SUIVI D'OBJET POUR LA TELEVISION INTERACTIVE

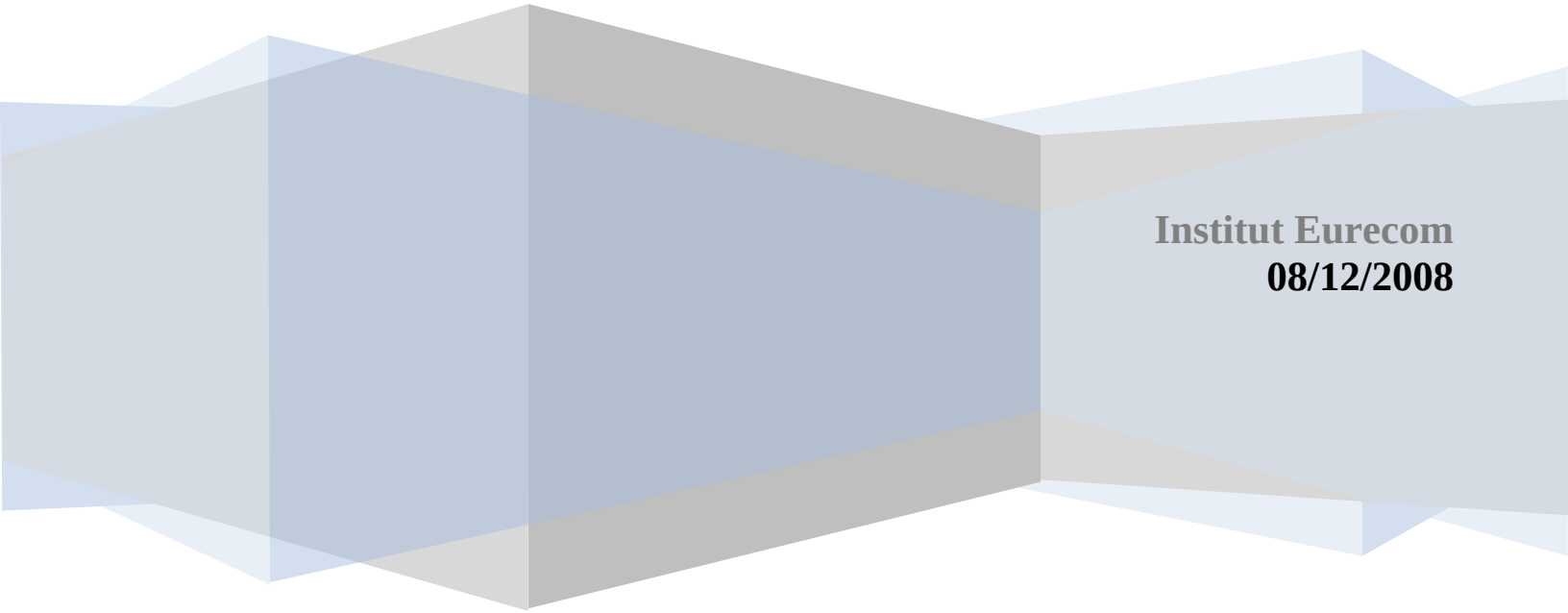
THESE

présentée en vue de
l'obtention du titre de

DOCTEUR EN INFORMATIQUE
de

L'école TELECOM ParisTech

Par
Trichet Rémi
Encadrant
Bernard Mérialdo



Institut Eurecom
08/12/2008

Remerciements

Je tiens à vivement remercier mon directeur de thèse, Bernard Mérialdo, pour sa grande disponibilité et sa très grande compétence tout au long de ces trois années. Je remercie également l'institut Eurecom de m'avoir donné les moyens d'effectuer cette thèse dans de bonnes conditions. Enfin je remercie les professeurs Liming Chen, Stéphane Marchand-Maillet, Béatrice Pesquet Popescu, et Philippe Joly d'avoir accepté de faire parti de mon jury.

Sommaire

Liste des figures :

1 Introduction

Le suivi d'objet est un domaine de recherche dont les premiers travaux remontent à la fin des années 80 et dont le développement s'est accru de façon exponentielle depuis. Ce phénomène peut être expliqué par des facteurs sociaux-économiques. D'une part, grâce à la loi de Moore, les puces semi-conductrices et donc le matériel informatique ne cessent de se miniaturiser, réduisant leur coût tout en augmentant leurs performances. Cela a conduit le prix des caméras et ordinateurs à baisser, rendant ainsi les systèmes de suivi abordables aux chercheurs et au marché des utilisateurs. De plus, leur utilisation croissante dans la vie de tous les jours pour des tâches auxquelles la capacité d'attention humaine a été prouvée inadaptée telles que le monitoring du trafic, ou l'asservissement visuel dans l'industrie s'est révélée être un franc succès. Mais l'impact le plus retentissant des systèmes de suivi est leur aide dans la lutte contre le crime ou les attaques terroristes. Après les événements du 11/09 aux Etats-Unis, et les attaques dans le métro londonien où les responsables ont pu être identifiés grâce au système de vidéo surveillance, les gouvernements de nombreux pays ont décidé d'accorder plus d'attention aux systèmes CCTV. En conséquence de tous ces facteurs, la demande pour des systèmes de suivi d'objet a été plus forte que jamais ces dernières années et de nombreux projets, européens ou internationaux, en incorporent le développement.

C'est notamment le cas du projet européen porTiVity visant à développer des services pour la télévision interactive au travers d'une plateforme prenant en charge le traitement de bout en bout. Ce système nécessite, entre autres, un outil d'annotation vidéo permettant au producteur de sélectionner et de suivre automatiquement un objet vidéo pour lui adjoindre des données supplémentaires auxquelles peut accéder ultérieurement l'utilisateur. Cet outil ayant à traiter tout type de vidéos impliquait la création d'algorithmes pour un suivi d'objet **générique** apte à gérer tous les types de difficultés envisageables. Les *points d'intérêts* ou *points-clés* nous sont apparus comme un outil adapté à la réalisation de cette tâche.

Après la justification de leur emploi, ce manuscrit s'articulera autours des diverses étapes d'un suivi d'objet s'appuyant sur des points d'intérêts. Dans chacune de ces étapes, nous présenterons nos contributions après un état de l'art circonstancié. Après avoir justifié le choix des points de Harris, nous présenterons notre modèle d'objet au travers de nombreuses améliorations visant à limiter les faiblesses des points-clés : un prétraitement ayant pour but d'exploiter au mieux les canaux couleurs, la conservation des points pendant *n* images limitant leur instabilité dans le temps, ou encore le délai d'utilisation afin de prévenir, dans une certaine mesure, les occultations. Nous détaillerons également un algorithme de labellisation des points d'intérêts différenciant les points de l'objet de ceux du décor afin de limiter l'influence de ce dernier sur le suivi. Enfin, nous présenterons « Fast Harris », un système accélérant l'extraction des points de Harris à l'aide d'une structure similaire à celle des ondelettes de Haar. La seconde partie de ce chapitre fera l'objet des descripteurs utilisés. Après justification de leur choix, nous montrerons les expériences effectuées afin d'optimiser leur paramétrage.

La première partie de la section 5 présentera notre méthode d'appariement normalisé des points-clés. Cet algorithme d'appariement utilise conjointement les descripteurs des points-clés et leurs relations spatiales, modélisées par une triangulation de Delaunay, pour accroître l'efficacité de l'appariement.

La seconde partie détaillera notre modèle de mouvement, un raffinement de la position de la boîte englobante exploitant la labellisation des points-clés ainsi qu'une adaptation du modèle de mouvement en fonction de l'encombrement de l'arrière-plan.

Enfin le dernier chapitre présentera les performances comparées de notre approche et d'algorithmes de référence en termes de précision du suivi. Nous donnerons aussi le temps d'exécution et la qualité du suivi pour quelques variantes de notre système.

2 Principes du suivi d'objet

Le suivi d'objet peut être défini comme « *la localisation spatiale et temporelle d'un objet au cours d'une séquence vidéo* », où un objet est « *un ensemble de pixels dépeignant la même représentation sémantique* » (ex : une voiture, un visage, un personnage…). Schématiquement, le principe consiste à extraire de l'objet des caractéristiques que l'on va s'efforcer de retrouver à chaque image de la séquence vidéo. Afin de restreindre le champ d'investigations, deux hypothèses sont communément acceptées. Tout d'abord, les formats vidéo actuels comprenant 25 où 30 images par seconde, on peut supposer le changement et le déplacement de l'objet minime entre deux images et limiter la zone de recherche au voisinage immédiat de sa dernière détection ; voisinage dont la taille est proportionnelle à la vitesse de l'objet. Ensuite, tout objet étant soumis aux lois de la physique de par son inertie, l'objet aura un mouvement d'une certaine fluidité. Afin de permettre au lecteur de mieux appréhender les mécanismes du suivi d'objet, ce chapitre décrira dans un premier temps les difficultés que les algorithmes sont susceptibles de rencontrer. Ensuite nous présenterons une classification des approches existantes. Nous ne décrirons pas ici les algorithmes en détail mais les avantages, inconvénients et domaines d'application des différentes familles de méthodes. En troisième partie, nous introduirons notre approche, induite par les contraintes du projet porTiVity dans lequel elle est développée. Enfin, nous débattrons les avantages et inconvénients des différents systèmes d'évaluation afin de justifier le choix de notre environnement de test.

2.1 Problématique du suivi d'objet

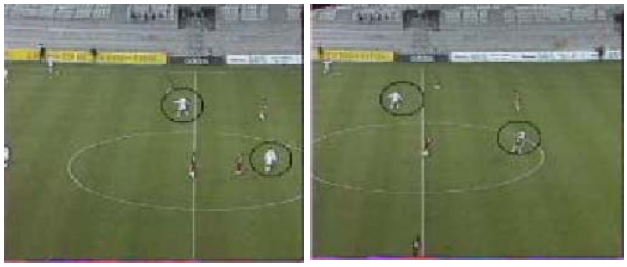
Les difficultés susceptibles de diminuer la qualité d'un suivi sont les suivantes :

Changement d'illumination : Il s'agit de la difficulté la plus répendue. Tout système de suivi doit donc pouvoir la gérer à un certain degré. Les changements d'illumination perturbent l'identification basée sur la valeur des pixels, particulièrement lorsque le changement d'illumination n'est pas uniforme.

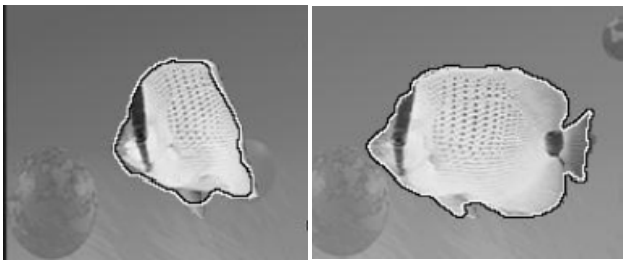
Changements d'échelle : Cette transformation change la taille de l'objet recherché et donc éventuellement du patron associé. La majeure partie des applications ne traitent pas ce cas de figure et considèrent une fenêtre de taille fixe. Le suivi est alors moins précis.

Occultations : Les occultations sont le cas de figure le plus difficile et la cause d'échec des algorithmes la plus fréquente. Elles peuvent être partielles ou totales. Si l'objet est partiellement masqué, une partie des caractéristiques de l'objet seront indisponibles, diminuant l'efficacité les algorithmes. Si l'objet est totalement masqué, seul le modèle de mouvement pourra présumer de la position de l'objet.

Mouvement de caméra : Ce désagrément perturbera la précision du modèle de mouvement. Pour pallier à ce problème, un prétraitement est parfois effectué pour extraire le mouvement global de l'image. Seuls les objets dont le mouvement diffère de celui-ci seront alors analysés.



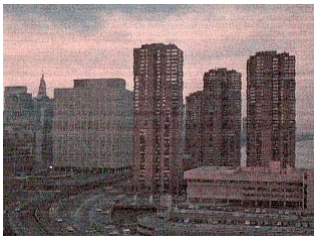
Déformation de l'objet : Cet événement modifie la quantité d'information disponible et le patron de l'objet, rendant fréquemment ce dernier caduc.



Objet de petite taille : Plus l'objet est petit, plus la quantité d'information disponible est réduite, plus son suivi pourra être influencé par les perturbations extérieures. De plus, les petits objets sont fréquemment rapides.

Objet en mouvement rapide : Il résulte d'un mouvement rapide le phénomène de « flou de bougé » gommant les contours et diluant les couleurs et textures. Les applications se basant sur ces traits seront donc pénalisées. De plus, lors d'un mouvement rapide, un suivi imprécis perdra plus facilement l'objet.

Vidéo de basse qualité : Ce type de vidéo sera sujet aux effets de flou et au bruit. Un prétraitement peu alors être effectué pour éliminer le bruit. Les effets de flou, comme expliqué précédemment compliqueront les suivis basés sur la couleur, la texture ou les contours.



Contrainte de temps réel : Cette difficulté n'est pas la moindre. En effet, la plupart des applications nécessiteront un compromis entre leur complexité et leur capacité à gérer les difficultés précédemment décrites. Cette contrainte sera donc un facteur majeur dans le choix de la technique à utiliser.

2.2 Etat de l'art des méthodes de suivi d'objet

Il existe d'innombrable techniques mises en œuvre pour réaliser le suivi d'objet, appliquées à des domaines aussi variés que la gestion et l'analyse du trafic, la robotique (asservissement visuel), la télévision interactive, la reconnaissance de gestuelle, la compression, ou encore la vidéo surveillance connaissant un essor important avec la recrudescence du terrorisme. Le choix de l'approche idoine sera fonction des connaissances à priori sur l'application et le type de vidéos traitées. Par exemple, une application de surveillance vidéo dans un métro impliquera une caméra fixe et un suivi d'humains dans un lieu clos. L'approche choisie pourra donc être sensible aux changements d'illumination, considérera une scène sans mouvement de caméras et utilisera certainement un modèle exploitant les caractéristiques physiques et une cinématique propres aux êtres humains.

Bien que chaque système de suivi ait les spécificités dictées par les contraintes du domaine d'application et les particularités des vidéos analysées, des classifications sont envisageables. Dans ce qui va suivre, nous nous focaliserons sur l'étude d'informations en deux dimensions issues d'une unique caméra. Les modèles d'objet 3D ou l'interpolation de données résultant de plusieurs vues ne seront donc pas considérées. Une classification possible se basant sur le type de caractéristique exploitée est proposée en Figure 1.

On distingue deux grandes familles de méthodes. Les méthodes basées patrons et celles basées traits. La première catégorie de techniques repose sur un modèle de la forme de l'objet et, en conséquence, est souvent utilisée dans le cadre d'un suivi d'objet rigide. Les méthodes sont généralement très rapides mais souffrent de graves lacunes, voire sont totalement inefficaces face à des difficultés communes. Cette classe regroupe le block matching, les méthodes basées sur un maillage (déformable ou non), ou sur le contour de l'objet, ou encore sur des modèles spécifiques au type d'objet suivi, tels que les modèles en bâton utilisés pour le suivi de corps humain.

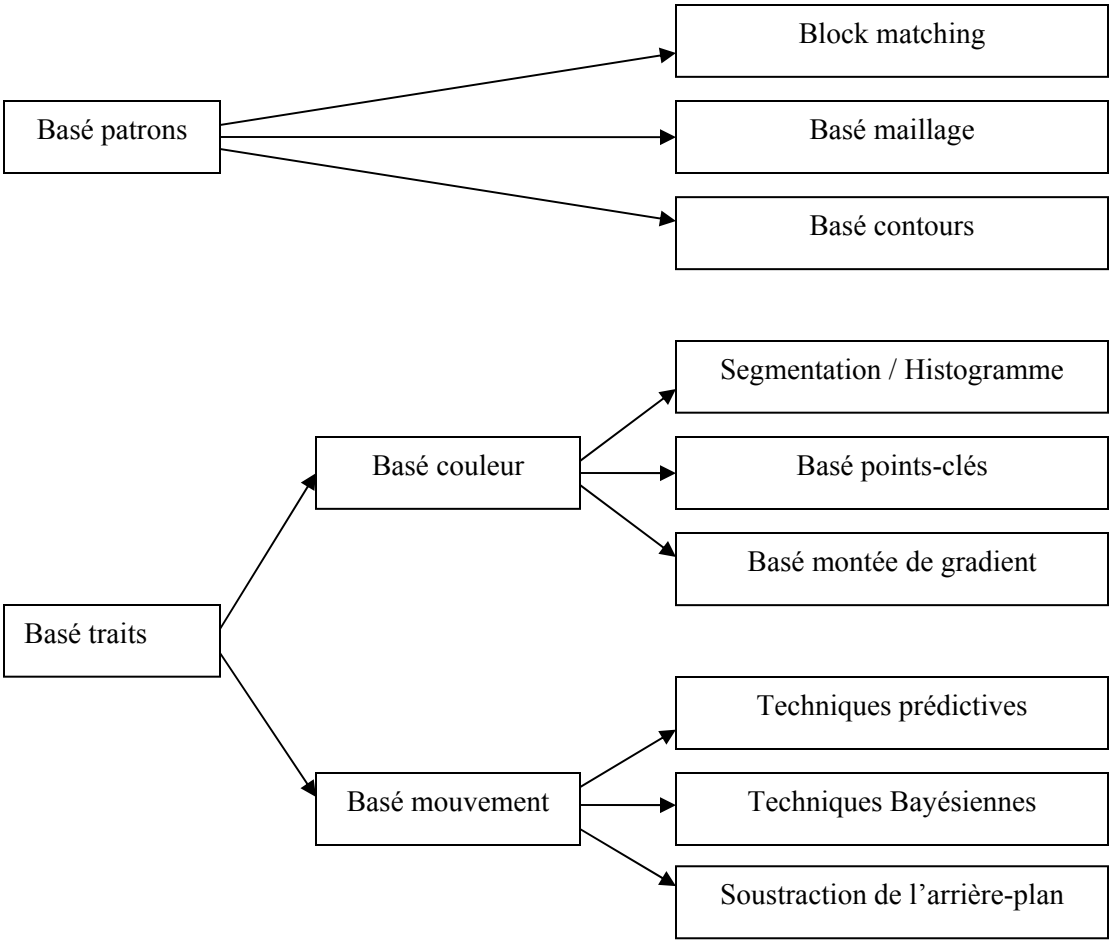


Figure 1: Une classification possible des techniques de suivi d'objet.

Les techniques de block matching **[Gya03][Har05]** procèdent en deux étapes. Elles commencent par isoler l'objet d'intérêt du fond en éliminant les régions animées du mouvement de la caméra. L'objet est ensuite découpé en blocs, des régions rectangulaires, que l'on va corréler d'une image à l'autre. Le procédé est illustré en Figure 2. Les résultats très rapides se payent par une faible tolérance à la plupart des difficultés et une localisation imprécise de l'objet.

Image initiale | Masque issu de la soustraction de l'arrière-plan | Corrélation des blocs

Figure 2: Déroulement d'un algorithme de bloc-matching **[Har05]**.

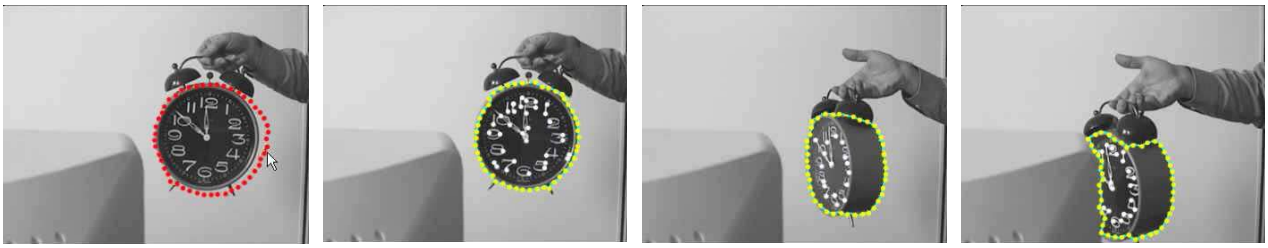
A l'inverse, les méthodes qui suivent les contours de l'objet **[Tec01][Tse03]** offrent une détection très précise de celui-ci. Elles utilisent des contours actifs ou « snakes ». Un snake V est un ensemble de *n* nœuds, ou points de contrôle $V=\{v_1,v_2,\ldots,v_n\}$ (donnant les coordonnées de points du contour) liés par une B-spline cubique. Pour une image I, la fonction d'énergie associée à cette courbe est la suivante :

$$E_{snake}(V) = \sum_{V_i \in I} E_{\text{int}erne}(v_i) + E_{externe}(v_i, I) + E_{contexte}(v_i, I)$$

L'énergie interne maintien la cohérence de la courbe. Elle contrôle les paramètres de raideur et de distance entre les points de contrôle du snake. Elle peut par exemple être modélisée par une mesure d'angle entre le point V_i et ses 2 voisins v_{i+1} et v_{i-1}. L'énergie externe gère l'adéquation aux données, autrement dit les caractéristiques de l'image. Puisque l'on recherche des contours, on pourra par exemple utiliser pour ce terme le gradient au voisinage du pixel V_i. L'énergie de contexte, encore appelée énergie de contrainte, représente les connaissances à priori qui sont possédées sur l'objet. Elle peut faire référence à une forme, une couleur, ou une intensité particulière de l'objet. Elle est fréquemment utilisée pour contrebalancer la tendance naturelle des snakes à se rétracter. La courbe optimale est celle qui minimise la fonction d'énergie $E_{snake}(V)$:

$$V_{opt} = \arg\min_{\{v_i\}} E_{snake}(V)$$

Cette famille de méthode nécessite, bien sûr, des contours hautement distincts. De plus, l'initialisation du snake est cruciale pour le bon déroulement de l'algorithme, le contour devant être initialement placé à proximité de l'optimal pour converger correctement. En conséquence, la méthode fonctionne très mal dans le cas d'un objet petit ou rapide et est perturbée par l'encombrement de l'arrière-plan ou les occultations, bien que des approches hybrides résolvent certains de ces problèmes **[Gou04]**. Un exemple est montré en Figure 3.

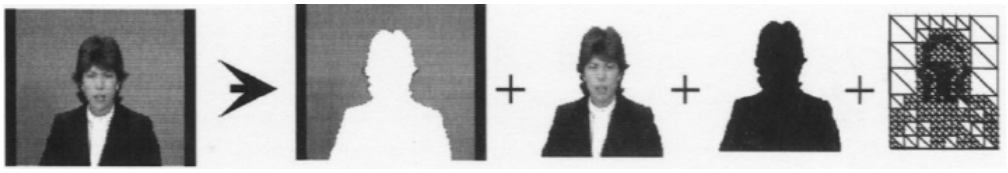


Initialisation suivi suivi suivi

Figure 3: exemple de suivi par snake **[Gou04].**

Les méthodes se basant sur un maillage sont un bon compromis, en termes de précision entre les deux précédentes. On distingue les maillages déformables **[Val04]** qui s'efforcent de suivre les changements de forme de l'objet des maillages non déformables **[Bad02]** qui corrèlent les régions de l'objet ainsi détectées (voir Figure 4). Elles offrent une bonne délimitation de l'objet à faible coût. Leur efficacité est directement dépendante du procédé de détection des nœuds.

Maillage non déformable



Maillage déformable

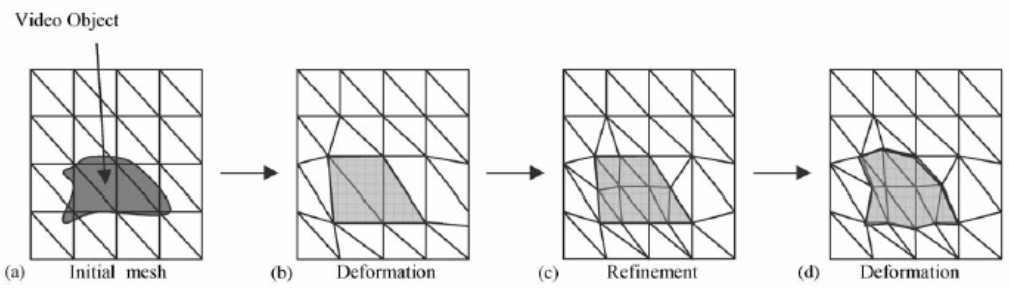
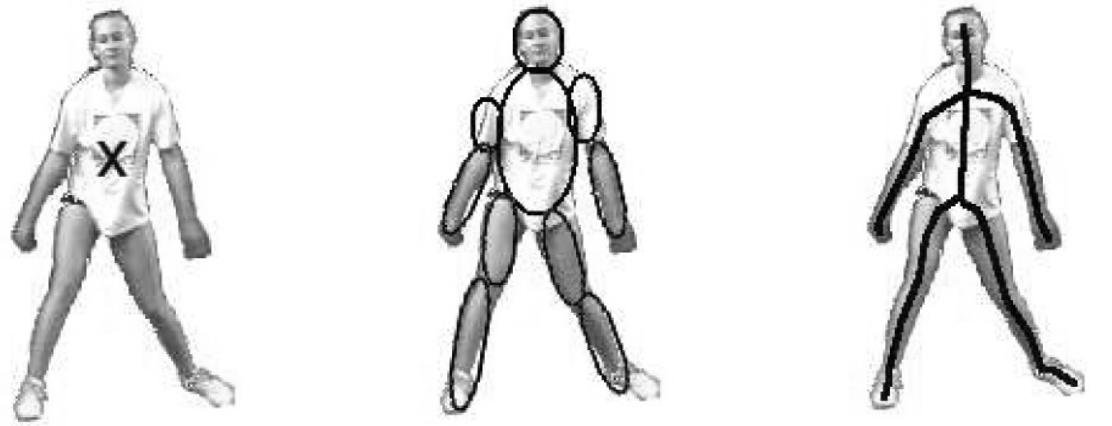


Figure 4: Illustration des methodes de maillage déformable et de maillage non déformable **[Bad02]**.

Le suivi de corps humain est une classe de famille qui cherche à capturer les larges mouvements de sujets pour une résolution donnée. Ce domaine ne recouvre pas les mouvements du corps à petite échelle comme les expressions faciales ou la gestuelle. Le corps humain est considéré dans son ensemble comme un unique objet et modélisé par une structure de squelette articulé ayant d'importants degrés de liberté. Ce type de méthode est dédié à des applications très spécifiques nécessitant une analyse précise de la cinématique corporelle telles que la reconnaissance du comportement, ou encore dans le cadre de diagnostiques médicaux ou l'analyse de performances des athlètes. Les difficultés d'un suivi de personne peuvent être estimées en termes de nombre de personnes présentent dans la scène, de nombre de croisement entre ces personnes, de la distance des sujets, et de la présence d'artéfacts nuisant à leur bonne détection (ombres, reflets, bruit,…). La majorité des méthodes se découpent en quatre parties : prédiction, synthèse, analyse d'image et estimation d'état. La tâche de prédiction considère les *n* états précédents pour augurer de la position du corps à l'état actuel. Elle permet l'intégration de connaissances dans le suivi. La composante de synthèse transfère la prédiction de la phase d'état à celui de mesure de l'image. Cette partie permet à l'analyse d'image de se focaliser sur un sous-ensemble de régions pour l'extraction d'un sous-ensemble de caractéristiques. Finalement, la phase d'estimation d'état calcule le nouvel état à partir des informations de l'image segmentée. Les modèles utilisés tirent avantage des connaissances anatomiques du corps humain. Les membres sont généralement représentés par des tiges rigides ou des blobs reliés par des joints. Le degrés de liberté de ces articulations est fonction des possibilités des membres dépeins, la difficulté résidant dans l'interprétation d'un mouvement 3D d'après des informations planaires (voir Figure 5). Les stratégies utilisées sont nombreuses **[Agg99][Sie04]** et apparaissent parfois dans des situations aussi variées que le suivi simultané de plusieurs personnes **[Zha02]** ou l'analyse de mouvement de foule **[Bey00]**.



Personne à modéliser | Modèle à base de blobs | Squelette de tiges rigides

Figure 5: modèles pour le suivi de personnes **[Yil06]**.

Les approches s'appuyant sur un trait particulier pour mener à bien leur suivi choisissent quasi systématiquement la couleur ou le mouvement. La couleur a l'avantage d'être une information immédiatement disponible donc facile à exploiter. Les méthodes exploitant les techniques de segmentation divisent l'image en régions homogènes selon un critère donné (qui peut être la couleur, la texture, la luminance…). Les techniques se distinguent en fonction de la primitive étudiée pour l'association selon le critère d'homogénéité choisi **[Ska94]**. On peut considérer les pixels, un choix populaire étant l'algorithme des nuées dynamiques flous **[Bez81]**. Une approche intéressante **[Shi00]** représente l'image au moyen d'un graph valué où les feuilles sont les pixels et les poids fonction de la similarité. La méthode divise alors le graph en n sous-graph disjoints correspondants aux n régions désirées. Les algorithmes par croissance de régions sont les plus connus. Partant d'un ensemble de pixels sélectionnés aléatoirement ou selon un critère donné, les régions s'étendent de proche en proche en fonction d'un critère d'homogénéité donné, et une étape ultérieure de fusion de régions regroupe les régions similaires afin d'obtenir un groupe de régions cohérent **[Mey92][Cav05]**. La dernière possibilité est de détecter directement les contours au moyen de divers opérateurs et de contraintes géométriques. Les histogrammes sont un autre type de méthode qui quantifient la couleur d'une région englobant l'objet. Certaines méthodes **[Qin05]** tenent de limiter l'absence de données spatiales en calculant un ensemble d'histogrammes sur un masque de régions. Toutes ces techniques fonctionnent en temps réel. Cependant, elles sont particulièrement sensibles aux changements d'illumination à cause de leur approximation de la colorimétrie de l'objet. Les occultations et déformations importantes de l'objet posent également problème en conséquence de la grossierté ou de l'absence d'information spatiale associée.

En revanche, Les méthodes de montée de gradients, et plus particulièrement le Mean-shift **[Com02][Jaf03]**, sont actuellement considérées comme les plus performantes. Le Mean-shift vise tout d'abord à construire une carte de similarité avec une couleur saillante de l'objet. La ressemblance locale entre l'objet et les différentes zones de l'image est évaluée par la distance de Bhattacharyya **[Bha43]** :

$$BC(p,q) = \sum_{x \in X} \sqrt{p(x)q(x)}$$

où p et q sont des distributions de probabilité sur le même domaine X. Cette ressemblance est ensuite modélisée par des Gaussiennes de probabilité. Puis, la carte de similarité est bâtie par sommation de ces *noyaux*. La position la plus probable de l'objet sera finalement obtenue par une montée de gradient depuis sa dernière position connue. Cette approche est particulièrement adaptée pour suivre les objets petits et rapides et gère efficacement les occultations. Toutefois, elle requiert une couleur saillante pour fonctionner correctement.

Les points-clés **[Har88][Mon98][Myk05]** constituent le troisième type d'approche exploitant un trait particulier. Ces points, enrichis de descripteurs locaux, sont localisés à des positions stratégiques de l'image, tel que des coins ou des extrema d'une fonction donnée. Ils font preuve d'une bonne robustesse aux transformations usuelles et sont adaptés à des situations où l'objet est confronté à des occultations partielles. L'objet est alors représenté comme un ensemble (fini ou non) de ces points. Toutefois leur calcul est coûteux et la stabilité du modèle problématique. En effet, l'objet est décrit par un ensemble de descripteurs locaux et non par un descripteur global. La distinction entre des points de l'objet entre eux et avec ceux du décor étant une tâche difficile, le mauvais appariement d'un faible nombre de points peut alors fausser le modèle et, à brève échéance, induire le système à suivre le décor ou un autre objet.

Le mouvement est un trait utilisé conjointement aux autres caractéristiques par la majorité des applications. Cependant, les techniques s'appuyant sur ce trait sont radicalement différentes et constituent une sous-famille distincte. Certaines méthodes dites « prédictives » analysent les déplacements antérieurs de l'objet pour augurer le prochain comme le célèbre filtre de Kalman **[May79]**. Elles reposent toutefois sur l'hypothèse d'une certaine fluidité du mouvement et échoueront à détecter un mouvement abrupte qui sera considéré comme incohérent. Les méthodes bayésiennes, quant à elles, évaluent les déplacements potentiels de l'objet en termes de probabilité. Elles font généralement appel aux champs de Markov aléatoires, ou encore aux filtres à particules **[Isa00][Pup05]**. La plupart de ces méthodes seront détaillées en **5.1.1**.

Les techniques de soustraction d'arrière-plan s'appuient sur un modèle d'une scène considérée comme globalement statique. Toute variation significative par rapport à ce modèle dans une région de l'image est interprétée comme un mouvement de l'objet d'intérêt (ou d'un objet) et renseigne donc sur la position (ou une position possible) de celui-ci. Afin d'avoir un modèle fiable et d'éviter le phénomène « d'écho » (illustré en Figure 6), l'arrière-plan est estimé d'après plusieurs images. Dans le système Pfinder **[Wre97]**, l'arrière-plan est représenté par la moyenne et la covariance de ses pixels qui sont récursivement mis à jour. Puisque de multiples couleurs peuvent être observées en un même point du décor à cause des effets d'ombre ou des mouvements périodiques de certains éléments de la scène (par exemple un arbre se balançant dans le vent), la modélisation de celle-ci avec une unique gaussienne n'est pas toujours un modèle pertinent ; particulièrement dans le cas de scènes d'extérieur. Une amélioration consistante dans le domaine fut donc l'utilisation de multiples gaussiennes **[Iva99][Sta00]** pour représenter l'arrière-plan, la décision de l'appartenance d'un pixel à l'objet ou au décor reposant typiquement sur la comparaison de celui-ci avec l'ensemble des gaussiennes du modèle. Une approche originale pour résoudre le problème de soustraction de l'arrière-plan a été proposée avec le système Wallflower **[Toy99]** qui analyse une image à trois niveaux : pixels, régions de couleur homogène et image. Un état de l'art plus étoffé sur la soustraction d'arrière-plan est donné en **[McI00]**. Cette famille de méthode a fait l'objet d'un récent regain d'intérêt dans le domaine qui nous intéresse en raison de sa grande robustesse aux changements d'illuminations et au bruit, de son aptitude à gérer les mouvements cycliques de l'arrière-plan, ainsi que de sa rapidité de calcul. Toutefois, et ce malgré les récentes tentatives pour pallier à cet inconvénient **[Zho03]**, une caméra statique est indispensable au bon fonctionnement du procédé. De plus, a majorité de la scène doit être immobile. La méthode est par exemple inapplicable dans l'éventualité de foules. Enfin le résultat est imprécis. Les algorithmes retournent généralement un ensemble disparate de régions duquel il faut extraire et inter-connecter les parties de l'objet. Ce type de technique est donc typiquement associé à d'autres traits dans le cadre du suivi d'objet.

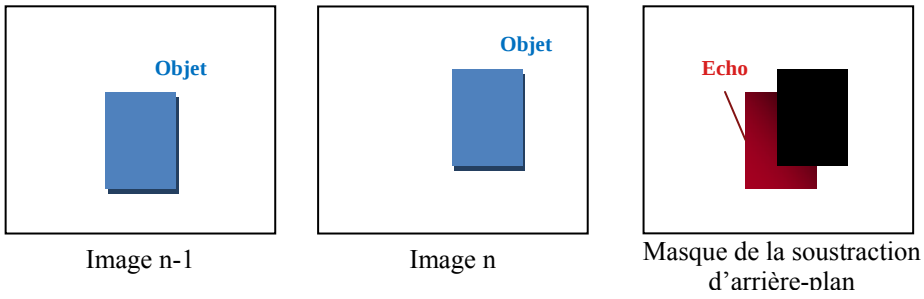


Figure 6: Illustration du phénomène d'écho.

Pour un état de l'art détaillé sur le suivi d'objet, consulter les travaux de Yilmaz **[Yil06]**.

2.3 Présentation de la méthode

2.3.1 Le projet porTiVity

Les plateformes mobiles comme les téléphones portables ou les PDAs ont acquis des capacités multimédia impressionnantes ces dernières années. La tendance, aiguillonnée par les compagnies de l'internet et des télécommunications est désormais à la télévision mobile. Mais la télévision mobile est également parfaitement adaptée pour l'interaction (manuelle). Elle a donc le potentiel pour être bien

davantage qu'une version miniaturisée de la télé traditionnelle. Mais quel type d'interaction est souhaitable pour ce type de dispositif ?

De précédents projets ont montré que l'interaction avec le contenu vidéo à l'échelle de l'objet était hautement attractive pour l'utilisateur **[GMF4iTV]**. D'autres ont enseigné que l'interactivité peut enrichir la diffusion de contenu **[SAMBITS]** et même être synchronisée avec un contenu additionnel fourni par d'autres canaux de distribution **[SAVANT].** Avec DVB-H **[DVB-H]** ou eDAB, une diffusion efficace des standards mobiles a été établie et UMTS permet des connections point à point vers les plateformes mobiles avec une bande passante rendant le streaming multimédia possible. Les capacités multimédia des dispositifs mobiles associées aux possibilités de diffusion vers des receveurs personnels ont permis de combiner les visions de ces projets pour réaliser quelque chose de nouveau : l'interaction directe avec des objets vidéo au travers de la diffusion de contenu vers des plateformes mobiles. En d'autres termes, un système télévision mobile basé sur l'interactivité avec des objets.

L'architecture du système porTiVity **[porTiVity]** est montrée en Figure 7. porTiVity ambitionne de réaliser une interactivité directe avec des objets en mouvement sur des plateformes mobiles (téléphones portables, pocket-PC…etc.). Le résultat est une scène interactive (réalisée en MPEG-4 LASeR **0**) permettant à l'utilisateur de cliquer sur des objets mis en surbrillance dans la vidéo afin de consulter l'information supplémentaire associée. Le contenu principal et la scène LASeR synchronisée sont envoyés au travers d'un premier canal (DVB-H) alors que le l'information supplémentaire est requise par l'utilisateur via un autre canal de diffusion (UMTS).

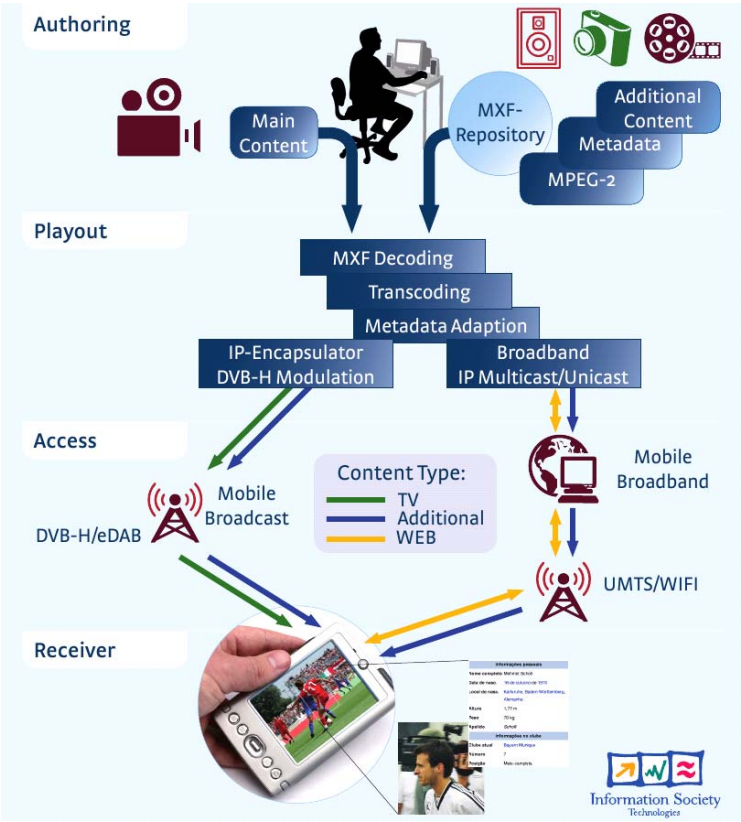


Figure 7: Architecture globale du système porTiVity.

Le système est divisé en trois composants :

1-Le système d'annotation est responsable du suivi des objets et de la création des scènes interactives LASeR (voir la Figure 8) qui rend les objets apparaissant sur la vidéo cliquables. L'annotation est effectuée en plusieurs étapes :

- La vidéo est automatiquement analysée par *l'outil d'analyse média*. Il extrait les métadonnées pour la navigation et l'agencement telles que les plans et les images-clés qui permettent ensuite une annotation semi-automatique rapide via le suivi d'objet et la redétection de plan ou d'objet. Plusieurs vidéos peuvent être analysées simultanément. Les métadonnées finales sont sauvées dans le format MPEG-7 **[MPEG-7]**.
- Les métadonnées extraites peuvent être visualisées et éditées à l'aide de *l'outil d'annotation vidéo*. La restructuration manuelle du contenu vidéo, par ajout d'information textuelle ou sémantique et annotation d'objets est possible. Le travail d'annotation peut être effectué de façon très efficace grâce aux fonctionnalités de suivi et de redétection d'objet, ainsi que la possibilité de réassigner les annotations déjà existantes. Les métadonnées produites sont également sauvées dans le format MPEG-7.
- Le service LASeR est ensuite généré par *l'outil d'application interactive* en se basant sur le fichier MPEG-7. L'interactivité est procurée en insérant des menus prédéfinis et des boutons de contrôle dans la vidéo, ainsi qu'en assignant de l'information supplémentaire aux objets mobiles. L'apparence de ces objets mobiles est également précisable. L'information supplémentaire peut être un lien vers une image, des données audio ou vidéo, une page HTML, ou une scène LASeR. Un script LASeR permettant des fonctionnalités telles que la personnalisation du contenu ou des jeux est aussi réalisable.

2-Le système d'affichage : le *système d'annotation* est connecté au *système d'affichage* au moyen de fichiers MXF contenant des AV multiplexés, l'information supplémentaire, et des métadonnées synchronisées.

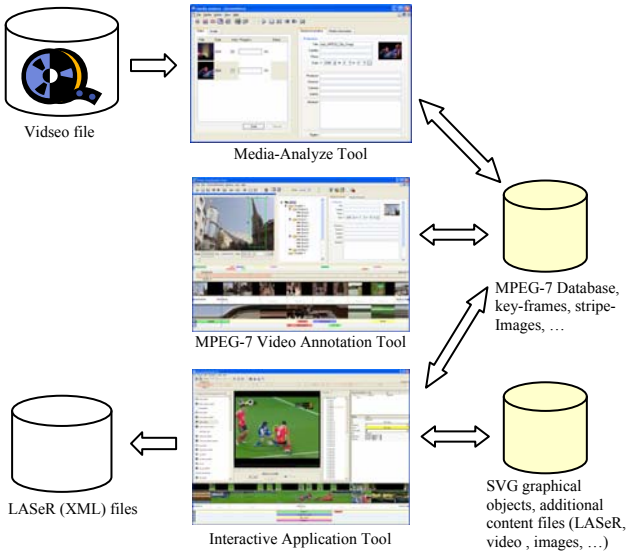


Figure 8: Architecture du sytème d'annotation.

