed by a threshold to detect motion. For each algorithm, two scalar versions and two "vector" versions were coded:

- s_0 scalar with no optimization, straight-forward coding,
- s_1 scalar with all possible optimizations,
- v_0 SWAR vector version with no optimization,
- v_1 SWAR with optimization like Loop unrolling, Register Rotation, strength reduction and computation factorization.

We provide cpp for classical image size, to point out the problem of cache behavior. For each algorithm, four ratios are also calculated:

- s_0/s_1 the impact of scalar optimization,
- v_0/v_1 the impact of SWAR optimization,
- s_1/v_1 the impact of SWAR switch for both optimised versions,
- s_0/v_1 the total acceleration from a basic/naive code to an optimised SWAR code.

We can see that the global speedup ($s0/v1$) is huge: from $\times 17$ for $\Sigma\Delta$ to $\times 60$ for 7×7 density filter. We can notice too that the code vectorization is the optimization technique that provides the highest speedup (line s_1/v_1: from $\times 6.8$ for ICM to $\times 15.6$ for density filter, while the scalar techniques all together provide a speedup (s_0/s_1) from $\times 1.2$ for $\Sigma\Delta$ to $\times 6.6$ for ICM. Such value of speedups make the use of optimization and vectorization to assert themselves for real-time computing on generalist purpose SWAR RISC processor.

Note that all the versions s_1, v_0, v_1 require some expertise from the developer. If these versions have been compiled with all optimization options of the compiler, without a little help, the compiler cannot achieve a level of performance higher than the s_0 version.

If we look in detail at the Fig. 13 that represents the cpp's evolution of ICM and $\Sigma\Delta$, for image sizes varying from 128×128 to $1{,}024 \times 1{,}024$, we can focus on two points. First there is a big gap in performance when image size increases and data do not fit in the cache. This phenomenon appears for different image size, depending on the algorithm (about 250×250 for $\Sigma\Delta$ and 350×350 for ICM). Then if both cpp are similar in the left part of the figure for small image sizes, the $\Sigma\Delta$cpp becomes 40% bigger than ICM cpp. The cpp value is multiplied by $\times 3.8$ between left and right part of the figure. This result is in contradiction with any complexity analysis: $\Sigma\Delta$ is more simple than ICM, but because it requires more images to be present at the same time in the cache and also because there is very few instruction to optimize, there is no possibility to optimize the code. The $\Sigma\Delta$ algorithm is a typical case of *memory bounded problem*. The performance decrease is more important than for ICM or other algorithms studied here. That raises another problem: SIMDization is efficient only when data fit in the cache, if the global speedup (s_0/v_1) is $\times 17.1$ for 256×256 images size, it is only $\times 5.2$ for $1{,}024 \times 1{,}024$ images size.

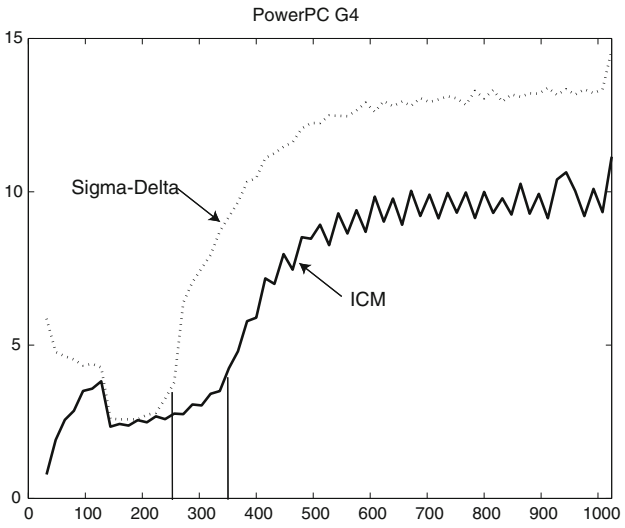


Fig. 13 ICM and $\Sigma\Delta$cpp on PowerPC G4

Finally, if we compare the cpp of the best version (v_1) of ICM or $\Sigma\Delta$ algorithm with the naive scalar version (s_0) of FD or even the optimised scalar version (s_1), we can see that the SIMDization makes complex algorithms like Markov Random Field relaxation, or $\Sigma\Delta$ filtering run faster than FD. From a qualitative point of view, this enforces the use of SWAR on general purpose RISC computer since such SIMD multimedia instructions make robust algorithm run faster than naive algorithm if this one is not optimised.