3-Les terminaux : l'information y est alors adaptée aux différents canaux de distribution et il est assuré que la synchronisation entre les différentes parties du contenu soit respectée lors de l'envoi vers les

terminaux. Les terminaux affichent enfin la scène LASeR et implémentent les fonctions vitales comme la fonction "time shift".

Un exemple de service utilisant la technologie porTiVity pour améliorer et transformer la diffusion télévisuelle traditionnelle et montrer le potentiel innovatif du projet est la vidéo « Spur & Partner » présentée à IFA (*Internationale Funkausstellung*, Berlin), *Medientage* (Munich) et IBC (International Broadcast Congress, Amsterdam). Cette vidéo fait partie d'une série policière pour enfants produit par la compagnie de télévision publique allemande ARD. La variante mobile de porTiVity montre une interaction simple et intuitive (voir Figure 9). En cliquant les objets en surbrillance, les enfants peuvent récolter des indices, répondre à des questions, pour finalement trouver le coupable. Ils peuvent également récupérer une solution vidéo sur demande, ou accéder à des pages web, comme un magazine de détective interactif.

Figure 9: Capture d'écran de la vidéo intéractive « Spur & Partner ».

2.3.2 Contraintes inhérentes au projet

La participation de l'institut Eurecom à ce projet concerne exclusivement le développement d'un outil de suivi d'objet dans le contexte d'un système de télévision interactive. Dans ce système, un producteur vidéo annotera un programme vidéo en définissant des objets vidéos dans la séquence, et en leur attachant de l'information (par exemple, du texte, des images, des références web, une vidéo,…etc.). Il pourra alors effectuer un suivi automatique de l'objet défini afin d'éviter les annotations inutiles des autres occurrences du même objet dans le plan. Au niveau de la réception, l'utilisateur regardant le programme vidéo pourra sélectionner les objets vidéo, et immédiatement accéder à l'information correspondante. Cet environnement implique certaines contraintes spécifiques pour le suivi d'objet :

- Pour que ce schéma soit réalisable, le coût de production supplémentaire dû à l'annotation manuelle doit être minimisé, afin que celle-ci soit aussi rapide que possible. L'idée est que le producteur identifie un objet à sa première occurrence, et qu'un système de suivi localise celui-ci pour le reste de la séquence. Puisque le producteur doit vérifier la validité du suivi, le système de suivi doit être **aussi rapide que possible** (et si possible plus rapide que le temps réel).
- Ce système doit être utilisable pour tout type de programme vidéo, le système de suivi ne doit donc pas être contraint à un type de vidéo particulier, ou à des caractéristiques spécifiques de l'objet suivi. Il doit reposer sur des techniques **génériques**, ou doit pouvoir s'adapter à la vidéo à annoter.
- Autant pour l'annotation que pour l'affichage à l'utilisateur, les objets vidéo doivent être identifiés par des **boîtes englobantes**. Cela limite le raffinement de la description de la boîte englobante, et impose son bon positionnement comme critère de performance du système de suivi. Et, bien que cette représentation de l'objet soit grossière, la précision de sa localisation reste évidemment une priorité.

Ces contraintes, et en particulier la **généricité** imposée ont déterminé nos choix tout au long du développement de notre système de suivi d'objet.

2.3.3 Justification du choix de la méthode

Notre approche est basée sur l'identification de points-clés dans la scène vidéo. Ces points, détaillés dans les deux chapitres suivants offrent de nombreux avantages en réponse aux contraintes de notre application. De par leur robustesse aux transformations usuelles, ils sont un outil fiable pour le problème d'un suivi générique. De plus, leur calcul est indépendant de la position de l'objet, de sorte que le calcul et l'appariement des points peut être fait off-line, laissant uniquement un minimum de calcul à faire pendant la session d'annotation.

2.3.4 Etapes du suivi

Notre système de suivi d'objet modélise donc l'objet grâce à un ensemble de points-clés. Le modèle est initialisé avec les points compris dans la boîte englobante de la première image. Par la suite, les points-clés extraits à chaque nouvelle image sont appareillés avec ceux du modèle. La boîte englobante est alors repositionnée en fonction du mouvement global de l'ensemble des points appariés. Enfin, le modèle est mis à jour. Lors de cette étape, les descripteurs des points appareillés du modèle sont remplacés par les descripteurs de leur point correspondant dans l'image, et les nouveaux points sont ajoutés au modèle. Un point de l'image est considéré comme nouveau si aucune correspondance ne lui est trouvée dans le modèle. Le système peut être résumé par les étapes suivantes communes à tout système de suivi d'objet :

Initialisation :
-extraction des traits pour la première image *(off-line)*
-tracé de la boîte englobante

Boucle Principale :
-extraction des traits *(off-line)*
-appariement avec le modèle *(off-line)*
-repositionnement de la boîte englobante
-mise à jour du modèle

Il est important de signaler que la majorité des opérations mentionnées ici peuvent être effectuées avant la session d'annotation. Les points-clés peuvent être calculés sur la totalité de scène pour chaque image (puisque l'on ignore pour l'instant quels objets seront annotés) et un appariement effectué entre les images consécutives. Les résultats seront ensuite stockés afin que durant la phase d'annotation, le seul travail restant soit le repositionnement de la boîte englobante et la mise à jour du modèle. Cette organisation des tâches répond aux prérequis pour un système de suivi rapide et fait des points-clés un outil privilégié pour le suivi dédié à l'annotation vidéo. Après avoir défini notre système d'évaluation, le reste de ce manuscrit s'articulera autours de ces grandes étapes d'un suivi d'objet.

2.4 Algorithmes de référence et système d'évaluation

2.4.1 Outils d'évaluation

Evaluer l'exactitude d'un suivi consiste à comparer deux patrons : la localisation de l'objet estimée par l'algorithme et une « vérité terrain ». Dans la pratique, ces patrons peuvent être une boîte englobante ou un masque des pixels appartenant à l'objet.

On distingue deux types de métriques. La première approche mesure la distance entre deux patrons et permet de définir la similarité entre deux masques X et Y du point de vue de leur recouvrement. Ce type de métrique se base fondamentalement sur trois ensembles de pixels qu'elle s'efforce de résumer :

- Les vrais positifs : les pixels de l'objet que l'algorithme identifie correctement.
- Les faux positifs : les pixels de l'objet identifiés comme étant ceux du décor.
- Les faux négatifs : les pixels du décor que l'algorithme considère comme faisant partie de l'objet.

La valeur des pixels présents n'est pas prise en compte. On utilise généralement la distance :

$$d_1(X,Y) = \frac{X \cap Y}{X \cup Y}$$

Toutefois la **distance de Chamfer** est aussi parfois utilisée :

$$d_2(X,Y) = \frac{1}{|X|}\sum_{x \in X} \min_{y \in Y} d(x,y)$$

Elle permet une meilleure précision dans le cas d'un recouvrement nul ou faible, comme le montre la Figure 10. En effet, et contrairement à la métrique précédente, la distance des pixels du masque retourné par l'algorithme au plus proche pixel de la « vérité terrain » sera ici prise en compte. Cela permet d'appréhender les notions de formes identiques et de proximité des centres, que la mesure précédente ne gère pas. Par contre, cette métrique est minimale dans le cas ou la forme Y est incluse dans la forme X. Le calcul de *max {d(X,Y) , d(Y,X)}* permet de pallier à ce désagrément. Toutefois, le résultat étant une distance et non un pourcentage, il est plus malaisé à interpréter. Pour convertir cette distance en pourcentage, il convient donc de définir une distance maximale D au delà de laquelle le pourcentage est nul. On a alors :

$$d_3(X,Y) = 1 - \frac{d_2(X,Y)}{D}$$

On choisit généralement D comme étant la plus grande distance possible entre le centre et un pixel du masque (voir Figure 11). Il faut cependant noter que cette conversion se fait au prix de la précision des résultats, qui seront d'autant plus dégradés que la hauteur et la largeur sont disproportionnées. Ce fait conduit à s'interroger sur l'utilisation de cette métrique puisque son avantage est une meilleure précision qu'il faudra ensuite dégrader si l'on désire pouvoir aisément interpréter les résultats.

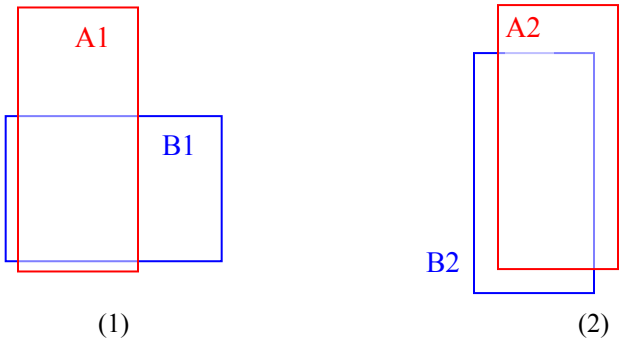


Figure 10: Soit A la vérité terrain et B l'estimation d'un algorithme. D'après la métrique d1 d1(A1,B1)=d1(A2,B2), alors que, subjectivement, la deuxième estimation est plus précise. La distance de Chamfer prend en compte ce cas de figure.

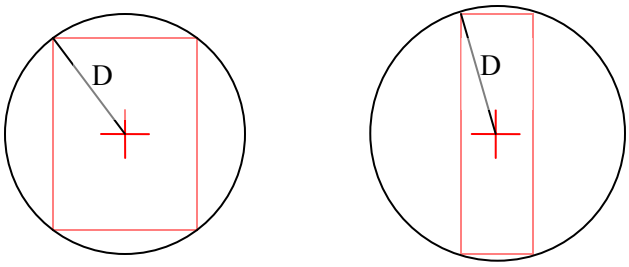


Figure 11: distance D maximale entre le centre de la boîte englobante et un pixel de celle-ci.

Afin de prendre en compte les notions recouvertes par la distance de Chamfer sans avoir souffrir des inconvénients précités, nous avons proposé notre propre mesure. Il s'agit d'une amélioration de la distance classique d_1 proposée au début, prenant en compte la distance entre les centres C_X et C_Y des deux boîtes englobantes. La formule est la suivante :

$$d_4(X,Y) = d_1(X,Y) \times \left(1 - \frac{K \times d(C_X, C_Y)}{2D}\right)$$

,avec $d(x,y)$ la distance euclidienne et K ($0 < K \leq \infty$) l'importance accordée à cette notion de distance des centres. Plus K sera élevé, moins l'éloignement entre les centres influera dans la mesure. Notons que :

$$\forall X, Y \quad d_4(X,Y) \leq d_1(X,Y)$$

Cette mesure est un outil pleinement satisfaisant pour comparer des algorithmes de suivi, car elle modélise plusieurs critères de comparaison et reste facile à interpréter. Elle est utilisée pour toutes les évaluations présentées dans ce manuscrit.

La seconde approche estime la précision d'une trajectoire. La méthode classique consiste à mesurer à chaque image la distance entre les centres des masques. Une moyenne est alors calculée sur les n images de la séquence étudiée pour obtenir une estimation globale de la qualité de son suivi. Afin de ne pas fournir des résultats trop aléatoires, cette méthode considère généralement que les patrons suivis sont de taille constante. Une quantification plus précise de la variation globale entre deux trajectoires a été proposée par Needham **[Nee03]**. Dans ses travaux, une variation temporelle et spatiale entre les trajectoires dans leur ensemble est recherchée avant d'étudier les différences entre les centres deux à deux.

Plutôt que résumer l'information, certains auteurs, tels que Schlögl **[Sch04]**, choisissent de présenter des résultats exhaustifs où le suivi est décrit en fonction des faux positifs et négatifs, des vrais positifs et de la précision de la trajectoire. Le lecteur est alors libre d'estimer la qualité de l'algorithme étudié selon une combinaison idiosyncrasique des critères d'évaluation fournis. Certains considéreront toutefois cette option comme une surcharge d'information.

Au delà du choix d'une métrique appropriée se pose le problème subjectif de la « vérité terrain » utilisée. Par exemple, « une boîte englobante doit-elle comprendre tous les pixels de l'objet ou alors maximiser le pourcentage de pixels de celui-ci par rapport à ceux de son environnement ? », ou « Sa taille sera-t-elle variable ? » sont des questions à se poser avant toute évaluation. Parallèlement, survient la tâche fastidieuse consistant à définir la vérité terrain. Bien que des logiciels tels que ODVis **[Doe00]**, VIPER **[Jay02]**, ou les outils des interfaces VIVID **[Col05]** ou VISOR **[Vez08]** aient été créés afin de simplifier et d'accélérer le processus, de considérables efforts sont tout de même requis. Black **[Bla03]** propose un palliatif original, avec son système de création automatique de « vérité terrain » qui sélectionne des boîtes englobantes fiables en se basant sur le mouvement et interpole les masques restants. Ce système fut cependant conçu dans le cadre de la surveillance et est donc restreint à des humains dans une zone dégagée, pour une caméra fixe.

Dans le reste de ce manuscrit, la qualité d'un suivi d'objet sur une séquence vidéo à l'image *t* sera donc évaluée comme le recouvrement moyen entre la vérité terrain et la boîte englobante retournée par l'algorithme testé, calculée sur les *t* précédentes images. Pour chaque vidéo, plusieurs mesures seront effectuées à intervalles de temps réguliers. Le gradient entre deux mesures sera donc aussi important que la qualité globale puisqu'il permettra d'inférer la perte de l'objet ou non.

2.4.2 Jeu d'essai

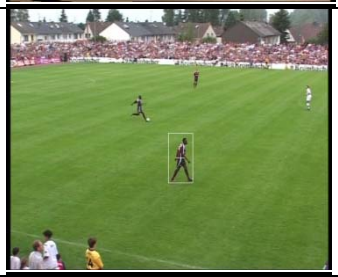
Dans le cadre d'un système de suivi générique, il était important de bâtir un jeu de vidéos tests recouvrant un éventail de difficultés et d'applications le plus vaste possible. C'est dans cette optique que nous avons choisi nos séquences. De plus, nous nous sommes concentré sur des séquences courtes et assez difficiles qui sont les plus susceptibles de départager les algorithmes et de mettre en valeur les apports de leurs variantes. Les séquences vidéos utilisées sont au format MPEG2, soit d'une résolution maximale de 720×576 pixels. Leur description et leur taille sont présentées en Table 1. Quelques images sont fournies en Table 2 afin d'en donner au lecteur une meilleure représentation.

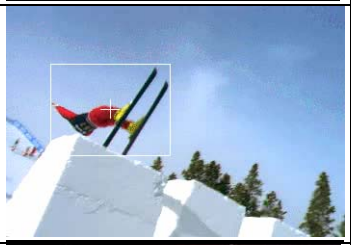
Table 1: Description du Jeu d'essai de vidéos utilisé dans le cadre du développement d'un suivi générique.

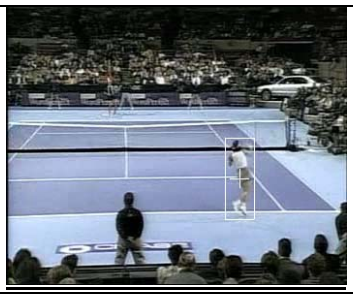
Name	Longueur (images)	Taille moyenne de l'objet (pixels)	Difficultés	Description
Fashion	120	540 × 150	Un flash.	Une femme approchant de la caméra, et se retournant.
Soccer	120	100 × 40	Deformations de l'objet.	Un joueur de foot driblant.
Cooking	60	150 × 170	Changement d'échelle, arrière-plan encombré.	Caméra tournant autour et se rapprochant d'une marmitte.
Jellifish	30	150 × 130	Déformations importantes de l'objet, faible contraste.	Une méduse nageant. Lumière jaune.
Frying pan	75	90 × 220	Arrière-plan encombré.	Un cuistot montrant une poêle.
Bottle	60	120 × 60	Occultation, arrière-plan encombré, rotation.	Une bouteille passant de main en main.
Surveillance	90	110 × 40	déformations de l'objet, arrière-plan très encombré.	Un homme traversant un couloir et déposant une malette.
Coast guard	120	30 × 100	Arrière-plan encombré et mobile (eau), objet sortant de l'écran, un tremblement de la caméra.	Deux bateaux se croisant sur un courts d'eau.
Skijump	90	100 × 100	Déformations de l'objet, mouvement intra-objet complexe.	Un skieur sautant et virevoltant.
Skicross	90	90 × 40	Occultations partielles, objet rapide.	Deux skieurs slalomant.
Picture	75	125 × 80	Objet en partie hors de l'écran.	Un détective montrant un dessin abstrait.
Tennis	240	80 × 30	Occultations partielles, objet petit, rapide et déformé.	Un match de tennis.
Foot from above	120	60 × 30	Mouvement de caméra et d'objet rapides, flou, objet petit et déformé.	Un match de foot vu du ciel.
Walking	60	105 × 45	Faible contraste, courtes occultations à 80%.	Deux piétons marchant sur le trottoir d'en face.
Cognac	30	100 × 50	Mouvement rapide et irrégulier,	Une femme agitant un

			objet flou, occultation.	flacon de Cognac.
Traffic in bombay	120	20 × 30	Objet minuscule, nombreux autres véhicules.	Traffic vu du ciel
Domatica	120	180 × 60	Occultation longue et 95% complète.	Un homme passant derrière une pile de caisses.

Table 2: Illustration du Jeu d'essai de vidéos utilisées dans le cadre du développement d'un suivi générique.

Name	Début	Milieu	Fin
Fashion			
Soccer			
Cooking			
Jellifish			
Frying pan			

Bottle			
Surveillance			
Coast guard			
Skijump			
Skicross			
Picture			

Tennis			
Foot from above			
Walking			
Cognac			
Traffic in bombay			
Domatica			

2.4.3 Algorithmes de référence

Nous avons comparé les performances de notre algorithme au Meanshift **[Com02]** présenté en **2.2** et à l'algorithme de Gabriel **[Gab05]**. Le premier est actuellement considéré comme le plus performant, particulièrement dans le cas d'objets petits et rapides dont la couleur est discriminante par rapport à l'arrière-plan. Il est très répandu dans la littérature et sert de référence à de nombreux algorithmes. Nous avons choisi la seconde méthode car l'approche est similaire à la nôtre (suivi d'un ensemble de points de Harris). Elle est détaillée dans le chapitre **6.1.2.5**.

3 Caractérisation des points d'intérêt

L'un des problèmes majeurs dans la plupart des systèmes de vision est l'extraction d'indices visuels significatifs dans les images ou les vidéos. Les points d'intérêts sont des points caractéristiques d'une image particulièrement riches en termes d'information qui répondent efficacement à cette problématique. Une centaine de points d'intérêts peuvent efficacement décrire les zones visuellement saillantes d'une image. Ceux-ci sont généralement calculés à des positions clés de l'image (coins, minimums ou maximums locaux,…), et complétés par des descripteurs d'image locaux, les rendant robustes aux transformations classiques (changements d'illumination, d'échelle, transformations affines, bruit,…) afin de mieux les identifier ultérieurement.

Dans le contexte de l'analyse d'image, ils offrent donc un large éventail de possibilités : résumer une image, caractériser efficacement une région tout en évitant les problèmes d'une segmentation, pouvoir décrire un objet d'une façon robuste aux occultations, etc. Cela fait des points d'intérêt un outil de plus en plus apprécié. A l'origine développés pour la robotique **[Mor80]**, leur utilisation est maintenant courante dans le cadre de la reconnaissance d'objet, de la recherche d'image par le contenu, de l'indexation ou du suivi d'objet dans les vidéos. Mais en conséquence de cet engouement, s'est développée une terminologie qui n'est pas toujours bien cloisonnée. Le but de ce chapitre est donc de définir ce que sont les points d'intérêts. Nous aborderons également les caractéristiques utilisées pour décrire et comparer les détecteurs ainsi que les mesures associées permettant d'en évaluer ces caractéristiques et leurs performances.

3.1 Définition des points d'intérêt

Tout d'abord, et afin d'éviter toute ambiguïté, il convient de définir ce qu'englobe le terme de *points d'intérêt* ou *points-clés* qui sera utilisé tout au long de ce manuscrit. En effet, un large panel de définitions, souvent contradictoires, sont utilisées dans l'état de l'art, et des termes spécifiques sont souvent employés en fonction de l'usage auxquels sont destinés les points extraits. Ici le terme de *points d'intérêt* ou *points-clés* sera utilisé pour tous points caractéristiques extraits d'une image, quelque soit l'utilisation pour laquelle il est prévu ou ses spécificités. De tels points sont supposés hautement distincts de leur voisinage et riches en terme d'information. D'un point de vue général, un détecteur de points d'intérêt peut être décrit par les deux critères suivants :

-La ***répétabilité*** : le détecteur est dit répétable si un même point est détecté dans une série d'images.
-la ***robustesse*** : La capacité d'un détecteur à résister aux transformations classiques qui sont le bruit, les effets de flou, les changements d'illumination, les translations, les rotations, les changements d'échelle et de point de vue.

Ces deux caractéristiques sont bien sûr liées dans la mesure où la répétabilité d'un détecteur chutera pour une transformation à laquelle il n'est pas robuste. Le taux de répétabilité est donc généralement exprimé pour une transformation donnée. Elles seront décrites de façon plus poussée dans les paragraphes suivant. Pour un état de l'art plus précis sur ces caractéristiques, on pourra consulter par exmple la thèse de **V. Gouet [Gou00]**.

A chaque point est conventionnellement associé une *région* qui sera le support spatial du descripteur le représentant. Par le terme *région*, nous définissons ici un voisinage centré sur le point d'intérêt dont les pixels n'ont pas nécessairement un trait commun, comme ce serait le cas dans une segmentation par exemple. Tout comme pour le point d'intérêt, la forme de sa région associée ne doit pas être altérée par les diverses transformations que l'image peut subir. En effet, si deux régions environnant

un même point ne recouvrent pas la même partie d'un objet, les descripteurs issus de ces régions ne pourront être identiques.

Pour résumer, les points d'intérêt offrent donc de nombreux avantages. Il est évident qu'une information compacte est plus facile à analyser, mais les points d'intérêt font plus que réduire l'information à l'essentiel : par leur faculté à être facilement détectés dans diverses situations, ils permettent une analyse où la majorité des fausses alarmes (les informations susceptibles de conduire à une conclusion erronée) sont éliminées des données. En cela, ils fournissent un outil fiable pour de nombreuses tâches de vision.

3.2 Répétabilité

Cette mesure représente la stabilité d'un point, et fut développée par **Schmid [Sch97]** dans sa thèse pour mettre en évidence la fiabilité du détecteur Harris. Sa définition fut la suivante : Soient deux images I_i et I_j d'une même scène 3D et M_i et M_j les matrices de projection correspondantes. La détection des points image p_i et p_j appartenant respectivement aux images I_i et I_j est parfaitement répétable si et seulement si il existe un point P de la scène tel que :

$$P_i = M_i P \text{ et } P_j = M_j P$$

Par exemple, si I_i et I_j sont les images d'une scène plane, il existe une homographie H_{ij} telle que dans le cas d'une répétabilité parfaite on ait :

$$p_j = H_{ij} \cdot p_i$$

Dans le cas général, mesurer la répétabilité d'un détecteur consiste donc à établir une relation entre p_i et p_j. Et le fait, qu'en principe, il n'en existe pas (il faut connaître les matrices de projection M_i et M_j) la rend difficile à estimer.

De plus, dans la pratique, suite aux approximations de la méthode, la répétabilité d'un détecteur est rarement parfaite. L'homologue p_j d'un point p_i devra donc être recherché dans un certain voisinage. Ce qui nous amène à considérer la notion de ***précision de la localisation*** des points d'intérêt pour un détecteur donné. Cette notion est distincte de celle de répétabilité. En effet, un détecteur ayant un taux de répétabilité élevé peut avoir une localisation imprécise et inversement. Toutefois, ces deux grandeurs sont généralement mesurées conjointement. La mesure de la répétabilité conventionnellement utilisée est celle proposée par **[Mik01, Mik02]** :

$$taux \quad de \quad répétabilité = \frac{nombre \quad de \quad correspondances}{nombre \quad de \quad correspondances \quad possibles}$$

Dans le cas de régions plutôt que de points, cette mesure est multipliée par le pourcentage de recouvrement moyen des surfaces.

3.3 Robustesse aux transformations usuelles

Afin d'être fiable, un détecteur doit être robuste à la plupart des transformations usuelles qui sont le bruit, les effets de flou, les changements d'illumination, les translations, les rotations, les changements d'échelle, et enfin les changements de points de vue. On prévient parfois le bruit ou les effets de flou par un prétraitement (filtre passe-haut ou passe-bas). Afin de rendre un détecteur robuste aux autres transformations, les points sélectionnés doivent avoir certaines caractéristiques particulières qui resteront invariantes malgré les changements en question. Leur principe est le suivant :

Robustesse aux changements d'illumination : Les variations d'illuminations peuvent avoir plusieurs origines telles que l'imprécision ou la défaillance des capteurs, des images prises sous des points de vue ou des éclairages différents. Que ce soit dans le cadre de la reconnaissance aussi bien que

du suivi d'objet, la confrontation à ce problème est très fréquente. La robustesse du détecteur à ces modifications est donc indispensable. Le principe consiste à sélectionner des points qui présentent des spécificités locales qui resteront présentes malgré des variations d'illumination. Il peut s'agir d'extremum locaux ou encore de pixels de contours (les coins). Malheureusement, si ces points sont toujours détectés lors de variations globales de la luminosité, certains détecteurs échouent dans le cas d'une variation locale.

Robustesse à la rotation : Tout comme pour les changements d'illumination les extrema locaux et les coins présentent des caractéristiques invariantes à la rotation. Ils constituent donc un choix privilégié pour résister à cette transformation.

Robustesse aux changements d'échelle : Le problème occasionné par le changement d'échelle ou de point de vue peut être assimilé à une modification du voisinage du point d'intérêt. Le détecteur ne se basera donc plus sur les mêmes informations pour déterminer si la région centrée sur le point traité est suffisamment discriminante pour que celui-ci soit jugé point d'intérêt. Si le détecteur n'est pas robuste à de telles transformations, la différence entre deux ensembles de points issus de ce détecteur sera proportionnelle à la magnitude de la transformation.

De même que l'on peut assimiler l'extraction de points susceptibles de résister aux changements d'illumination ou aux rotations comme une recherche d'extrema locaux sur une fonction I dépendant de l'intensité (typiquement I étant la fonction d'intensité ou la dérivée de l'intensité), on peut considérer la recherche de points invariants aux changements d'échelle comme la recherche d'extrema locaux sur une fonction f de l'espace-échelle. En pratique, cela correspond à calculer le même point d'intérêt pour plusieurs échelles distinctes puis à conserver celui pour lequel la fonction f définie par la méthode admet un maximum local (Figure 12). Le choix de la fonction f est important. Le pic doit être prononcé et de préférence unique pour un choix de l'échelle caractéristique fiable (Figure 13).

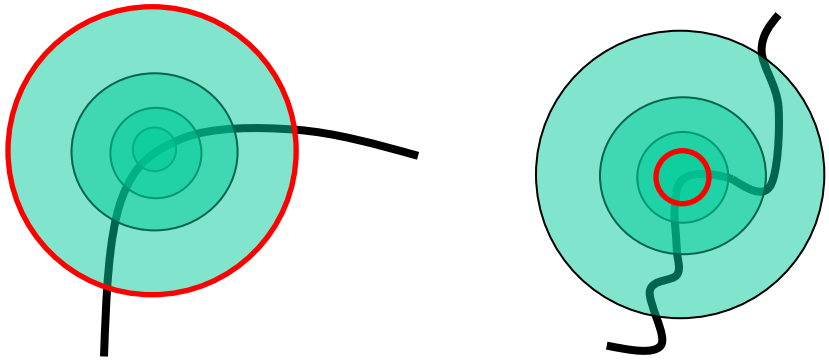
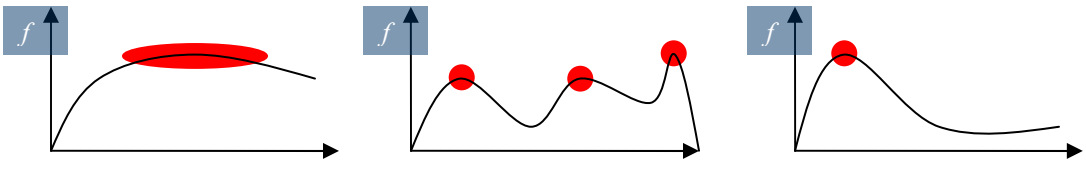


Figure 12: Sélection d'échelle caractéristique pour un même point à 2 échelles différentes.



(a) (a) (b)

Figure 13: Illustration de l'importance du choix de la fonction f descriptive de l'espace échelle (a) fonction mal choisie (b) fonction bien choisie.

Robustesse aux changements de point de vue : Dans le cas d'un changement de point de vue, le changement d'échelle n'est pas uniforme mais différent dans chaque direction. Les algorithmes utilisés pour le changement d'échelle, considérant qu'une telle transformation est uniforme, ne sont plus adaptés (Figure 14: (a) Région circulaire sur une image de référence (b) Région circulaire sur le même point de la même image après un changement de point de vue. La zone recouverte est différente (c) Région elliptique recouvrant la même zone que celle de l'image de référence.

). Deux approches sont alors possibles. La première consiste à caractériser l'image par des régions comportant des spécificités identifiables quelque soit le point de vue. La région ainsi extraite est ensuite souvent ramenée à une ellipse, les calculs ultérieurs étant plus aisés s'ils se basent sur un support de forme élémentaire. Rappelons que toute région peut être approximée par une ellipse grâce à ses moments d'ordre 2. La deuxième méthode fonctionne de manière similaire à la recherche d'échelle caractéristique. De même que déterminer une échelle invariante pour une fonction donnée revient à déterminer une région circulaire, on essaiera ici d'identifier une ellipse spécifique au voisinage du point.

(a) (b) (c)

Figure 14: (a) Région circulaire sur une image de référence (b) Région circulaire sur le même point de la même image après un changement de point de vue. La zone recouverte est différente (c) Région elliptique recouvrant la même zone que celle de l'image de référence.

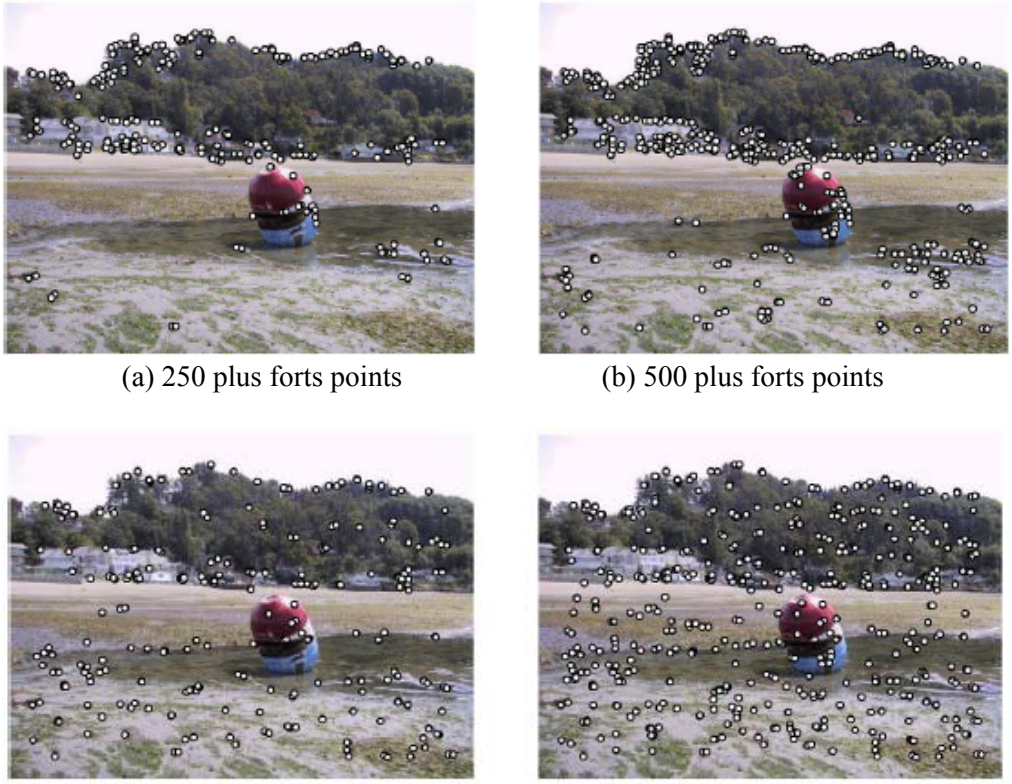
3.4 Caractérisation des points-clés pour le suivi d'objet

En surcroit des possibilités inhérentes des points d'intérêts énoncées dans le chapitre précédent, certaines caractéristiques ayant trait au nuage de points sont requises dans le cadre spécifique de leur emploi pour le suivi d'objet. Ces caractéristiques sont la densité, la répartition et la fiabilité des points. Leur paramétrisation est, bien sûr, interdépendante et fonction du type d'objet suivi et de l'application désirée. Leurs effets sur le suivi d'objet sont décrits dans cette partie.

Densité des points : La densité de points au voisinage d'un objet d'intérêt nécessaire au bon fonctionnement de son suivi sera directement dépendante de la complexité du suivi et du type d'application. Par exemple, un suivi de fluide va requérir une densité de points-clés beaucoup plus élevée que pour un suivi de véhicule sur l'autoroute. S'il est donc difficile d'établir une loi fiable sur le sujet, certaines règles restent toutefois de mise. Tout d'abord, une densité minimum de points sera toujours requise d'une part pour assurer un l'apport minimum vital d'information à l'algorithme, mais aussi, à défaut de garantir une répartition homogène des points, pour augmenter les chances qu'au moins un point soit présent dans chaque zone de l'objet. Ensuite, la fiabilité de l'information étant toujours préférable à sa quantité, l'accroissement de la densité des points, ne devra pas se faire au prix de sa validité. Enfin, nos expériences dans ce domaine ont montré que, passé un certain seuil, la densité de points n'apportait plus rien au suivi. Autrement

dit, une surcharge d'information sera inutile, voire nuisible au bon fonctionnement de l'algorithme.

Répartition des points : Une répartition naturelle des points d'intérêts conduira à une répartition non-uniforme du nuage, les points se massant dans les zones à forte variation locale et fuyant les parties homogènes de l'image (voir Figure 15). Cette répartition reflète généralement l'intérêt visuel de la scène. Cet agencement est usuellement préférable pour le suivi d'objet puisqu'il conduit à une forte densité de points sur l'objet. Toutefois, une répartition homogène des points recèle également des avantages. Deux méthodes permettent l'obtention d'une telle structure du nuage de point : découper l'image en blocs et ne conserver que les *n* meilleurs points par blocs (selon un critère donné), ou alors forcer un voisinage vierge de points de rayon *r* autour de chaque candidat retenu. Cet agencement réduit au minimum les ambigüités lors de l'appariement des points. Toutefois, il n'existe pas de garanti, que le point sélectionné comme représentant de son voisinage ait une forte répétabilité. Autrement dit, le meilleur point-clé du voisinage ne sera pas nécessairement toujours le même. Donc, cette stratégie diminue la quantité d'information au voisinage de l'objet sans pour autant en accroître de façon conséquente la fiabilité. Elle sera donc préférable dans le cadre d'une application où une répartition homogène des points-clés est plus clairement souhaitée, telle que l'indexation d'image par exemple.



(a) 250 plus forts points (b) 500 plus forts points

(c) 250 plus forts point, r = 24 (d) 500 plus forts points, r = 16

Figure 15: (a-b) Résultats d'une la détection de points-clés classique (c-d) et d'une détection homogène avec r le rayon de suppression au voisinage des points retenus. **[Bro05]**

Fiabilité des points : Comme énoncé dans la partie **3.1**, la fiabilité d'un point d'intérêt est fonction de sa *robustesse* et de sa *répétabilité*. Cette dernière a été définie dans le cadre de la reconnaissance d'objet comme « l'aptitude à détecter un même point dans une **série** d'images ». Mais cette définition a été créée pour des scènes fixes. Dans le cadre du suivi d'objet où une notion de

mouvement rentre en compte, celle-ci se ramène à « l'aptitude à détecter un même point dans une **suite successive** d'images ». Il est à noter que dans ce cas particulier, la répétabilité de certains détecteurs montre une différence significative.
La fiabilité d'un détecteur de points est définie comme la fiabilité moyenne du nuage de points détecté. Si certains des points détectés sont fiables, une quantité de bruit, variant selon les algorithmes, est également générée. Les méthodes susceptibles d'accroître la fiabilité des algorithmes cherchent donc à filtrer ces points indésirables. Comme mentionné dans le paragraphe sur la densité des points, les algorithmes, pour être performants, nécessitent une quantité minimale de points. L'efficacité d'un détecteur de points d'intérêts dans le cadre du suivi d'objet sera donc fonction de sa fiabilité et de la quantité de points extraits. De plus, le filtrage des points indésirables demande du temps de calcul supplémentaire.

3.5 Caractérisation de régions d'intérêt

Les régions d'intérêt répondent aux mêmes critères de distinctivité et de richesse d'information que les points-clés. Leur seule différence réside dans un support spatial plus étendu. Il en résulte avant tout des primitives avec un taux de répétabilité plus élevé que pour des points-clés puisque les diverses variations de géométrie et d'illumination entre plusieurs images sont plus à même de perturber la présence d'un point que d'une région. De plus les régions extraites constituent également le support nécessaire au calcul des descripteurs associés. Toutefois les régions sont généralement extraites selon un critère d'homogénéité qui les rende plus difficile à différencier les unes des autres que pour des points-clés. De plus, la forme de la région peut être sujette à d'importantes variations d'une image à l'autre, ce qui conduit généralement à approximer les régions par des ellipses. Enfin la densité de primitives extraites est largement inférieure dans le cas de région que dans celui de points-clés. La question de savoir si l'utilisation de régions d'intérêt doit être préférée à celle de points-clés dépend une fois de plus du type de traitement induit par l'application considérée.

4 Détection de points d'intérêts et descripteurs

4.1 Etat de l'art

4.1.1 Détection de points d'intérêts

Les algorithmes de détection de points d'intérêts peuvent se diviser en trois catégories : la détection de coins qui exploite des propriétés géométriques de l'image, la détection par mesure d'énergie qui recherche des points d'intérêt le plus descriptif possible (i.e. dont le voisinage est riche en terme information), et la détection d'extrema locaux qui sélectionne les points caractéristiques d'une fonction dépendant de l'intensité de l'image.
Pour un état de l'art ou une comparaison de différents points d'intérêts existants, voir **[Mik03, Mik05-2]**. Leur comparaison dans le cadre de scènes non planaires est aussi disponible **[Fra04]**.

4.1.1.1 Détection de coins

Les bons points d'intérêts sont ceux qui sont faciles à détecter, répétables (qui ont une haute probabilité d'être présents dans d'autres images de l'objet considéré sous un angle différent) et dont le voisinage est suffisamment caractéristique pour pouvoir décrire une région invariante. Typiquement, les coins ont ces avantages, d'où leur très large utilisation dans la littérature.

4.1.1.1.1 Le détecteur de Moravec (1980)

Le détecteur de **Moravec [Mor80]** est l'un des premiers détecteurs de points d'intérêts qui fut, à l'origine, développé pour la robotique. Son principe est d'analyser une fenêtre centrée sur un pixel et de déterminer les changements moyens d'intensité E(x,y) dans le voisinage considéré lorsque la fenêtre se déplace dans diverses directions (x,y). La fonction considérée est la suivante :

$$E(x,y) = \sum_{u,v} w(u,v)\left|I(x+u,y+v) - I(u,v)\right|^2$$

, avec *w* la fenêtre considérée (w(u,v)=1 dans la fenêtre, w(u,v)=0 en dehors de la fenêtre), et I(u,v) l'intensité du pixel (u,v).

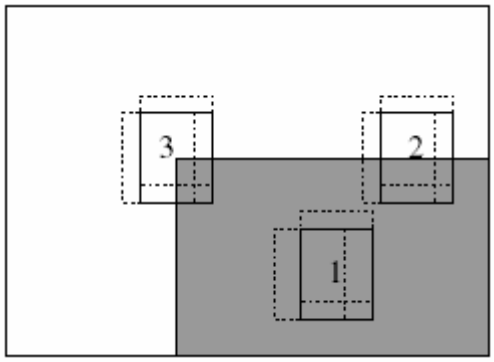


Figure 16: Les différents cas de figure traités par le détecteur de Moravec, (1) La surface uniforme, (2) Le point de contours, (3) Le coin.

Trois cas de figure peuvent intervenir lors de l'application de cette fonction. Si la zone considérée est uniforme (Figure 16), le changement d'intensité sera faible quel que soit la direction de glissement de la fenêtre. Par contre, si le voisinage considéré contient un contour (Figure 16), l'intensité moyenne de la fenêtre variera fortement pour une translation de celle-ci perpendiculaire au contour et faiblement pour une translation parallèle. Enfin, pour le cas d'un coin présent dans la fenêtre (Figure 16), la fonction E aura de fortes valeurs quelle que soit la direction du mouvement.

Toutefois, ce détecteur souffre d'inconvénients qui ne lui permettent de fonctionner que pour des cas limités :

- Ce détecteur est directionnel (anisotrope)
- Le voisinage rectangulaire induit une réponse bruitée.
- Ce détecteur répond trop fortement aux contours car seul le minimum de E est pris en compte pour chaque pixel.