In the next subsection, we will focus on the implementation of these algorithms on the Retina and on the Mesh to finally compare them from an *embedded point of view*: frame rate and power consumption.

Table 8 shows, for PowerPC G4, that scalar optimizations are as important as SWAR optimizations: $\times 7$! As usual, the most efficient optimization is the highest level optimization: the algorithmic transform by LUT utilization provides a speedup of $\times 3.6$. Caches have also an important impact on performance whether the data fits in the cache (256×256) or not (512×512 and more).

4.3 Retina benchmarks

The results presented in this section, related to computation time and energy, have been measured on our experimental device composed of the 200×200 PAR *Pvlsar34* connected to an *Excalibur* board EPXA1, used to control the PAR and to perform higher level computations. The measures have been made using an oscilloscope and a logic

analyzer except for density operator where figures are estimated, not measured.

Table 9 details the cost in time of the different functions. The first column represents the number of Boolean instructions, the second column the number of clock cycles and the third column the time, in ms. The acquisition corresponds to the time of photo-transduction, during which no operation is performed. This time, measured here in normal conditions of our laboratory, naturally varies according to the lighting conditions. For the following functions (digital conversion, ΣΔ estimation and spatial binary morphology) the computation time only depends on frequency of the retina. For the spatial processing (binary morphology), three different sizes are considered for the largest radius of the structuring element set used both by the alternated sequential filter and the density operator.

If we only consider the computation time, the overall time consumed by the PAR is approximately 3.5 ms per frame, among which 2 ms for the CNA and 1.5 ms for the algorithm itself. These times are measured for a control frequency of 5 MHz. This means that, if we discard the acquisition time (which can make sense for a PAR observing a strongly lighted scene, for which the 2 ms of the CNA are sufficient as acquisition time), then a frame rate of 285 images/s is attainable at 5 MHz. Conversely, if the frame rate of 25 images/s is sufficient, then the control frequency can be lowered to 440 kHz, thus reducing proportionally the computing power.

At 5 MHz, the computing power of the whole device (PAR + EPXA1 board) has been measured at less than 1 W, from which only 100 mW is consumed by the PAR circuit and its cortex controller (external micro controller, Fig. 5) and the rest by the EPXA1 board. This means that the computing power of the PAR-based vision system can certainly be lowered significantly by developing specific controlling ASIC instead of using off-the-shelf development kit.

Table 7 Implementation on PowerPC G4: cpp and gain

Algorithm	FD	ΣΔ	ICM	Density3	Density7
cpp of scalar and vector versions					
s_0	28.7	50.5	112.2	27.2	114.0
s_1	17.5	40.5	16.9	12.3	30.1
v_0	3.4	4.5	3.4	1.9	5.7
v_1	1.2	3.0	2.5	1.5	2.5
Gain between scalar and vector versions					
s_0/s_1	×1.6	×1.2	×6.6	×2.2	×3.9
v_0/v_1	×2.9	×1.5	×1.4	×1.2	×2.7
s_1/v_1	×15.0	×13.7	×6.8	×8.2	×15.6
s_0/v_1	×24.6	×17.1	×44.9	×18.3	×60.0

Table 8 PowerPC G4 optimizations impact for ICM

Version	cpp	Gain
Basic	147	–
LUT	41	×3.6
Internal loop unrolling	32	×1.3
External loop unrolling	20	×1.6
SIMD vectorization	2.6	×7.7

Table 9 Computation costs of the different algorithmic functions of ΣΔ detection on *Pvlsar34*