4.1.1.1.2 Le détecteur de Harris (1988)

Prenant en compte ces désavantages, **Harris et Stephen [Har88]**, améliorèrent le principe du détecteur de Moravec, pour créer, en 1988, le détecteur de coins parmi les plus connus et largement utilisés dans la litterature **[Gab05][Isl05-1][Bau00]**. Tout comme le détecteur de Moravec, le détecteur de Harris est basé sur la variation de la luminance au voisinage d'un pixel. Celle-ci est exprimée sous la forme :

$$[x, y] M . \begin{bmatrix} x \\ y \end{bmatrix}$$

, avec M la convolution des dérivées premières de l'image I_x et I_y avec une gaussienne $G(\sigma)$. Elle est basée sur les moments du second ordre μ_{11}, μ_{02}, μ_{02}, et s'exprime sous la forme :

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} = G(\sigma) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

La matrice M, dite du gradient moyen décrit la façon dont la luminance se comporte autour du point considéré. Une configuration de coin correspondant à une forte variation dans toutes les directions se traduit par des valeurs propres de M fortes et de même importance. Dans le cas d'un contour, une des deux valeurs propres est prépondérante, et la variation n'est forte que selon une direction.
Afin d'éviter le coûteux calcul des valeurs propres, la valeur d'intérêt d'un pixel est calculée avec :

$$R=Det(M)-kTr(M)^2$$

Où $Det(M)=AB-C^2$ et $Tr(M)= A+B$.
Cette grandeur favorise les configurations où les deux valeurs propres sont fortes et d'importances égales. R est donc positif dans les régions de coins, négatif dans les zones de contours et petit dans les régions homogènes. Enfin, une détection de maximum local permet de retenir les points identifiés comme des coins.

Trois facteurs entrent en ligne de compte. Tout d'abord, le paramètre k, qui influence le nombre de points détectés et qui, expérimentalement, produit de meilleurs résultats pour une valeur entre 0,04 et 0.06. Ensuite, la taille de la fenêtre de lissage et, pour finir, la méthode d'estimation de la dérivée. Ce détecteur est invariant à la rotation et à la translation, résistant aux changements d'illumination. De plus, les travaux de **Schmidt & Mohr [Sch98],** en 1998, identifièrent le détecteur de Harris comme le détecteur de points d'intérêt le plus répétable parmi les détecteurs existants. Toutefois, ses performances s'écroulent lors de changements d'échelle.

Par la suite, de nombreuses améliorations furent apportées à cet algorithme. **Montesinos & al. [Mon98]** le généralisèrent aux images couleurs en modifiant la matrice M de la façon suivante :

$$M = G(\sigma) * \begin{bmatrix} (R_x^2 + G_x^2 + B_x^2) & (R_x R_y + G_x G_y + B_x B_y) \\ (R_x R_y + G_x G_y + B_x B_y) & (R_y^2 + G_y^2 + B_y^2) \end{bmatrix}$$

où $G(\sigma)$ est un lissage gaussien de taille σ.

Les travaux de **Dufournaud [Duf00]** montrèrent qu'il est également possible de le rendre robuste aux changements de résolution. Dans cette méthode, les points de Harris sont tout d'abord calculés à différentes échelles en faisant varier la taille σ da la gaussienne, en modifiant la fonction de Harris :

$$M = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} = \sigma_D{}^2 G(\sigma_I) * \begin{bmatrix} I_x{}^2(x,\sigma_D) & I_x I_y(x,\sigma_D) \\ I_x I_y(x,\sigma_D) & I_y{}^2(x,\sigma_D) \end{bmatrix}$$

avec σ_D l'*échelle de différenciation* et σ_I l'*échelle d'intégration*. Les dérivées locales sont calculées à l'issue d'un lissage gaussien de taille σ_D et sont ensuite moyennées par une fenêtre gaussienne de taille σ_I au voisinage du point étudié. L'étape suivante consiste à apparier à l'image à identifier chaque ensemble de points associé à une échelle. Le principal défaut de cette méthode est que l'image à identifier doit être comparée à plusieurs images (une pour chaque échelle), ce qui augmente considérablement le temps de calcul.

Mikolajczyk [Mik01] apporta une réponse à ce problème avec le détecteur de Harris-Laplace. L'avantage de ce détecteur est l'identification, pour chaque point, d'une « échelle caractéristique ». Ainsi, un point d'intérêt peut être défini par une position (x,y) et une échelle σ. Une seule image représente l'ensemble des points d'intérêts et leurs résolutions respectives. La méthode consiste à extraire des points d'intérêts sur différentes échelles (toujours calculées par des lissages gaussiens de différentes tailles) avec le détecteur de Harris, pour ensuite ne conserver que les points pour lesquels le Laplacien est un maximum local dans l'espace-échelle. Les points ainsi trouvés sont donc plus discriminants et les auteurs montrèrent que leur répétabilité surpassait celle des points déjà existants (Figure 17).

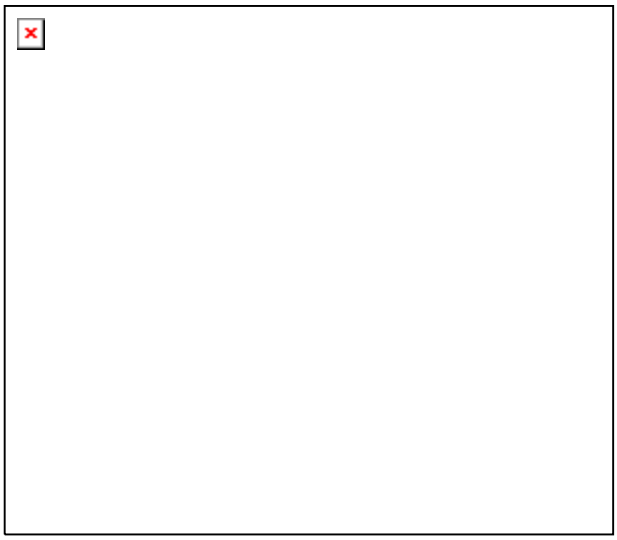


Figure 17: Le taux de répétabilité de différents détecteurs.

Le détecteur Harris-affine le rendit ensuite robuste aux transformations affines **[Mik02]**. Dans ce cas de figure, les auteurs considèrent que la région subit alors deux changements d'échelle distincts, chacun selon une direction donnée. Ces deux transformations sont quantifiées par σ_I, l'*échelle d'intégration* et σ_D, l'*échelle de différentiation*. Afin de simplifier les calculs, les auteurs émettent l'hypothèse peut contraignante que $\sigma_I = s.\sigma_D$. L'échelle d'intégration est calculée à partir de la fonction de Harris-Laplace vue dans le paragraphe précédent et l'échelle de différenciation est estimée grâce à la mesure d'isotropie Q suivante :

$$Q = \frac{\lambda_{\min}(M)}{\lambda_{\max}(M)}$$

où $\lambda_{min}(M)$ et $\lambda_{max}(M)$ sont la plus petite et la plus grande des valeurs propres de la matrice M des moments du second ordre. Cette valeur varie sur l'intervalle [0…1] avec 1 pour une structure parfaitement isotrope. La recherche d'une telle structure pour une échelle d'intégration caractéristique donnée nous permet donc d'obtenir une échelle de différentiation propre au voisinage d'un point.

L'algorithme consiste donc à évaluer successivement ces deux transformations tout en respectant l'hypothèse $\sigma_I = s.\sigma_D$ jusqu'à stabilisation des paramètres σ_I et σ_D. Son déroulement pour un point x est le suivant :

1-On initialise M

2-Tant que le critère de convergence $1 - \frac{\lambda_{\min}(M)}{\lambda_{\max}(M)} \leq \varepsilon$ *n'est pas satisfait :*

2-1-On calcule l'échelle caractéristique σ_I avec la fonction de Harris-Laplace.
2-2-On sélectionne dans l'intervalle [0.5,…,0.75] le rapport s qui maximise la mesure d'isotropie Q.
2-3-On localise le point x sur la position de ce voisinage qui maximise la fonction de Harris.
2-4-Fort des nouvelles valeurs de σ_I et σ_D, on met à jour M.

Bénéficiant de la répétabilité du détecteur Harris, cette méthode est parmi les plus efficaces. Elle offre une robustesse accrue aux transformations usuelles pour un temps de calcul acceptable (1 à 2 secondes pour une image 800 × 640).

4.1.1.1.3 Le système de suivi de KLT (1981)

Le système de suivi KLT (Kanade, Lucas et Tomasi) **[Bru81][Tom91][Shi94]** est un algorithme qui vise à détecter des « caractéristiques optimales pour le suivi ». Une telle caractéristique est définie comme discriminante, non-redondante et dont la phase d'extraction ne doit pas être séparée de celle de suivi. Les primitives choisies par les auteurs comme comportant ces qualités sont des patchs texturés avec de fortes variations d'intensités sur les deux axes *x* et *y*. Notons que cette approche, similaire à celle du détecteur de Harris recherche des régions carrées de taille prédéfinie plutôt que des points. Les caractéristiques retenues seront donc beaucoup moins nombreuses, mais plus discriminantes que les points de Harris. De plus, cette concision peut être ultérieurement exploitée en éliminant les patchs dont la mesure de dissimilarité est trop faible. Un exemple est montré en Figure 18.

Figure 18: Image test et caractéristiques KLT correspondantes détectées.

La fonction d'intensité étant notée par *g(x,y)*, on considère la matrice de variation d'intensité sur un patch donnée :

$$Z = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix}$$

Un patch est retenu comme candidat si ses deux valeurs propres λ_1 et λ_2 sont supérieures à un seuil prédéfini λ. Le nombre *n* de caractéristiques à suivre est fixé par l'utilisateur. En conséquence, les patchs sont classés en fonction de leur qualité définie par :

$$min\ (\lambda_1, \lambda_2) > \lambda$$

Afin d'éviter tout recouvrement, une distance minimum séparant deux patchs est fixée. Si un patch est perdu, l'algorithme peut optionnellement le remplacer par un nouveau.

Une mesure de dissimilarité supportant des transformations affines permet à l'algorithme de comparer les patchs du modèle à ceux de l'image actuelle.

4.1.1.1.4 Méthodes robustes aux transformations affines

Typiquement, rendre des points d'intérêts invariants aux transformations affines consiste à les enrichir de descripteurs robustes à ces transformations. Ce qui est le cas des moments d'ordre supérieur à deux. Généralement, le voisinage de points d'intérêts existants est approximé par une ellipse et un groupe de moments invariants est calculé pour ce voisinage, comme par exemple pour les travaux de **Baumberg [Bau00]** inspirés de dérivés gaussiennes ou ceux de **Zisserman et Schaffalitzky [Zis01],** basés sur une fonction polynomiale.

Tuytelaars et Van Gool [Tuy04] présentèrent une méthode originale. Pour tout coin p retenu comme étant un point d'intérêt, la région recouverte par un parallélogramme pp_1qp_2 est étudiée avec l'accroissement de sa taille, q étant le point opposé à p, p_1 et p_2 les points suivant le contours tels que leur vitesse relative l_1 et l_2 soit la même (Figure 19).

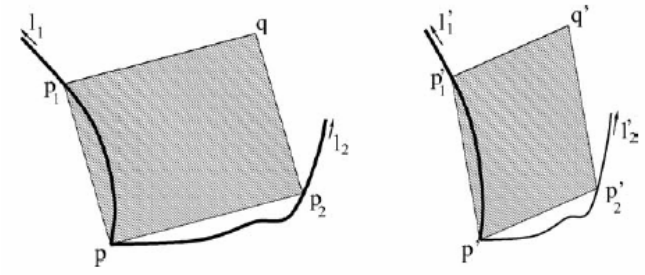


Figure 19: Construction d'une région affine à partir d'un coin défini comme point d'intérêt.

Leur vitesse relative est calculée par la formule :

$$l_i = \int abs\left(\det(p_i^{(1)}(s_i)\,p - p_i(s_i))\right) ds_i \quad \text{i=1,2}$$

avec s_i un paramètre de courbe arbitraire et $p_i^{(1)}(s_i)$ la première dérivée de $p_i(s_i)$. la région Ω retenue est celle pour laquelle la fonction d'analyse utilisée atteint un maximum. Trois fonctions sont proposées :

$$f_1(\Omega) = \frac{M_{00}^1}{M_{00}^0} \qquad f_2(\Omega) = \left|\frac{\det(p_1 - p_g \quad p_2 - p_g)}{\det(p - p_1 \quad q - p_2)}\right| \times \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)^2}}$$

$$f_3(\Omega) = \left|\frac{\det(p - p_g \quad q - p_g)}{\det(p - p_1 \quad q - p_2)}\right| \times \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)^2}}$$

$$\text{avec } M_{pq}^n = \int_\Omega I^n(x,y) x^p y^q dxdy \text{ et} \qquad p_g = \left(\frac{M_{10}^1}{M_{00}^1}, \frac{M_{01}^1}{M_{00}^1}\right)$$

$f_1(\Omega)$ représente l'intensité moyenne de la région Ω. $f_2(\Omega)$ et $f_3(\Omega)$ sont composées de deux termes. Le premier dépend du centre de gravité pondéré par l'intensité des pixels (Figure 20) et le deuxième contrebalance la dépendance à l'intensité.

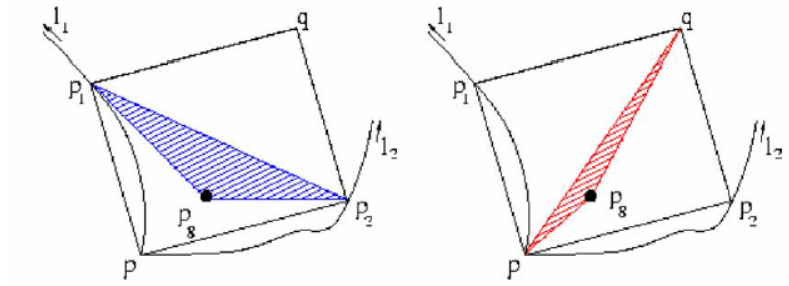


Figure 20: Interprétation physique du premier terme de f2(Ω) (gauche) et de f3(Ω) (droite).

Une autre approche, suite logique des travaux existants, mais seulement testée récemment **[Bro02][Fra05]** consiste à apparier des groupes de points plutôt que des points isolés pour caractériser la région ou l'image d'intérêt. Cette méthode procède en deux étapes. Tout d'abord, les points d'intérêts sont extraits à partir du détecteur adéquat. Dans un deuxième temps, le nuage de points détecté est découpé en sous-ensembles distincts. Dans sa méthode, **Brown [Bro02]** propose la transformée de Hough

comme algorithme de classification, alors que **Fraundorfer [Fra05]** élabore un algorithme inspiré des *MSER* (voir **4.1.1.2.3** pour leur description) accroissant la stabilité des sous-ensembles créés. Cette technique accroît la fiabilité du détecteur de point d'intérêt utilisé, puisque seul trois points sont nécessaires pour correctement identifier une région et son mouvement affine. La mise en correspondance des groupes de points reste alors efficace malgré la possible mauvaise redétection de certains points.

4.1.1.1.5 Autres détecteurs de coins

D'autres détecteurs furent développés par la suite en se référant à la notion de coin. **Smith [Smi95]**, partant du principe qu'au niveau d'un coin, on est en présence de deux objets de luminance différente, développa le détecteur SUSAN. Analysant le voisinage d'un pixel, ce détecteur considère, avec une certaine tolérance, la zone de même intensité que le pixel (appelée *noyau*). Ainsi l'intérêt d'un pixel sera fonction de la taille de son noyau : pour un pixel d'angle de 90 degrés, le noyau occupera le quart du voisinage, alors que pour un contour, on obtiendra un noyau d'environ la moitié du voisinage.

Zitova [Zit99] proposa un algorithme basé sur la variation de la luminance par rapport à la luminance moyenne M(i,j) au voisinage d'un pixel (i,j). En chaque point $(i_1, j_1),\ldots,(i_k,j_k)$ d'un cercle centré sur (i,j), on évalue en suivant le sens des aiguilles d'une montre la grandeur suivante :

$$R=I(i_1, j_1)-M(i_1, j_1)$$

avec $I(i_1, j_1)$ l'intensité du pixel (i_1, j_1). Si le signe de *R* change 2 fois au cours de cette analyse, on considère que le cercle coupe 2 objets de différente luminance en (i_a,j_a), et (i_b,j_b). Si l'angle $\alpha=((i_a,j_a),(i,j),(i_b,j_b))$ est égal à $\pi/2$, à un paramètre *d* près, le pixel (i,j) est alors marqué comme étant un coin (Figure 21).
La totalité des pixels de l'image sont traités par cet algorithme et les coins trop proches sont éliminés pour ne garder que les *n* pixels les plus significatifs.

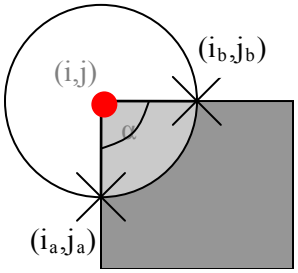


Figure 21: Illustration de l'algorithme de Zitova **[Zit99]**.

Trajkovic [Tra98] se base également sur le principe de la variation de la luminance au voisinage d'un pixel. Son idée est de comparer l'intensité des pixels aux extrémités des différents diamètres du cercle centré sur le pixel à analyser, le nombre de valeurs différentes permettant de conclure à la présence d'un coin ou non. Une technique d'interpolation circulaire et une analyse «coarse to fine » rendent cet algorithme plus rapide que ses prédécesseurs malgré une robustesse sensiblement moins bonne.

4.1.1.1.6 Détecteurs basés sur les ondelettes

Rappelons que les ondelettes donnent des informations sur une image à différentes échelles. Elles sont obtenues grâce à deux fonctions calculées sur des blocs de plus en plus précis de l'image. La *fonction*

d'échelle donne la valeur moyenne du signal sur l'intervalle considéré et la *fonction de détails* code la différence entre la fonction d'échelle et la valeur réelle du signal.

Bhattacharjee [Bha99] proposa un algorithme à base d'ondelettes détectant les fins de segments. Son détecteur utilise une dérivée de l'ondelette de Morlet suivante :

$$ES_1(x,y) = \frac{1}{4} x e^{-(\frac{x^2+y^2}{4} + \frac{y_1}{4}(y_1 - 2iy))}$$

y_1 contrôlant la sélectivité de fréquence du filtre. Pour chaque pixel, cette ondelette est calculée pour une série d'orientations et le couple orientation-réponse (appelé « jeton ») correspondant à la meilleure réponse est retenu. Enfin, les maxima locaux parmi l'ensemble des jetons sont conservés comme points clés de l'image.

Loupias [Lou00][Seb01] quant à lui, opta pour une analyse de l'image de la résolution la plus grossière à la plus fine en se basant sur des ondelettes orthogonales à support compact. La pertinence de chaque bloc de l'image est calculée en fonction de la valeur du signal de détail (représentant la variance sur le bloc). Si un bloc est retenu, l'opération est répétée sur son bloc-fils pour lequel la pertinence est maximale (Figure 22). Ce processus est appliqué itérativement jusqu'à la plus petite échelle (4 pixels). Le point d'intérêt choisi est celui, parmi les 4 pixels finaux, dont le gradient est le plus élevé. Il est ensuite caractérisé par ses trois premiers moments couleurs.
Cette méthode permet d'obtenir des points essentiellement situés sur les contours et les arêtes. **Sebe & al [Seb02]** la testèrent pour les ondelettes de Haar et de Daubechie et montrèrent que les meilleurs résultats sont obtenus avec les ondelettes de Daubechies. De plus, cet algorithme est plus performant en termes de taux de répétabilité et d'information apportée que le détecteur de Harris.

Figure 22: Support spatial des coefficients pertinents suivis.

4.1.1.2 Détection d'extrema locaux

Tout comme les coins, les extrema peuvent également servir de points d'intérêts. Ces points ne peuvent être extraits avec autant de précision que les précédents, car les extrema sont souvent peu marqués. Toutefois, contrairement aux coins, majoritairement présent sur les contours et les zones fortement texturées, ils sont généralement répartis de façon homogène dans l'image. Cet avantage est particulièrement appréciable dans le cas de suivi d'objet dont le contour est bien souvent déformé au cours de la vidéo et donc mal adapté pour l'extraction de points fiables. Néanmoins, la détection des extrema étant très sensible au bruit, l'extraction de ces points d'intérêts nécessite souvent un filtrage préalable de l'image.

4.1.1.2.1 Scale Invariant Features Transform (SIFT) de Lowe (1999)

Mikolajczyk. K, Schmid [Mik03] ont évalués en 2003 une variété d'approches et identifièrent les SIFT de **Lowe [Low99][Low04]** comme les points d'intérêts donnant les meilleurs résultats. La première étape de la méthode consiste à trouver les extrema locaux grâce à une pyramide de différences de gaussiennes. Soit A le produit de la convolution de l'image d'origine par un noyau gaussien de taille $\sigma = \sqrt{2}$. Et soit B le produit de la convolution de A par une gaussienne de taille $\sigma = \sqrt{2}$, ce qui équivaut à la convolution de l'image d'origine par un noyau gaussien de taille $\sigma = 2$. La fonction de différence de gaussienne est obtenue par soustraction de B par A. La pyramide de gaussienne est obtenue en appliquant ce même filtrage à chaque niveau de la pyramide construite par rééchantillonnages successifs de l'image B. Un pixel est ensuite retenu comme étant un maximum local (respectivement minimum local) si il a une valeur maximale (respectivement minimale) parmi ses 8 voisins directs et ses 18 voisins des niveaux supérieurs et inférieurs (Figure 23).

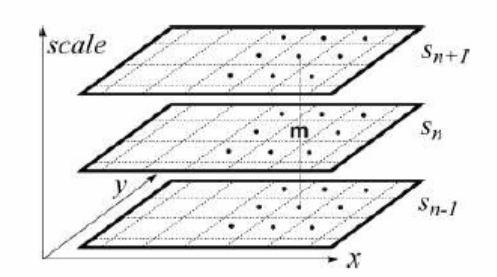


Figure 23: Détection des maxima et minima locaux. Le pixel (marqué d'un m) est comparé à ces 26 voisins ; les 8 voisins à la même échelle et les 18 voisins aux échelles inférieure et supérieure.

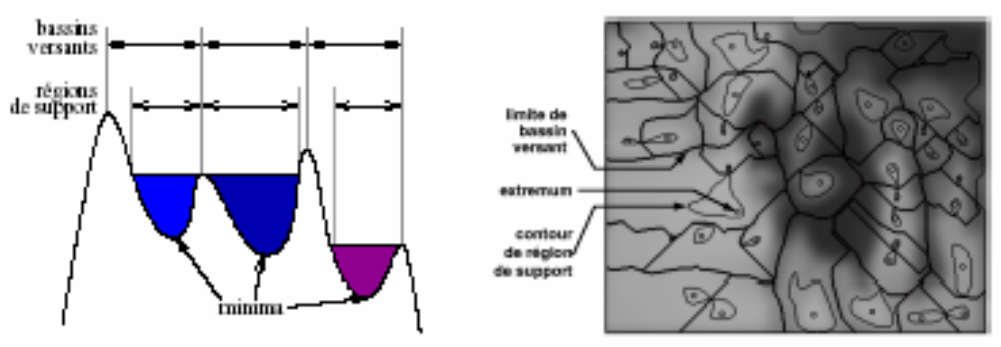
La deuxième étape consiste à filtrer les points d'intérêts qui ne sont pas stables, soit les points ayant un faible contraste et les points de contour.

4.1.1.2.2 Blobs issus de la théorie de l'espace-échelle de Lindeberg (1996)

Les blobs issus de la théorie de l'espace-échelle développé par **Lindeberg [Lin96]** et utilisés par **Megret [Meg01]** se basent sur l'analyse d'extrema locaux d'images filtrées L(σ) obtenues par convolution de l'image d'origine I avec des noyaux gaussiens g_σ de taille σ :

$$L(\sigma) = g_\sigma * I \text{ où } g_\sigma\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Afin d'éviter une surcharge de calculs, le nombre d'images filtrées est limité et leur calcul est effectué avec la transformée de Fourier rapide.
On associe à chaque minimum local une région de support. Celle-ci est calculée par inondation mais sa limite ne dépasse pas la plus petite valeur du contour du bassin versant correspondant (Figure 24). Un processus analogue (mais indépendant du calcul des minima) est effectué pour les maxima en traitant le signal opposé.



(a) (b)

Figure 24: Régions de support et bassin versant (a) Analogie avec le cas monodimensionnel (b) Blobs de maxima d'une image avec σ=8.

4.1.1.2.3 Méthodes invariantes par transformations affines

Bien que la plupart des méthodes soient robustes à la rotation et à la translation, elles ne résistent pas aux autres transformations affines. C'est pourquoi les travaux plus récents, particulièrement dans le domaine de la vision stéréo (avec plusieurs caméras), s'orientent vers des points d'intérêts qui soient robustes à tout type de transformation affine. Le principe consiste à extraire plutôt qu'un point, une région caractéristique qui sera ensuite approximée par une primitive (généralement une ellipse) dont le centre sera le point-clé et la zone recouverte le support du descripteur associé. Ce type d'approche offre deux avantages. Une région est, tout d'abord, une primitive plus répétable qu'un point de part sa taille. Ensuite, elle permet la sélection d'un support de descripteur optimal plutôt qu'une zone circulaire de rayon prédéterminé. Par contre elle peut souffrir d'imprécision lors de sa détection, tout particulièrement dans le cas d'effets de flous.

Tuytelaars et Van Gool [Tuy04] proposent une méthode qui offre également l'avantage de palier au problème de la localisation imprécise d'un extremum local (Figure 25). Après la détection d'extrema, la fonction d'intensité suivante est calculée sur une série de rayons émanant de chaque extremum :

$$f_I(t) = \frac{abs(I(t) - I_0)}{\max\left(\frac{\int_0^t abs(I(t) - I_0)dt}{t}, d\right)}$$

avec t la position sur l'arc, $I(t)$ l'intensité à la position t, I_0 l'extremum d'intensité, et d une valeur ajoutée pour éviter la division par 0.

Figure 25: Robustesse de l'extraction de région à une localisation imprécise d'un extremum d'intensité. **[Tuy04]**

Le premier maximum local rencontré pour chaque rayon par f_I est conservé. Tous les points retenus sont ensuite liés et la région conservée est l'approximation par une ellipse de cette forme irrégulière. La taille de l'ellipse est ensuite doublée pour obtenir une région plus riche en information (Figure 26). Les deux principaux axes de l'ellipse sont ensuite déterminés pour permettre la rotation.

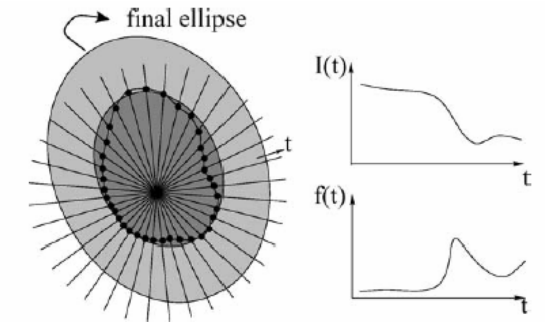


Figure 26: Représentation de la méthode de Tuytelaars et Van Gool **[Tuy04]** appliqué à un point.

Une approche aussi simple qu'originale, s'inspirant de l'algorithme de *la ligne de partage des eaux* fut présentée par **Matas & al [Mat02-2]** avec les *régions d'extrema maximalement stable (MSER)*. Cet algorithme se base sur le seuillage de l'intensité de l'image. On étudie l'intensité pour un seuil décroissant (de 255 jusqu'à I_0) et les composantes connexes de l'image sont extraites. Les *régions d'extrema maximalement stable* sont les composantes connexes obtenues lorsque leur croissance en fonction de l'intensité admet un minimum local (i.e lorsque la pente est minimale). La région extraite est ensuite approximée par une ellipse et caractérisée par des moments invariants. Des *régions* d'*extrema maximalement stable* similaires sont calculables pour une étude croissante du seuil (de 0 à I_0). Cet algorithme a l'avantage d'offrir à la fois une grande rapidité (le calcul des régions prend environ 0.6 secondes pour une image 800 × 640) et un taux de répétabilité élevé pour toutes les transformations sauf les effets de flou. Notons toutefois que le résultat n'est pas un ensemble de points mais de régions. Une description explicite de cette région engendrerait un descripteur de trop grande dimension, impropre au stockage et à la comparaison. L'information est donc ensuite approximée afin d'obtenir un descripteur compact. Il en résulte une perte de précision du descripteur. Un autre problème de la méthode est le faible nombre de primitives extraites. Pour pallier à ce défaut, la communauté scientifique s'est logiquement orientée vers l'extraction de MSER couleurs **[Don06-2][For07]**.

4.1.1.3 Détection par mesure d'énergie

Ces points d'intérêts, se caractérisent par l'extraction de l'information visuelle d'une image (ou d'un objet) et sont conçus pour donner un « résumé » de celle-ci (ou de celui-ci) comme le montre la Figure 27. Bien que, tout comme les coins ils ne soient pas répartis de façon homogène dans l'image, le fait qu'ils soient concentrés sur les régions saillantes et délaissent les zones de moindre intérêt (comme l'arrière-plan) leur donne un fort pouvoir discriminant.

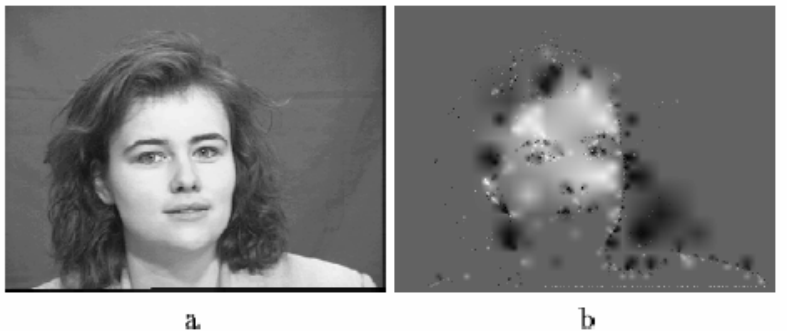


Figure 27: (a) Image initiale (b) Image résumée par ses points d'intérêts (extrait de **[Bre99]**).

4.1.1.3.1 Scale saliency (2001)

Cette méthode de **Kadir & Brady [Kad01]** mesure la pertinence d'une région à différentes échelles et sélectionne l'échelle pour laquelle la pertinence est optimale. Plus précisément, en chaque pixel est calculé un histogramme d'intensité de D colonnes sur un voisinage circulaire de rayon s. L'entropie de chacun de ces histogrammes est calculée et le maximum est retenu. Plus formellement :

$$E(x,s) = -\sum_{d \in D} p(d,x,s) \log_2 p(d,x,s)$$

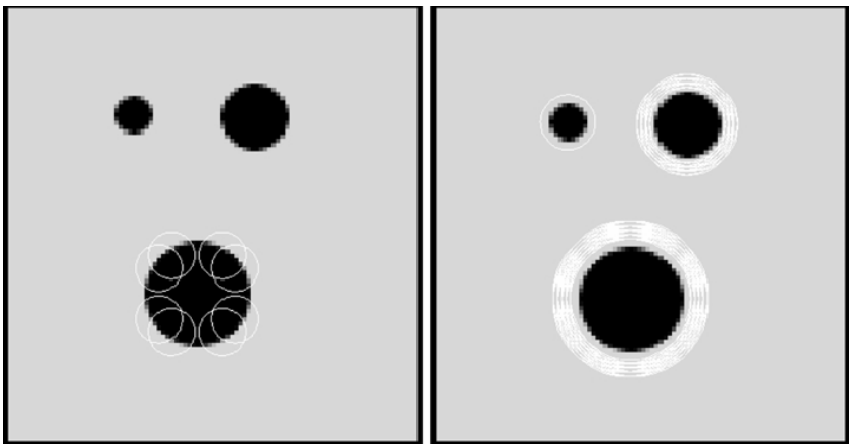
avec E l'entropie, $p(d,x,s)$ la probabilité que l'intensité d soit présente dans l'entourage du pixel situé à la position x, pour une échelle s. Les n pixels ayant la valeur la plus élevée sont conservés comme points d'intérêt de l'objet. Toutefois, la consistance de ce détecteur est difficile à évaluer. Il en résulte le calcul de beaucoup de points similaires dans le voisinage d'un point d'intérêt. Les auteurs utilisent la variation des statistiques du descripteur en fonction de l'échelle pour évaluer cette similarité. En pratique, ils utilisent la somme des valeurs absolues des variations de la fonction de densité de probabilité $p(d,x,s)$:

$$W_D(x,s) = \sum_{d \in D} \left| \frac{\partial}{\partial s} p(d,x,s) \right| dd$$

L'entropie est ensuite pondérée par cette fonction comme suit :

$$S(x,s) = E(x,s) \times W_D(x,s)$$

La Figure 28 illustre l'efficacité du procédé.



(a) (b)

Figure 28: Zones pertinentes: (a) Sélectionnées en utilisant le pic d'entropie (b) Sélectionnées en utilisant le pic d'entropie pondéré par la somme des valeurs absolues des variations sur le pic.

Par la suite **Kadir, Zisserman & Brady [Kad04]** adaptèrent la méthode pour la rendre robuste aux changements de point de vue en remplaçant la région circulaire par une région elliptique. Calculer un histogramme, puis l'entropie pour chaque pixel d'une image est un procédé algorithmiquement très coûteux, et il faut environ de 30 secondes pour déterminer les régions pertinentes d'une image de résolution 800 × 640.

4.1.1.3.2 Points d'intérêts issus de la mesure du contraste (1999)

Cette méthode originale mise au point par **Bres [Bre99]** est basée sur la mesure du contraste. Une pyramide de contraste est construite en trois passes. Tout d'abord, une première pyramide représentant la luminance locale est construite de façon ascendante (l'image la plus large à la plus petite).

$$G_k(P) = \sum_{M \in fils(P)} w(M) G_{k-1}(M)$$

où w est une fonction normalisée de poids simulant une pyramide gaussienne. Une deuxième pyramide représentant la luminance globale est créée de façon descendante.

$$B_k(P) = \sum_{Q \in père(P)} W(M) G_{k+1}(Q)$$

avec W une fonction normalisée de poids. Enfin, deux moyens sont proposés pour construire la pyramide de contraste:

- Une formule usuelle $C_k(P) = \frac{G_k(P)}{B_k(P)}$
- Une deuxième permettant d'avoir 0 comme borne inférieure et des valeurs similaires pour les intensités faibles et fortes $C_k^*(P) = Min\left(\frac{|G_k(P) - B_k(P)|}{B_k(P)}, \frac{|G_k(P) - B_k(P)|}{255 - B_k(P)} \right)$

Les points d'intérêts sont ensuite obtenus par une recherche des maxima locaux aux différentes échelles de la pyramide.

4.1.1.3.3 Points issus de l'intérêt visuel (1999)

Itti [Itt99] s'appuie sur la notion d'intérêt visuel pour extraire de l'image les zones visuellement saillantes, autrement dit celles attirant le regard d'un individu observant l'image sans a priori (sans connaissances particulières sur le contexte de l'image). Ce système se base sur l'analyse de critères bas niveaux (couleur, orientation, contraste, intensité,…). Une carte de traits, représentant l'information visuelle de chacun de ces critères est calculée et l'ensemble de ces cartes est ensuite combinée pour former une carte de saillance représentant les zones d'intérêt visuel (Figure 29). Les extrema locaux de cette carte dépassant un seuil pré-établi sont retenus comme points d'intérêt. Ce type de technique nécessite toutefois de lourds calculs et doit être réservée pour des applications spécifiques à la détection d'intérêt visuel d'une image.

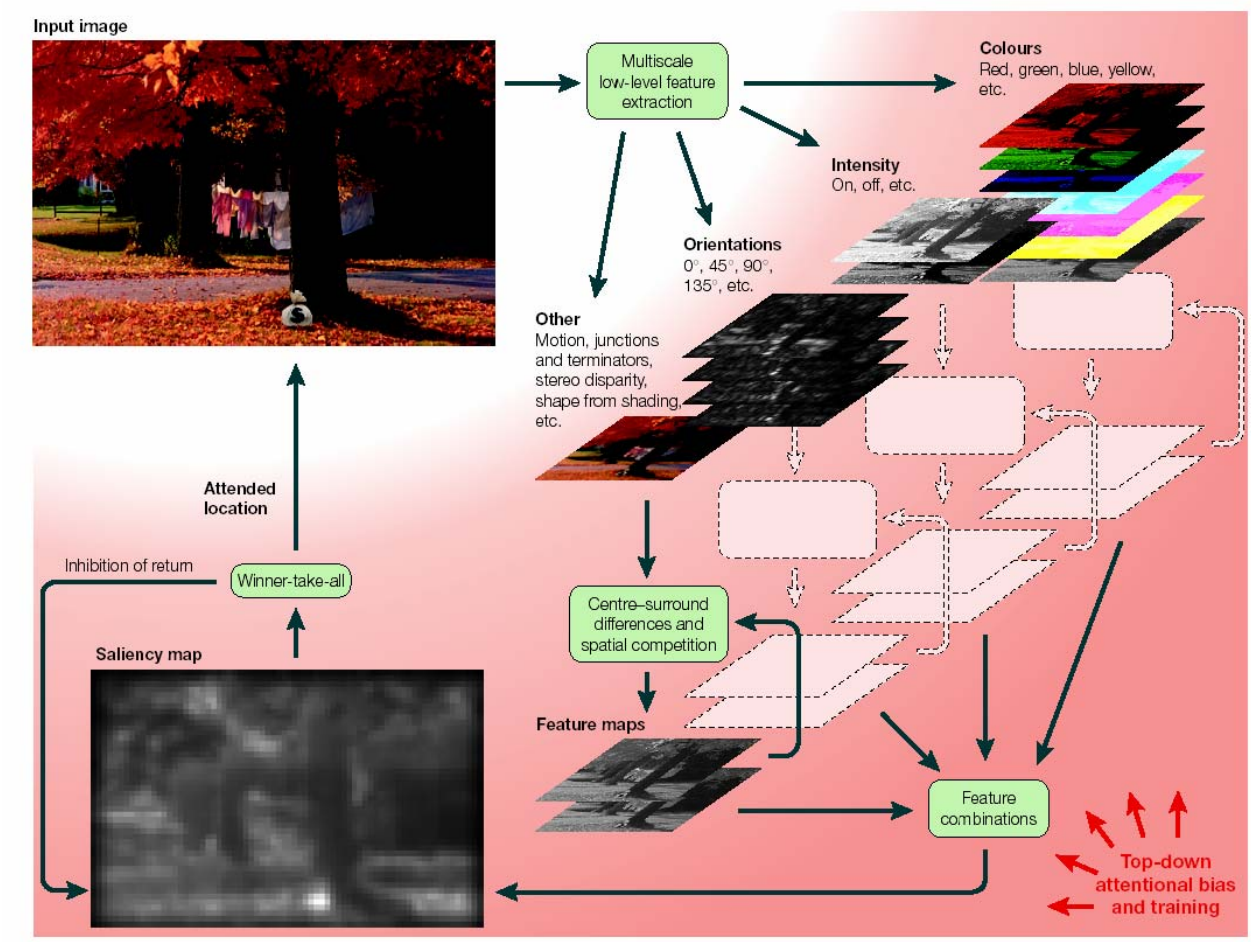


Figure 29: Diagramme d'un modèle d'attention visuelle.

4.1.2 Descripteurs

Une spécificité de l'appariement de points pour les problèmes de vision est qu'il est possible d'enrichir le point extrait d'informations issues de son voisinage. De tels descripteurs locaux ont pour but de faciliter l'appariement ultérieur entre les points en fournissant une signature caractéristique pour chacun d'eux.

Bien sûr, tout comme pour la détection de points d'intérêt, pour être fiables, ces descripteurs doivent être résistants aux transformations classiques : changements d'illumination, bruit, rotations, changements d'échelle, points de vue différents. L'obtention d'une échelle caractéristique est simple : il suffit de réutiliser celle calculée par le détecteur de points d'intérêt. La méthode pour rendre le descripteur invariant à la rotation consiste à le calculer sur une région circulaire centrée sur le point d'intérêt puis à sélectionner une direction caractéristique afin de pouvoir comparer deux régions par la suite. Toutefois, une orientation dominante n'étant pas nécessairement présente, cette méthode ne fonctionne donc pas toujours. Pour pallier à cette défaillance, certains auteurs ont mis au point des descripteurs invariants à la rotation. L'invariance au changement de point de vue est effectuée de manière similaire à cela prêt que la région de calcul du descripteur est elliptique ou approximée par une ellipse. Notons toutefois, que lorsque le descripteur calculé n'est pas invariant aux transformations affines, la comparaison de deux régions elliptiques nécessite une étape de normalisation préalable (Figure 30), innovation qui fut apportée par

Baumberg [Bau00]. Enfin il n'existe pas de méthode générale pour obtenir l'invariance au changement d'illumination, elle dépend du descripteur choisi et du modèle d'illumination considéré.

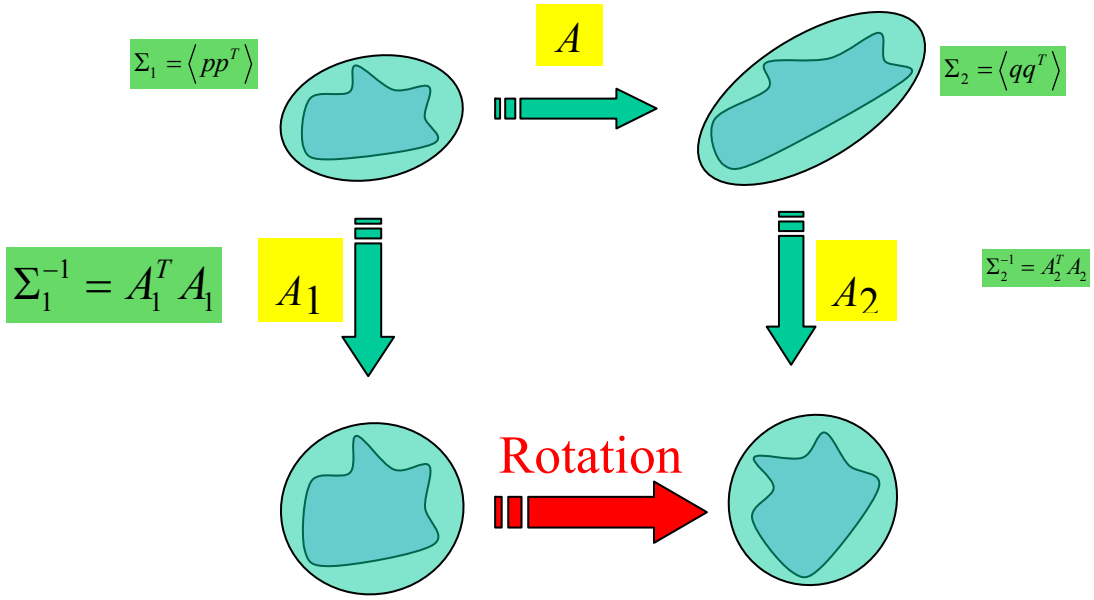


Figure 30: Procédé de normalisation d'une ellipse.