Function	#*i*	#*c*	*t* (ms)
Acquisition	0	0	15
Digital conversion	64	2.5k	2
ΣΔ Estimation	160	6.5k	1.3
Spatial binary morphology			
3 × 3 ASF	25	1k	0.2
5 × 5 ASF	58	2.3k	0.5
7 × 7 ASF	108	4.3k	0.9
3 × 3 Density operator	36	2.2k	0.4
5 × 5 Density operator	95	6k	1.2
7 × 7 density operator	152	9.6k	2.0

Table 10 cpp and frame processing rate of Associative Mesh (for 200 × 200 images)

Algorithm	cpp	*t* (μs)	Frame rate
Frame Difference	0.5	1.024	977×10^3
ΣΔ	35	70	14.3×10^3
ICM	70	140	7.14×10^3
Geodesic reconstruction	0.46	0.94	1.07×10^6
3 × 3 Morphological opening	0.19	0.39	2.58×10^6
7 × 7 Morphological opening	0.44	0.91	1.11×10^6
3 × 3 Density operator	0.32	0.65	1.55×10^6
7 × 7 Density operator	10.21	20.9	47.8×10^3

4.4 Associative Mesh results

The results provided in the following section were simulated using a 90 nm technology parameter. On the Associative Mesh, the ICM cpp is about 70 for one ICM relaxation (varying from 70 to 80, depending on the degrees of SIMD and virtualization) and 35 for ΣΔ.

For the morphological operator, the Associative Mesh cpp is higher than PowerPC G4 cpp because of 1-bit implementation of `PLUS` − association. But with SIMD distributed processing power, it has the higher frame processing rate, even with virtualization.

The Mesh achieves spectacular performance. The bandwidth offered by its internal busses allow the Mesh to achieve a frame processing rate of 24,800 images/s. This number could, however, be impacted by the performances and/or synchronization with the video sensor. Another physical limitation is the number of incident photon impact(s) on the associated sensor.

4.5 Synthesis benchmarks

We only take into account the computation time and discard the acquisition time, the conversion time, the transfer time, and the power consumption of these operations. We are aware that results are a bit unfavorable to the PAR as the acquisition and conversion is integrated into itself contrary to the Mesh and the PowerPC. Right now, there is no way to get better results so the synthesis benchmarks will focus on the computation time to evaluate architecture performance. Considering the power consumption of a sensor—which is about 500 mW for both acquisition and conversion—the simulation and execution times will change but one order of magnitude between the PAR, the Mesh and the PowerPC performances will still exist.

Two configurations of benchmarks have been done (Table 11):

#1 $\Sigma\Delta$ + Markovian relaxation (four iterations of ICM),
#2 $\Sigma\Delta$ + morphological post-processing (geodesic reconstruction, 3 × 3 density or 3 × 3 ASF, depending on the architecture).

In the configuration #1, $\Sigma\Delta$ is considered as a pre-processing algorithm used to provide a better initialization for ICM than classical Frame Difference algorithm. In configuration #2, $\Sigma\Delta$ is a "stand alone" algorithm with a post processing step to remove the remaining noise. The choice of spatial regularization algorithm has been done to be coherent with the architecture capabilities, i.e.:

- geodesic reconstruction on the Associative Mesh, since it is the strongest algorithm, by far and its implementation is efficient on Associative Mesh (compared to the implementation of the other architectures)
- 3 × 3 ASF on the PAR, since ASF is the most efficient operator on the retina.
- 3 × 3 density on SWAR CPU, to get a complexity that is comparable to PAR complexity, keeping in mind that after optimizations, SWAR 5 × 5 and 7 × 7 operators are quite as fast as the 3 × 3 operator.

To assess the performance of the retina, we only take into account the processing time (1.3 + 0.2 = 1.5 ms) not the total time (acquisition + conversion + processing). The reason is that both acquisition and conversion times are unknown for the PowerPC G4 and the Associative Mesh. This leads to a frame rate of 667 images/s. The estimation of energy consumption is based on this assumption.

Table 12 presents the energy consumption of the three architectures for the configurations #1 and #2. We can notice that specialized architectures are by far more efficient than the general purpose processor—even with SWAR computation: performance ratios are all greater than ×10. This also means that even a 50% error, about the estimation of PowerPC G4 power consumption, is definitively not a problem.