Le choix du descripteur n'est pas arbitraire. Il dépend d'une part du détecteur de points qui a été choisi. Par exemple, un descripteur basé sur les contours ne sera pas adapté pour caractériser les régions issues de l'algorithme des *régions d'extrema maximalement stable* (voir **4.1.2.3**) qui comportent de faibles variations d'intensité. D'autre part, les contraintes de l'application et des images utilisées jouent également un rôle primordial. Les descripteurs seront-ils calculés inline ou offline ? Quelles transformations l'image sera-t-elle susceptible de subir ? Quelle information les points d'intérêts devront-ils véhiculer en priorité ? Ce sont autant de questions qui détermineront la complexité de calcul permise, la robustesse aux différentes altérations, ou encore le type d'information sur laquelle le descripteur se fondera.

Lors de cette étape on pourra même s'interroger sur la taille de la région. En effet, plus une région sera grande, plus elle comprendra d'information, plus elle sera susceptible de décrire avec précision le voisinage du point d'intérêt. En contrepartie, d'avantage de pixels du décor pourront dégrader la description de l'objet et le recouvrement de régions, donc la redondance d'information sera plus probable. En pratique cependant, la plupart des auteurs choisissent des régions de grande taille pour décrire le voisinage d'un point.

Nous présenterons dans ce chapitre la plupart des descripteurs existants ainsi que leurs avantages et inconvénients. Leur comparaison est donnée dans **[Mik05-1]**. Mais dans la mesure où celle-ci ne les confronte que pour les cas les plus difficiles des transformations, les résultats obtenus ne sont valables qu'à titre indicatif dans le cadre du suivi d'objet où les transformations sont toujours minimes. Nous ne les détaillerons donc pas ici.

4.1.2.1 Descripteurs couleur

La couleur représente une information privilégiée dans la mesure où elle est facile d'accès et constitue une information essentielle dans la caractérisation d'une région. Et, bien que la résistance aux transformations usuelles des descripteurs basés sur la couleur soit discutable, ils sont souvent choisis.

Le descripteur le plus simple consiste à conserver les valeurs d'intensité au voisinage d'un point. L'appariement entre deux points est alors effectué par corrélation croisée. Rappelons que la corrélation croisée consiste à calculer une valeur de similarité, le coefficient de corrélation croisée C_i, entre le bloc R de l'image de référence et un bloc S dans l'image étudiée pour toutes les positions possibles de S sur un intervalle donné. L'objet recherché est identifié comme se trouvant à la position i pour laquelle la valeur du coefficient de corrélation croisée C_i est maximale. Le coefficient de corrélation croisée calculé avec un décalage $\Delta = (\Delta x,\Delta y)$ par rapport au bloc de l'image de référence est donné par la formule :

$$C_{R,S}(\Delta x,\Delta y) = \frac{\sum_{x,y}\left[I_R(x,y)-M_R\right]\times\left[I_S(x+\Delta x,y+\Delta y)-M_S\right]}{\sqrt{\sum_{x,y}\left[I_R(x,y)-M_R\right]^2\times\sum_{x,y}\left[I_S(x+\Delta x,y+\Delta y)-M_S\right]^2}}$$

avec M_R et M_S les valeurs d'intensité moyennes respectivement sur le bloc de l'image de référence et sur le bloc de l'image étudiée. Il est possible de rendre ce descripteur robuste aux rotations en considérant une région circulaire, il est également possible de le rendre robustes aux changements d'échelle en adaptant la taille de la région à l'échelle caractéristique détectée. Par contre la corrélation croisée reste très sensible aux variations d'illumination.

Par leur capacité à regrouper l'information, les histogrammes offrent une certaine stabilité face aux transformations de l'image, et la division de l'espace en parties distinctes apporte la robustesse face à d'importantes altérations locales. Les histogrammes sont de plus une représentation facile à mettre en place, et de nombreuses méthodes optent pour ce descripteur. Par exemple, la méthode de **Qin [Qin05],** « l'*Angular Radial Partitionning Intensity* » se base sur un histogramme 2D appliqué dans le voisinage centré sur le point d'intérêt et sous divisé en régions angulaires (Figure 31). La première dimension représente l'index de la sous région angulaire et la deuxième l'intensité de l'image (Figure 31). L'histogramme est ensuite normalisé afin de le rendre robuste aux changements d'illuminations.
La comparaison de deux régions P_i et Q_j s'effectue ensuite à l'aide de la fonction suivante :

$$COST_{ij} = \frac{1}{2}\sum_{n=0}^{N}\sum_{m=0}^{M}\frac{\left[h_i(n,m)-h_j(n,m)\right]^2}{h_i(n,m)+h_j(n,m)}$$

avec $h_i(n,m)$ la valeur de l'histogramme angulaire de l'image i en (n,m).

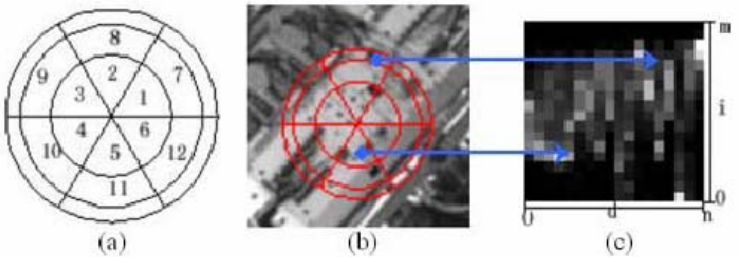


Figure 31: (a) Le descripteur est divisé en 18 sous-régions (par clarté, seules les 12 premières sont représentées ici) (b-c) Représentation en histogramme 2D, l'index des sous-régions en abscisse, l'intensité en ordonné. **[Qin05]**

Il est toutefois prouvé que, à cause des *effets de blocs* (ou *aliasing*), les histogrammes sont moins robustes que les autres méthodes aux déformations classiques (changement d'illumination, d'échelle,..).

Une solution fut apportée à cet inconvénient par le descripteur *spin-image* développé par **Lazebnik & al [Laz05]** qui étendit l'influence d'un pixel donné à plusieurs barres de l'histogramme. Dans leur méthode, le support de la région est divisé en cercles concentriques et un histogramme 2D est créé en fonction de l'intensité *i* et de la distance *d* par rapport au centre du support. La contribution *C* d'un pixel *x* sur le calcul de la valeur d'une barre *(d,i),* comparable à une gaussienne est donnée par :

$$C(x) = \exp\left(-\frac{\left(|x - x_0| - d\right)^2}{2\alpha^2} - \frac{|I(x) - i|^2}{2\beta^2} \right)$$

avec x_0 le centre et α, β les paramètres de lissage de l'influence du pixel en fonction de la distance au centre. Cette idée originale permet de pallier efficacement au problème de discontinuité entre les barres de l'histogramme.

Matas [Mat01], quant à lui, se proposa d'aborder le problème à l'aide d'un invariant couleur robuste aux changements géométriques et aux changements d'illumination : le « *multimodal neighborhood signature* » (MNS). Dans sa méthode, le calcul des descripteurs prend place après la division de l'image en un quadrillage régulier, et la recherche dans chacune de ces cases (appelées voisinage) d'extrema locaux, à l'aide de l'algorithme du *mean-shift*. Seuls les voisinages multimodaux sont retenus pour le calcul des descripteurs. Pour chaque voisinage retenu, un vecteur de RGB est calculé avec chaque combinaison de paire de modes. L'ensemble des vecteurs ainsi créé est ensuite découpé en classes et un vecteur représentatif est généré pour chaque classe.
Partant du principe que l'illumination est constante sur un voisinage réduit, l'invariant couleur choisi est le suivant :

$$r_1 = (r_R^1, r_G^1, r_B^1) = \left(\frac{R_i^1}{R_j^1}, \frac{G_i^1}{G_j^1}, \frac{B_i^1}{B_j^1} \right).$$

La distance adoptée entre deux vecteurs r_1 et r_2 est :

$$d(r_1, r_2) = \frac{1}{3}\left(d_{fr}(r_R^1, r_R^2) + d_{fr}(r_G^1, r_G^2) + d_{fr}(r_B^1, r_B^2) \right)$$

$$\text{avec } d_{fr}(p,q) = \frac{|a \times d - b \times c|}{a + b + c + d} \text{ où } p = \frac{a}{b} \text{ et } q = \frac{c}{d}.$$

Ce modèle suppose toutefois une orientation identique entre les deux images. Afin de parer à ce défaut, les auteurs proposèrent le descripteur suivant :

$$r = \left(\frac{R_i G_j}{G_i R_j}, \frac{G_i B_j}{B_i G_j} \right).$$

Cette méthode est exécutée avec différentes tailles de grilles pour la rendre robuste aux changements d'échelle.

Pour d'avantage de renseignements sur l'utilisation de la couleur pour calculer des invariants locaux, voir **[Gro97]**

4.1.2.2 Descripteurs basés sur l'orientation

Les descripteurs SIFT (*Scale Invariant Features Transform*) de **Lowe [Low99]** rendent le point d'intérêt robuste aux déformations classiques par calcul d'un histogramme 4*4*8 des 8 orientations possibles au voisinage de l'extremum considéré (Figure 32). Il s'agit donc d'un vecteur de 128 dimensions, ce qui lui confère un important pouvoir discriminatoire. En effet, les SIFT sont robustes aux changements d'échelle,

à la rotation, à la translation et aux transformations affines. L'invariance à l'illumination est obtenue en normalisant le descripteur par la racine carrée de la somme des carrés des composants.

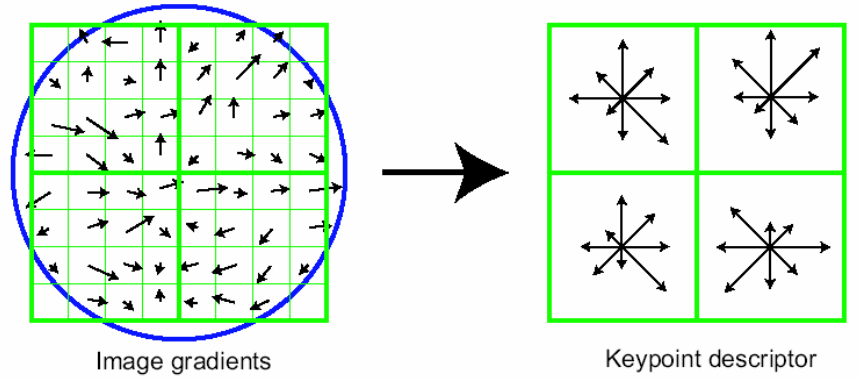


Figure 32: Le descripteur d'un point-clé est calculé à l'aide de l'histogramme 4*4*8 du voisinage centré sur le point-clé de 8 orientations possibles.

PCA-SIFT de **Ke & Sukthankar [Ke04]** améliora ensuite cette approche. L'analyse en composantes principales (PCA) permettant de garder l'essentiel de l'information du vecteur descriptif (36 dimensions au lieu de 128) et d'éliminer le bruit, PCA-SIFT permet de créer un descripteur plus discriminant accroissant significativement aussi bien l'efficacité que les temps de calcul des SIFT. Dans le même esprit, **Mikolajczyk & Schmid [Mik05-1]** présentèrent GLOH (Gradient Location and Orientation Histogram). Dans cette alternative, les SIFT sont calculés sur une grille en coordonnées polaires comprenant 3 divisions radiales et 8 angulaires (sauf au centre) et la magnitude du gradient est calculée sur une échelle de 16 valeurs. Il en résulte un vecteur de 17*16, soit 272 dimensions, ensuite réduit à 128 dimensions par l'analyse en composante principale. Dans leur étude sur les descripteurs **Mikolajczyk & Schmid [Mik05-1]** montrèrent que les résultats obtenus avec les SIFT et leurs dérivés surpassaient ceux des autres descripteurs. **Grabner [Gra06]** accéléra considérablement le processus d'extraction de ces descripteurs en utilisant une structure d'image d'intégrale et en remplaçant les différences de gaussiennes (DoG) par des différences de moyennes (DoM). Les résultats obtenus sont sensiblement moins bons pour une extraction en moyenne 12 à 14 fois plus rapide.

Ce principe fut repris par **Dalal et Triggs [Dal05]** pour le développement des HOGs (Histogram of Oriented Gradient). Leur stratégie consiste simplement à découper la région d'intérêt en portions élémentaires appelées cellules qui sont ensuite caractérisées par l'histogramme d'orientation de gradient de leur zone recouverte. Cette technique n'a été appliquée qu'au cas particulier de la reconnaissance de piéton (où les contours sont particulièrement saillants), mais elle allie efficacité et simplicité puisque les résultats outrepassent les méthodes déjà existantes pour un temps de calcul restreint. De plus les auteurs ont étudié, dans le cadre de leur application, l'influence de différents paramètres tels que la préférence d'une cellule circulaire à une autre rectangulaire, ou l'importance d'une normalisation préalable.

4.1.2.3 Moments

Il est possible de calculer des invariants en combinant différent moments et leurs propriétés. Comme constaté en **1.1**, les moments centrés sont invariants à la translation, l'invariance aux changements d'échelle peut être obtenue par les moments normalisés ou le rapport de deux moments du même ordre, et, enfin les moments complexes permettent d'obtenir l'invariance à la rotation. Traiter les changements d'illumination est plus délicat. On l'effectue en général en normalisant la variance de l'intensité ou en la soustrayant par la moyenne.

Décrire une région à l'aide d'un groupe de moments invariants soulève aussi la question du choix de leur ordre maximal. En effet, les moments d'ordre élevé sont ceux qui décrivent le plus précisément une région, mais ils sont aussi sensibles au bruit et signifient un temps de calcul plus long. C'est pourquoi les auteurs préfèrent généralement utiliser plus de moments des premiers ordres plutôt que d'avoir recourts à ceux d'ordre élevé.

Les premiers travaux sur les moments invariants furent réalisés par **Hu en 1962 [Hu62]**. En ce basant sur la théorie des invariants algébriques, celui-ci établit le premier théorème des moments 2D invariants qui fut utilisé pour dériver ses sept invariants 2D aux mouvements de rotation :

$$I_1 = \mu_{20} + \mu_{02}$$

$$I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$I_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

$$I_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$I_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})\left((\mu_{30} + \mu_{12})^2 - 3(\mu_{03} - \mu_{21})^2\right) + (3\mu_{21} - \mu_{03})(\mu_{03} + \mu_{21})\left(3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2\right)$$

$$I_6 = (\mu_{20} - \mu_{02})\left((\mu_{30} + \mu_{12})^2 - (\mu_{21} - \mu_{03})^2\right) + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})$$

$$I_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})\left((\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right) - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})\left(3(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2\right)$$

Par la suite, il fut prouvé que ce théorème comportait quelques erreurs, et, depuis les années 80, d'autres méthodes de calcul des moments invariants d'ordre 2 ou 3 furent proposées. Toutefois, aucune de ces méthodes ne permet leur calcul dans un espace de dimension *k* quelconque. De plus, quelque soit la technique utilisée pour les calculs, les moments peuvent être dépendants les uns des autres. En effet, la somme, le produit, et le rapport de deux invariants à une transformation donnée sont aussi des invariants à cette même transformation. Il est donc nécessaire d'établir une règle pour éviter toute redondance. Bien que les méthodes actuelles permettent d'éliminer efficacement les dépendances, leur identification demeure un problème irrésolu.

Un autre inconvénient des moments est qu'il n'existe pas de moment invariant aux transformations affines comme le prouvèrent **Van Gool & al [VGal95]**, dans leurs travaux sur les méthodes de Lie. La région étudiée doit donc d'abord être normalisée pour éliminer les déformations issues du changement de point de vue.

Nous citerons toutefois les moments couleurs généralisés développés par **Mindru & al [Min03]** qui sont une adaptation aux canaux couleurs des moments sur les niveaux de gris. On définit un moment généralisé d'ordre $p+q$ et de degré $a+b+c$ par :

$$M_{pq}^{abc} = \sum_V x^p y^q \left[R(x,y)\right]^a \left[G(x,y)\right]^b \left[B(x,y)\right]^c$$

avec *R(x,y), G(x,y), B(x,y)* les réponses respectives de chaque canal couleur R,G,B pour le point *(x,y)*. Ils caractérisent la forme et la distribution de couleur sur la région considérée d'une manière uniforme. De plus, le fait d'exploiter les canaux couleurs, leur permet, à un même ordre que les autres moments, d'extraire plus d'information de l'image. Ils peuvent donc décrire une région de façon plus précise et le coûteux calcul de moments d'ordre supérieur n'est pas nécessaire.

Les moments invariants ont donc un grand pouvoir discriminant mais au prix de calculs importants. Pour un état de l'art plus détaillé sur les moments invariants, se référer à **[Flu06]**.

4.1.2.4 Invariants différentiels

Le voisinage d'un point peut être décrit par un ensemble de dérivées calculées par convolution de l'image avec des dérivées gaussiennes. Cet ensemble de dérivées est nommé le *local jet* par **Koenderink**

et Van Doorn [Koe87]. Pour une image I, à l'échelle σ, le *local jet* d'ordre N à un point $x=(x_1,x_2)$ est défini par :

$$J^N[I](x,\sigma)=\{\mathrm{L}_{\mathrm{i1},\ldots\mathrm{in}}(\mathrm{x},\sigma)\,|\,(x,\sigma)\in \mathrm{I}\times R^+; n=0,\ldots,N\}$$

, avec $L_{i1,\ldots in}(x,\sigma)$ la convolution de l'image avec la dérivée gaussienne $G_{i1,\ldots in}(x,\sigma)$ relative aux variables $i_1,\ldots,i_k$ (dans une image $i_k \in \{x,y\}$), et σ la taille du lissage gaussien appliqué pendant le calcul des dérivées. Par la suite, **Schmid [Sch97]** utilisa les composants du *local jet* pour déterminer un vecteur V d'invariants différentiels défini par :

$$V[0\ldots8]=\begin{bmatrix} L \\ L_iL_i \\ L_iL_{ij}L_j \\ L_{ii} \\ L_{ij}L_{ji} \\ \varepsilon_{ij}(L_{jkl}L_iL_kL_l - L_{jkk}L_iL_lL_l) \\ L_{iij}L_jL_kL_k - L_{ijk}L_iL_jL_k \\ -\varepsilon_{ij}L_{jkl}L_iL_kL_l \\ L_{ijk}L_iL_jL_k \end{bmatrix}$$

, avec L_i les éléments du *local jet* et ε_{ij} la matrice 2D défini par $\varepsilon_{12=}\,\varepsilon_{21=}1$ et $\varepsilon_{22=}\varepsilon_{11=}0$.
Il faut remarquer que la première composante représente la luminance moyenne, la seconde le carré de la magnitude du gradient, et la quatrième le Laplacien.

Les *filtres complexes* proposés par **Schaffalitzky & al [Sch02]** diffèrent des dérivées gaussiennes par un changement linéaire de la réponse du filtre (Figure 33) et sont issus de l'équation suivante :

$$K_{mn}(x,y)=(x+iy)^m(x-iy)^n G(x,y)\,.$$

Ils sont calculés pour $m+n \le 6$ et $m \ge n$, donnant un total de 16 *filtres complexes*. Leur principale propriété est l'invariance à la rotation. En effet, sous une rotation d'angle θ, les deux quantités complexes $z_1 = x+iy$ et $z_2 = x-iy$ subissent la transformation :

$$z_1 \to e^{i\theta}z_1 \quad et \quad z_2 \to e^{-i\theta}z_{21}$$

On a donc :

$$K_{mn} \to e^{i(m-n)\theta}K_{mn}$$

Afin de pouvoir comparer deux régions ayant subi une rotation, la direction principale considérée est celle pour laquelle la valeur absolue du coefficient $m-n$ est maximale. L'invariance aux changements d'illumination est obtenue en ramenant préalablement la moyenne de l'intensité sur la région considérée à zéro, puis en normalisant l'intervalle de valeurs à 1. Les invariants sont finalement obtenus en prenant la valeur absolue des réponses des filtres.

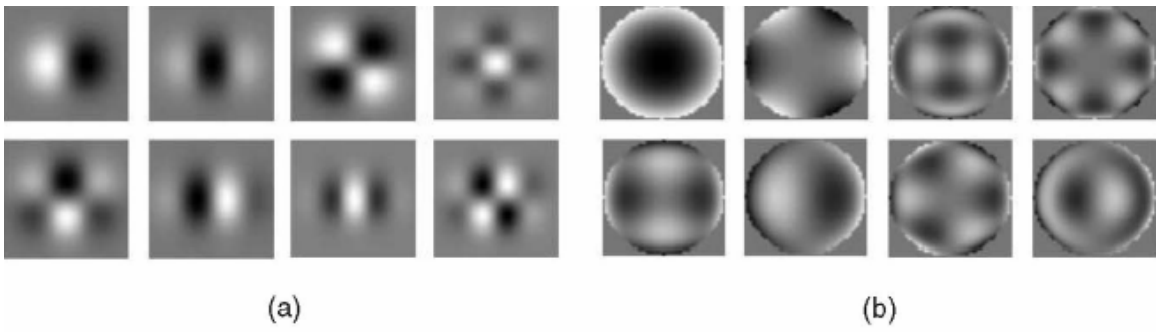


Figure 33: Filtres basés sur des dérivées (a) Dérivées gaussiennes jusqu'à l'ordre 4 (b) Filtres complexes jusqu'à l'ordre 6. **[Sch02]**

Une autre approche pour obtenir une invariance à la rotation serait de trouver la réponse d'un filtre sous toutes les orientations possibles. Une façon triviale d'obtenir ce résultat serait d'appliquer plusieurs versions du même filtre ne différant les unes des autres que par une légère rotation.
Une méthode plus efficace serait de n'appliquer que quelques filtres et d'interpoler entre les réponses. Les *filtres orientables* **[Fre91]** se basent sur ce principe. En utilisant la base de filtres idoine et en appliquant une règle d'interpolation appropriée, il est donc possible de déterminer la réponse d'un filtre sans avoir à la calculer explicitement. Un tel filtre est dit *orientable* et s'écrit :

$$f^{\theta}(x,y) = \sum_{j=1}^{M} k_j(\theta) f^{\theta_j}(x,y)$$

où $f^{\theta_j}(x,y)$ est le j^{eme} filtre de la base de M filtres et $k_j(\theta)$ la fonction d'interpolation associée. Le rôle de ces fonctions est de pondérer les filtres de la base pour obtenir le filtre orienté voulu. L'architecture présentée est donnée en Figure 34.

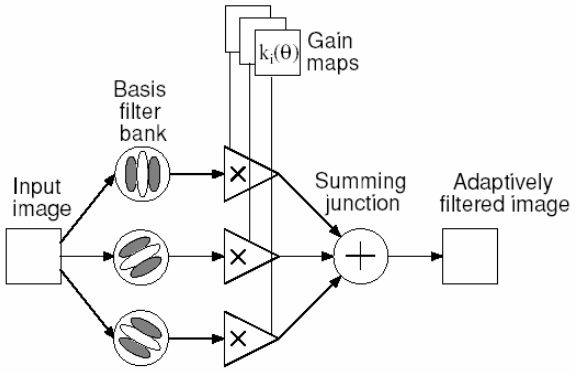


Figure 34: L'architecture des filtres orientables. **[Fre91]**

Bien que de nombreuses fonctions soient orientables, les bases choisies se composent généralement de dérivées gaussiennes car elles constituent un outil efficace dans de nombreux domaines d'analyse d'images et sont séparables selon l'axe x-y. Il est également possible de définir des pyramides de filtres orientables en appliquant, à plusieurs échelles, une même base de filtres sur une image. Comme lors d'une transformée en ondelettes, tous les filtres ne sont alors que des rotations et des étirements d'un seul et unique filtre. Les filtres orientables offrent l'avantage de demander peu de calculs tout en donnant des résultats assez précis. Dans sa comparaison des descripteurs, **Mikolajczyk [Mik03]** les présenta comme un bon compromis entre rapidité et efficacité.

4.2 Contribution

4.2.1 Détection de points d'intérêts

Cette section décrit le modèle d'objet adopté pour notre système de suivi. Nous justifierons le choix des points-clés et amènerons progressivement la façon dont nous décrivons un objet au travers des différentes limitations inhérentes à un système de suivi se reposant sur des points-clés et des améliorations que nous avons choisi pour leur répondre. Pour finir, la dernière partie décrit un système accéléré d'extraction de points-clés.

4.2.1.1 Choix des points-clés

Le choix des points-clés à utiliser pour notre système de suivi est fonction de leur efficacité et de leur temps de calcul. Les points de Harris **[Har88]** détaillés en **4.1.1.1.2** dans la littérature offrent ces deux avantages. De plus, suite à leur large utilisation dans la littérature, de nombreuses variantes et améliorations ont été développées. Dès lors, plusieurs questions, ce sont posées à nous. Tout d'abord le choix de l'utilisation de l'information couleur. Dans le cadre d'un système de suivi générique, tous types de vidéos doivent être considérées. Or, certaines séquences, telles que des images comportant des plages de couleurs différentes mais d'intensité identique, imposent l'utilisation de la couleur pour garantir une extraction correcte des points-clés. Nous avons donc opté pour des points de Harris couleur **[Mon98]**. La seconde variation envisageable concernait le rapport entre la fiabilité et le coût algorithmique du détecteur. En effet, le détecteur de Harris-Laplace **[Mik01]** extrait des points beaucoup plus robustes et moins nombreux mais au prix de calculs supplémentaires. Aucun argument ne nous permettant de choisir une variante plutôt qu'une autre, nous avons conservé les deux types de détecteurs pour nos expérimentations jusqu'à ce qu'une limitation de l'un ou de l'autre lors de l'exécution d'un de nos algorithmes nous permette de choisir.

4.2.1.2 Prétraitement : adaptation de l'algorithme à la couleur

La couleur étant une information riche et immédiatement disponible, de nombreux algorithmes de traitement d'image se basent sur ce trait. Toutefois, les techniques considèrent généralement les espaces couleurs utilisés (RGB, HSV,…) comme étant d'importance égale. Dans la pratique, les couleurs sont fréquemment réparties de façon inégale, ce qui conduit à des canaux plus discriminants que d'autres. Par exemple, la Figure 35 montre des exemples d'images où une couleur est prédominante. L'information est donc concentrée sur un ou deux canaux et une exploitation équilibrée de chacun des canaux minimisera donc celle-ci. Par contre, si on accorde à chaque canal une importance proportionnelle à sa discrimination, l'information extraite est plus riche et les performances de l'algorithme utilisé s'en trouvent accrues. Plus précisément, on cherche à définir pour chaque canal couleur i un poids $P(i)$ représentant son importance, tel que.

$$\sum_{i=1}^{n} P(i) = n$$

, avec n le nombre de canaux couleurs. Bien que cette méthode ait été développée pour un système de suivi avec l'espace colorimétrique RGB, elle est applicable à tout type d'algorithme se basant sur la couleur.

Afin de comparer l'importance relative de chaque canal couleur, nous avons défini deux caractéristiques susceptibles d'influer sur celle-ci : sa taille et sa saillance. La taille représente l'étendue et l'intensité du canal de couleur dans l'image. La taille T*(i)* d'un canal *i* est égale à la somme des intensités des pixels de l'image *Im* pour ce canal couleur:

$$T(i) = \sum_{p \in \mathrm{Im}} I(p)$$

La saillance modélise l'attraction visuelle d'un canal couleur. Elle se caractérise par un histogramme prononcé sur une ou plusieurs valeurs (présence de pic(s)). Afin de déterminer cette grandeur, nous calculons le Kurtosis pour chacun des canaux. Le Kurtosis *K(i)*, ou moment centré de degré 4 représente le degré d'aplatissement d'une distribution. Un Kurtosis inférieur à zéro représentera une distribution plate alors qu'un Kurtosis supérieur à zéro caractérisera une distribution pointue. Plus formellement :

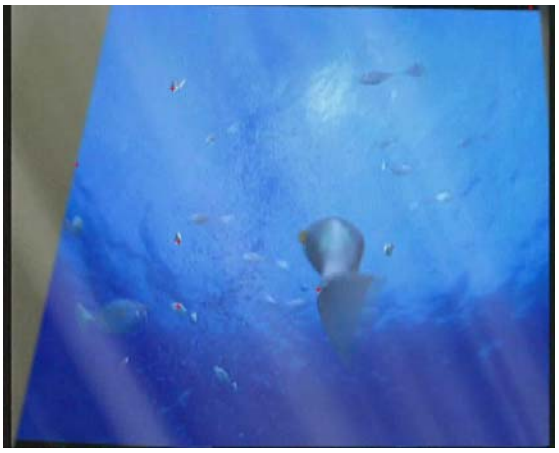
$$K(i) = \frac{\sum_{x=0}^{n} (x - Ci)^4 h(x)}{\left(\sum_{x=0}^{n} (x - Ci)^2 h(x)\right)^2} - 3 \qquad K(i) \in [0, +\infty[$$

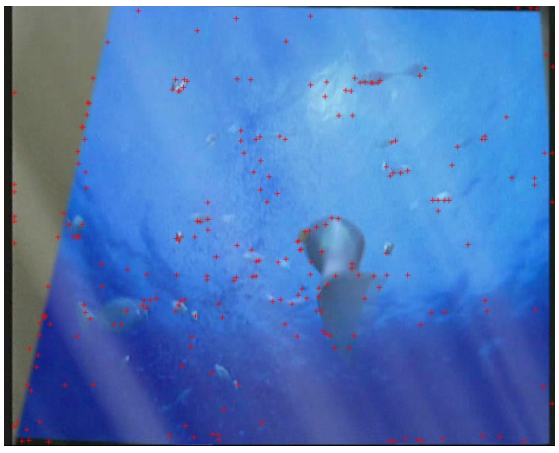
, avec *n* le nombre de colonnes (255 dans le cas de l'espace de couleur RGB) de l'histogramme considéré, *h(x)* la x$^{\text{ième}}$ valeur de la distribution, et *Ci* le centre de la distribution. Le poids *P(i)* associé à chaque canal couleur est ensuite obtenu en combinant la taille et la saillance par la formule :

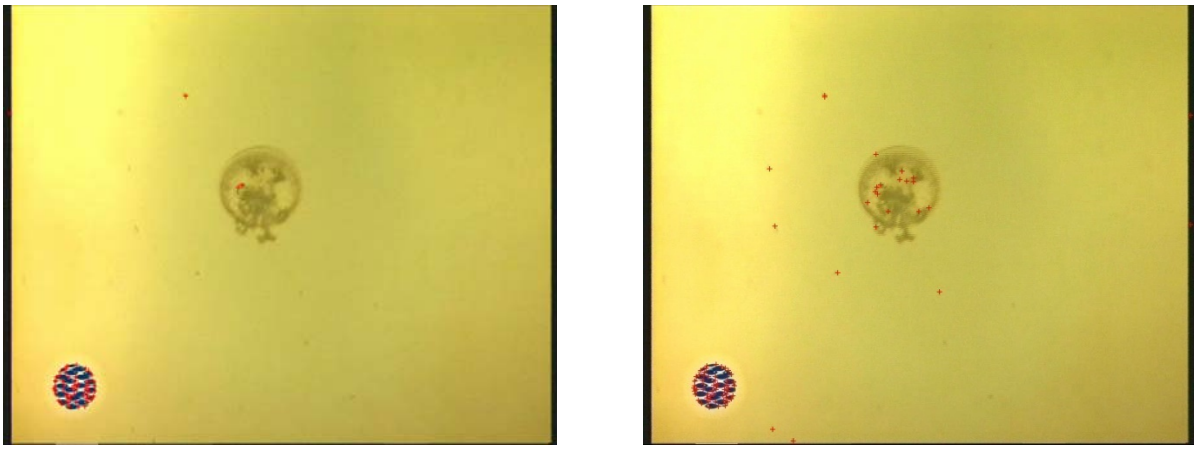
$$P(i) = T(i) \times (1 + K(i))$$

Cette combinaison est conçue afin de favoriser les canaux répondants à ces deux caractéristiques. Pour finir, les poids sont normalisés afin de satisfaire la contrainte initiale $\sum_{i=1}^{n} P(i) = n$

Nous avons utilisé cet algorithme comme un prétraitement sur notre système de suivi se basant sur des points de Harris couleur. Les résultats sont présentés en Figure 35.







(a) (b)

Figure 35: Points de Harris extraits en rouge (a) Sans adaptation des canaux couleurs à l'image (b) Avec adaptation des canaux couleurs à l'image.

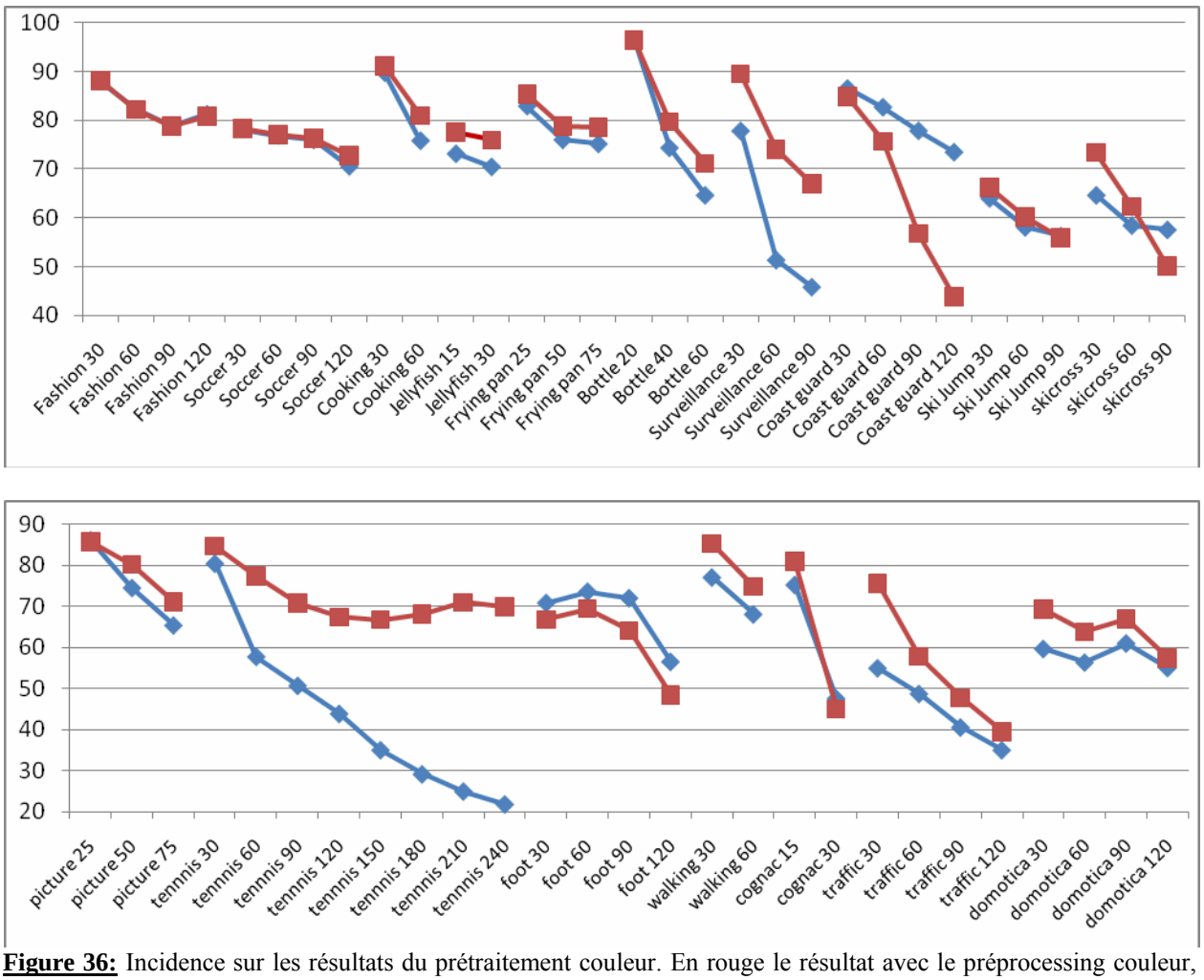


Figure 36: Incidence sur les résultats du prétraitement couleur. En rouge le résultat avec le préprocessing couleur, en bleu sans le préprocessing couleur. Tests effectués sur 17 vidéos. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne (en pourcentage de recouvrement) du suivi sur le nombre d'images indiquées. La valeur en abscisse représente la séquence vidéo testée ainsi que le nombre *n* d'images traitées.

La valeur moyenne des résultats sans le prétraitement couleur est de 65.29%, et de 71.29% avec le préprocessing couleur, validant ainsi l'apport de ce traitement. Le temps moyen d'exécution est d'une seconde.

4.2.1.3 Instabilité temporelle

Nous avons vu au paragraphe **3.1** que l'efficacité d'un détecteur de points-clés peut être mesurée par sa robustesse et sa répétabilité. Toutefois cette dernière mesure a rarement été utilisée dans le cadre de vidéos et pour des transformations usuelles bien définies (translation, rotation, changement d'échelle,…). A l'issue de nos expériences sur l'utilisation de points-clés pour le suivi, nous avons également constaté une instabilité temporelle de ceux-ci. En effet, dû à diverses petites altérations locales propres aux vidéos (flous issus des mouvements, de la qualité de la vidéo ou des défauts des capteurs), de nombreux points sont susceptibles de disparaître pendant une ou plusieurs images pour ensuite réapparaître. Afin de palier à ce désavantage, notre modèle conserve les points pendant *k* images, i.e. si un point du modèle n'est pas apparié pendant *k* images, alors il est éliminé du modèle. Il en résulte une augmentation du taux d'appariement en fonction du temps de conservation des points (Figure 37). Toutefois, plus un point est conservé longtemps sans être apparié, plus son descripteur reste longtemps sans être mis à jour, plus le risque d'appariement erroné augmente. En conséquence, si les points sont conservés trop longtemps, le nombre d'appariement sera plus important, mais la précision du suivi n'en sera pas pour autant meilleure (Figure 38). De plus, une longue conservation des points conduit à un ensemble de points du modèle de plus grande cardinalité, donc un algorithme plus lent. Nous avons donc choisi *k=3*, offrant un bon compromis entre précision du suivi et temps de calcul.

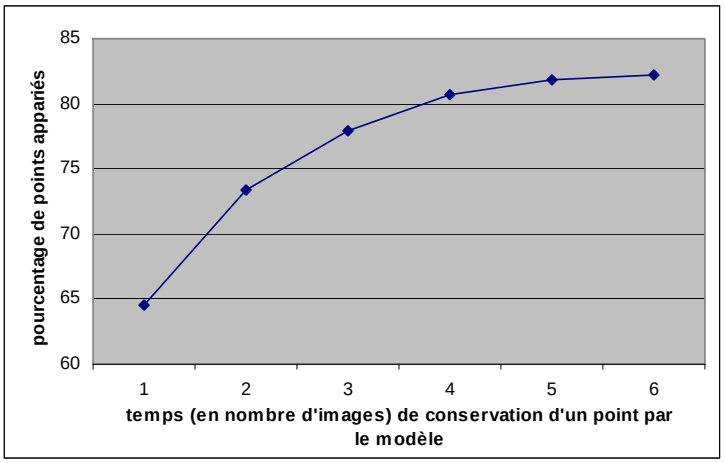


Figure 37: Pourcentage de points appariés en fonction du temps de conservation des points.

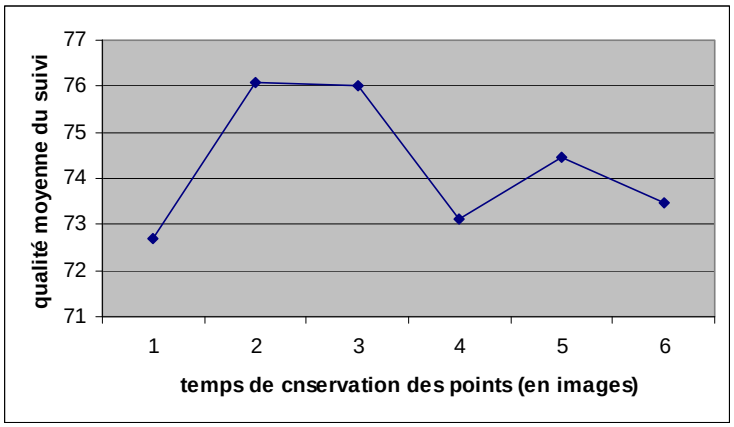


Figure 38: Qualité du suivi en fonction du temps de conservation des points.

4.2.1.4 Labellisation des points

Afin de localiser l'objet d'une image à l'autre les algorithmes de suivi se doivent de définir, pour l'image de référence *t*, la zone représentative de l'objet dont on va extraire les informations nécessaires à l'identification de l'objet dans l'image *t+1*. Si cette zone est un masque précis délimitant parfaitement l'objet, la totalité de l'information obtenue sera fiable. Toutefois les méthodes permettant d'obtenir une telle frontière, comme la segmentation **[Cav05]**, les « snakes » **[Tec01]**, ou des techniques basées maillage **[Val04]**, ne sont efficaces ou ne fonctionnent que dans des cas de figures restreints (contours marqués, forts contrastes entre l'objet et le décor). C'est pourquoi la majorité des méthodes existantes localisent grossièrement l'objet, généralement à l'aide d'une boîte englobante. La surface couverte par l'objet sera donc souvent inférieure à celle de la boîte englobante. Une partie de l'information extraite ne sera donc pas celle de l'objet et perturbera le fonctionnement des algorithmes. On parle de *distracteurs*. Plus la proportion de *distracteurs* sera grande, plus les performances des algorithmes chuteront. Deux facteurs déterminent cette quantité. Tout d'abord, la proportion de la surface de la boîte englobante n'appartenant pas à l'objet, fonction de la compacité de l'objet. Ensuite, l'encombrement de l'arrière-plan (ou de l'avant-plan dans le cas d'occultations) par d'autres objets. Ces zones du décor ont des couleurs distinctes, des contours, ou encore de forts contrastes, représentant une forte concentration d'information erronée qui influe de façon conséquente sur le résultat.

Les algorithmes d'estimation de paramètres **[Zha95][Ste99][Mal06]** sont généralement capables de gérer une certaine quantité de *distracteurs*. Toutefois, dans le cas de suivi d'objet, particulièrement lorsque celui-ci est déformable, son mouvement peut être sujet à d'importantes variations internes. Et il est difficile d'établir un modèle de mouvement prenant en compte ces fluctuations qui sont donc traitées par les algorithmes comme du bruit ou des *distracteurs* suivant leur magnitude. Les modèles de mouvement du décor et de l'objet deviennent alors ardus à distinguer et leurs informations respectives sont souvent amalgamées, perturbant le bon déroulement du suivi.

En définitive, le traitement des *distracteurs* reste donc un problème ouvert dans le domaine du suivi d'objet. Généralement traité comme du bruit par les algorithmes, peu de travaux ont été effectués afin de limiter préalablement leur influence. Une astuce couramment employée consiste à utiliser une boîte englobante ellipsoïdale plutôt que rectangulaire (voir Figure 39) plus proche de la forme de la plupart des objets. Mais cela reste insuffisant dans la plupart des cas.

Figure 39: Boîtes englobantes rectangulaire et ellipsoïdales d'un objet. La boîte englobante ellipsoïdale a une plus grande proportion de pixels de l'objet.

Dans notre cas particulier d'un suivi basé sur des points-clés, une phase de labellisation survient entre l'appariement et l'évaluation du déplacement global de l'objet. L'idée est similaire à celle de l'association de données **[Bar88]**. Nous cherchons à différencier les points de l'objet de ceux de l'arrière-plan. Nous leur attribuons un label afin de spécifier si ce sont des points « objet » ou des points « décor ». Seul les points labellisés « objet » seront ultérieurement pris en compte pour évaluer le mouvement de l'objet. Un algorithme classique considèrerait les points à l'intérieur de la boîte englobante comme appartenant à l'objet et les points extérieurs à la boîte englobante comme solidaires du décor. Un tel procédé est sujet aux inconvénients énoncés plus haut. Nous avons donc tenté de pousser plus loin l'analyse des points présents afin de juger de leur probabilité d'appartenir à l'objet ou au décor. Le problème peut se résumer ainsi : étant connus deux ensembles de points A et B respectivement issus des images successives t et $t+1$, les appariements entre ces deux ensembles de points, et les labels des points de A, déterminer les labels des points de B. On va donc obtenir deux sous-ensembles de points labellisés : B_O, les points de l'objet, et B_D, les points du décor. Le label L_p d'un point p sera sa probabilité ($0 \leq L_p \leq 1$) que celui-ci fasse partie de l'objet. Tout point dont le label sera supérieur à 0.5 sera considéré comme partie intégrante de l'objet. A l'inverse, les points de label inférieur à 0.5 seront estimés appartenant au décor.

Il est tout d'abord important de constater que le mouvement de la boîte englobante ne sera estimé que d'après les points appariés. L'affectation d'un label aux autres points peut donc être repoussée si l'on ne dispose pas de suffisamment d'information (sans appariement, l'information de mouvement est indisponible) pour aboutir à une décision fiable. En conséquence, on ne disposera pas nécessairement du label de tous les points de A, et tous les points de B ne seront pas forcement labellisés.

Ces deux ensembles de points B_O et B_D sont déterminés en fonction de leur homogénéité par rapport à certaines caractéristiques. Le choix et la combinaison de ces caractéristiques, ainsi que le procédé d'évaluation de leur homogénéité (local ou global) est capital sur la discrimination de la classification. Nous avons expérimenté plusieurs traits susceptibles de servir ce dessein. Nous présentons ici leur intérêt ainsi que les techniques élaborées pour les exploiter au mieux :

L'appariement du point : Si un point est apparié et que son correspondant est déjà labellisé, la probabilité que son label soit le même est donc très élevée. Deux points associés n'auront réellement un label différent que dans le cas d'un faux appariement. Dans ce cas, affecter de manière certaine le même label propagera irrévocablement l'erreur. La question d'une influence partielle ou totale de l'appariement sur l'affectation du label se pose donc. On opte toutefois généralement pour une confiance absolue en l'appariement de façon à avoir un premier groupe de points servant de référence pour déterminer le label des autres points.

La couleur : la couleur est une information facile à exploiter et souvent discriminante. Elle constitue donc un choix privilégié. De plus, dans notre cas un descripteur est associé à chaque point. Deux variantes sont possibles : la classification globale des points en deux groupes selon les descripteurs couleurs ou la classification de chaque point indépendamment par comparaison avec les *k* plus proches voisins labellisés « objet » et les *k* plus proches voisins labellisés « décor ». Nos expériences ont rapidement montré que la première possibilité est à bannir. En effet, elle se base sur l'hypothèse d'homogénéité des couleurs de l'objet et du décor. Or, dans la pratique, ils peuvent être constitués de plusieurs couleurs caractéristiques. De plus, une des couleurs de l'objet peut être plus similaire du décor que du reste de l'objet, conduisant à une classification désastreuse. Nous avons donc choisi une évaluation locale de la similarité basée sur la couleur. Après la comparaison du descripteur du point à ceux des *k* plus proches voisins « objet » et « décor », on obtient deux valeurs de similarité, S_O et S_D. La probabilité P_C que le point fasse partie de l'objet estimée d'après la couleur est ensuite donnée par la formule :

$$P_C = S_O/(S_O+S_D)$$

Le paramétrage de *k* est un compromis entre la quantité d'information récoltée et sa fiabilité. En effet, *k* élevé signifiera un grand nombre de points utilisé pour la mesure. Toutefois, l'objet peut très bien être constitué de plusieurs couleurs et n'être homogène que localement. Donc, si *k* est trop important, certains points utilisés seront trop éloignés du point à évaluer et leur information risque de ne plus être discriminante. Nous avons fixé *k* à 3 pour équilibrer ces deux influences.

Le mouvement : Par définition, le mouvement de l'objet est indépendant de celui du décor. Ce trait peut donc également servir. Ici aussi, nous nous sommes interrogés sur le choix d'une évaluation locale ou globale. Le principe d'une estimation locale est, tout comme pour la couleur, une comparaison aux *k* voisins les plus proches pour chaque label. Cependant, la localisation des points est imprécise, et, si pour un ensemble de points de grande cardinalité, l'estimation du mouvement est fiable, ce n'est pas le cas pour un petit sous-ensemble. Les résultats qui en découlent sont donc faussés. En conséquence, on préférera une estimation globale qui implique de trouver deux vecteurs de mouvement, le premier correspondant au mouvement du décor, le deuxième à celui de l'objet. La probabilité qu'un point appartienne à l'objet ou au décor d'après son déplacement sera donc fonction de sa similarité avec ces deux vecteurs. Dans la pratique, le mouvement de l'objet est déjà calculé pour chaque image. Seul le mouvement du décor reste à évaluer à l'aide des points dont le label est déjà connu. Sa mesure doit seulement permettre de le distinguer de l'objet, et ne nécessite donc pas une précision importante. Le déplacement du décor au voisinage de l'objet est donc assimilé à une translation. Il est calculé par une moyenne élaguée afin d'éliminer les valeurs bruitées (une version simplifiée du calcul du mouvement de l'objet décrite en **5.2.2.1**). Après calcul des écarts types σ_x et σ_y aux mouvements du décor m_x et m_y (estimés à l'image précédente) selon les axes respectifs *X* et *Y*, on élimine tout point *(x,y)* qui ne satisfait pas la condition suivante :

$$(m_x - x) < v\sigma_x \quad et \quad (m_y - y) < v\sigma_y$$

, avec *v* la variabilité de mouvement tolérée. Le temps entre deux images étant faible, les mouvements de l'objet et du décor sont fréquemment proches, nous avons donc fixé v à 1 afin de restreindre le plus possible le nombre de points de l'objet dans l'évaluation du mouvement du décor. Avec m_O, m_D, et m_p les mouvements respectifs de l'objet, du décor et du point, la probabilité P_M que le point fasse partie de l'objet d'après son mouvement est donnée par :

$$\mathrm{P_M} = \left(\mathrm{P_M^X} + \mathrm{P_M^Y}\right)/2$$

$$\mathrm{P_M^a} = \mathrm{abs}(\mathrm{m_O} - \mathrm{m_p})/(\mathrm{abs}(\mathrm{m_O} - \mathrm{m_p}) + \mathrm{abs}(\mathrm{m_D} - \mathrm{m_p}))$$

Avec *a* l'axe considéré et *abs()* la fonction valeur absolue.

La position du point : La position du point s'est révélée être le critère décisif pour l'évaluation de son label. De nombreuses méthodes, locales et globales furent explorées. En premier lieu, la position du point par rapport à son voisinage. Nous avons envisagé l'utilisation de la triangulation de Delaunay décrivant

les relations de voisinage pour déduire le label du point de celui de ces voisins à l'image précédente. Toutefois, dans de nombreux cas de figure, peu de points voisins sont labellisés, ou les points voisins ont tous le même label, ce qui ne nous permet pas de juger de manière équitable l'appartenance à chacune des deux catégories. Nous avons donc essayé une comparaison se basant sur la distance spatiale avec les *k* plus proches voisins de l'image précédente pour chacun des deux labels possibles. Cependant, ce procédé ne tient pas compte de la notion d'intériorité (voir Figure 40).

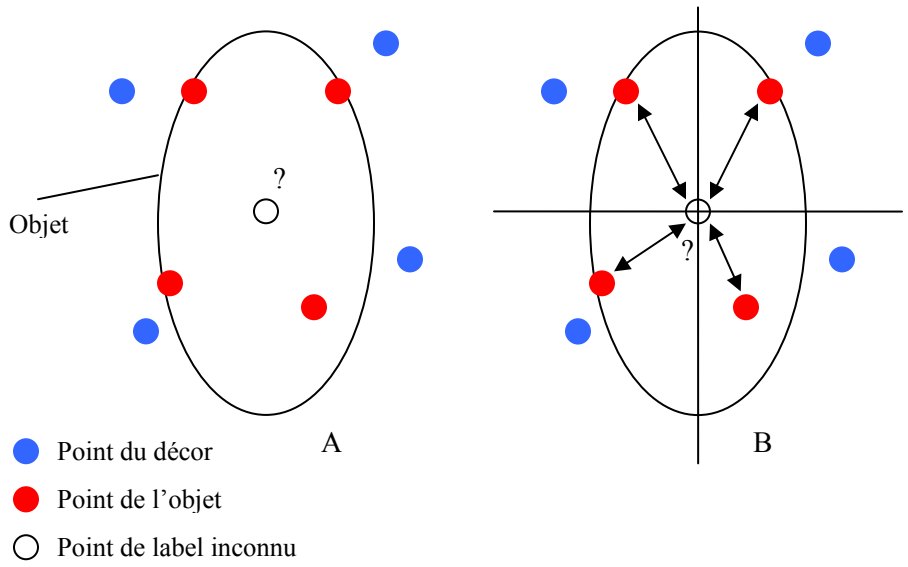


Figure 40: Caractérisation de la notion d'intériorité. Le point au label inconnu, entouré de point de l'objet fait partie de l'objet. (A) Le test des labels basés sur la distance spatiale avec les k plus proches voisins ne prend pas en compte cette notion. (B) L'interpolation bilinéaire sur la valeur du label du point le plus proche de chacun des quatre quadrants issus du point à estimer modélise cette notion.

Pour pallier à ce défaut, nous avons divisé le voisinage du point en quatre quadrants, et effectué une interpolation bilinéaire de la valeur du label entre quatre points, chacun étant spatialement le plus proche du point étudié dans son quadrant. Mais là encore, nous nous sommes heurtés à une défaillance. Ce procédé n'était pas assez stable. Le système, trop sensible aux erreurs, finissait invariablement par dévier de la trajectoire optimale. Nous nous sommes donc orientés vers une analyse globale du mouvement.

Ce système se base sur l'hypothèse que le label des points n'est incertain que pour une petite partie des points. Les points largement dans l'intérieur de la boîte englobante sont considérés comme des points de l'objet en proportion à leur proximité du centre de celle-ci. De même, plus les points sont éloignés du bord de la boîte englobante, plus ils sont estimés faire partie du décor. Ainsi, seule une ceinture au niveau du cadre sera étudiée (voir Figure 41). La position du point n'étant plus discriminante dans cette zone, l'évaluation du label du point est laissée à d'autres critères. La définition de ces trois zones est cruciale pour le bon fonctionnement de l'algorithme car elle permet d'induire deux ensembles initiaux de points « objet » et « décor » qui serviront de référence pour évaluer les points de la zone incertaine. De plus, des zones labellisées de façon constante stabilisent l'algorithme en limitant la propagation des erreurs de jugement. Deux paramètres entrent en jeu : les distances extérieure et intérieure de la ceinture au bord de la boîte englobante. La boîte englobante recouvrant entièrement l'objet, la première peut être fixée de façon constante (dans notre cas, 4 pixels). En revanche, la limite intérieure variera en fonction de la forme de l'objet suivi et de la quantité de *distracteur*s dans son voisinage.

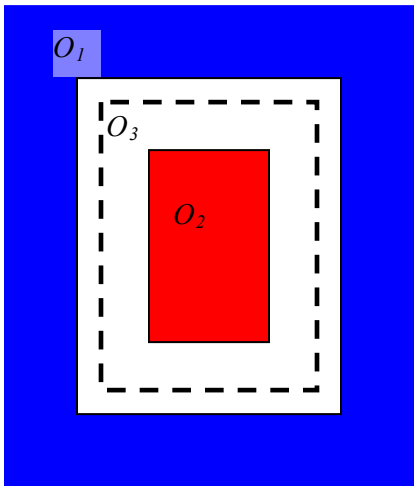


Figure 41: Labellisation des points par rapport à leur position. En pointillé la boîte englobante. En bleu la zone O_1 où les points sont considérés comme faisant partie du décor. En rouge la zone centrale O_2 où les points sont jugés appartenir à l'objet. Seule les points de la zone blanche O_3 restent incertains.

Nous avons donc tiré enseignement de ces expériences pour créer la méthode de labellisation des points suivante. Soit, pour un point p donné, $P_C(p)$, $P_M(p)$, $P_P(p)$, et $P_A(p)$ les probabilités que le point soit jugé appartenir à l'objet se basant respectivement sur la couleur, le mouvement, sa position, et le point auquel il est associé. Soit $L(p)$ le label de p. Soient O_1, O_2, les zones de points jugés appartenir respectivement au décor, à l'objet, et O_3 la ceinture de points dont le label ne peut être déduit de la position. Notre algorithme est donc le suivant :

Première image :
Pour tout point p $L(p) = P_P(p)$

Autres images :
1- Calcul du mouvement du décor d'après les points de O_1
2- Pour tout point p
Si $p \in O_1$ *ou si* $p \in O_2$, *alors*
$$L(p) = (W_{1C}P_C(p) + W_{1P}P_P(p) + W_{1A}P_A(p))/(W_{1C} + W_{1P} + W_{1A})$$
Sinon
Si p est apparié
$$L(p) = (W_{2C}P_C(p) + W_{2M}P_M(p) + W_{2A}P_A(p))/(W_{2C} + W_{2M} + W_{2A})$$
Sinon L(p) n'est pas calculé

W_{1C}, W_{1P}, W_{1A}, W_{2C}, W_{2M}, W_{2A} sont les poids associés aux probabilités de jugement se basant respectivement sur la couleur, la position, le point associé et le mouvement pour chacune des deux évaluations. Nous utilisons la paramétrisation contrainte suivante : $W_{1C} = 1$, $W_{1P} = 3$, $W_{1A} = 2$, $W_{2C} = 1$ $W_{2M} = 1$, $W_{2A} = 1$.

Un exemple est présenté en Figure 42.

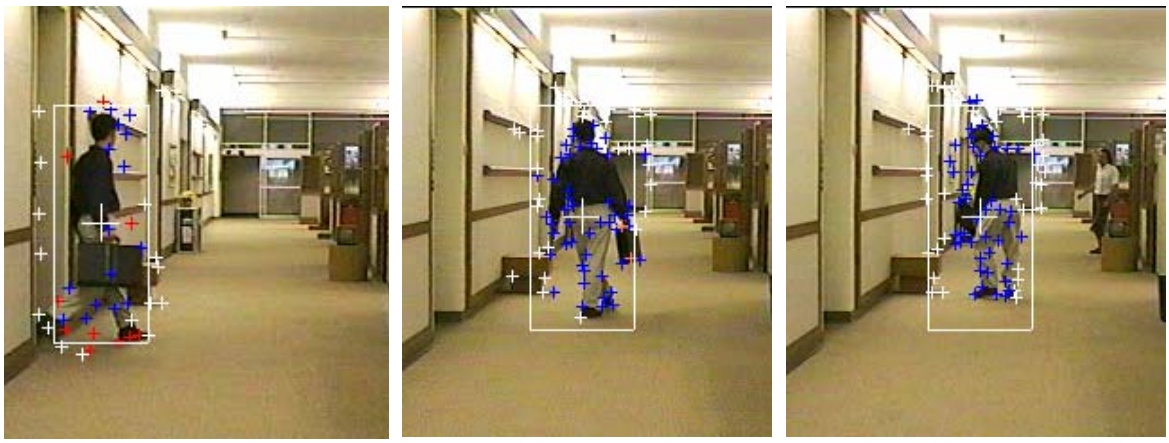


Figure 42: Labellisation des points pour les images 2, 30 et 60 de la séquence « surveillance ». En bleu les points-clés labélisés « objet », en blanc les point-clés « décor » et en rouge les points-clés au label indéterminé.

Cependant, l'objet évolue parfois dans un environnement non encombré. Prenons l'exemple d'un joueur de foot courant sur une pelouse uniformément verte. La quasi-totalité des points-clés détectés appartiennent alors à l'objet. Dans ce cas de figure, l'algorithme classique présenté au début de ce chapitre est mieux adapté que notre technique d'étiquetage des points.

Le choix de l'algorithme dépend entièrement d'une détection préalable du taux de *distracteurs*. Nous la déterminons en fonction de la proportion *tauxDist* de points appariés *nbDec* dont leur antécédent est labellisé « décor » par rapport au nombre total de points appariés *nb*. Les deux facteurs susceptibles de détériorer la qualité de la mesure sont un trop faible nombre de points et leur instabilité temporelle (phénomène détaillé en **4.2.1.3**). Une estimation d'après plusieurs images antérieures (l'importance accordée à l'information décroissant en fonction de son ancienneté) permet de contrebalancer l'instabilité temporelle des points. Quant aux mesures se basant sur un trop faible nombre de points, elles ne sont tout simplement pas prises en compte. L'algorithme de décision est le suivant :

Si (tauxDist = 0) *tauxDist =nbDec/nb;*
Sinon
Si (nb>10) *tauxDist = (tauxDist +nbDec/nb)/2;*
Si (tauxDist >SeuilE et nbDec>1 et nb>nbDec+1)
Lancer notre algorithme de labellisation
Sinon
Lancer l'algorithme classique de labellisation

La variable *SeuilE* représente la quantité de *distracteurs* tolérée. Ce test est effectué à chaque image, ce qui permet de détecter le passage d'une zone homogène à une zone où le décor est encombré.

L'inconvénient majeur de cet algorithme découle de la définition d'une zone centrale à la boîte englobante qui est étiquetée « objet ». En effet, ce procédé nuit gravement à la détection des occultations. Néanmoins, notre algorithme de suivi, pour être générique, s'appuie sur l'hypothèse nécessaire du traitement d'objets hautement déformables. En conséquence, si les occultations partielles sont gérées efficacement, les occultations totales sont assimilées à une déformation de l'objet et doivent être détectées par un mécanisme indépendant. Cet inconvénient n'en est donc pas un puisque le problème était déjà présent.

Dans le cas particulier d'un suivi développé spécifiquement pour l'annotation vidéo, la labellisation des points-clés soulève un deuxième désavantage. L'appariement des points-clés, effectué jusqu'ici lors d'une étape préalable, repose désormais sur l'information issue de la labellisation. Or, l'obtention de cette information implique la connaissance de la position de la boîte englobante qui n'est

disponible que pendant le suivi. En d'autres termes, l'application de la labellisation interdit l'appariement off-line des points-clés.

L'efficacité de cet algorithme est dépendant de deux paramètres. Le premier est la taille *T* de la zone analysée O_3 par rapport à celle de la boîte englobante. Elle représente la compacité de l'objet. Un livre épousant parfaitement les contours rectangulaires de la boîte englobante nécessitera un paramétrage de *T* proche de 0%. A l'inverse, le suivi d'un objet hautement déformable, tel qu'une personne en pleine course impliquera une grande zone d'incertitude quand au label des points-clés. Une paramétrisation optimale de *T* devrait donc se faire en fonction de la compacité de l'objet. Toutefois nous sommes dans le cas générique et ne disposons pas de connaissances sur l'objet ni d'un moyen d'en estimer la compacité. Nous avons donc déterminé expérimentalement la taille *T* optimale de la zone d'analyse sans connaissances a priori sur l'objet. Elle est mesurée par rapport à la plus grande dimension (hauteur, largeur) de la boîte englobante. Le jeu d'essai est constitué de 6 séquences vidéos à l'arrière-plan encombré : « cooking », « cognac », « frying pan », « Bottle », « surveillance », et « coast guard ». Les résultats sont présentés en Figure 43. Nous avons choisi la taille de la ceinture analysée à 20% de la boîte englobante.

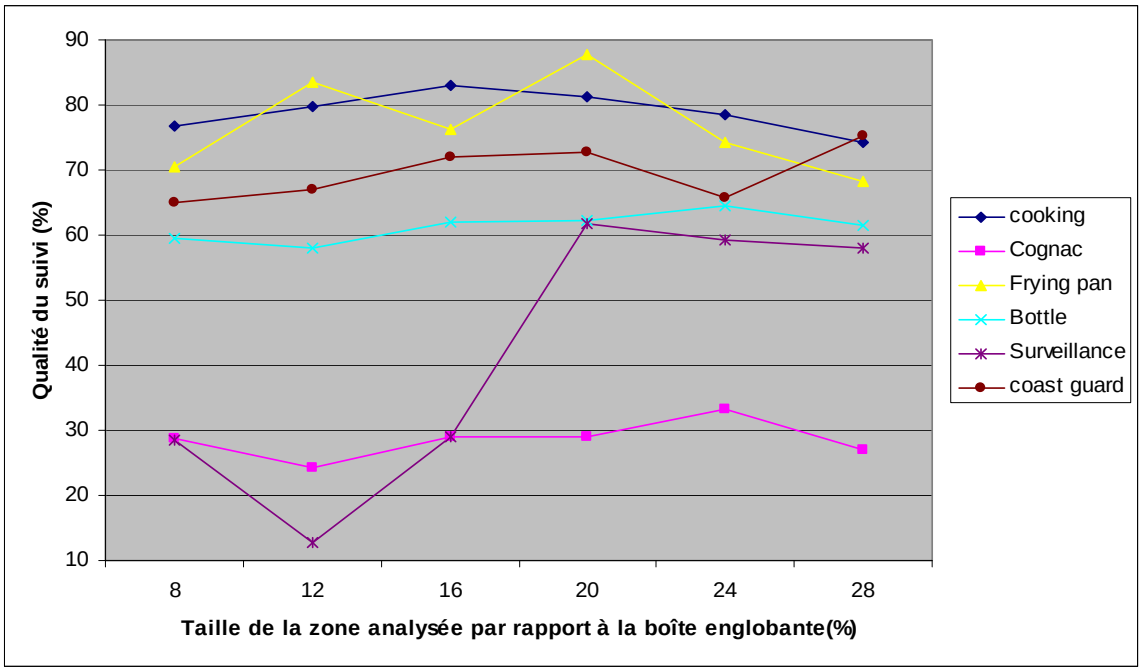


Figure 43: Influence de la taille *T* de la ceinture analysée O_3 sur la qualité du suivi.

La seconde variable déterminante dans le bon déroulement de l'algorithme est la quantité de *distracteurs* tolérée *seuilE*. Ce seuil marque la limite supposée de l'évolution de l'objet entre un milieu uniforme et un milieu encombré. Il déclenche en conséquence l'application de l'algorithme de labellisation approprié. Afin de trouver la valeur optimale de *seuilE*, nous avons effectué des tests de qualité de suivi pour différentes valeurs de ce seuil sur les 10 séquences vidéos encombrées ou non suivantes : « fashion », « soccer », « cooking », « cognac », « jellyfish », « frying pan », « Bottle », « surveillance », « coast guard », et « skijump ». La Figure 44 récapitule les résultats.

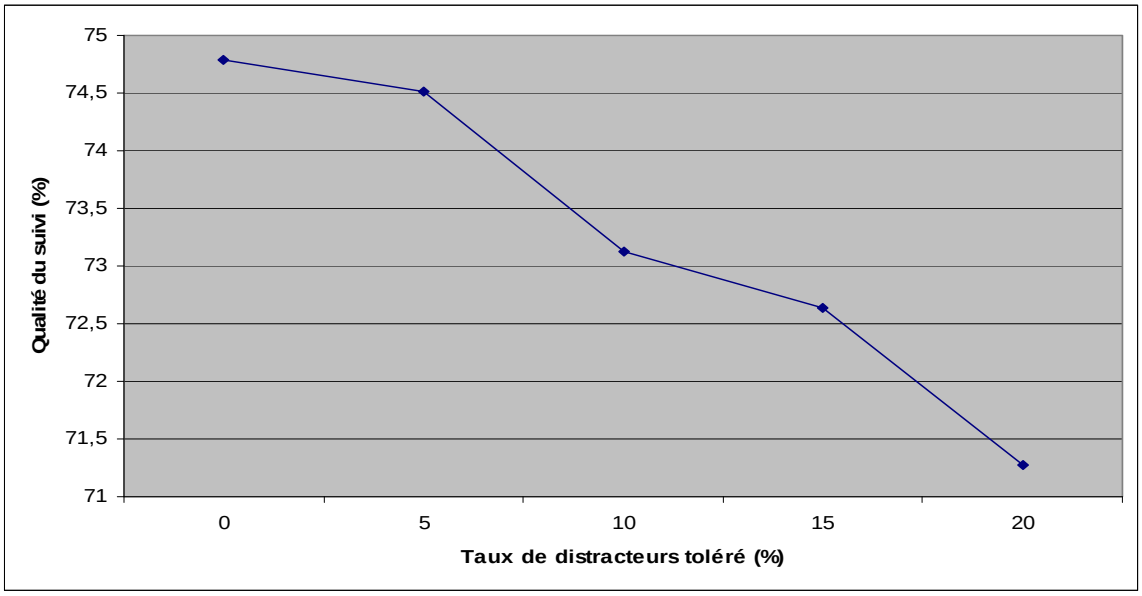


Figure 44: Influence du taux de distracteurs toléré seuilE sur la qualité du suivi. Moyenne de la qualité du suivi calculée sur 10 séquences vidéos.

Notons qu'un taux de *distracteurs* toléré de 0% signifie une application permanente de notre algorithme de labellisation, alors qu'une tolérance supérieure au taux de *distracteurs* équivaudra à une labellisation élémentaire décrite en début de paragraphe. En conséquence de quoi, cette expérience valide de façon significative notre algorithme puisque l'accroissement de la qualité du suivi est proportionnel à la fréquence de l'application de notre système de labellisation. Nous avons adopté une tolérance de 5% de *distracteurs*, un seuil inférieur impliquant des calculs inutiles pour une amélioration non significative de 0.2% des résultats.

4.2.1.5 Délai d'utilisation des points-clés

Le modèle de points-clés présenté jusqu'ici comporte certains désavantages. En effet, si la mise á jour fréquente des descripteurs et l'ajout systématiques des nouveaux points-clés nous permet de gérer d'importantes déformations susceptibles de survenir, elle le rend vulnérable aux occultations. De plus, cette vulnérabilité est accentuée par la labellisation (voir le chapitre précédent **4.2.1.4**). Il est également possible d'améliorer le modèle en pondérant les points-clés en fonction de leur fiabilité. La fiabilité des points-clés dépend, dans notre modèle, de l'ancienneté de leur denier appariement ainsi que de leur fréquence d'appariement. En effet, le risque de faux appariement pour les points-clés associés fréquemment ou dont l'information est récente étant plus faible, ceux-ci doivent être privilégiés lors de l'étape d'appariement.

Afin de pallier à ces lacunes, nous avons introduit un délai d'utilisation des points-clés. Le principe consiste à n'ajouter un point-clé au modèle qu'après *n* frames d'observations afin de garantir l'appartenance du point ajouté à l'objet et non au décor ou à un éventuel objet occultant. A l'initialisation, les points-clés sont normalement incorporés au modèle. Par la suite, à tous nouveau point est affecté un compteur d'observation initialisé à zéro. Pour chaque appariement, le compteur est incrémenté. A l'inverse, si un point n'est pas apparié, le compteur est décrémenté à concurrence de 0. De plus, une fois le seuil *n* dépassé, cette évaluation est maintenue afin de ne prendre en compte le point que si sa fiabilité persiste. Cependant, sa valeur est bornée inférieurement par n-2 afin d'éviter les variations abusives.

Ce système a l'inconvénient de pénaliser les points au comportement stable en retardant leur incorporation au modèle de *n* frames. Afin de contrebalancer ce problème, leur compteur est incrémenté de 1.5 si leur dernier appariement date de la précédente image. La Figure 45 montre les résultats de l'algorithme pour plusieurs jeux d'essai avec différentes valeurs de *n*. Une durée de $n = 4$ ou $n = 6$ améliore la qualité du suivi sur l'ensemble du jeu d'essai.

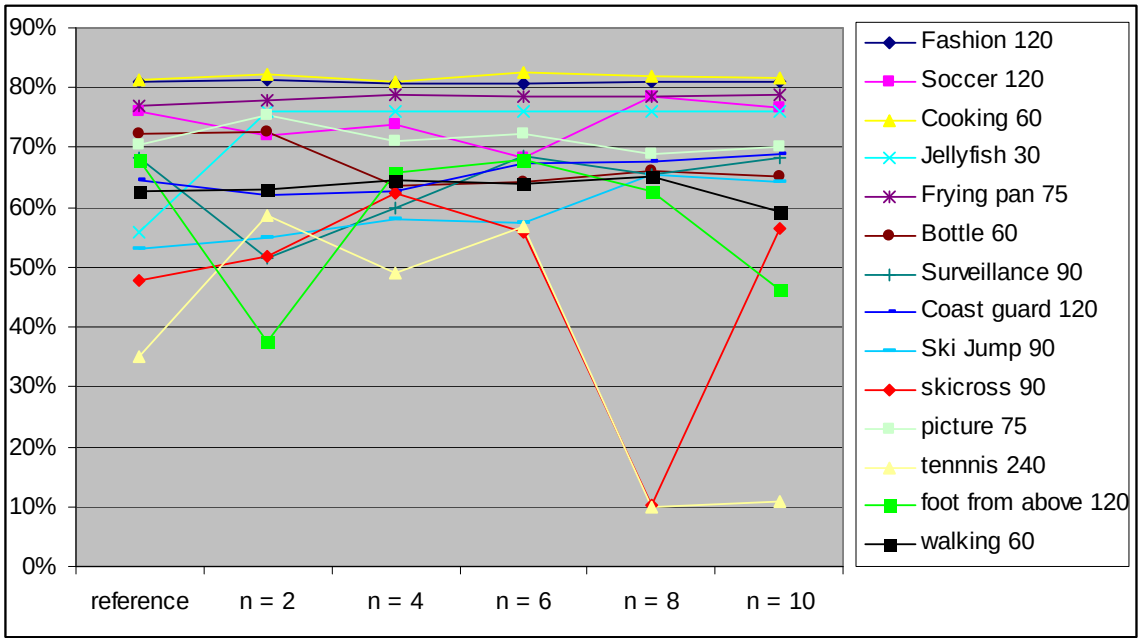


Figure 45: Résultats en pourcentage de recouvrement moyen de la boîte englobante du nouveau modèle de point-clés pour 14 séquences vidéos. L'algorithme de référence sans contrainte est comparé à l'ajout contraint des nouveaux points-clés pour une durée *n* variant de 2 à 10. Les séquences testées ainsi que leur taille sont données en légende.

Cette modification permet aussi une meilleure évaluation des labels et donc une meilleure différenciation entre l'objet et le décor.

4.2.1.6 Mise à jour du modèle

Cette étape consiste à mettre à jour les points du modèle en fonction des appariements effectués entre l'image observée et le modèle. Les données fournies sont donc deux ensembles de points (celui de l'image observée et celui du modèle) dont certains sont appariés. Trois tâches distinctes sont opérées.

Tout d'abord, la gestion des points présents. Elle consiste à éliminer les points conservés par le modèle depuis plus de *k* itérations sans être appariés et à ajouter au modèle les points de l'image observée qui n'ont pas été appariés.

Le second rôle est la mise à jour des descripteurs. Pour chaque point du modèle apparié, le descripteur de son homologue dans l'ensemble de l'image observée lui est affecté.

La dernière fonction consiste à ajuster le vecteur mouvement des points du modèle. Dans le cas où le point a été associé, la mise à jour est aisée. Dans le cas contraire, elle repose sur une hypothèse. Pour ce faire, plusieurs techniques peuvent être mises en œuvre. Une méthode brutale considérera le mouvement comme inchangé. Mais cette approche ne fonctionnera correctement que pour un mouvement uniforme. En effet, l'erreur entre le mouvement réel et le mouvement supposé s'avérera d'autant plus grande que l'irrégularité du mouvement sera prononcée. Une autre solution est d'affecter le mouvement global de l'objet à tout point non apparié. L'erreur entre le mouvement réel et le mouvement supposé sera

ainsi réduite en cas de mouvement irrégulier (voir Figure 46). La dernière possibilité consiste à inférer le déplacement d'un point en fonction du mouvement des points de son voisinage.

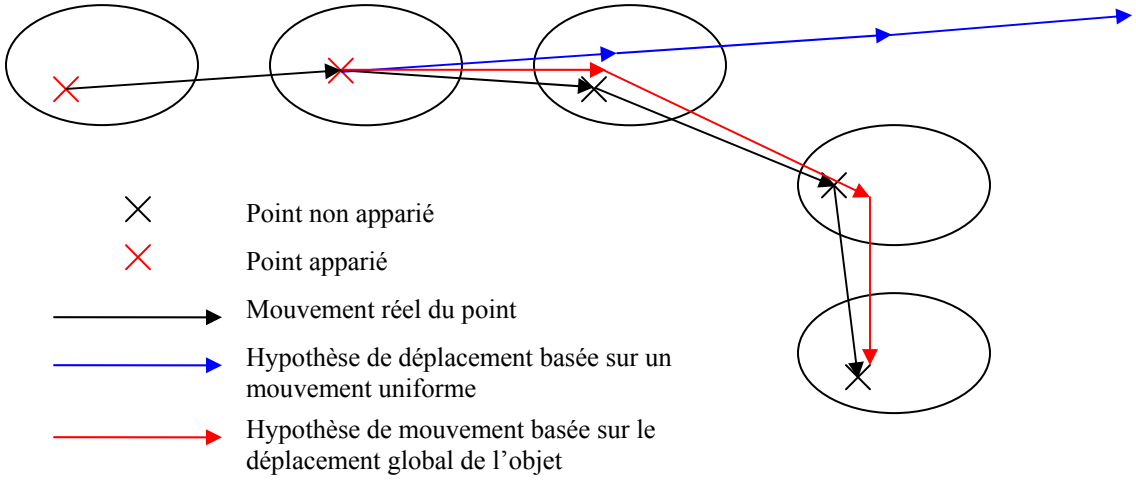


Figure 46: Illustration des hypothèses de mouvement.

Le principe consiste à affecter au point un mouvement fonction du déplacement de ses plus proches voisins appariés. Afin de créer des relations de voisinages, une triangulation de Delaunay est effectuée sur le semis de points (voir l'annexe **8.3**). Selon le critère de l'angle minimal, celle-ci offre en effet une triangulation de qualité. Et de telles triangulations permettent une bonne modélisation de la proximité entre les points. L'étape suivante vise à trouver les trois points appariés les plus proches du point étudié. Tout d'abord chaque point non apparié est éliminé de la triangulation. Ensuite, son triangle englobant est déterminé (voir l'annexe **8.3**). Un voisin n'est pris en compte que si son label (« objet » ou « décor ») est le même que celui du point étudié. Deux cas de figure peuvent alors survenir :

- Aucun voisin n'a le même label que le point. On se ramène alors au cas précédent et le mouvement global de l'objet est affecté au point.
- Un ou plusieurs voisins ont le même label que l'objet. On affecte alors au point la moyenne des mouvements de ses voisins pondéré par leur distance.

4.2.1.7 Points de Harris rapides

Les points-clés sont un outil de plus en plus repondu dans la littérature. Initialement développés pour la robotique, leur utilisation s'est maintenant étendue à des domaines aussi diverses que l'indexation, la compression, les résumés d'images, et bien sûr le suivi d'objets. Toutefois, dans la plupart de ces applications, le temps d'exécution est également un facteur important et leur extraction peut prendre jusqu'à une seconde par image. Dans le cas particulier du suivi d'objet le temps réel est une contrainte fréquente. Nous nous sommes donc intéressés à la possibilité d'accélérer ce processus.

Notre méthode repose sur l'idée de l'identification préalable de zones susceptibles de contenir des points-clés et la restriction de leur extraction à ces seules zones. En se basant sur l'hypothèse que les points sont extraits dans des zones à forte variance, nous avons développé un algorithme s'inspirant des ondelettes de Haar (dont le principe est détaillé en annexe **8.2**) permettant le calcul de cette variance sur plusieurs échelles. De plus, nous utilisons cette structure en échelle pour un accès rapide aux zones d'intérêt. Nous avons testé cette technique à l'extraction de points de Harris couleur, mais elle reste applicable à tout type de points-clés dont l'extraction respecte l'hypothèse de variance énoncée plus haut.

L'algorithme repose donc sur l'étude de la variance sur *n* échelles. Chaque étage de la structure est constituée de blocs carrés englobant systématiquement quatre blocs de l'étage inférieur (ou quatre pixels dans le cas du premier étage). Ainsi, la structure sera un arbre de blocs dont chaque bloc a une surface quatre fois plus importante que chacun de ses quatre blocs-fils. Pour chaque bloc *(i,j)* est calculé sa variation $V_n[i][j]$ et ses moyennes $M_n^R[i][j]$, $M_n^G[i][j]$, $M_n^B[i][j]$ pour chaque plage de couleur. Ils sont déterminés par les formules suivantes :

$$V_n[i][j] = (\max R[i][j] - \min R[i][j]) + (\max G[i][j] - \min G[i][j]) + (\max B[i][j] - \min B[i][j]) + \max P[i][j]$$

Avec (pour $X = \{R,G,B\}$):

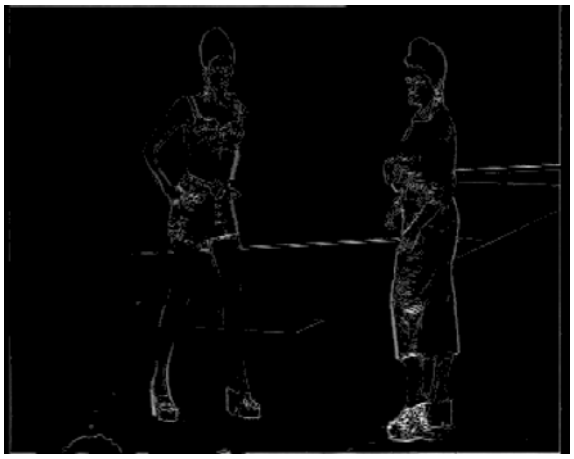
$$\max X[i][j] = \max (M_{n-1}^X[i][j], M_{n-1}^X[i][j+1], M_{n-1}^X[i+1][j], M_{n-1}^X[i+1][j+1])$$
$$\min X[i][j] = \min (M_{n-1}^X[i][j], M_{n-1}^X[i][j+1], M_{n-1}^X[i+1][j], M_{n-1}^X[i+1][j+1])$$
$$\max P[i][j] = \max (V_{n-1}[i][j], V_{n-1}[i][j+1], V_{n-1}[i+1][j], V_{n-1}[i+1][j+1])$$

$$M_n^X[i][j] = (M_{n-1}^X[i][j] + M_{n-1}^X[i][j+1] + M_{n-1}^X[i+1][j] + M_{n-1}^X[i+1][j+1]) / 4$$

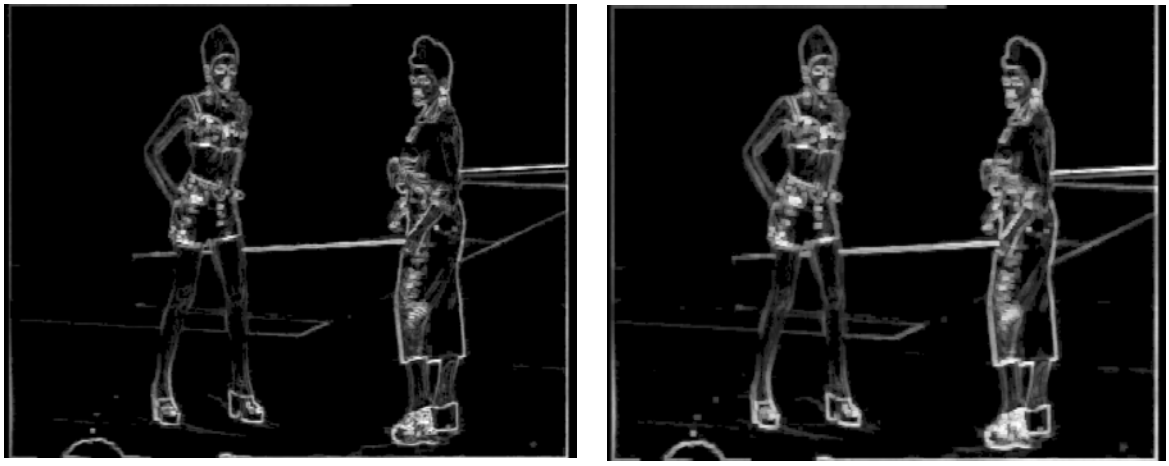
Le paramètre *maxP[i][j]* représente l'information sur la variance issue des échelles précédentes. La valeur $V_n[i][j]$ présentée par chaque bloc sera donc la meilleure variance cumulée sur plusieurs échelles par rapport au trajets sous-jacents dans la structure d'arbre. Les échelles les plus fines étant plus importantes que les plus grossières, leur valeur de variance respective est pondérée en conséquence. Ainsi, cette valeur nous informe sur l'intérêt de parcourir la structure sous-jacente. En effet, si cette valeur est en dessous d'un certain seuil *S*, aucun des pixels englobés ne sera un candidat potentiel à l'extraction de points-clés. Parcourir la structure plus avant s'avérera dès lors inutile. Ce seuil modélise le compromis entre rapidité de l'algorithme et quantité de points retrouvés. Nous avons effectué nos expériences avec un seuil de 30.