4.6 Benchmark analysis

Before concluding, we focus, for each architecture, on the impact of the optimizations and the efficiency of the implementation.

RISC PowerPC G4

- From a point of view of embedded system, Altivec is well-adapted to complex algorithm like ICM relaxation: the ratio with Associative Mesh is ×35.7 for configuration #1 and ×88.6 for configuration #2. That could lead people to redesign SIMD Mesh PE architecture with an Altivec-like SWAR Instruction Set Architecture. For example a sub-set with only integer and also with restriction within the cross-bar capabilities could be integrated on a FPGA.
- For RISC, SWAR is very efficient, since a complex and robust algorithm like those proposed in the

Table 11 Benchmarks results for configurations #1 and #2

Image size	PowerPC G4	Retina	Mesh
Configuration #1: $\Sigma\ \Delta$ + *4 ICM*			
Frame rate	1,178	–	24,800
Real-time Freq (MHz)	21	–	0.504
Energy (μJ)	8,500	–	201.6
Configuration #2: $\Sigma\ \Delta$ + *morpho*			
Frame rate	3,436	667	184,000
Real-time Freq (kHz)	7,300	188	68
Energy (μJ)	2,900	150	27.1

Table 12 Energy comparison for configurations #1 and #2

Image size	128	256	512	1,024
Configuration #1: $\Sigma\ \Delta$ + *4 ICM*				
G4/AM	×36	×42	×139	×155
Configuration #2: $\Sigma\ \Delta$ + *morpho*				
G4/PAR	×14.4	×18.0	×61.3	×72.1
G4/AM	×89	×107	×329	×395
PAR/AM	×5.5	×5.5	×5.5	×5.5

configuration 1 and 2, are running faster, after SIMDization, than naive Frame Difference.
- Another point for fair comparison, is the cache size of a RISC. We can see that the G4 is efficient (cpp low) for size up to 300 × 300. This means that for smaller size, the G4 efficiency is underestimated, from an embedded point of view, since it will work fine with smaller cache. Not only we can apply a down-clocking frequency for its embedded version, but we can also reduce its cache (both will decrease power consumption).
- Down-clocking for *System on Chip*: Altivec frequency could be as low as 10 MHz for both configurations and for 128 × 128 and 256 × 256 images.

Retina

- The cost of the *serial-bit ALU* is a problem for arithmetic operators. A 8-bit ALU would have a great impact on performance, but will also have a negative impact of size and power consumption of the retina. A material *full adder* may be a golden mean to have good arithmetic performance.
- Asynchronous logic and graph manipulation is a *must have* for specialized architecture, not only for low level operations, but also and especially for middle level operations with irregular processing like the morphological reconstruction. Next generation of artificial retina should integrate such kind of silicon graph management.

Associative Mesh

- Computation results show that the Associative Mesh is well suited for both configurations. Each sequence of algorithms takes advantage of one of the Mesh's architectural characteristics. For configuration 1, the massively parallel resources easily handle the amount of computation required by the ICM relaxation. For configuration 2, the dynamic reconfiguration of the graph's structure allows to efficiently represent the objects, while the asynchronous implementation of global operations guarantees a fast processing of the geodesic reconstruction. In both cases, frame rates are quite spectacular.
- A remarkable aspect of the algorithms implementation on the Associative Mesh, in contrast with PowerPC G4 (and to a lesser extent, with retina) is that computation time is quasi-independent of the images' size or the detected object's shape.
- The major drawback is the hardware cost of the Mesh to process big images when compared to the other architectures. Still, vision SoC implementation of a 256 × 256 Associative Mesh is compliant with current technology and only requires 3 times more transistors than a PowerPC G4 for a ×20 faster computation.
- With such performance, reducing the clock frequency by a factor 10 could still allow to process more than two thousand 256 × 256 images/s with a power consumption under 1 W. The Associative Mesh could then be used in association with a HD camera on a SoC platform.

5 Conclusion: future architectures

We have presented the implementation of robust sets of operator for Motion Detection, based on Markov Random field, Sigma–Delta filtering and morphological operators like opening, density and Alternate Sequential Filter. These algorithms have been used to emphasize the intrinsic qualities and drawbacks of these architectures (Sect. 5.1) and then to envision the specification of future architectures, first with SWAR paradigm (Sect. 5.2), second with FPGA-based customization (Sect. 5.3) and finally with many-core reconfigurable processor (Sect. 5.4).

5.1 Pros and cons of the three architectures

- SWAR is efficient for low level algorithm. Currently used in RISC processor and also present and customized in some SoCs.
- Asynchronous Associative Networks. This model of computation is extremely efficient for both power consumption and intermediate levels algorithms. It is efficient for power consumption because asynchronism mechanism. It is also interesting for speed since, in our case, up to 40 asynchronous associative operators can be executed during 1 synchronous cycle. It is also efficient for intermediate level of processing because associative networks can be reconfigured and then, an operator can be applied through a graph, to any connected components. Any kind of irregular and CPU intensive algorithms can be handled efficiently, like geodesic reconstruction, watershed segmentation and of course, connected component labeling.
- Retinas are very low power embedded architecture. For tight integration and an optimised connection between sensor and calculator, retinas outperform SoC and Vision SoC systems like FPGA + sensor. But, right now they are limited to regular processing. Integrating an associative network inside a retina will allow to use such a kind of machine for intermediate level algorithm. So a quite complete image processing chain could fit into a high parallel and versatile system.

5.2 SWAR enhancement

Nowadays, the two main solutions to computer architecture limitations are: increasing RISC performance or customizing FPGA.

When RISCs have replaced CISCs using architectural optimizations like pipeline, registers and cache, the RISC motto was more instructions per cycle, because they were using less complex instructions that can be fetched, decoded and executed faster than CISCs can. As at this time, it was commonly accepted that clock frequency can go higher and higher. The easiest way, thanks to the technology, was to increase the clock frequency. At the same time two evolutions of RISC were released: the *superscalar* architecture (multi ALU/FPU per chip) and the *VLIW* (Very Long Instruction Word) like the Intel Itanium or the Texas Instrument C6x DSP family. But since a few years, clocks frequency does not increase as much as before. The new RISC motto could be "more instruction per second". The General Purpose solution is the multicore approach (see Sect. 5.4) And the Domain Specific solution is SWAR extension. As we can see in Table 7, the speedup provided by Altivec—up to $\times 60$—released in 1998 is by far, greater than the current number of cores inside a processor in 2008. As SWAR implementation requires *few* transistors because of the very simple control structure due to SIMD model, one very efficient way to increase performance could be "more SWAR into RISC"

- longer registers: 256 bits or even 512 bits, to process more data per cycle,
- more smart instructions: Altivec has a very useful *vector permutation unit* that provides powerful instructions like `vec_sel` that is an aesthetic way to perform a SIMD *if then else* condition (replacing masks computations and combinations) or `vec_perm` that can permute data with any kind of pattern (SSE can only do regular patterns of interlacing). `vec_perm` is used for computing unaligned vectors, matrix multiplications and even for sorting data. Such a kind of unit should be present in any SWAR architecture,
- more specialized or dedicated instructions like Altivec `vec_sum`, `vec_msum` that performs reduction into a register and the SSE2 `_mm_sad_epu8(a,b)` that performs a sum of absolute difference (SAD) between two registers. This instruction is used in every correlation algorithm based on block-matching. Adding such an instruction has been studied into [29].