Le principal problème connu des ondelettes de Haar sont les effets de bloc. En effet, dû à la décomposition en bloc, l'information répartie à leur frontière de deux blocs, ne sera pas détectée. Pour contrebalancer cet effet, essentiellement présent au plus bas niveau de la structure, l'analyse des premiers blocs 2×2 se fait sur une zone que l'on dilate de *p* pixels (on étudie donc un bloc 4×4 pour *p=1*). Si la détection de zone susceptible de contenir des points-clés est plus précise (voir Figure 47), cette amélioration se fait au prix d'un coût de calcul supplémentaire.

(a)



(b)



(c) (d)

Figure 47: Illustration de l'influence du paramètre de voisinage p. Les pixels blancs sont les pixels retenus pour l'extraction de points-clés. (a) Image analysée. Résultats pour (b) p = 0 (c) p = 1 (d) p = 2.

Afin d'évaluer cet algorithme appelé *points de Harris rapides* ou *FastHarris*, nous avons comparé le détecteur de Harris avec et sans la construction de cette structure. Nous avons évalué l'algorithme en nous basant sur deux critères : la différence de temps d'exécution et le pourcentage de points identiques. Deux pixels sont considérés comme identiques si la distance entre leurs positions sur les deux images est inférieure ou égale à 1 pixel. Nous avons étudié l'influence du paramètre *p*, du nombre d'étages de la structure, ainsi que du seuil *S* de sélection des blocs sur son efficacité. Les tests ont été effectués sur un ensemble de 340 images différentes issues de 5 vidéos, certaines images majoritairement homogènes, d'autres présentant de fortes variations. Bien évidemment, ce type d'algorithme fonctionnera mieux pour des images majoritairement homogènes, pour lesquelles il permettra d'éviter le traitement des zones uniformes. Ce que nous avons tenté d'établir au travers de ces expériences est un jeu de paramètres donnant des résultats satisfaisants pour tout ensemble d'images traité sans connaissances a priori sur leur contenu. Les résultats sont présentés dans les Figure 48.

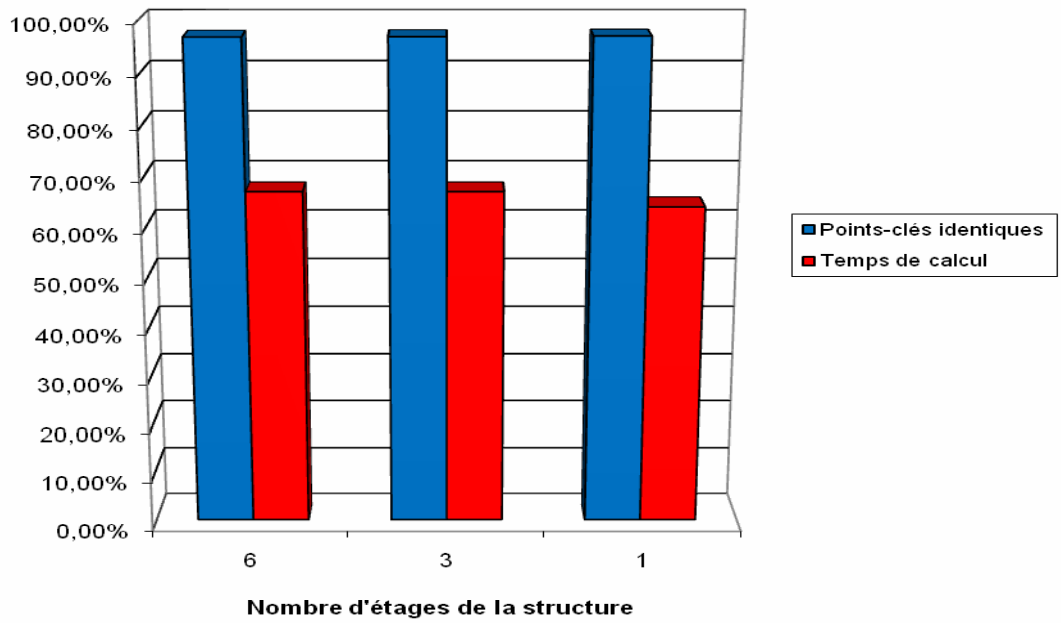


Figure 48: Influence du nombre d'étages de la structure sur l'algorithme *FastHarris*. Les mesures sont le temps de calcul et le pourcentage de points identiques par rapport au détecteur de Harris seul. Test effectué sur un ensemble de 340 images. Chaque test est la moyenne de 3 exécutions pour p égal de 0 à 2 et S=20.

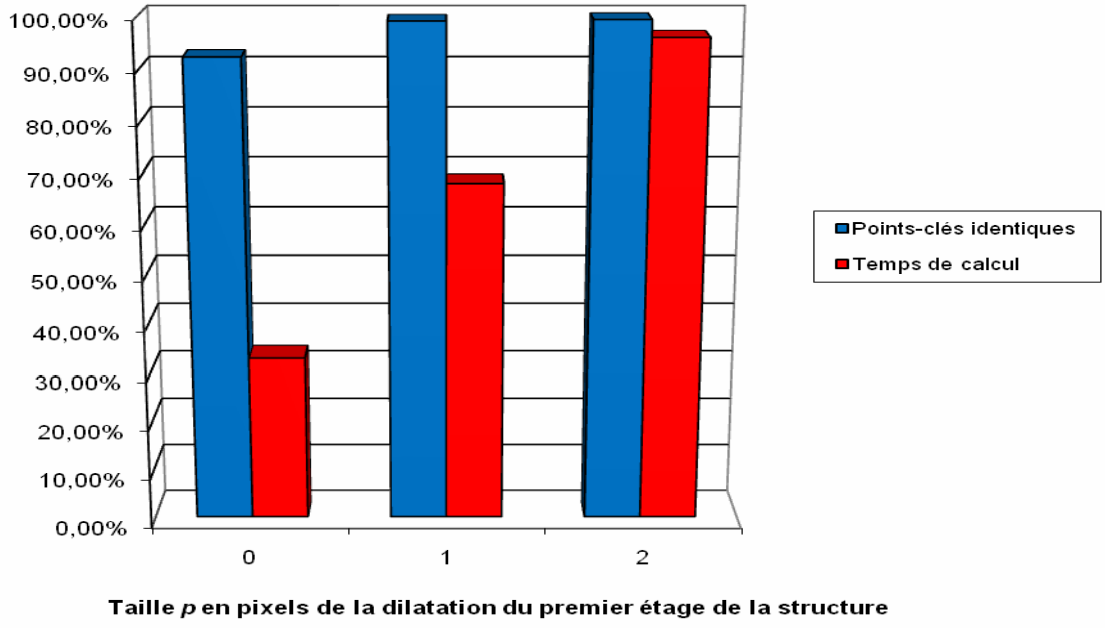


Figure 49: Influence du paramètre p sur l'algorithme *FastHarris*. Les mesures sont le temps de calcul et le pourcentage de points identiques par rapport au détecteur de Harris seul. Test effectué sur un ensemble de 340 images. Chaque test est la moyenne de 3 exécutions pour une structure de 1, 3 ou 6 étages et S=20.

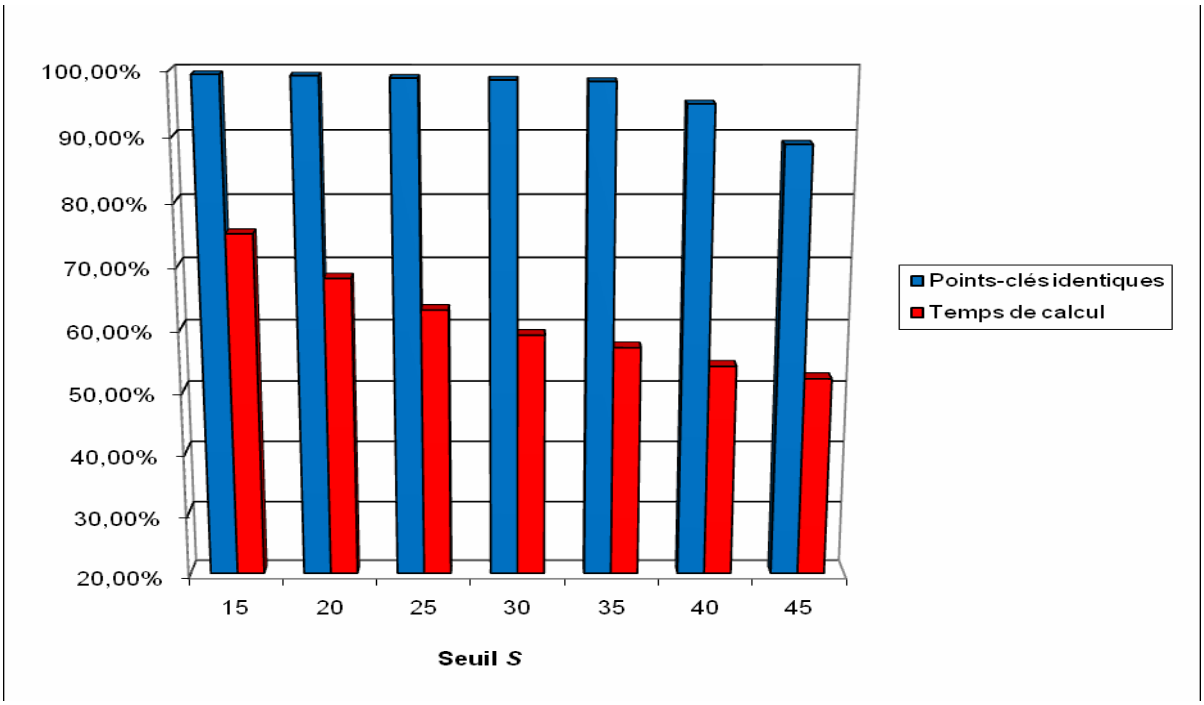


Figure 50: Influence du Seuil S sur l'algorithme *FastHarris*. Les mesures sont le temps de calcul et le pourcentage de points identiques par rapport au détecteur de Harris seul. Test effectué sur un ensemble de 340 images. Chaque test est effectué pour une structure de 3 étages et p=1.

On constate que le nombre d'étages de la structure ne semble pas avoir, en moyenne, d'influence sur les résultats de l'algorithme. Ce résultat peut être expliqué par le fait que pour une image avec de fortes variations, cette structure s'avérera inutilement coûteuse puisque toute l'image devra être analysée. A l'inverse, pour une image avec de larges zones homogènes, celle-ci sera avantageuse. Une connaissance à priori sur la complexité des images traitées permettra d'optimiser l'utilisation de la structure. Le paramètre *p,* en revanche, accroît les résultats pour un temps de calcul plus élevé. Fixer *p* à 1 et *S* à 30 semble un bon compromis puisqu'il permet de retrouver en moyenne 98.48% de points identiques tout en diminuant les temps d'exécution de 42%. L'intégration de cette structure au suivi, son impact sur la qualité du suivi, ainsi que les temps de calculs résumltant seront présentés en **6.3.1.**

4.2.2 Descripteurs

4.2.2.1 Choix et performances du descripteur

Une étude évaluant la performance des descripteurs **[Mik03]** a montrée que les descripteurs SIFT **[Low99]** sont ceux qui donnent les meilleurs résultats pour la reconnaissance d'objet. Toutefois nous ne les utilisons pas dans notre système de suivi pour trois raisons.

Tout d'abord, ceux-ci impliquent un vecteur descripteur de trop grande dimension (128 valeurs pour chaque point). Leur utilisation pour des vidéos nécessiterait une mémoire colossale (25 images par seconde et de 100 à 1000 points par images). Une contremesure à cet inconvénient serait l'utilisation de PCA-SIFT **[Ke04]**, effectuant une analyse en composantes principales sur le vecteur caractéristique afin d'en réduire la taille aux valeurs les plus discriminantes. Bien qu'offrant une alternative intéressante, cette méthode se fait aux frais de calculs supplémentaires et requiert une diminution limitée du nombre de

dimensions afin de rester fiable. Les meilleurs descripteurs de faible dimension, toujours selon l'étude **[Mik03]** sont les moments invariants ou les « filtres orientables » **[Fre91]**.

De plus, si les SIFT sont les mieux adaptés pour la reconnaissance d'objets, ils ne sont pas nécessairement optimaux pour le suivi où les variations entre deux images sont souvent minimes. Dans ce cas de figure, des descripteurs moins discriminants peuvent donner de tous aussi bons résultats.

Enfin, afin de pouvoir pleinement utiliser notre adaptation de l'algorithme à la couleur (voir **4.2.1.2**) des descripteurs couleurs sont préférables.

Notre choix s'est donc orienté vers les moments couleur généralisés **[Min99][Min03]**, offrant une description compacte de la couleur au voisinage d'un point (voir **4.1.2.3**). Rappelons qu'un moment généralisé d'ordre $p+q$ et de degré $a+b+c$ est défini par:

$$M_{pq}^{abc} = \sum_{V} x^p y^q \left[R(x,y)\right]^a \left[G(x,y)\right]^b \left[B(x,y)\right]^c$$

avec *R(x,y), G(x,y), B(x,y)* les réponses respectives de chaque canal couleur R,G,B pour le point *(x,y)*. Le dernier choix à faire était donc celui de la base de moments. De nombreuses bases de moments ont été proposées dans la littérature **[Bid02][Min99][Tuy04][Flu06]** mais ceux-ci sont employés dans le cadre de la reconnaissance d'objet. Les critères de constitution d'un jeu de moments idoine pour le suivi d'objet sont différents :

- La représentation doit être la plus compacte possible non seulement pour éviter la redondance d'information mais aussi pour des raisons de temps d'exécution. En effet, plus le nombre de dimensions du descripteur est réduit, moins il y aura de calculs.
- Le descripteur doit, bien sûr, être robuste aux diverses transformations susceptibles de survenir dans l'application. Dans le cas d'un suivi générique, tout type de transformation est susceptible de survenir. Toutefois, l'intervalle de temps qui sépare deux images étant réduit ($1/25^{eme}$ ou $1/30^{eme}$ de seconde) la magnitude de la transformation sera faible. Contrairement à la reconnaissance d'objets, l'utilité d'un descripteur apte à gérer d'importantes déformations sera donc limitée. De plus, certaines difficultés comme les transformations affines deviennent trop délicates à détecter et sujettes à erreurs. Elles ne sont donc généralement pas traitées.

La question est surtout de trouver la base de moments qui parvient à un équilibre entre ces deux critères. Nous avons donc mis au point trois jeux de moments répondant à divers degrés à ces deux critères. La première base est constituée des 18 moments précédemment utilisés par Tuytelaars **[Tuy04]** :

$$\begin{array}{lll}
inv[1] = M_{00}^{110} / M_{00}^{000} & inv[2] = M_{00}^{011} / M_{00}^{000} & inv[3] = M_{00}^{101} / M_{00}^{000} \\
inv[4] = M_{10}^{100} / M_{00}^{100} & inv[5] = M_{10}^{010} / M_{00}^{010} & inv[6] = M_{10}^{001} / M_{00}^{001} \\
inv[7] = M_{01}^{100} / M_{00}^{100} & inv[8] = M_{01}^{010} / M_{00}^{010} & inv[9] = M_{01}^{001} / M_{00}^{001} \\
inv[10] = M_{11}^{100} / M_{00}^{100} & inv[11] = M_{11}^{010} / M_{00}^{010} & inv[12] = M_{11}^{001} / M_{00}^{001} \\
inv[13] = M_{20}^{100} / M_{00}^{100} & inv[14] = M_{20}^{010} / M_{00}^{010} & inv[15] = M_{20}^{001} / M_{00}^{001} \\
inv[16] = M_{02}^{100} / M_{00}^{100} & inv[17] = M_{02}^{010} / M_{00}^{010} & inv[18] = M_{02}^{001} / M_{00}^{001}
\end{array}$$

Les trois premiers représentent la moyenne de chaque plage de couleur, les invariants 4 à 10 sont les coordonnées des centres de gravités des trois distributions de couleur, et les 9 derniers des caractérisations de second ordre. Les 2^{eme} et 3^{eme} bases proposent des descripteurs similaires mais étudiant de manière moins détaillée la distribution de couleur. La 2^{eme} base élimine les 9 derniers invariants de la première pour les remplacer par des invariants caractérisant les plages de couleurs.

$$inv[1] = M_{00}^{110} / M_{00}^{000} \qquad inv[2] = M_{00}^{011} / M_{00}^{000} \qquad inv[3] = M_{00}^{101} / M_{00}^{000}$$
$$inv[4] = \left(M_{00}^{200} \times M_{00}^{000}\right)/\left(M_{00}^{100}\right)^2 \qquad inv[5] = \left(M_{00}^{020} \times M_{00}^{000}\right)/\left(M_{00}^{010}\right)^2 \qquad inv[6] = \left(M_{00}^{002} \times M_{00}^{000}\right)/\left(M_{00}^{001}\right)^2$$
$$inv[7] = M_{00}^{110} /\left(M_{00}^{100} \times M_{00}^{010}\right) \qquad inv[8] = M_{00}^{011} /\left(M_{00}^{001} \times M_{00}^{010}\right) \qquad inv[9] = M_{00}^{101} /\left(M_{00}^{100} \times M_{00}^{001}\right)$$
$$inv[10] = M_{10}^{100} / M_{00}^{100} \qquad inv[11] = M_{10}^{010} / M_{00}^{010} \qquad inv[12] = M_{10}^{001} / M_{00}^{001}$$
$$inv[13] = M_{01}^{100} / M_{00}^{100} \qquad inv[14] = M_{01}^{010} / M_{00}^{010} \qquad inv[15] = M_{01}^{001} / M_{00}^{001}$$

La 3eme base n'est composée que des 9 premiers invariants de la base précédente se focalisant sur la description des plages de couleurs :

$$inv[1] = M_{00}^{110} / M_{00}^{000} \qquad inv[2] = M_{00}^{011} / M_{00}^{000} \qquad inv[3] = M_{00}^{101} / M_{00}^{000}$$
$$inv[4] = \left(M_{00}^{200} \times M_{00}^{000}\right)/\left(M_{00}^{100}\right)^2 \qquad inv[5] = \left(M_{00}^{020} \times M_{00}^{000}\right)/\left(M_{00}^{010}\right)^2 \qquad inv[6] = \left(M_{00}^{002} \times M_{00}^{000}\right)/\left(M_{00}^{001}\right)^2$$
$$inv[7] = M_{00}^{110} /\left(M_{00}^{100} \times M_{00}^{010}\right) \qquad inv[8] = M_{00}^{011} /\left(M_{00}^{001} \times M_{00}^{010}\right) \qquad inv[9] = M_{00}^{101} /\left(M_{00}^{100} \times M_{00}^{001}\right)$$

Les performances de notre algorithme de suivi générique ont été testées pour chacun de ces jeux de descripteurs avec les points de Harris-Laplace. Les résultats sont présentés en Figure 51.

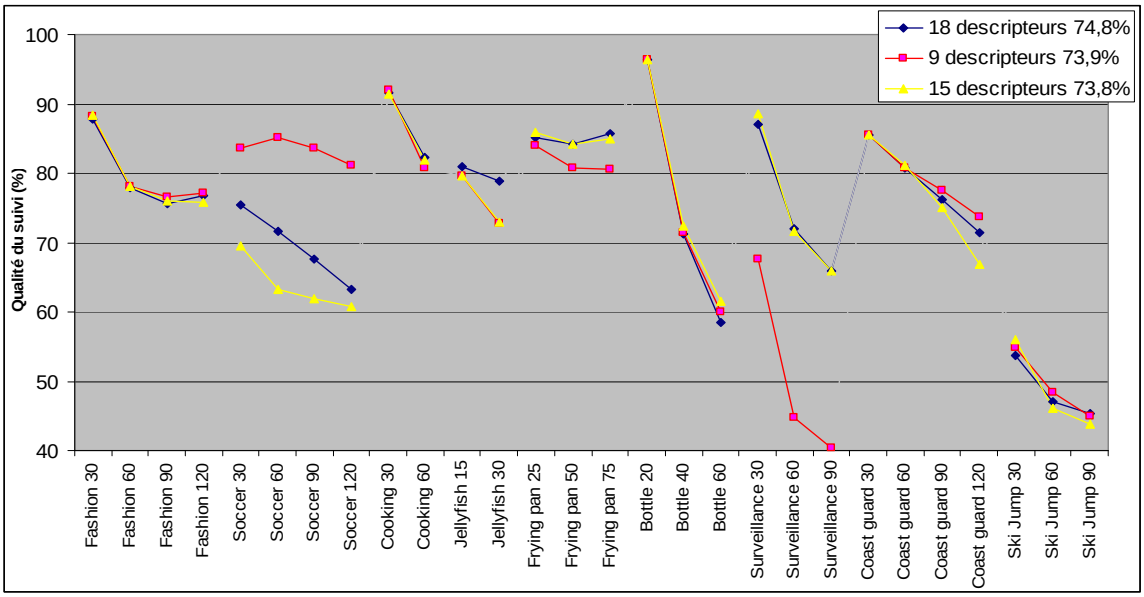


Figure 51: Performances du système de suivi pour différentes bases de descripteurs composées respectivement de 9, 15, et 18 invariants avec les points de Harris-Laplace. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des tests est donnée dans l'intitulé des courbes.

On constate que, bien que la base de 18 descripteurs soit, en moyenne, sensiblement meilleure sur l'ensemble des tests, les résultats globaux diffèrent peu. Par contre ceux-ci peuvent beaucoup varier d'une vidéo à l'autre. Par exemple, alors que la base de 9 invariants est un sous-ensemble de celle de 15, on constate une prédominance marquée de la première pour la séquence « Soccer » et de la seconde pour la séquence « Surveillance ». Ce comportement met en valeur l'importance des variations pouvant survenir dans un suivi pour des modifications minimes.

Un autre paramètre pouvant influer sur les résultats et dont le réglage obéit à des lois différentes dans le cadre de la reconnaissance d'objet est la taille du voisinage sur lequel ces descripteurs sont extraits. En effet, en reconnaissance d'objet, on considère un rayon relativement large de façon à obtenir un descripteur riche en termes d'information. Par contre, le suivi d'objet traite des objets en mouvement

plutôt que des objets vus sous des angles ou dans des décors différents. Leur voisinage sera donc changeant. Donc, si une taille minimum est nécessaire afin de garantir une information fiable, un voisinage large sera en revanche généralement néfaste pour la robustesse du descripteur. Nous avons conduit un ensemble d'expériences afin de trouver le rayon optimal de la région circulaire à partir de laquelle sont extraits les descripteurs. La Figure 52 montre les résultats de notre algorithme de suivi pour un descripteur calculé sur des régions de rayon de 3, 5, 7, et 9 pixels. Les tests sont effectués avec les points de Harris-Laplace et les rayons proportionnels à l'échelle caractéristique détectée.

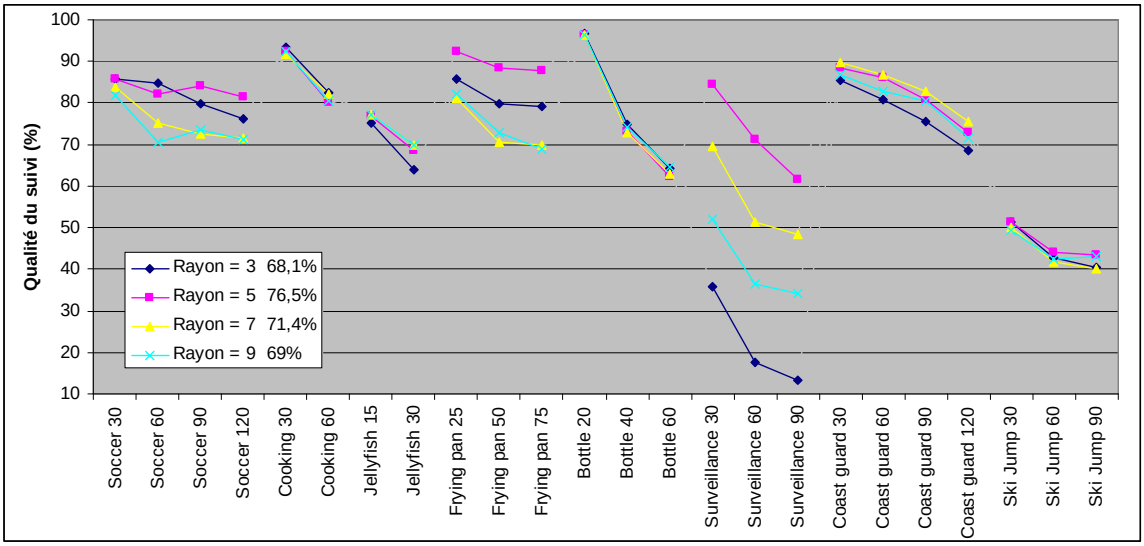


Figure 52: Performances du système de suivi avec les points de Harris-Laplace pour des rayons de régions sur lesquelles les descripteurs sont extraits de respectivement 3, 5, 7, et 9 pixels. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des tests est donnée dans l'intitulé des courbes.

Les résultats indiquent clairement un rayon de 5 pixels comme adéquat pour un suivi d'objet utilisant des moments couleurs généralisés. Bien que l'on ne puisse être catégorique sur ce point (chaque descripteur ayant ses spécificités), on peut supposer que ce résultat puisse s'étendre à d'autres descripteurs.

Le descripteur retenu pour notre algorithme calcule la base de 18 moments généralisés couleurs sur un voisinage circulaire de 5 pixels autour du point d'intérêt.

4.2.2.2 Distance

Nous utilisons la distance de Mahanalobis dont nous rappelons ici le calcul. Soit un ensemble de N données caractérisées par un vecteur de dimension p. Rappelons que la covariance entre deux variables X et Y se calcule par :

$$Cov(X,Y) = \frac{1}{N}\sum_{i=1}^{N}\left(x(i)-\bar{x}\right)\left(y(i)-\bar{y}\right)$$

Avec $\bar{x}$ et $\bar{y}$ la moyenne des variables sur l'ensemble des données. Cette grandeur indique à quel point ces deux variables varient ensemble. On peut alors calculer la matrice de corrélation Σ comparant chacune des p variables 2 à 2. La distance de Mahanalobis mesurant la similitude de deux échantillons i et j de l'ensemble étudié se calcule par :

$$d_M(x(i),x(j)) = \sqrt{\left(x(i)-x(j)\right)^T \sum^{-1}\left(x(i)-x(j)\right)}$$

Avec $(x(i)-x(j))^T$ de dimension *(1×p)*, Σ^{-1} de dimension *(p×p)*, et $(x(i)-x(j))$ de dimension *(p×1)*. Bien que coûteuse en temps de calcul, cette mesure est très précise puisqu'elle tient compte de la corrélation globale sur l'ensemble des données pour juger de la similarité de deux données. Ainsi, deux composantes similaires influeront peu sur la mesure si leurs variables associées sont fortement corrélées. Les résultats sont donc nettement meilleurs que, par exemple, la distance euclidienne. Notons que, ici encore, la dimension *p* du descripteur utilisée est le paramètre ayant le plus d'influence sur le temps de calcul de la distance.

5 Suivi de points d'intérêt et estimation des paramètres du modèle

5.1 Etat de l'art

5.1.1 Suivi de points d'intérêt

Le suivi de primitive simple ou de points est un domaine qui a de nombreuses applications dans le cadre de la robotique, l'exploitation d'images radar, le suivi ou l'indexation dans les vidéos. Le principe est d'apparier des données prises à intervalle de temps régulier pour créer des trajectoires. Afin de lever les ambiguïtés (i.e. le suivi d'un point *x* à l'instant *t* admet plusieurs points candidats à l'instant *t+1* qui sont susceptibles de lui être appariés) des contraintes sont définies. Ces contraintes peuvent faire référence aux caractéristiques des primitives, à leur mouvement, ou bien aux relations entre elles. Les algorithmes peuvent être classés en fonction des contraintes qu'ils définissent et de leur manière de les exploiter.

5.1.1.1 Principes du suivi de primitives

5.1.1.1.1 Problématique du suivi de primitives

Le suivi de primitives consiste à apparier des ensembles de points pris à intervalles de temps régulier, afin de créer des trajectoires physiquement cohérentes. Les points ne représentent qu'une information de position ponctuelle de primitives dont on essaye de retracer le mouvement. A chaque étape, on va donc établir des correspondances entre un ensemble de *n* points d'un modèle et un ensemble de *m* points d'une image observée. Le modèle généralement utilisé est l'ensemble de points de l'image précédente. Le problème se ramène donc à apparier les points des images *t* et *t+1*.

Comme un point n'a qu'une représentation physique, pour chaque image, un point ne peut être associé qu'à une seule trajectoire. Toutefois, aucune hypothèse n'est faite quand à leur fiabilité ou la primitive à laquelle ils sont associés. A causes de perturbations des capteurs, de défaillances de l'algorithme, d'occultations de la primitive suivie, de sortie du point suivi du champ de la caméra, ou de changement dans les caractéristiques de la primitive à détecter, des trajectoires peuvent apparaître, disparaître ou être imprécises par rapport au mouvement réel de la primitive étudiée. Un point à une image *t*, peut donc avoir, soit un, soit aucun correspondant à l'image *t+1*, et des points de l'image *t+1* peuvent ne pas avoir de trajectoire associée. De plus, pour un site donné, plusieurs points du modèle pourront être candidats à l'appariement. La difficulté du problème consiste donc à lever ces ambiguïtés.

5.1.1.1.2 La régularité du mouvement

De part leur inertie propre, la trajectoire des objets est rarement erratique. Leur mouvement présente une certaine continuité qui peut d'autant plus facilement être appréhendée dans les vidéos que le faible

intervalle temporel entre les images permet une vision fluide du mouvement. En se basant sur ce principe, on peut lever des ambiguïtés d'associations entre trajectoires et points. Il faut toutefois noter qu'au moins trois images sont nécessaires pour obtenir suffisamment d'informations pour caractériser la régularité du mouvement. Les modèles proposés sont généralement des modèles linéaires : vitesse constante, accélération constante, ou encore minimisation de la déviation par rapport à la trajectoire. Afin de lever le maximum d'ambiguïtés, il serait plus efficace de minimiser les contraintes d'associations trajectoires / primitives de façon globale plutôt que de considérer chaque trajectoire indépendamment (Figure 53). Toutefois, une telle stratégie est très coûteuse en terme de temps de calcul et les auteurs choisissent en conséquence de traiter les affectations primitive / trajectoire de manière isolée. Ces algorithmes sont donc sous-optimaux.

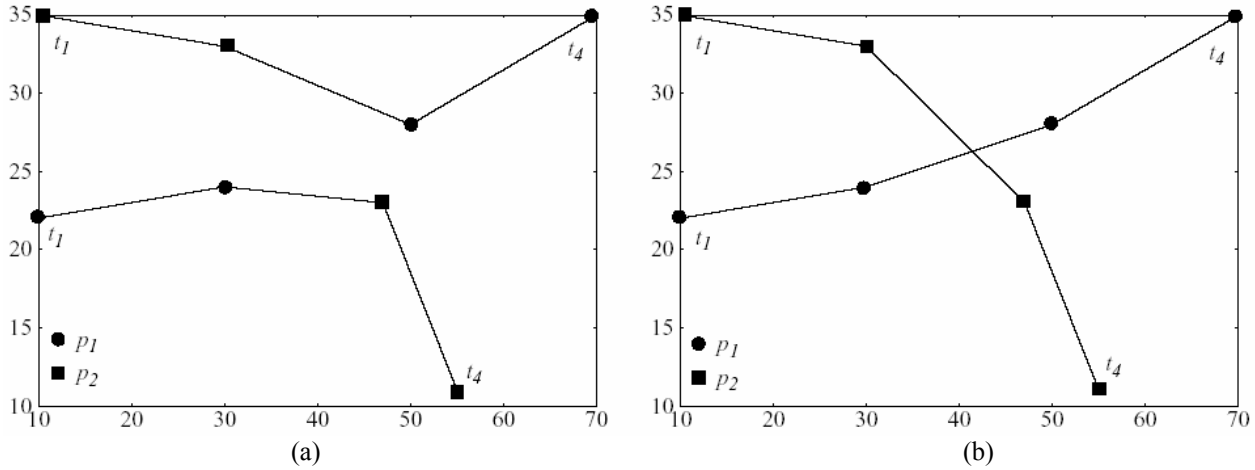


Figure 53: Les points représentent les mesures de deux primitives en mouvement suivies au cours du temps, les lignes représentent les associations. (a) Algorithme où les trajectoires sont analysées de manière indépendante. (b) Algorithme minimisant la déviation par rapport à la trajectoire pour l'ensemble des primitives suivies. Ce dernier parvient, contrairement à la méthode d'analyse locale à retrouver le mouvement correct.

En ce basant sur le même principe de régularité du mouvement, certaines méthodes, dites de prédiction, interprètent les positions d'une primitive dans les images passées pour émettre une hypothèse, avec une certaine incertitude, sur sa position actuelle. Une fenêtre de recherche centrée sur la position supposée de la primitive suivie est alors définie. Sa taille varie en fonction de l'incertitude de la prédiction et de la vitesse de déplacement supposée de la primitive. Si l'incertitude de la prédiction est totale (i.e. aucune prédiction ne peut être faite) on peut centrer la fenêtre sur la position précédente de la primitive (Figure 54).

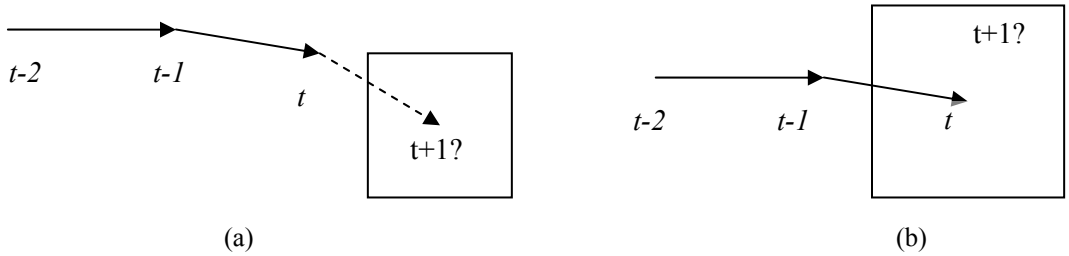


Figure 54: Fenêtres de recherche (a) Avec prédiction sur le mouvement (b) Sans aucune prédiction.

Une méthode classique de prédiction du mouvement, développé en 1960, est le *filtre de Kalman*. La force de cette méthode vient de son caractère récursif. Les mesures passées sont tout d'abord utilisées pour prédire la position de la primitive suivie, puis cette prédiction est corrigée en fonction de la mesure

(bruitée) de la position actuelle. Ce cycle *prédiction / correction* se répète jusqu'à la fin de l'algorithme (Figure 55).
La phase de *prédiction* consiste à estimer la prochaine position de la primitive suivie d'après un modèle linéaire de mouvement et en fonction des *n* mesures passées. On ajoute également un terme d'incertitude sur le mouvement, issue de l'incertitude sur la position et de la vitesse de la primitive suivie, et modélisant l'erreur entre le modèle de mouvement et la position observée. Ce terme permet de calculer la taille de la fenêtre de recherche.

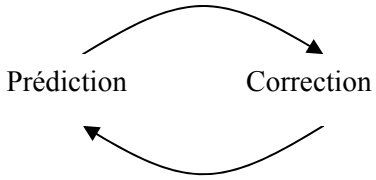


Figure 55: Cycle de Kalman. La phase de *Prédiction* projette le modèle dans le temps en fonction des mesures passées. La phase de *Correction* ajuste la prédiction en fonction des mesures actuelles.

La phase de *correction* permet, après obtention des mesures se référant à la position prédite (tempérées par la variable de bruit de mesure), d'ajuster la trajectoire. Elle est modifiée suivant les valeurs du bruit de mesure et d'incertitude sur le modèle. On rapprochera d'autant plus le point estimé du point mesuré que le bruit de mesure est faible par rapport à l'incertitude sur le mouvement. Enfin, on met à jour l'incertitude sur le mouvement. Plus la correction apportée au point prédit a été faible, plus la prédiction était donc avisée, plus l'incertitude sur le mouvement diminue.

La littérature sur le filtre de Kalman est très riche, de très nombreux travaux furent effectués tels que le *filtre de Kalman étendu* qui adapte le filtre de Kalman à des systèmes non linéaires. Pour une introduction plus détaillée sur le filtre de Kalman, consulter les travaux de **Maybeck [May79]** ou de **Welch [Wel04]**.

5.1.1.1.3 Comparaison des caractéristiques de primitives

Cette deuxième contrainte est une particularité du traitement d'image. Des caractéristiques de couleur, de texture ou de forme peuvent être extraites d'un point d'intérêt ou de son voisinage afin d'accroître sa spécificité. De plus, comme le montre les travaux de **Megret [Meg02]**, il est possible d'étudier l'évolution temporelle des relations spatiales entre les points, plutôt que de considérer un point (i.e. une trajectoire) comme une entité isolée. Par exemple, dans la méthode de **Gabriel [Gab05]**, les points de l'objet sont suivis en fonction de leur apparence locale et leur position par rapport au centre de gravité des points associé à l'objet. De plus cette technique a l'avantage que seules 2 images sont nécessaires pour estimer la régularité de la distance d'un point par rapport à d'autres, alors que trois images minimum sont requises dans le cas de la caractérisation par le mouvement.

5.1.1.2 Les méthodes d'association de primitives

5.1.1.2.1 Les conditions du suivi

Avant de décrire les différents algorithmes existants, il est utile d'avoir à l'esprit les différents paramètres qui influent sur la qualité d'un suivi. Selon les applications, les besoins sont différents et les capacités des méthodes varient.

Le paramètre qui a le plus d'influence sur les résultats est sans doute la densité des points. Plus le nuage de points est dense, plus, pour une primitive à apparier, il y aura de candidats possibles, ce qui augmente la complexité des opérations effectuées et le risque d'erreur. En revanche, cette plus grande quantité d'information induit également une meilleure précision sur le modèle de mouvement à extraire.

Une autre caractéristique des algorithmes est leur capacité à traiter les cas difficiles tels que les occultations des trajectoires, l'absence d'information à l'initialisation, l'ajout ou la disparition de primitives à suivre (Figure 56), l'évolution des caractéristiques associées aux primitives au fur et à mesure du suivi. Ces cas particuliers produisent généralement des changements importants dans les données à traiter et déroutent certains modèles.

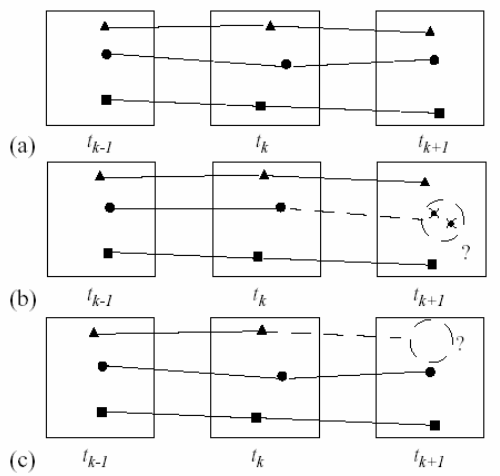


Figure 56: Les points représentent les mesures de trois primitives en mouvement suivies au cours du temps, les lignes représentent les associations. (a) Toutes les primitives sont détectées à chaque image. (b) Apparition d'une fausse alarme. (c) Une primitive n'est pas détectée.

Les hypothèses faites sur le mouvement sont également importantes. Les suppositions faites sur la vitesse du mouvement détermineront la taille de la fenêtre de recherche, alors que les connaissances (ou l'absence de connaissance) sur la régularité du mouvement sont des données capitales dans le choix de l'algorithme à utiliser.
Enfin, dernier paramètre, le temps de calcul est, dans la mesure où le choix des algorithmes est un compromis entre complexité et rapidité, un facteur déterminant dans la précision du suivi. Le nombre de points suivis influencera aussi ce dernier paramètre.

La plupart de ces paramètres sont déterminés par le type de scène étudiée et de l'application désirée. Le nombre de points à suivre ou à apparier sera déterminé par le domaine d'application. Par exemple, le suivi d'objets demande un nombre variable de points suivant la taille de l'objet suivi, allant de quelques dizaines à des centaines, alors que plusieurs centaines seront nécessaires dans le cadre de l'indexation, et plusieurs milliers pour le suivi de fluides, où les primitives sont associées à des particules en suspension. De plus, les informations issues des connaissances sur la vidéo ou les images traitées permettent de déterminer un grand nombre de paramètres. Par exemple des images satellites détermineront la vitesse et sa régularité des primitives ou le suivi de personnes dans un lieu public informera sur les possibilités d'occultations, d'apparitions ou de disparitions de trajectoires. Le choix de l'algorithme est donc à effectuer en fonction du type de scène à étudier et de l'application voulue.

5.1.1.2.2 Les algorithmes basés sur un mouvement régulier

Ces méthodes se basent sur le critère le plus simple et se focalisent sur la correspondance entre primitives et trajectoires. Malgré une hypothèse de travail identique, les techniques et leurs limitations varient. Certains algorithmes ne traitent pas les occultations, d'autres ont des contraintes d'initialisation ou ne gèrent pas les fausses alarmes. Une comparaison des méthodes est disponible dans les travaux de **Chetverikov [Che98]** et ceux de **Veenman [Vee01]**.

L'algorithme présenté ici, initialement développé par **Sethi et Jain [Set87]**, fut ensuite amélioré par **Salari et Sethi [Sal90]** afin de prendre en compte les erreurs de détection. Cette méthode est initialisée en associant les paires de points les plus proches. Ensuite des correspondances entre les paires de points sont échangées itérativement en vue d'améliorer un critère de régularité du mouvement. L'algorithme s'arrête à la stabilisation du système. Bien que ne prenant pas en compte les cas d'occultation, cette méthode a l'avantage de simuler une prise en compte globale du problème.

Une approche différente fut proposée par **Rangarajan et Shah [Ran91]**. Leur modèle suppose le nombre de points fixe, la méthode peut ainsi traiter efficacement les occultations ou les détections manquées, mais échoue lors de la présence de fausses alarmes. Cet algorithme trie l'ensemble des associations possibles en fonction d'un critère de régularité de mouvement. Cela permet de traiter en premier les correspondances les plus sures et de laisser pour la fin les cas problématiques. De plus, cette analyse n'étant pas itérative, est plus rapide.