5.3 FPGA-based customization

Processor customization, as defined for reconfigurable architectures [39] and embedded systems, have to be explored. A customizable processor is a General Purpose Processor (GPP) embedded into a FPGA which cores can be enhanced. Most major FPGA manufacturers now provide solutions with softcore customizable FPGA (NIOS 2 for Altera, microblaze for Xilinx). Such technologies have room for improvements like adding new instructions, new customized format [37] for specific domain application [38] but also new dedicated blocks. With a compiler like C2H for Altera FPGA or DIME-C for Xilinx, a complete C function can be compiled into a VHDL block and be directly called inside a C code. GPP and its accelerators can then be seen as a full system on a chip. With these two levels of customization (instruction and hardware function) one can envision to add a new specialized instruction at the C level or new hardware function. A new instruction could be `b = sigmaDelta(a,b)` that compares `a` and `b` and increment/decrement `b` according to the result of the comparison. Such a function will remove *if then else* structure that stalls/flushes the pipeline. On the other hand an hardware function could implement a morphological operator. Such hardware implementation can be much more faster than the sequential execution of the instructions that compose it, as no more "register to register" stage is required at each cycle like it is the case for pipeline execution. One of the best example of processor customization (not softcore but ASIP) is the Tensilica Xtensa architecture [37].

5.4 Many-core reconfigurable processor

If classic systems are able to race against Moore's Law (bigger caches, more complex branch predictors, more hardware optimizations), they are slowly but steadily losing the efficiency race. The amount of transistors involved in those systems keep increasing, but most of them are only used to stay on par with Moore's Prediction. They could be used more efficiently if GPP were no so "General Purpose" but also target some specific domains like computer vision or multimedia. The solution to this problem is brought by recent technology advances by combining both above solutions: designing reconfigurable parallel processor.

This leads to the fact that different models of computation have to be implemented in order to fit a given domain constraints. For example, the PAR can execute various regular algorithms in a very fast and efficient way even if the PAR itself is only composed of simple processors with few bits of memory. Similarly, the Mesh, thanks to its asynchronous network can handle irregular algorithms. Another example of such a processor is the PIMM [25]. PIMM is dedicated to morphological operations and use, an explicit hardware queue model to execute algorithms—like geodesic reconstruction or watershed

segmentation—faster than a GPP, which are the most used architecture for such tasks but are, in fact, less efficient.

Currently, multi-cores approach is the leading solution for Thread Level Parallelism. But these processors are designed for regular processing, irregular processing still being out of their range. This also applies to CELL processor and GPU: Cell is *just* a 9-core heterogeneous RISC processor and GPU—from Nvidia or ATI—are still dedicated to regular processing even if they are going to be more flexible when used with a Stream Computing language—CUDA [11] and Brook GPU [6].

Next generation will include specialized/dedicated logic to tackle the problem of GPP inefficiency. Adding a custom part of logic (100–200 millions of transistors) is definitively no more a problem, compared to the total size of a CPU (800 millions of transistors for current Intel quad core). One of the most promising architecture of this kind is the Intel Polaris/Larrabee architecture from Terascale project [41]. Polaris has a hierarchical bus to connect PEs together, PEs include an 512-bit SWAR unit and can be reconfigured, for 3D graphic processing or cryptography.

We believe that adding a custom part of logic into GPP (100–200 millions of transistors is now just a *part* of) dedicated to irregular processing would be a solution to the problem of GPP inefficiency.

Acknowledgment We wish to thanks Joel Falcou for his valuable help.

References

1. Azencott, R.: Simulated Annealing: Parallelization Techniques. Wiley, New York (1992)
2. Bellon, A., Derutin, J.-P., Heitz, F., Ricquebourg, Y.: Real-time collision avoidance at road-crossings on board the Prometheus-ProLab 2 vehicle. Intell Vehicles, pp. 56–61 (1994)
3. Biancardi, A., Segrovia-Martinez, M.: Adaptative segmentation of MR axial brain images using connected components. In: International Workshop on Visual Form, Capri, pp. 295–302 (2001)
4. Bouthémy, P., Lalande, P.: Recovery of moving object in an image sequence using local spatiotemporal contextual information. Opt Eng **32**(6), 1205–1212 (1993)
5. Blelloch, G.: Vector Models for Data-Parallel Computing. MIT Press, Cambridge (1990)
6. BrookGPU: http://ati.amd.com/technology/streamcomputing
7. Caplier, A., Bonnaud, L., Chassery, J.-M.: Robust fast extraction of video objects combining frame differences and adaptive reference image. Int Conf Image Process, pp. 785–788 (2001)
8. Ceccarelli, M., Petrosino, A.: A parallel fuzzy scale-space approach to the unsupervised texture separation. Pattern Recognit Lett **23**(5), 557–567 (2002)
9. Cohen, F.S., Cooper, D.B.: Simple parallel hierarchical and relaxation algorithms for segmenting non-causal Markovian random fields. IEEE PAMI **9**(2), 195–219 (1987)
10. Cray X1: http://www.cray.com/products/x1/specifications.html
11. CUDA: http://www.nvidia.com/cuda
12. Denoulet, J., Mérigot, A.: System on chip evolution of a SIMD architecture for image processing. In: Computer Architecture and Machine Perception, New Orleans, 12–16 May 2003. doi: 10.1109/CAMP.2003.1598175
13. Denoulet, J., Mérigot, A.: Evaluation of a SIMD architecture dedicated to image processing. Global Signal Process (2004)
14. Denoulet, J., Mostafaoui, G., Lacassagne, L., Mérigot, A.: Implementing motion Markov detection on General Purpose Processor and Associative Mesh. In: Computer Architecture and Machine Perception, Palermo, pp 288–293 (2005)
15. Diefendorff, K., Dubeyn, P.K., Hochsprung, R., Scales, H.: Altivec extension to PowerPC accelerates media processing. In: IEEE Micro (2000)
16. Ducourthial, B., Merigot, A.: Parallel asynchronous computations for image analysis. Proc IEEE **90**(7), 1218–1229 (2002)
17. Ducourthial, B., Constantinescu, G., Merigot, A.: Implementing image analysis with a graph-based parallel computing model. Computing. In: Supplementum, GbR'97, Workshop on Graph based Representation, Lyon, pp. 111–121 (1997)
18. Dudek, P.: An asynchronous cellular logic network for trigger-wave image processing on fine-grain massively parralel arrays. IEEE Trans Circuits Syst II Express Briefs **53**(5), 354–358 (2006)
19. Dulac, D., Merigot, A., Mohammadi, S.: Associative meshes: a new parallel architecture for image analysis applications. In: Computer Architecture and Machine Perception, News Orleans, pp 393–399, 15–17 December 1993
20. Forchheimer, R., Astrøm, A.: Near-sensor image processing: a new paradigm. IEEE Trans Image Process **3**, 736–746 (1994)
21. Daniel Hillis, W., Tucker, L.W.: The CM-5 connection machine: a scalable supercomputer. Commun ACM **36**(11), 31–40 (1993)
22. Galilee, B., Mamalet, F., Renaudin, M., Coulon, P.Y.: Parallel asynchronous watershed algorithm—architecture. IEEE Trans Parallel Distrib Syst **18**(1), 44–56 (2007)
23. Gies V., Bernard T.: Increasing interconnection network connectivity for reducing operator complexity in asynchronous vision systems. In: International Conference on Discrete Geometry for Computer Imagery, Poitiers, pp.1–10 (2005)
24. Gies, V., Bernard, T.M.: Tree extension of micro-pipelines for mixed synchronous-asynchronous implementation of regional image computations. In: Proceedings of SPIE, vol 5677, Sensors and Camera Systems for Scientific and Industrial Applications VI, SPIE (2005)
25. Klein, J.-C., Lemonnier, F., Gauthier, M., Peyrard, R.: Hardware implementation of the watershed zone algorithm based on a hierarchical queue structure. In: IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras (1995)
26. Komuro, T., Ishii, I., Ishikawa, M., Yoshida, A.: A digital vision chip specialized for high-speed target tracking. IEEE Trans Electron Devices **50**(1), 191–199 (2003)
27. Lacassagne, L., Milgram, M., Garda, P.: Motion detection, labeling, data association and tracking, in real-time on RISC computer. In: International Conference on Image Analysis and Processing, Venise, pp 520–525 (1999)
28. Lalande, P., Bouthemy, P.: A statistical approach to the detection and tracking of moving objects in an image sequence. Eur Signal Process Conf **2**, 947–950 (1990)
29. Limousin, C., Sebot, J., Vartanian, A., Drach, N.: Architecture optimization for multimedia application exploiting data and thread-level parallelism. J Syst Arch EUROMICRO J **51**(1), 15–27 (2005)