L'originalité de la méthode de **Chetverikov [Che98]** tient à l'utilisation de triplet d'images pour une analyse plus précise de la régularité du mouvement. Son fonctionnement est le suivant. A chaque point est associé une fenêtre de recherche centrée sur celui-ci et dépendant de la vitesse maximale accordée aux primitives suivies. Donc les points des images précédentes et suivantes sont inscrits dans ce cercle. A chaque étape (initialisation comprise), on considère, localement pour un point x à l'instant t, l'ensemble des triplets sur l'intervalle [t-1 ; t+1] comprenant le point x et on choisit le meilleur selon le critère de régularité établit. Soit T_x ce triplet. Ensuite, tous les triplets sur le même intervalle de temps [t-1 ; t+1] comprenant au moins un point commun avec T_x sont étudiés et le meilleur est retenu. Ce procédé, illustré en Figure 57 sélectionne le meilleur triplet parmi tous ceux localement en concurrence.

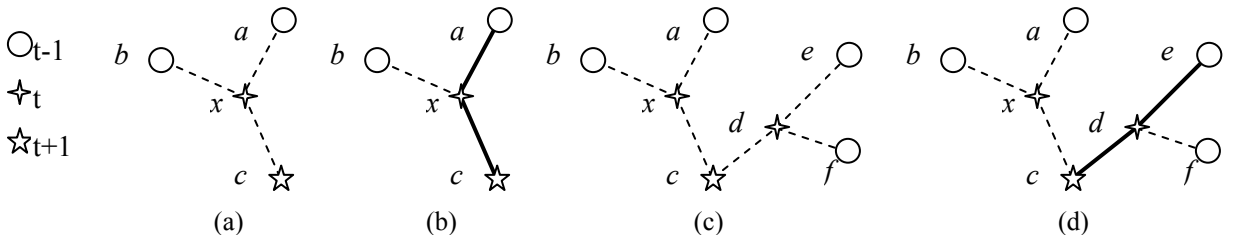


Figure 57: Recherche locale du meilleur triplet. (a) Evaluation des triplets associables à x (b) Conservation du meilleur triplet *axc* (c) Etude des triplets ayant au moins un point commun avec *axc*, ici exemple des triplets issues de *c* (d) Conservation du meilleur triplet *cde*.

On répète ensuite ce processus sur les points du nouveau triplet. Une fois tous les points de l'intervalle [t-1 ; t+1] appariés, la même technique est appliquée à l'intervalle [t ; t+2] puis itérativement à tous les autres intervalles jusqu'au dernier. Seule la mise en correspondance des points du premier triplet est coûteuse puisque, par la suite les deux premiers points sont déjà associés.
Les occultations sont traitées en dernière étape par prolongement des trajectoires interrompues en avant et en arrière de un ou deux pas temporels, et en définissant des zones de recherche fonctions de la vitesse maximale et de la déviation maximale.

Veenman [Vee01] quant à lui, associe points et trajectoire par minimisation d'une fonction de coût sur la totalité de l'image. La fonction, choisie expérimentalement, est la régularité du mouvement sur un triplet de point au niveau local et la moyenne des coûts locaux au niveau global. Le problème des occultations est solutionné par l'utilisation de points fictifs créés par interpolation entre le dernier point de la trajectoire et le(s) point(s) candidat(s) de reprise de la trajectoire. Il faut toutefois remarquer que toutes les trajectoires interrompues sont traitées comme une occultation, ce qui n'est pas nécessairement le cas. Les auteurs prouvent que les performances de cet algorithme dépassent celles des autres méthodes se basant sur un mouvement régulier.

5.1.1.2.3 Suivi multi-hypothèses (MHT)

Le suivi multi-hypothèses, développé par **Reid [Rei79]** en 1979, est une méthode qui traite de l'association trajectoire/points. Il s'agit de la méthode multi-hypothèses la plus utilisée, particulièrement dans le cas des vidéos. Il est basé sur la génération, à chaque étape du suivi, des différentes possibilités induites par la situation. Tous les cas de figure sont examinés; la poursuite d'une trajectoire existante, l'éventualité d'une occultation, l'apparition ou la disparition de nouveaux traits géométriques, ou bien la présence de fausses alarmes due à des données erronées. On effectue des hypothèses locales pour chaque point ainsi que des hypothèses globales, tenant compte des contraintes d'association pour des ensembles de points. Les *k* meilleures hypothèses sont ensuite retenues. Ce qui fait l'intérêt du suivi multi-hypothèses est justement le fait que le choix de l'hypothèse adoptée est différé. Cette meilleure exploitation du temps est très importante. En effet, l'étude des possibilités sur plusieurs images permet de rassembler plus d'informations afin de lever davantage d'ambiguïtés d'association.
La gestion des hypothèses peut être représentée par un arbre de décision où chaque niveau est une image et chaque branche une hypothèse. Les feuilles du niveau le plus bas sont donc les hypothèses actuellement retenues (Figure 58). Ainsi, à chaque étape de l'algorithme, l'arbre d'hypothèses est mis à jour. D'une part, en ajoutant *k* nouvelles hypothèses issues des données de la dernière image. Et, d'autre part, en éliminant *k-1* anciennes hypothèses suite à une prise de décision, ce qui se traduit par l'élagage des sous-arbres correspondants.

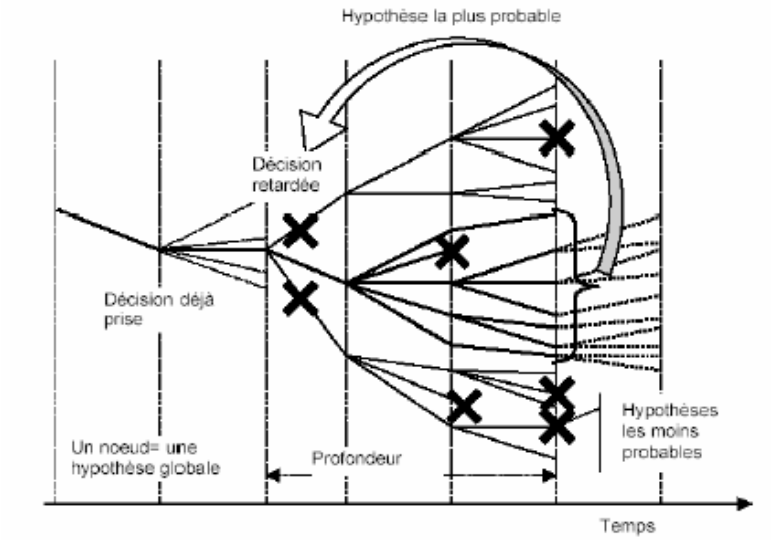


Figure 58: Décisions dans l'arbre multi-hypothèses.

Ce type d'algorithme a donc l'avantage de traiter la totalité des cas envisageables tout en traitant plus efficacement les ambiguïtés. Cette performance se fait par contre au prix de la complexité. Deux paramètres influent sur le temps de calcul. La largeur de l'arbre d'hypothèses, autrement dit, le nombre *k* d'hypothèses conservées à chaque étape et la profondeur de l'arbre, soit le nombre d'itérations nécessaires à la prise de décision. Dans l'implémentation de **Cox [Cox96]**, le choix des *k* meilleures hypothèses est traité comme un problème d'affectation linéaire, ce qui permet de gagner un temps de calcul non négligeable.

De nombreuses variantes sont disponibles. **Gauvrit [Gau97]** proposa une approche probabiliste et une approche combinatoire ainsi qu'une classification des algorithmes multi-hypothèses. Un bon exemple d'utilisation du MHT est la méthode de suivi d'objets de **Tissainayagam [Tis04]** où un premier algorithme de MHT est utilisé pour grouper les contours de mêmes objets, puis un deuxième pour suivre des points issus de la carte de contours ainsi formée.
Dans le cas du suivi d'objet, cette technique peut également être appliquée à l'objet et non aux trajectoires. Les hypothèses correspondent alors à divers possibilités de trajectoires, d'occultation, de changement d'échelle,etc.

5.1.1.2.4 Filtres à particules

Les techniques de filtrage particulaire sont des méthodes de simulation séquentielle de type Monte-Carlo qui estiment le vecteur d'état d'un système Markovien que l'on suppose soumis à des variations (de type gaussiennes ou non). La naissance de ces méthodes remonte à la parution de l'algorithme CONDENSATION de **Isard et Blake[Isa98]**. Etant donné x_t ce vecteur d'état et z_t les observations associées à l'instant *t*, le filtre particulaire va estimer la densité de probabilité a postériori $p(x_t \mid z_t)$ de x_t selon z_t. Pour ce faire, un ensemble de *N* échantillons $x_t(i)$, appelés « particules » vont être sélectionnés dans l'espace des solutions envisageables. Le positionnement des particules est généralement fonction de deux composantes : une composante en rapport avec le modèle à caractériser et une composante aléatoire, le poids accordé à ces deux facteurs étant laissé a la discrétion de l'utilisateur. La pertinence de ces particules par rapport au modèle recherché va ensuite être évaluée en accord avec les observations, et un poids $w_t(i)$ attribué à chacune d'elles en conséquence. Finalement, la densité de probabilité sera évaluée à partir du nuage de particules :

$$p(x_t|z_t) = \sum_{i=1}^{N} w_t^i \partial\left(x_t - x_t^i\right) \qquad \sum_{i=1}^{N} w_t^i = 1$$

L'algorithme classique de filtrage particulaire se déroule en trois étapes :

1-génération des particules
2-mise à jour du poids associé aux particules
3-ré-échantillonnage du nuage de particules

L'initialisation consiste à définir un ensemble de particules représentant la distribution à priori $p(x_0)$ en plaçant dans l'espace des solutions un ensemble de particules pondérées représentant les connaissances possédées sur le vecteur d'état du modèle. Les poids sont fixés à $w_0(i)=1/N$. Par la suite, à chaque instant *t*, disposant de la mesure z_t et de la description particulaire $\left\{x_{t-1}^i, w_{t-1}^i\right\}$, on effectue les trois étapes énoncées précédemment. Tout d'abord on « propage » les particules en modifiant le vecteur d'état $x_t(i)$ en fonction du vecteur d'état précédent $x_{t-1}(i)$ et d'une perturbation aléatoire μ :

$$x_k^i = q\left(x_t \middle| x_{t-1}^i, \mu\right)$$

Ensuite, les poids des particules sont mis à jours :

$$w_t^i = w_{t-1}^i p\left(x_t|z_t\right)$$

Préalablement à une étape de normalisation afin de garantir $\sum_{i=1}^{N} w_t^i = 1$. Tout algorithme de ce type souffre du phénomène de dégénérescence, c'est-à-dire qu'après quelques itérations quelques particules ont un poids très élevé et toutes les autres ont des poids quasi nuls. Afin de contrebalancer ce comportement, une étape de ré-échantillonnage peut être insérée en fin d'itération. Les particules dont le poids est en deçà d'un seuil prédéfini sont alors éliminées et, pour chacune de ces évictions, une des particules à poids élevé est dupliquée. Cette étape permet de limiter l'exploration de l'espace des solutions aux seules zones pertinentes.

L'application des filtres à particules à un système de suivi est particulièrement efficace lorsque des hypothèses de déplacement de l'objet précises ne peuvent être établies. On définit le vecteur d'état x_t par la position (et éventuellement l'échelle) de l'objet, et l'observation associée z_t comme la valeur du descripteur extrait à cette position. Dans ce cadre, énormément de variantes ont été proposées modifiant la fonction de mise à jour des poids, de propagation des particules, ou encore pour une meilleure utilisation du descripteur (influence spatiale, mise à jour du descripteur,…etc.). Les articles **[Bre05][Ris04][Yil06]** offrent une vision d'ensemble des variations possibles.

5.1.1.2.5 Méthodes exploitant des modèles de points-clés

Ce type de techniques enrichissent le suivi de caractéristiques extraites de l'ensemble du nuage de points afin d'accroître la fiabilité du suivi. Dans la méthode de **Gabriel [Gab05]**, le mouvement de l'objet est déduit du mouvement du centre de gravité des points associés. Supposant que l'on dispose des points issus des images *t* et *t+1* ainsi que de leur appariement, le centre de gravité est calculé en plusieurs étapes. Tout d'abord, un calcul préliminaire du centre de gravité G_1 est effectué à partir des points de l'image *t+1*. Un second centre de gravité G_2 est prédit du mouvement global de l'objet entre l'image *t-1* et l'image *t*. Ce point G_2 est la position théorique de l'objet à l'image *t+1* d'après son mouvement précédent. Les points sont ensuite considérés un par un. Ceux ayant une influence trop marquée sur l'éloignement de G_1 de sa position théorique G_2 sont éliminés et G_1 est recalculé. Ce filtrage permet de réduire l'influence des appariements erronés. Le mouvement de l'objet retenu est la différence de position entre les centres de gravités des images *t* et *t+1*.

« Flocks of features » est une approche originale au problème des objets déformables récemment proposée par **Kölsch et Turk [Kol05]**. Elle s'inspire de l'observation du vol de groupes d'oiseaux. Leurs formations conservant en effet constamment une certaine cohésion spatiale bien qu'aucun individu ne se trouve trop près de son voisin. La méthode de Kölsch et Turk fonctionne par analogie, assimilant le nuage de points-clés de l'objet à suivre à une formation de volatiles. Pour ce faire, les auteurs ont établies trois règles de positionnement des points :

1. Aucun point ne doit être plus proche qu'une distance d_1 des ses voisins.
2. Aucun point ne doit être plus éloigné qu'une distance d_2 du centre de gravité de l'ensemble de points.
3. La cardinalité de l'ensemble de points doit rester constante.

Tout point ne pouvant satisfaire les deux premières règles, a un mouvement par rapport au reste de l'objet considéré comme aberrant. Il est alors traité comme *distracteur* ou faux appariement et éliminé. Afin de respecter la troisième règle, il est remplacé par un nouveau point. Cette technique permet de gérer d'importantes déformations internes à l'objet (voir Figure 59). Dans leur méthode, les auteurs se basent sur les points KLT **[Bru81][Tom91][Shi94]** dont le déplacement est estimé à partir du flot optique, afin d'obtenir un algorithme temps-réel. Toutefois, on peut supposer la méthode applicable à tout type de points-clés.

Figure 59: Un exemple de l'algorithme « flocks and features » appliqué au suivi de main. Les petits points représentent les points-clés, le gros point, le centre de gravité. **[Kol05]**

Une telle approche implique toutefois une échelle de l'objet constante afin de pouvoir garantir la règle de cardinalité ainsi que la présence d'une couleur discriminante pour l'objet. Cette méthode fut par la suite associée à un filtre particulaire **[Hoe06]** pour parfaire l'estimation de la position de l'objet.

Une dernière approche intéressante proposée par **Donoser [Don06]** s'appuie sur les *régions d'extrema maximalement stable* ou *MSER* (voir **4.1.1.2.3**) pour effectuer un suivi robuste. Une partie de la contribution des auteurs repose sur une structure d'arbre (voir aussi **[Mur06]**) modélisant les hypothèses de l'algorithme *MSER* et en maximalisant donc l'efficacité. Le reste de leur contribution optimise l'application des *MSERs* dans le cadre du suivi, exploitant l'hypothèse d'une variation minime entre deux images. L'arbre d'hypothèses peut alors être taillé et le spectre des valeurs d'intensités analysé par l'algorithme contraint à des valeurs proches de celles du candidat recherché. Cette adaptation astucieuse des *MSERs* au problème du suivi d'objet conduit à un algorithme fiable pour un traitement d'images en moyenne 10 fois plus rapide que pour *MSER* seul.

5.1.2 Estimation de paramètres du modèle

L'estimation de paramètres consiste à évaluer les variables d'un modèle d'après un ensemble de données. D'un point de vue plus concret, on peut définir ce type de technique comme l'évaluation globale d'une caractéristique d'après des données locales ou ponctuelles. Toutefois, des erreurs dans l'extraction de ces informations ou la présence de données appartenant à un autre modèle, créent la présence de données aberrantes appelées *distracteurs (outliers)*. De plus, des imprécisions des capteurs ou dans l'extraction des données peuvent conduire à des données bruitées. Notons que l'on ne dispose généralement pas d'informations a priori sur la quantité de bruit ou de *distracteurs*. Plus formellement, le problème se définit comme suit :

Pour un ensemble x de n vecteurs de données $\{x_1,\ldots,x_n\ ,\ y\}$ et pour un modèle $f()$ constitué de m paramètres $\{p_1,\ldots,p_m\}$, on cherche le jeu de paramètre p qui explique au mieux les données :

$$Argmin(E(p)) = Argmin(f(x \mid p)\text{-}y)$$

, sachant que celles-ci peuvent être corrompues par du bruit ou des données aberrantes.

On peut donc distinguer deux caractéristiques essentielles dans les algorithmes d'estimation de paramètres : l'**exactitude** et la **robustesse**. L'exactitude de l'extraction du modèle sera la capacité de l'algorithme à décrire précisément un modèle à partir de données exemptes de bruit ou de *distracteurs*. Elle dépend principalement de la précision de définition du modèle. La robustesse de l'algorithme sera sa capacité à gérer bruit et *distracteurs*. La résistance d'un algorithme aux distracteurs est mesurée grâce à la notion de **point de rupture**. Le point de rupture d'un algorithme est définit comme étant le pourcentage maximum de *distracteurs* que celui-ci peut supporter sans que son estimation soit faussée. Il existe deux stratégies distinctes afin d'obtenir un algorithme robuste. La première consiste à éliminer préalablement les *distracteurs* pour ensuite estimer les paramètres du modèle à partir des données restantes. La seconde méthode traite simultanément les problèmes de la gestion du bruit et de l'élimination des données aberrantes.

Dans le cas du suivi d'objet, on cherche à extraire un modèle de mouvement d'un ou plusieurs ensembles d'association de primitives. Ce processus revient donc à estimer un mouvement global (ou des mouvements locaux) à partir de primitives précédemment associées. Les *distracteurs* seront ici des primitives appartenant à l'environnement ou à un autre objet, ou encore mal associées. Le bruit sera issu des imprécisions des capteurs et de détection des points-clés. Parmi les nombreuses techniques proposées dans la littérature, nous ne nous focaliserons ici que sur des méthodes satisfaisant les contraintes de temps de calcul du suivi d'objet. Nous détaillerons la méthode des moindres carrés, la transformée de Hough, les M-estimateurs, les LMS (Least Median of Squares), et enfin l'algorithme RANSAC (RAndom SAmple Consensus). Pour un état de l'art sur le sujet, consulter **[Zha95][Ste99][Mal06]**.

Le lecteur désireux de se renseigner sur des méthodes plus performantes, mais aussi plus coûteuses en termes de temps de calcul, tels que les algorithmes ALKS, MUSE, MINPRAN, RESC, et MDPE se référera à **[Wan04]**.

5.1.2.1 La méthode des moindres carrés

Cette méthode, créée par **Gauss et Legendre [Rou87]** repose sur l'évaluation de la différence entre les données *x* et le modèle *f()*.Cette erreur ou *résidu* peut être positive, négative ou nulle.

$$r_i = f(x_i) - y$$

L'idée est de mesurer l'ajustement du modèle aux données en effectuant la somme des carrés des résidus de l'ensemble des données. Si cette somme est petite, l'ajustement est considéré comme étant bon. À l'inverse, si elle est grande, l'ajustement est mauvais. Parmi tous les jeux de paramètres *p* possibles du modèle, celui qui offre le meilleur ajustement d'un ensemble de données est celui qui minimise l'erreur totale *E* suivante :

$$E(p) = \sum_i r_i^2(p)$$

Un exemple est présenté en Figure 60.

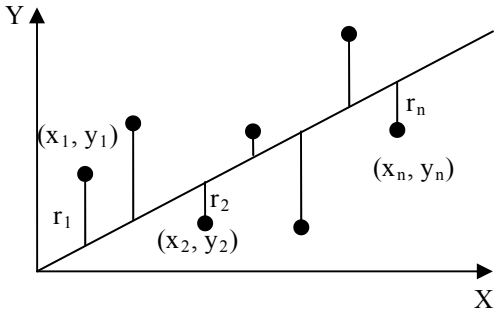


Figure 60: Exemple d'ajustement d'une droite par la méthode des moindres carrés.

On appelle modèle des moindres carrés le modèle qui vérifie cette propriété. Supposons que l'on tente d'expliquer un ensemble de *n* points (x_i, y_i) par un modèle linéaire d'équation $Y = a_0 + a_1 X$. Les constantes a_0 et a_1 peuvent être déterminées en résolvant le système d'équations suivant :

$$\begin{cases} \sum Y = a_0 n + a_1 \sum X \\ \sum XY = a_0 \sum X + a_1 \sum X^2 \end{cases}$$

Ce système peut alors être résolu à l'aide des formules suivantes:

$$a_0 = \frac{\sum Y \sum X^2 - \sum X \sum XY}{n\sum X^2 - \left(\sum X\right)^2} \quad et \quad a_1 = \frac{n\sum XY - \sum X \sum Y}{n\sum X^2 - \left(\sum X\right)^2}$$

Il est bien évidemment possible d'expliquer les données avec d'autres modèles plus complexes, le choix étant fonction de l'information dont on dispose sur les données. Le système d'équations et sa résolution seront alors similaires. De même, il est possible d'adapter ce procédé à des données de plus de trois variables.

Cette méthode est très efficace lorsque les données sont corrompues par du bruit mais, en revanche, est très sensible aux *distracteurs* (son point de rupture est de 0%). Afin de pallier à ce désavantage, une étape préliminaire peut être ajoutée afin d'éliminer les données aberrantes.

Pour conclure, la méthode des moindres carrés est une référence en matière de gestion du bruit, mais des études ont montré que même une unique donnée aberrante peut gravement perturber l'algorithme. Elle est donc à réserver pour le traitement de données comportant un taux faible ou nul de *distracteurs*. En conséquence, des algorithmes plus robustes ont été proposés au cours des dernières décennies.

5.1.2.2 La méthode des moindres médians des carrés

Afin de pallier à l'incapacité de la méthode des moindres carrés à gérer les *distracteurs*, la méthode robuste LMS (Least Median of Squares) **[Rou84]** fut proposée. Elle minimise la fonction de coût suivante :

$$E(p) = médian\left(\sum_i r_i^2(p)\right)$$

, avec p un jeu de paramètres possibles. Si cette fonction est très robuste (son point de rupture théorique est de 50%), il est en revanche impossible d'établir une formule ou un système d'équation permettant de trouver le meilleur jeu de paramètre minimisant $E(p)$. Le problème doit alors être résolu par une recherche dans l'espace des paramètres. Une recherche exhaustive étant trop coûteuse, l'algorithme doit s'appuyer sur un espace des paramètres possibles. L'exploration de cet espace est très délicate car, le médian n'étant pas différenciable, des méthodes d'exploration de l'espace des solutions telles que la montée de gradient, les algorithmes génétiques, ou encore le recuit simulé, sont ici très difficiles à mettre en œuvre. De plus, si le gradient est faible au niveau de la médiane, la convergence peut être très lente. Les approches envisageables afin de sélectionner des jeux de paramètres pertinents sont alors limitées et sous-optimales :

- Effectuer un tirage aléatoire de m sous-ensembles de n données et déterminer le meilleur jeu de paramètres pour chacun de ces sous-ensembles.
- Déterminer un sous–ensemble probable à partir de connaissances a priori sur le modèle. Par exemple, dans le cadre du suivi d'objet, le mouvement de l'objet sera réduit à un certain intervalle induit par ses déplacements et son accélération sur les précédentes images, ainsi que l'incertitude sur son mouvement actuel.

Ce défaut doublé du fait que la méthode n'est guère stable en la présence de bruit, comme noté dans **[Rou87]**, ne rend guère la méthode attractive pour des problèmes de vision par ordinateur. En conséquence, Rousseeuw **[Rou87]** proposa par la suite la méthode LTS (Least Trimmed Squares) qui minimise la fonction de coût suivante pour un ensemble de q données :

$$E(p) = \sum_{i=1}^{q} r_i^2(p)$$

, où les résidus sont triés en fonction de leur importance. Cette fonction est beaucoup plus facile à minimiser, mais ces résultats sont bien sûr dépendants de la valeur de *q*. Une connaissance à priori du nombre de *distracteurs* permet d'en optimiser l'efficacité. Dans le cas contraire, une mesure couramment utilisée de l'écart-type est le *Mad* (Median Absolute Deviation) :

$$\sigma = 1.4826 \times median\left(r_i - median(r_i)\right)$$

Cette variante constitue un bon compromis entre rapidité et efficacité.

5.1.2.3 Les M-estimateurs

Une technique populaire et robuste d'estimation des paramètres sont les M-estimateurs, que l'on peut considérer comme une généralisation de la méthode des moindres carrés. En effet, alors que cette dernière tente de minimiser, les M-estimateurs remplacent le carré des résidus par une autre fonction des résidus, de sorte que, afin de rendre la méthode plus stable à la présence de *distracteurs,* l'on cherche à minimiser :

$$E(p) = \sum_{i=1}^{q} \rho\left(r_i(p)\right)$$

ρ est une fonction symétrique, positive avec un unique minimum égal à 0. Minimiser cette fonction de coût revient à trouver le vecteur de paramètres $p = [p_1,\ldots,p_m]$ qui résout le système d'équations suivants :

$$\sum_i \psi(r_i)\frac{\partial r_i}{\partial p_j} = 0, \quad pour \quad j = 1,...,m$$

, avec $\psi(r_i) = \rho'(r_i)$. Cette fonction représente l'influence de la donnée r_i sur l'estimation des paramètres *p*. Par exemple, pour la méthode des moindres carrés, $\rho(r)=r^2$, la fonction d'influence sera donc $\psi(r)=2r$, ce qui signifie que l'influence d'une erreur sur l'estimation des paramètres augmentera linéairement avec l'importance de l'erreur. Ce qui prouve que la technique des moindres carrés ne soit pas stable en présence de *distracteurs*. Afin de limiter l'influence des données aberrantes, on cherche donc à construire des M-estimateurs tels que la fonction $\psi(r)$ soit bornée. Si l'on introduit maintenant la fonction de poids :

$$w(r) = \psi(r) / r$$

qui reflète la confiance en chaque donnée de faire partie du modèle à décrire. En remplaçant $\psi(r)$ dans l'équation précédente, elle devient alors :

$$\sum_i w(r_i) r_i \frac{\partial r_i}{\partial p_j} = 0, \quad pour \quad j = 1,...,m$$

Ces équations conduisent à l'algorithme IRLS (Iteratively Reweighted Least Squares) qui successivement, calcule les poids $w(r_i^{k-1})$ à partir de l'estimation courante des paramètres p^{k-1}, puis les nouveaux paramètres p^k (et donc les nouveaux résidus r_i^k) à l'aide des poids mis à jours. L'estimation initiale des paramètres peut être obtenue de nombreuses façons, des connaissances sur le modèle où l'exécution d'un algorithme simple telle que la méthode des moindres carrés étant les plus courantes.

Le choix du M-estimateur est bien sûr capital. La Figure 61 ci-dessous représente les M-estimateurs les plus couramment utilisés.

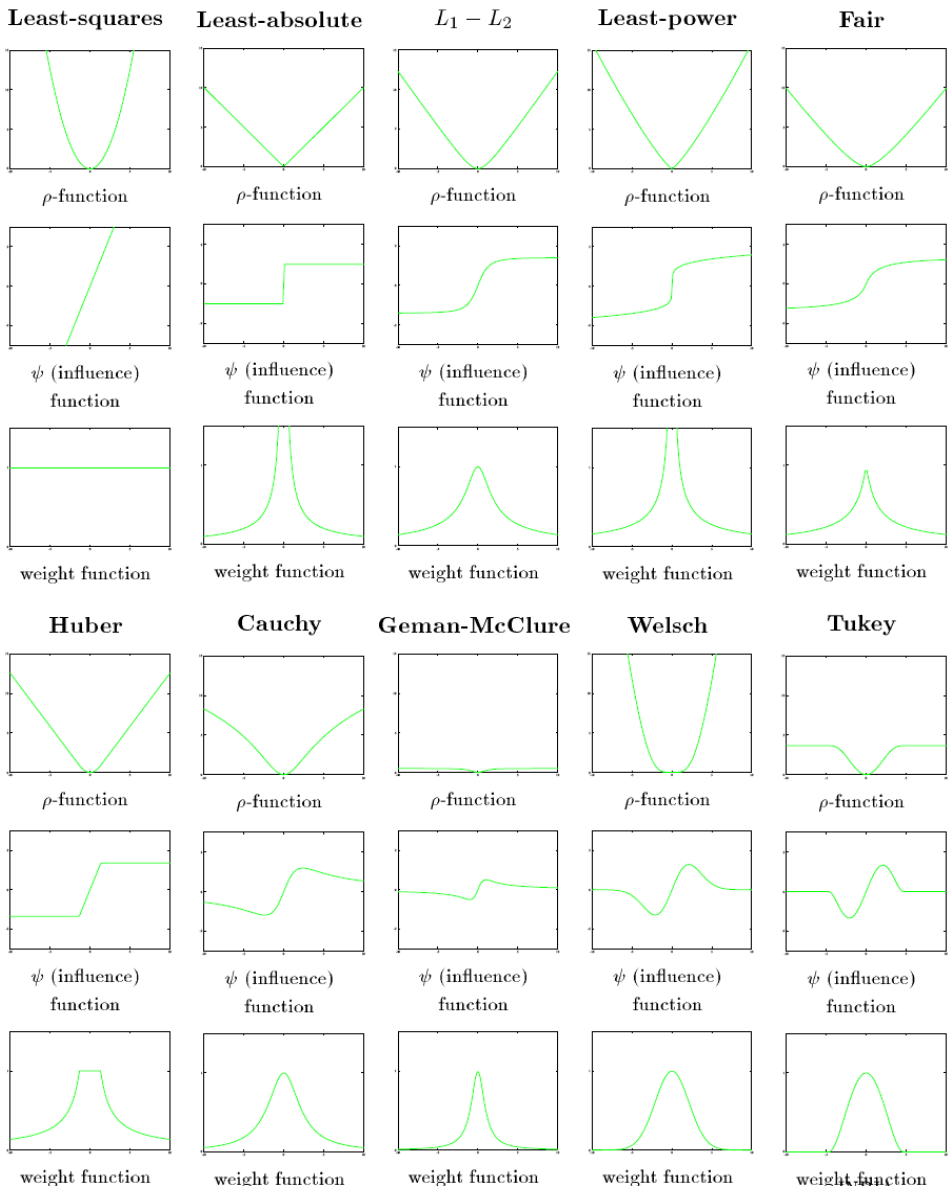


Figure 61: Représentation graphique des fonctions de quelques M-estimateurs communs.

Il est ardu de conseiller l'utilisation d'une fonction, particulièrement dans la mesure où le choix de l'estimateur dépend du problème à traiter. Notons toutefois que les M-estimateurs L_p, de Huber, et de Tukey sont ceux qui donnent généralement les meilleurs résultats.

Bien qu'ayant un point de rupture théorique de 0%, les M-estimateurs donnent de très bons résultats, ce qui en fait des outils largement utilisés dans la littérature.

5.1.2.4 La transformée de Hough

La transformée de Hough **[Hou59]** est une des plus vieilles méthodes utilisée en vision par ordinateur. Plutôt que de trouver un jeu de paramètres qui explique au mieux les données selon un certain critère, l'originalité de la méthode consiste à transposer les données dans l'espace des paramètres puis à élire le jeu de paramètres expliquant le plus de données. Il s'agit donc d'un algorithme de vote ou chacune des

donnée incrémente les compteurs associés aux différents jeux de paramètres qui expliquent cette information. A l'issue de l'algorithme, les compteurs aux valeurs les plus élevées représenteront les jeux de paramètres les plus probables pour expliquer les données. Par exemple, supposons que pour un ensemble de points donné, on cherche à construire les droites les plus probables, en termes de nombres de points passant par cette droite. Pour chaque point, on va donc incrémenter les compteurs de l'ensemble des droites auxquelles il appartient. Les compteurs cumulant la valeur la plus élevée correspondront aux droites les plus probables.

Un désagrément de la méthode est la gestion du bruit. En effet, des données bruitées vont conduire à incrémenter des compteurs correspondants à des jeux de paramètres différents, bien que proche dans l'espace des paramètres. Pour reprendre l'exemple de la recherche de droites dans un ensemble de points, de points localisés imprécisément résultera l'incrémentation de plusieurs droites géométriquement similaires là où des points parfaitement placés éliraient une seule droite. Plusieurs contremesures ont été proposées :

- Plutôt que d'élire les paramètres à partir du pic d'une valeur, il est possible de prendre en compte un voisinage (en utilisant une gaussienne par exemple).
- Des techniques de classifications peuvent être utilisées pour différencier les candidats.

Cette technique est très largement utilisée et de nombreuses variantes sont disponibles **[Ill88]**. L'une, communément utilisée, consiste à modéliser la confiance dans un jeu de paramètres de façon probabiliste en tenant compte du bruit. Au lieu d'incrémenter un compteur par un entier, la valeur ajoutée est alors un réel prenant en compte l'incertitude sur l'information. Les performances de l'algorithme sont alors considérablement accrues.

La transformée de Hough est donc une méthode très robuste qui permet de traiter aussi bien le bruit qu'un nombre important de *distracteurs*. Toutefois elle souffre de deux limitations. Tout d'abord la nécessité d'avoir une quantité d'information bien plus importante que le nombre de paramètres à déterminer pour avoir un algorithme fiable. Deuxièmement, le temps de calcul, prohibitif pour un algorithme de vision par ordinateur.

5.1.2.5 L'algorithme RANSAC (1981)

L'algorithme RANSAC (RANdom SAmple Consensus) est un algorithme robuste pouvant gérer une importante quantité de *distracteurs* pour apparier des données. Il fut inventé par **Fischler & Bolles [Fis81]** en 1981. Il s'agit d'un estimateur aléatoire capable de gérer une importante quantité de bruit. Il a largement été appliqué à de nombreux problèmes de la littérature tels que l'appariement de points ou la détection de primitives géométriques. Plus précisément, pour un ensemble M de m données, RANSAC cherche un modèle décrit par un vecteur de paramètres $\{x_i\}$ tel qu'un sous ensemble K de M, satisfasse ce modèle. Son algorithme est le suivant :

1-Sélectionner aléatoirement un ensemble N de n données (n<m).
2-Estimer $\{x_i\}$ d'après les données de N.
3-Trouver combien de données de M satisfont $\{x_i\}$. Soit K cet ensemble.
4-Si K est suffisamment grand, terminer avec succès.
5-Sinon, répéter les étapes 1 à 4 i fois.
6-Echec de l'algorithme si cette étape est atteinte.

Si les données contiennent plusieurs structures, alors, après avoir trouvé chaque modèle, on élimine les données qui lui sont associées et on relance l'algorithme.

De nombreuses améliorations ont été apportées à cet algorithme. Dans la méthode MLESAC présenté par **Torr et Zisserman [Tor00],** l'estimation de la correspondance entre l'ensemble N de

données tirées aléatoirement et le modèle est effectuée, non pas par décompte du nombre d'éléments de l'échantillon satisfaisant les contraintes du modèle, mais par probabilité que cette distribution soit cohérente avec le modèle. En pratique, une distribution gaussienne de probabilité est associée à chaque appariement de données, la probabilité de validité du modèle par rapport à ces données est ensuite évaluée suivant cette distribution. Par la suite, **Tordoff et Murray [Tor02]** basèrent leur système de correspondance guidé (donc impliquant une distribution non uniforme) sur cet algorithme.

Plus récemment, **Chum et Matas [Chu05]** proposèrent l'algorithme PROSAC (PROgressive SAmple Consensus) plus rapide que son successeur. Il est basé sur deux améliorations. Tout d'abord, les données sont triées dynamiquement par rapport à une fonction de qualité. Les données de forte qualité sont ainsi sélectionnées en priorité, afin d'accroître les chances d'obtenir un ensemble *N* pertinent. De plus, afin de diminuer le temps de calcul, les tirages sont effectués sur des ensembles de plus en plus grands. Des résultats similaires à ceux de la méthode RANSAC sont obtenus environ cent fois plus rapidement.

Malgré sa capacité à traiter une importante quantité de *distracteurs* et de disparition / apparition de points, cet algorithme nécessite de préciser au préalable la quantité de *distracteurs* supposée. De plus, bien qu'il puisse traiter efficacement jusqu'à 80% de *distracteurs*, la gestion du bruit est reléguée au second plan.

5.1.2.6 La longueur de description minimale (minimum description length)

La longueur de description minimale (MDL) est une technique à l'origine employée en théorie de l'information qui minimise le nombre de bits utilisés pour encoder un signal. L'idée repose sur l'élimination des redondances dans le code. Par analogie avec un problème d'estimation de probabilité maximale classique où on cherche à maximiser la fonction de probabilité $P(x \mid p)$, avec x les données et p les paramètres du modèle, on cherche ici à minimiser $-\log P(x \mid p)$. De plus, on inclut également la fonction $L(p)$ représentant la longueur de codage du jeu de paramètres p. Cette partie va évaluer la complexité du jeu de paramètres et contrebalancer la fonction d'ajustement précédente afin d'éviter des modèles invraisemblables. En effet, comme le montre la Figure 62, un sur-ajustement du modèle aux données n'est pas souhaitable dans la mesure où une telle structure serait par trop sensible à l'influence du bruit et des *distracteurs*. Cette combinaison de deux types de fonctions aux objectifs opposés fait l'originalité de l'approche.

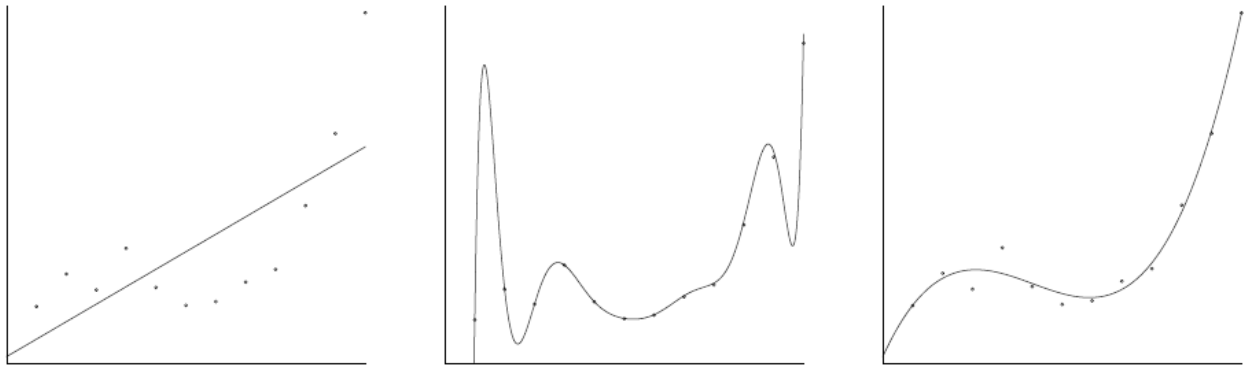


Figure 62: Illustration de la nécessité de contrôler la complexité d'un modèle par l'ajustement d'une courbe polynomiale à un ensemble de points. (a) Sous-ajustement. (b) Sur-ajustement. (c) Ajustement optimal.

La longueur de description minimale est alors donnée par la fonction :

$$mdl(x,p) = \min(-\log(P(x \mid p)) + L(p))$$

Le principal avantage de cette approche est la possibilité de complètement décrire la structure du modèle de part non seulement la valeur des paramètres mais aussi leur nombre. Il est dès lors envisageable d'ajouter un paramètre λ représentant la quantité de *distracteurs* afin d'en optimiser la gestion. En revanche il n'existe pas de méthode analytique pour résoudre ce système. Tout comme pour la méthode des moindres carrés des médians, il faut alors explorer l'espace des solutions à l'aide de techniques approximatives. Pour plus de détails sur l'utilisation du critère MDL aux problèmes d'estimation de paramètres, se référer à **[Gru00]**.

5.2 Contribution

5.2.1 Suivi de points d'intérêts

Rappelons que l'on cherche à établir des correspondances entre un ensemble de *n* points d'un modèle et un ensemble de *m* points (que l'on appellera ici des *« sites »*) d'une image observée. Le modèle généralement utilisé est l'ensemble de points de l'image précédente. Le problème se ramène donc à apparier les points des images *t* et *t-1*.

5.2.1.1 Fenêtre de recherche

A chaque nouvelle image, les points sont recherchés dans une fenêtre rectangulaire dont la taille est proportionnelle à leur mouvement. Celle-ci est initialisée comme un carré de 20 pixels d'arête centré sur la position du point-clé puisque, pour la première image, aucune information de mouvement n'est encore disponible. Pour le reste de l'algorithme, la fenêtre de recherche est centrée sur la position supposée du point, déduite de son mouvement à l'image précédente. La hauteur (respectivement la largeur) de la fenêtre est égale à 4 fois son mouvement avec un minimum de 6 pixels (voir Figure 63). Ces dimensions représentent l'incertitude maximale quant à la position estimée du point pour des vidéos d'une résolution inférieure ou égale à 720×576 pixels.

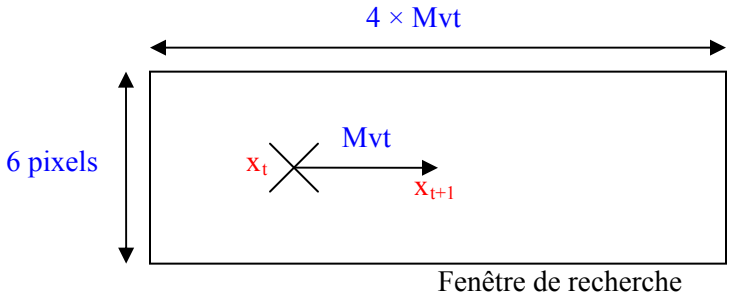


Figure 63: Illustration du calcul de la taille et de l'emplacement de la fenêtre de recherche utilisée pour l'appariement des points.

Il est important de justifier la raison d'une incertitude sur la mesure aussi grande ainsi que pourquoi nous n'utilisons pas le très répandu filtre de Kalman. Cette technique repose sur l'hypothèse d'une certaine fluidité de mouvement de la part de la primitive suivie. Si cette hypothèse se vérifie presque toujours dans le cas du suivi d'un objet, elle n'est plus vraie à l'échelle des points-clés. En effet, d'une part à cause des imprécisions de localisation, d'autre part à cause des déformations internes de l'objet, leurs mouvements peuvent être sujets à de brusques variations. Par exemple, si, l'on se focalise

sur le pied d'une personne en train de marcher ou courir, celui-ci va décrire de façon cyclique une trajectoire en forme de demi-cercle (levé de la jambe) suivi d'une brusque interruption (pose du pied). Le filtre de Kalman échouera à suivre ce type de mouvement. Le paramétrage d'une grande incertitude nous permet donc de répondre à cette difficulté du suivi. De plus, cela nous offre également plus de chances de retrouver le point après une perte ou un faux appariement. En contrepartie le nombre d'ambigüités survenant au cours de l'appariement est multiplié, ce qui amène le besoin d'un algorithme d'appariement plus fiable.