30. Lohier, F., Lacassagne, L., Garda, P.: A New Methodology to Optimize DMA Data Caching: Application toward the Real-time Execution of an MRF-based Motion Detection Algorithm on a multi-processor DSP. International Conference on Signal Processing Applications and Technology, March 1999, Phoenix USA
31. Manzanera, A., Richefeu, J.: A robust and computationally efficient motion detection algorithm based on Sigma–Delta background estimation. In: Proceedings Indian Conference on Computer Vision, Graphics and Image Processing, Kolkata (2004)
32. Manzanera, A., Richefeu, J.: A new motion detection algorithm based on Sigma–Delta background estimation. Pattern Recognit Lett **28**(3), 320–328 (2007)
33. Mérigot, A.: Associative nets model: a graph based parallel computing model. IEEE Trans Comput **46**(5), 558–571 (1997)
34. Mérigot, A., Zavidovique, B.: Image analysis on massively parallel computers: an architectural point of view. Int J Pattern Recognit Image Anal **6**(3), 387–399 (2002)
35. Nadrag, P., Manzanera, A., Burrus, N.: Smart retina as a contour-based visual interface. In: ACM Distributed Smart Cameras Workshop, DSC'06, Boulder (2006)
36. Sébot, J., Drach-Temam, N.: Memory bandwidth: the true bottleneck of SIMD multimedia performance on a superscalar processor. In: Proceedings of the 5th International Euro-Par Conference on Parallel Processing, pp 439–447, Springer, Berlin (2001)
37. Piskorski, S., Lacassagne, L., Bouaziz, S., Etiemble, D.: Customizing CPU instructions for embedded vision systems. IEEE Comput Arch Mach Percept Sensors (2006)
38. Piskorski, S. , Lacassagne, L., Kieffer, M., Etiemble, D.: Efficient floating point interval processing for embedded systems and applications. Int Symp Sci Comput Comput Arithmetic Valid Numer (2006)
39. List of reconfigurable architectures http://www.site.uottawa.ca/~rabielmo/personal/rc.html
40. Rodriguez-Vazquez, A., Linan-Cembrano, G., Carranza, L., Roca-Moreno, E., Carmona-Galan, R., Jimenez-Garrido, F., Dominguez-Castro, R., Espejo Meana, S.: ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs. IEEE Trans Circuits Syst **51**(5), 851–863 (2004)
41. Intel Tera-scale project: http://techresearch.intel.com/articles/Tera-Scale/1449.htm
42. Zavidovique, B., Stamon, G.: Bilevel processing of multilevel images. In: Proceedings PRIP'81, Dallas (1981)

Author Biographies

Lionel Lacassagne is an assistant professor in Fundamental Electronics Institute (IEF), University of Paris Sud (France). His research areas deal with High Performance Computing and Image Processing applied to embedded systems and especially the benchmarking of vision systems. The research activities are done in AXIS team (Architecture, Control, Communication, Vision, Systems).

Antoine Manzanera is an assistant professor in the Electronics and Computer Science Lab. at ENSTA, Paris (France). His activity field is Image Processing and Computer Vision algorithmics, particularly for Real-time and Embedded systems. His research interests are discrete geometry, mathematical morphology, motion analysis, and image models.

Julien Denoulet is an assistant professor at University Pierre and Marie Curie in Paris (France). He received his PhD in 2004 from University of Paris-Sud, working on massively parallel architectures for image processing. His current activities take place in the SYEL (Electronic Systems) group and deal with SoC-AMS modeling and methodologies for design exploration on reconfigurable SOC platforms.

Alain Merigot is professor of Computer Engeneering at the University of Paris Sud in Orsay (France) and researcher at the Fundamental Electronics Institute and Digiteo Labs. His research interests are computer architecture and image processing.