5.2.1.2 Appariement normalisé exploitant les relations de voisinage entre les points

Le choix de notre technique d'appariement s'est trouvé particulièrement restreint par le contexte du projet PorTiVity et les primitives utilisées. En effet, toute approche algorithmiquement coûteuse, tel que le suivi multi-hypothèse, nous était interdite par l'impératif de rapidité du système. De plus, l'instabilité temporelle des points-clés rendait inappropriée les approches exploitant la notion de trajectoires, nécessitant une continuité minimum. Notre système se base donc sur un type d'appariement classique entre deux ensembles de points issus d'images consécutives.

Lors de nos premières expériences, nous avons constaté une insuffisance de l'algorithme d'appariement dans plusieurs cas de figure. Par exemple, certains points, bien que parfaitement localisés n'étaient pas appariés car la distance calculée entre les descripteurs était un peu en dessous du seuil d'appariement. Ou encore dans le cas de points spatialement proches, les descripteurs, étant souvent similaires, conduisaient parfois à des inversions de points lors de l'appariement. Ces erreurs faussaient par la suite le modèle de mouvement. Afin de pallier au problème nous avons décidé d'élargir nos critères d'appariement en ne nous basant pas exclusivement sur l'information visuelle mais également sur l'information spatiale.

Le principe consiste à utiliser les relations de voisinage entre les points et non uniquement les descripteurs pour décider de la correspondance entre un point du modèle et un point de l'image observée. Les relations de voisinages sont modélisées à l'aide d'une triangulation de Delaunay (voir l'annexe **8.3**). En effet, de par le critère de qualité de l'angle minimum, la triangulation de Delaunay est prouvée avoir en moyenne, des triangles peu aplatis, plus adaptés pour représenter la notion de proximité. La triangulation est effectuée sur les points de l'image observée plutôt que sur ceux du modèle. Deux facteurs ont influencé cette décision. Premièrement, le nombre de points de l'image observée étant toujours inférieur ou égal au nombre de points du modèle, une telle triangulation sera plus rapide. Deuxièmement, cela permet de mettre à jour le mouvement des points non appariés en fonction de celui de leurs voisins appariés (voir **4.2.1.6**).

Afin d'exploiter les relations de voisinage conjointement aux informations visuelles, nous avons mis au point un algorithme normalisé inspiré d'une technique émergeante : les « champs discriminants aléatoires » **[Kum04][Kum05][Leo05]**. Bien que notre modèle de l'objet prenne en compte les points issus de plusieurs images (voir **4.2.1.3**), pour des raisons de clarté, nous considérerons ici le modèle classique ne traitant qu'une seule image antérieure. On comparera donc les n points du modèle $\{x_{t-1}^i\}, \quad i \in 1...n$, issus de l'image *t-1* aux m sites de l'image t observée $\{x_t^j\}, \quad j \in 1...m$. L'algorithme se décompose en deux parties. Dans un premier temps, on initialise, les *potentiels appariements* PA_0. Ce score évalue la possibilité d'associer un point à un site. Cette première itération ne se base que sur les descripteurs visuels.

$$P_0(x_t^j, x_{t-1}^i) = sim(desc(x_t^j), desc(x_{t-1}^i)) \quad 0 \le P_0(x_t^j, x_{t-1}^i) \le 1$$

, avec *sim(desc1,desc2)* la métrique de comparaison de deux descripteurs *desc1* et *desc2* et *desc(x)* la fonction retournant le descripteur du point *x*. A l'issue de cette étape, on a défini, pour chaque site de l'image observée, un ensemble de k ($k \ge 0$) points candidats à l'appariement. Afin de limiter les calculs

ultérieurs, seuls les trois points les plus similaires sont conservés ($k \leq 3$). Dans un deuxième temps, on va recalculer de manière itérative, et pour chaque candidat à l'appariement, le *potentiel d'appariement* PA_i en se basant sur le *potentiel d'interaction* PI_i. Cette grandeur estime la cohérence d'une hypothèse d'appariement (i.e. l'appariement d'un point candidat à un site) en fonction de la crédibilité de la configuration environnante. Un appariement n'est donc plus considéré de manière isolée mais conjointement aux appariements de son voisinage. Quantifier la l'éventualité d'une configuration implique d'évaluer celle de chaque voisinage. Cette tâche est réalisée grâce à quatre paramètres :

- Les potentiels d'appariement $PA_{i-1}(x_t^j, x_{t-1}^i)$ et $PA_{i-1}(x_t^l, x_{t-1}^k)$ de chacun des deux points x_{t-1}^i et x_{t-1}^k pour leur site respectifs x_t^j et x_t^l.
- L'angle $\left(\overrightarrow{x_t^j x_{t-1}^i}, \overrightarrow{x_t^l x_{t-1}^k}\right)$ formé par les vecteurs $\overrightarrow{x_t^j x_{t-1}^i}$ et $\overrightarrow{x_t^l x_{t-1}^k}$ constitués du point du modèle et de son associé dans l'image observée. Cette valeur est comprise entre 0 et 180 degrés. Cet angle est ensuite converti en un score P_{angle}, un angle de 180 degrés étant assimilé à une totale impossibilité d'occurrence et un angle de 0 degré induisant une confiance absolue dans l'appariement.
- La différence entre les distances euclidiennes des vecteurs $\overrightarrow{x_t^j x_{t-1}^i}$ et $\overrightarrow{x_t^l x_{t-1}^k}$. La première mesure possible serait :

$$d_1(x_t^j, x_{t-1}^i, x_t^l, x_{t-1}^k) = \frac{\min\left(d(x_t^j, x_{t-1}^i), d(x_t^l, x_{t-1}^k)\right)}{\max\left(d(x_t^j, x_{t-1}^i), d(x_t^l, x_{t-1}^k)\right)}$$

, avec *d(a,b)* la distance euclidienne entre les points *a* et *b* :

$$d(a,b) = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$$

Toutefois, cette distance donne de mauvais résultats dans le cas où *d* est petite (typiquement, 1 à 10 pixels). Par exemple, deux groupes de points distants respectivement de 1 et 3 pixels seront considérés comme éloignés selon cette distance (d1=0.33). Dans ce cas de figure, on préférera la distance suivante :

$$d_2(x_t^j, x_{t-1}^i, x_t^l, x_{t-1}^k) = 1 - \left|d(x_t^j, x_{t-1}^i) - d(x_t^l, x_{t-1}^k)\right| / 10$$

Afin de couvrir toutes les possibilités, nous avons donc opté pour la distance suivante :

$$d_3(x_t^j, x_{t-1}^i, x_t^l, x_{t-1}^k) = \max\left(d_1(x_t^j, x_{t-1}^i, x_t^l, x_{t-1}^k), d_2(x_t^j, x_{t-1}^i, x_t^l, x_{t-1}^k)\right)$$

L'importance accordée à chacun de ces paramètres peut être ajustée par un système de poids. Mais, la pertinence de ces quatre facteurs varie en fonction de la régularité du mouvement de l'objet suivi, elle-même dépendante du type de vidéo étudiée. Comme notre système tend à être générique, tout type de vidéo doit être attendu. Nous avons donc fixé des poids équivalents pour chacun des paramètres. Leur moyenne offre ensuite une estimation satisfaisante de la possibilité d'un voisinage, excepté lorsque $x_{t-1}^i = x_{t-1}^k$ qui est une configuration impossible. En effet, un même point ne peut être affecté à deux sites. Dans ce cas son score est nul. Plus formellement, on a :

$$PI_i\left(x_t^j, x_{t-1}^i \middle| x_t^k, x_{t-1}^{kl}\right) = \begin{cases} \left(PA_{i-1}(x_t^j, x_{t-1}^i) + PA_{i-1}(x_t^l, x_{t-1}^k) + P_{angle} + d_3\right)/4 & si \quad x_{t-1}^i \neq x_{t-1}^k \\ 0 & si \quad x_{t-1}^i = x_{t-1}^k \end{cases}$$

, avec x_{t-1}^{kl} désignant le candidat *l* à l'appariement avec le site x_t^k.

Afin d'évaluer la possibilité globale d'une configuration, pour un site x_t^j et un point candidat à l'appariement x_{t-1}^i fixés, les potentiels d'interactions pour tous les voisins ayant des candidats doivent donc être combinées. Et ce, pour la totalité des configurations possibles. En effet, si pour chaque site, il existe jusqu'à trois points candidats possibles, il en résultera une multitude de voisinages possibles. Deux formulations s'offrent alors à nous :

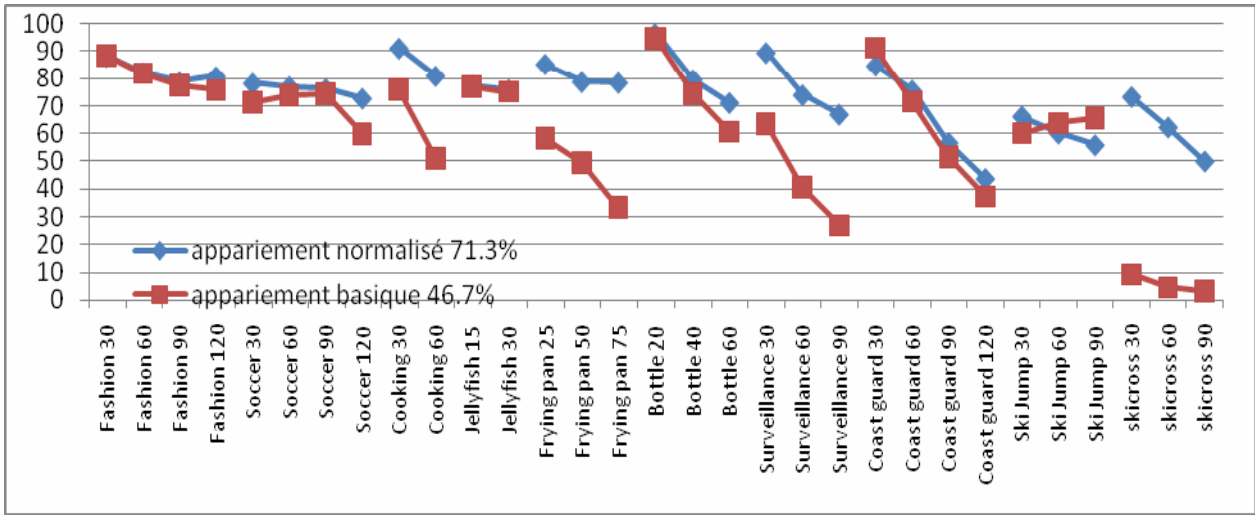
$$PA_i\left(x_t^j, x_{t-1}^i \middle| x_t^k, x_{t-1}^{kl}\right) = \sum_l \left[\max_k \left(PI_i\left(x_t^j, x_{t-1}^i \middle| x_t^k, x_{t-1}^{kl}\right)\right)\right]$$
$$PA_i\left(x_t^j, x_{t-1}^i \middle| x_t^k, x_{t-1}^{kl}\right) = \sum_l \left[\prod_k PI_i\left(x_t^j, x_{t-1}^i \middle| x_t^k, x_{t-1}^{kl}\right)\right]$$

Toujours, afin de respecter le critère d'unicité de l'appariement, seules des configurations crédibles sont étudiés, i.e. des cas où un même candidat n'est pas affecté à deux sites distincts. La première possibilité favorise la configuration la plus probable alors que la seconde considère l'information issue de configurations possibles de manière équilibrée. Toutefois lorsque l'une de ces configurations à un score très faible, le score globale PA_i s'en trouve d'autant amoindrie, même si une autre configuration est très probable. En d'autres termes, cette seconde possibilité favorise les cas où la majorité des configurations sont crédibles. Or, en pratique, il existe fréquemment une configuration prédominante pour de multiples cas mineurs. La dernière formulation est donc à bannir. Nous avons implémenté la première.

Les potentiels d'appariement sont donc recalculés itérativement jusqu'à arrêt de l'algorithme. Deux critères d'arrêt sont possibles :

- Effectuer n itérations, puis, affecter à chaque site, le point ayant le score d'appariement la plus élevée, si celle-ci est supérieure à un seuil s_1 fixé par l'utilisateur. On notera que dans le cas particulier ou $n=0$, on est ramené à un algorithme d'appariement classique ne se basant que sur les descripteurs.
- Continuer l'algorithme jusqu'à ce que, pour chaque site, une hypothèse d'appariement soit retenue. On peut considérer une hypothèse d'appariement comme validée si un point candidat a un score supérieur à un seuil s_2 donné, ou si le score de tous les points candidats est inférieure à un seuil s_3.

La Figure 64 montre les performances de notre système de suivi (avec $n = 1$) comparées au même algorithme utilisant un appariement basique, résolvant les conflits par un système de type « winner takes all » qui se contente d'associer chaque point au meilleur candidat possible. Ces résultats soulignent l'ampleur de l'apport de l'appariement normalisé exploitant les relations de voisinages entre les points.



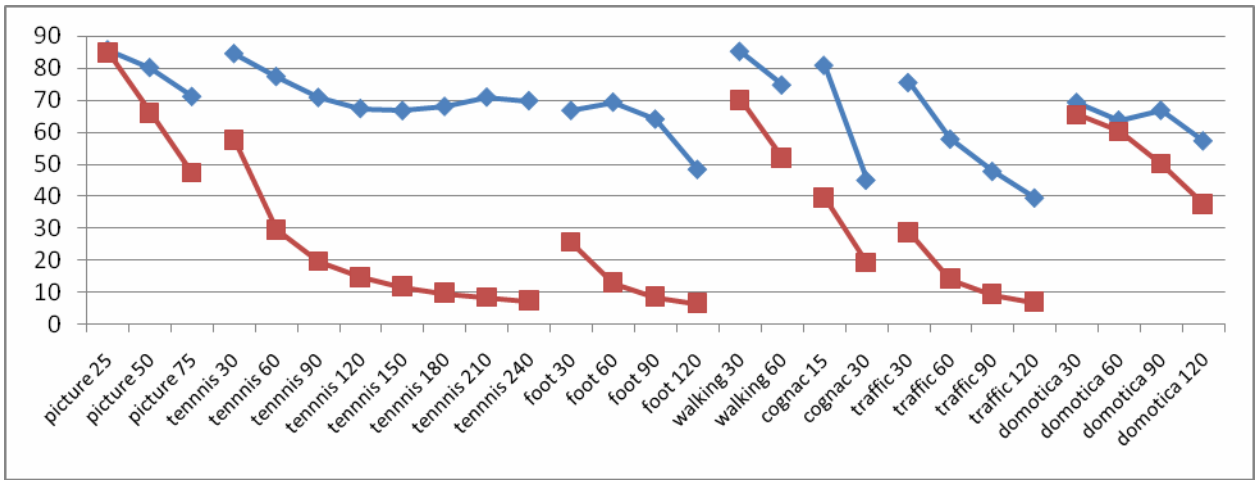


Figure 64: Performances comparées du suivi à base de points-clés avec l'appariement normalisé, et du même suivi avec un appariement basique. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne (en pourcentage de recouvrement) du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des tests est donnée dans l'intitulé des courbes.

5.2.2 Estimation de paramètres du modèle

5.2.2.1 Replacement de la boîte englobante basé sur le mouvement

Il existe peu d'algorithmes de suivi utilisant des points-clés. Les travaux de Gabriel & al **[Gab05]** sont les plus probants en la matière. Cet algorithme prend comme entrée les points appariés à l'image actuelle. Il élimine tout d'abord les points éloignés du mouvement moyen, considérant ceux-ci comme des appariements erronés. Le centre de gravité de ces points est ensuite calculé pour le modèle et l'image étudiée. Afin d'éliminer davantage de faux appariements, les points passent ensuite au travers d'un second tamis. Si l'élimination d'un point modifie de manière significative la position du centre de gravité, alors ce point est considéré comme un faux appariement et n'est pas pris en compte. La dernière étape consiste à replacer la boîte englobante sur l'image actuelle de façon à ce que sa position relative au centre de gravité soit la même que pour la bounding box du modèle. Cette méthode comporte deux défauts. Tout d'abord, elle ne prend pas en compte les changements d'échelle. Ensuite, dû à l'instabilité temporelle des points-clés énoncée plus haut (voir **4.2.1.3**), des points peuvent apparaître et/ou disparaître d'une image sur l'autre modifiant la position du centre de gravité. Ce problème ajouté à l'imprécision de localisation des points précisée en **3.2**, nuira à la précision de détection du centre de gravité. Comme le placement de la boîte englobante est toujours effectué par rapport à celle de l'image précédente, les erreurs se propageront au fur et à mesure des images.

Pour éviter la plupart de ces inconvénients, nous avons préféré un algorithme se basant sur un modèle de mouvement. Nous présenterons dans cette partie l'algorithme de base et dans les sections suivantes les diverses améliorations.

Connaissant les positions *(x,y)* des points-clés pour le modèle *A* et l'image actuelle *B,* l'algorithme détermine par la méthode des moindres carrés (voir **5.1.2.1**) les valeurs *a0 et a1* de translation, *a2 et a3* de changement d'échelle, et *a4 et a5* de rotation qui explique au mieux le mouvement. Plus formellement, on a :

$$\begin{pmatrix} x_A \\ y_A \end{pmatrix} = \begin{pmatrix} a2 & a4 \\ a5 & a3 \end{pmatrix} \times \begin{pmatrix} x_B \\ y_B \end{pmatrix} + \begin{pmatrix} a0 \\ a1 \end{pmatrix}$$

Afin de faciliter le calcul d'un mouvement local, nous avons fixé *a4=a5=0* et cherchons à identifier celui-ci en priorité par une translation, puis à expliquer l'erreur persistante par un changement d'échelle.
Afin de pouvoir fonctionner correctement, cet algorithme doit reposer sur des données fiables. Quatre facteurs influent sur la qualité de l'information :

- Le nombre de points : En effet la localisation des points étant imprécise (voir **3.2**), le mouvement détecté d'un point peut différer de son déplacement réel d'un ou plusieurs pixels. Il est donc indispensable de disposer d'un grand nombre de données pour pouvoir caractériser le déplacement de l'objet de manière fiable.
- Les faux appariements : Ils représentent un ajout de bruit, et doivent, dans la mesure du possible être purgés des données. Tout comme l'algorithme de Gabriel cette méthode est précédée d'un traitement visant à ne pas prendre en compte les points susceptibles d'être des appariements erronés. Après calcul des moyennes m_x *et* m_y et des écarts types σ_x *et* σ_y respectivement selon les axes *X* et *Y*, on élimine tout point *(x,y)* qui ne satisfait pas la condition suivante :

  $$(m_x - x) < v\sigma_x \quad et \quad (m_y - y) < v\sigma_y$$

  , avec *v* la variabilité de mouvement toléré ($0 < v < 3$). Plus la valeur de *v* sera faible, plus la quantité de faux appariements éliminés sera importante. En revanche, une valeur de *v* plus élevée tolérera une plus grande variabilité interne du mouvement de l'objet et fournira une plus grande quantité de points. Nous avons donc fixé *v* à 2 afin de privilégier ces derniers avantages tout en éliminant les valeurs incohérentes. Afin de mieux caractériser la notion de fluidité du mouvement, il est possible de remplacer les moyennes m_x *et* m_y par les mouvements axiaux de l'objet calculés à l'image précédente. Cette variante limite la variabilité du mouvement, et, bien que l'initialisation sans connaissances à priori sur l'objet soit délicate, donne de meilleurs résultats.
- Les *distracteurs* : Ce sont les zones riches en information et voisines de l'objet mais qui ne font sémantiquement pas partie de ce dernier. Par exemple un autre personnage croisant celui que l'on suit ou un décor de forêt à l'arrière-plan. Les *distracteurs* génèrent donc des données qui vont perturber les algorithmes et constituent avec les faux appariements la première cause d'échec des algorithmes. Nous avons constaté que, à cause de leur vecteurs mouvements nuls ou de direction opposé à celle de l'objet, leur présence diminuait la magnitude du vecteur mouvement estimé. Afin de contrebalancer cette tendance, le vecteur de mouvement obtenu pour l'objet est multiplié par une constante *d* représentant l'interférence des *distracteurs* dans l'estimation du mouvement. Nous avons expérimentalement fixé *d* d'après les meilleurs résultats obtenus sur 8 vidéos (comportant une quantité de *distracteurs* variable) à *d=1.15*.
- L'âge de l'information associée : Notre modèle conservant une mémoire des points (voir **4.2.1.3**), leur dernière occurrence peut dater de une à trois images. Plus l'information date, plus elle risque d'être imprécise. Cependant, plus on limitera l'information à des points dont la dernière occurrence est récente, plus leur nombre sera restreint. Nous ne nous basons donc sur les points datant de l'image précédente que si leur nombre *n* est suffisant (typiquement *n=5*). Dans le cas contraire, tous les points sont exploités.

Les résultats montrent que l'algorithme se comporte généralement mieux que celui de Gabriel & al, excepté dans le cas où la totalité de l'objet suivi est soumis à des variations locales de mouvement comme le montre le cas de la vidéo « méduse ». Toutefois il souffre également d'une propagation des erreurs au fur et à mesure du suivi. En conséquence, après une perte de précision momentanée, l'algorithme aura des difficultés à se recentrer sur l'objet, problème particulièrement ennuyeux dans le cas de longues séquences.

5.2.2.2 Gestion des occultations

Une quantité minimum d'information est requise pour que tout algorithme de ce type fonctionne correctement. Dans le cas où le nombre de points labéllisés « objet » détectés est inférieur à ce minimum vital (fixé à 6), l'objet est alors considéré comme occulté. Dans ce cas, et tant que cette quantité minimum de points-clés n'est pas à nouveau détectée, le système ne repose plus que sur le modèle de mouvement. Aucun traitement des points-clés extraits ni mise à jour du modèle n'est plus effectué. Le mouvement de l'objet est considéré constant, et égal au dernier déplacement estimé.

5.2.2.3 Optimisation en fonction du label des points (objet ou décor)

Si notre modèle de mouvement s'avère efficace dans le cas de mouvement uniforme, il souffre de limitations lorsque différents vecteurs mouvement associés à différentes parties de l'objet rentrent en jeu. Ce cas de figure peut survenir lors du suivi d'objets déformables, mais aussi dans le cas d'objets rigides soumis à un mouvement complexe, comme une rotation par exemple. La différenciation des mouvements issus de l'environnement de ceux issus de l'objet est alors plus délicate. De plus, la densité des points n'étant pas homogène, une partie de l'objet ayant un taux de points plus élevé, biaisera l'estimation du mouvement de l'objet vers le mouvement de cette partie.

Nous disposons cependant d'une autre information permettant de replacer la boîte englobante : le label des points. Rappelons qu'à l'issue de notre fonction de labellisation (voir **4.2.1.4**), à chaque point apparié est affecté une valeur réelle représentant la probabilité que ce point fasse partie de l'objet suivi ou de l'arrière-plan. L'échelle de valeurs, comprise entre 0 et 1 modélise la confiance dans l'appartenance du point à l'objet (1 signifiant la certitude que le point fasse partie de l'objet). Si l'algorithme de mouvement permet d'expliquer le mouvement global de l'objet, le placement de la boîte englobante peut être peaufiné en fonction du label des points environnant le cadre. Cela nous permet d'adapter la forme et la position de la boîte englobante en fonction des déformations de l'objet.

L'algorithme se base sur l'affectation d'un label aux zones périphériques (intérieure et extérieure) de la boîte englobante. Le label d'une zone, que nous appelons « qualité », est calculé comme la moyenne des labels des points inscrits dans cette zone. Le principe de l'algorithme consiste à évaluer la qualité de zones limitrophes au cadre de la boîte englobante, pour différentes tailles. Si une zone extérieure (respectivement intérieure) est jugée être une zone de l'objet (respectivement du décor), la boîte englobante est bougée en conséquence. Cette modification étant souvent la résultante d'un déplacement local de l'objet, en particulier des bords de l'objet, elle n'est pas prise en compte dans le calcul du mouvement global de l'objet.

Deux facteurs influent sur cette mesure de qualité : Tout d'abord le nombre de points pris en compte qui influence directement la fiabilité de la mesure. Ensuite, le fait que cette mesure sera biaisée par notre algorithme de labellisation. En effet, cet algorithme considère que la zone où le label des points est incertain est très étendue dans la boîte englobante et très peu à l'extérieure de celle-ci. Il y aura donc plus de points « décor » dans la boîte englobante que de points « objet » à l'extérieur. L'optimisation de la boîte englobante en fonction des labels aura donc plus tendance à la rétrécir.

Tenant compte de ces comportements, nous avons développé un algorithme d'optimisation de la boîte englobante. Pour chacun des quatre cotés de la boîte englobante, On teste la possibilité d'un rétrécissement (ou d'une dilatation) de deux ou quatre pixels. Si la qualité de la zone est supérieure à un certain seuil *Seuilqualité*, le changement est retenu comme candidat. Le candidat de meilleure qualité est ensuite appliqué. Les algorithmes de test de réduction et de dilatation sont les suivants :

Réduction de la boîte englobante (seuil initial Seuilqualité = 2×S)
Ajout = 0.5-qualité;
Si (Ajout > Seuilqualité/nb)

Enregistrer la modification;
Seuilqualité = ajout×nb;

Dilatation de la boîte englobante (seuil initial Seuilqualité = S)
Ajout = qualité-0.5 ;
Si (Ajout > Seuilqualité/nb)
Enregistrer la modification;
Seuilqualité = ajout×nb;

A l'issue de cette étape, nous disposons donc d'une information de réduction ou de dilatation pour chacun des quatre bords de la boîte englobante. Chaque axe est traité indépendamment. Afin de limiter les déformations abusives, nous appliquons les règles suivantes pour le replacement de ces bords :

1- si les déformations *d1* et *d2* des bord sont dans des directions opposées alors la boîte englobante est rétrécie ou agrandie d'un nombre de pixels *M* en accord avec la plus petite magnitude des deux déformations $M = min(abs(d1), abs(d2))$.

2- sinon (les deux déformations *d1* et *d2* sont dans la même direction). la boîte englobante est alors recentrée selon les déformations détectées par application d'un mouvement de $M = (d1+d2)/2$ à chacun des deux bords.

Ce procédé peut donc effectuer un changement d'échelle **ou** un recentrage de la boîte englobante. Une variante envisageable serait d'effectuer les deux modifications l'une après l'autre : Tout d'abord, changement d'échelle puis recentrage. Toutefois, nous préférons ne pas le faire afin de parer à d'éventuelles imprécisions. Le changement entre deux images étant minime, si un recentrage est réellement nécessaire, il sera redétecté à l'image suivante.
La Figure 65 illustre ce procédé :

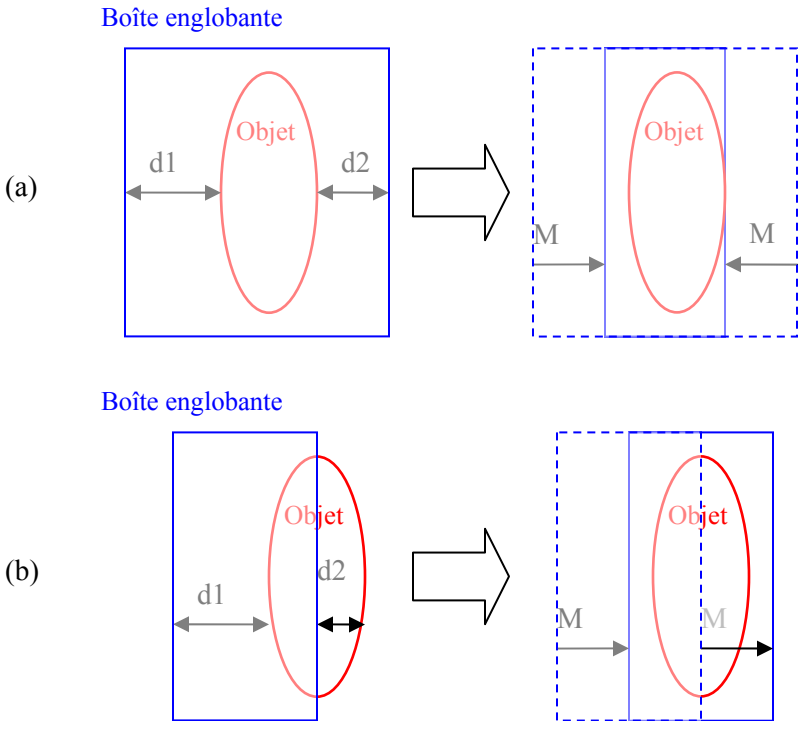


Figure 65: Exemple d'optimisation de la position de la boîte englobante se basant sur la labellisation. (a) Réduction (b) Recentrage.

L'efficacité de l'algorithme repose entièrement sur l'ajustement du seuil *Seuilqualité*. Il est initialisé à une valeur deux fois plus importante pour la réduction que pour la dilatation, afin d'éviter de favoriser le premier par rapport au deuxième. De plus, Ce seuil s'adapte en fonction du nombre points entrant en jeu. Plus le nombre de points sera élevé, plus la mesure sera considérée fiable, plus le seuil sera faible. Pour chaque modification retenue, le seuil est mis à jour afin de ne plus considérer que des modifications de meilleure qualité.

Cet algorithme est aussi adaptable dans le cas d'un environnement non encombré (peu ou aucun des points-clés extraits appartiennent au décor). On se base alors sur le nombre de points présents dans la zone analysée sans tenir compte de leurs labels. Si une zone intérieure à la boîte englobante n'a aucun point-clé, on enregistre une réduction du bord en conséquence. De même, si une zone extérieure à la boîte englobante a plus d'un point, on enregistre la dilatation du bord correspondant. On accorde la priorité à la dilatation sur la réduction, puis au candidat de plus grande amplitude.

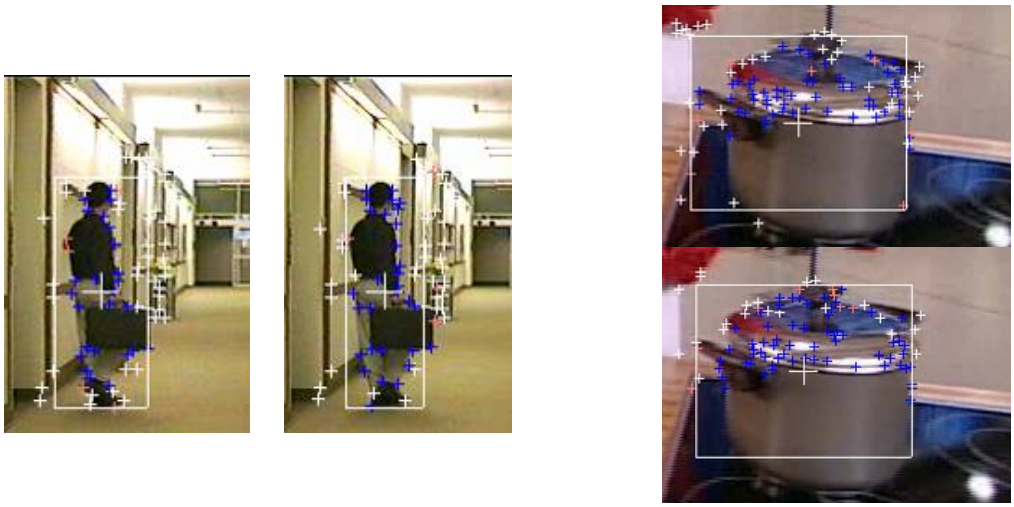
En appliquant les règles ci-dessous similaires à celles énoncées pour un environnement encombré, il est possible d'obtenir un algorithme d'optimisation de la position de la boîte englobante fiable pour un environnement non encombré.

$$Réduction\ M = min(abs(d1)-p, abs(d2)-p)$$
$$Dilatation\ M = min(abs(d1)/d, abs(d2)/d)$$
$$Recentrage\ M = (d1+d2)/2*d$$

Le nombre de pixels p et le facteur d ont pour but d'accroître la fiabilité des transformations. Ces variables vont minimiser les modifications. Donc seules les modifications détectées sur plusieurs images, donc fiables, seront pleinement effectuées. Nous utilisons $p=4$ et $d=2$. Un avantage de cette variante est l'absence de paramètres à ajuster.

Toutefois, un algorithme générique peut avoir à faire face à des environnements encombrés aussi bien qu'uniformes. Il est même possible que l'environnement change au cours de la vidéo (un personnage passant devant un arbre par exemple). Notre algorithme doit donc pouvoir s'adapter en fonction du décor. Pour cela nous utilisons la mesure d'encombrement détaillée en **4.2.1.4**. Le seuil du choix d'un algorithme plutôt que l'autre est arbitrairement fixé à 5% d'encombrement. Cette utilisation simultanée des deux variantes de l'algorithme conduit toutefois à des défaillances lorsque le taux d'encombrement oscille aux environs du seuil. En effet le comportement des deux variantes sera différent pour une même configuration, les modifications étant beaucoup plus contraintes dans le cas d'optimisation basée sur les labels. Afin d'harmoniser les comportements, nous restreignons les transformations issues de l'algorithme opérant dans un environnement non encombré au recentrage de la boîte englobante.

La Figure 66 montre des exemples de replacement et redimensionnement de la boîte englobante dans un environnement encombré et les Figure 67 illustrent son recentrage dans une scène non encombrée.



(a) (b)

Figure 66: Examples dans un environement encombré (a) Redimensionnement de la boîte englobante. Images 7 et 8 de la séquence « Surveillance ». (b) Déplacement de la boîte englobante de 4 pixels vers la droite. Images 35 et 37 de la séquence « Cooking ». En bleu les points-clés labélisés « objet », en blanc les point-clés « décor » et en rouge les points-clés au label indéterminé.

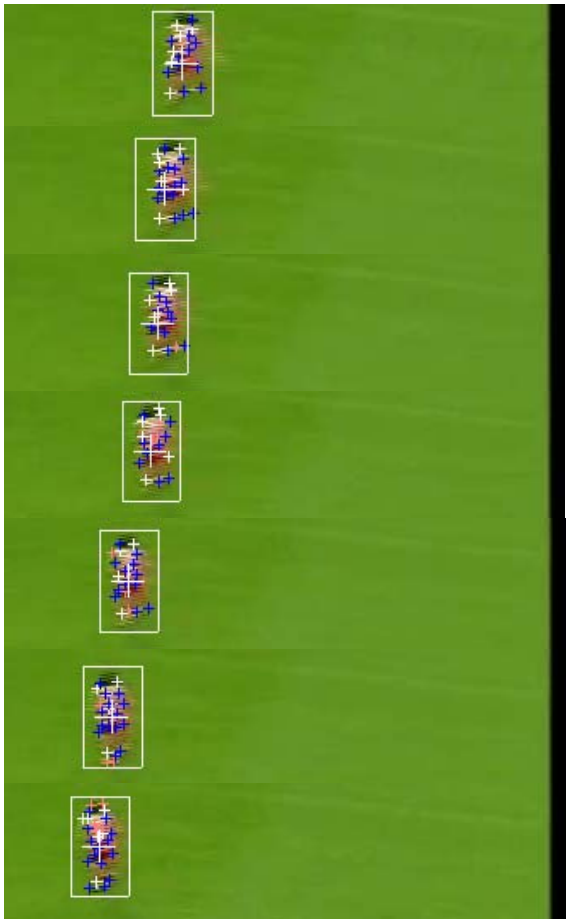


Figure 67: Suivi d'un joueur de football grâce au recentrage de la boîte englobante. Images 20 à 30 de la séquence "football from above". En bleu les points-clés labélisés « objet », en blanc les point-clés « décor » et en rouge les points-clés au label indéterminé.

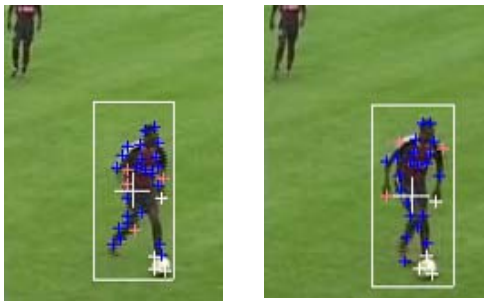


Figure 68: Recentrage de la boîte englobante. Images 18 et 21 de la sequence "soccer". En bleu les points-clés labélisés « objet », en blanc les point-clés « décor » et en rouge les points-clés au label indéterminé.

Nous avons testé cet algorithme pour deux types de points-clés (les points de Harris et de Harris-Laplace détaillés en **4.1.1.1.2**) dont les densités sont différentes, ainsi que pour différentes valeurs du paramètre *Seuilqualité,* dans le cas des environnements encombrés. Les résultats sont présentés dans les Figure 69.

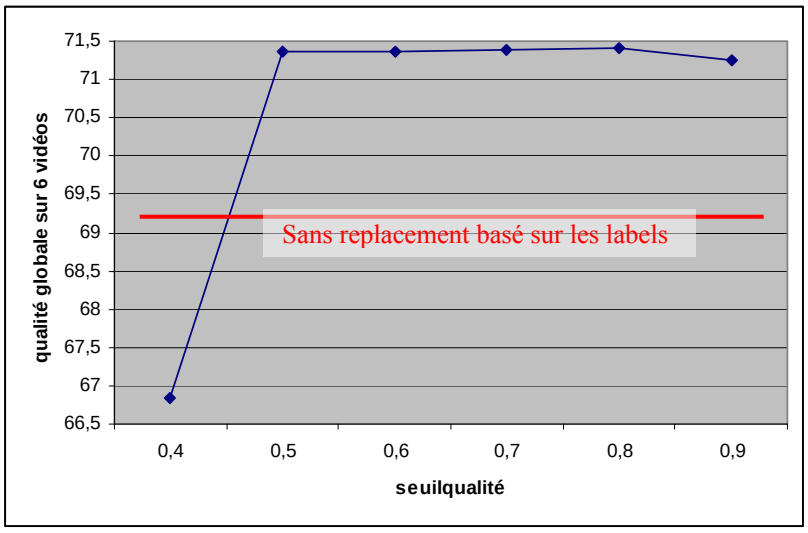


Figure 69: Résultats pour l'algorithme d'optimisation de la boîte englobante en fonction des labels avec les points de Harris. Tests sur 6 vidéos avec un arrière plan encombré : cooking, frying pan, bottle, surveillance, skijump, coast guard.

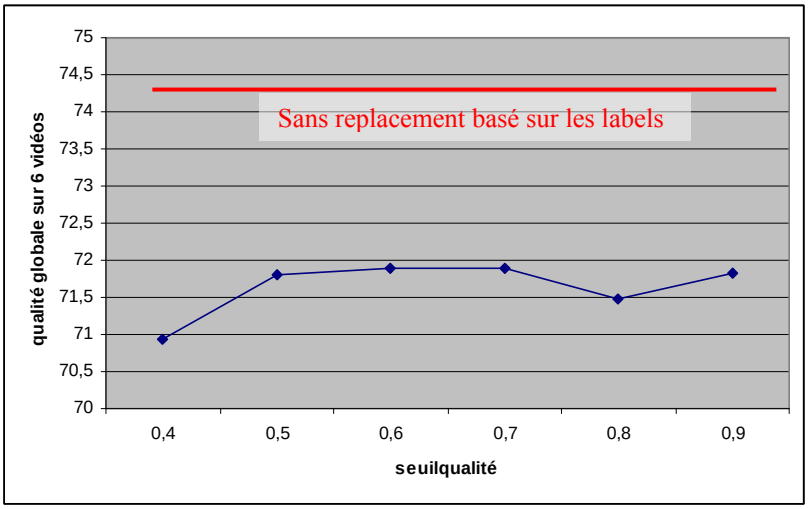


Figure 70: Résultats pour l'algorithme d'optimisation de la boîte englobante en fonction des labels avec les points de Harris-Laplace. Tests sur 6 vidéos avec un arrière plan encombré : cooking, frying pan, bottle, surveillance, skijump, coast guard.

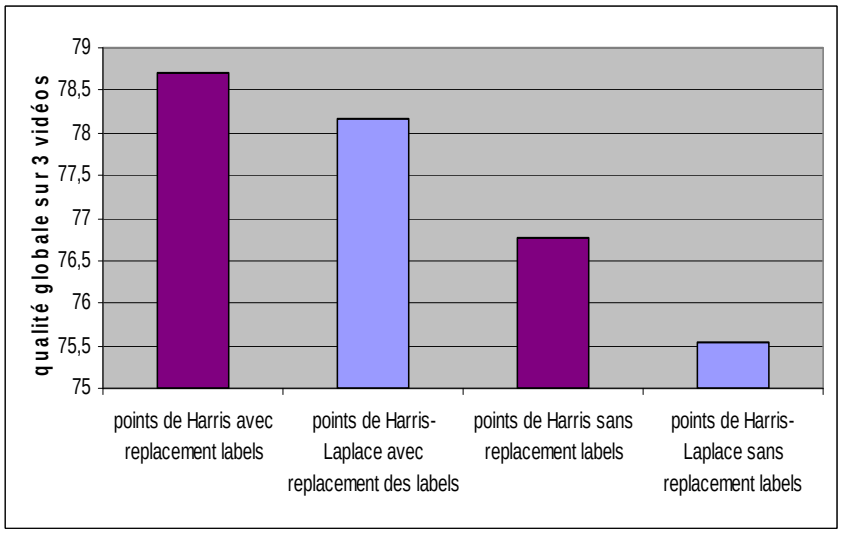


Figure 71: Résultats pour l'algorithme d'optimisation de la boîte englobante en fonction des labels avec les points de Harris et Harris-Laplace. Tests sur 3 vidéos avec un arrière plan non encombré: fashion, soccer, et jellyfish.

Si les résultats avec les points de Harris sont encourageants, ceux-ci sont décevants dans le cas des points de Harris-Laplace. Cela est dû à une trop faible densité de points pour que l'algorithme s'avère efficace. En revanche, dans le cas d'un environnement non encombré l'algorithme améliore les performances du système de suivi quelque soient les points utilisés. On constate également que le seul paramètre important *Seuilqualité*, n'a pas, en moyenne, une forte influence sur les résultats. Cela démontre que la méthode n'est que faiblement contrainte au réglage du paramètre. Cette variable a néanmoins son importance. Elle représente le compromis entre la quantité de modifications effectuées et leur sécurité. Pour une valeur élevée, seules des modifications fiables seront autorisées. A l'inverse, un faible seuil de qualité induira de nombreux replacements de la boîte englobante, mais au risque d'erreurs.

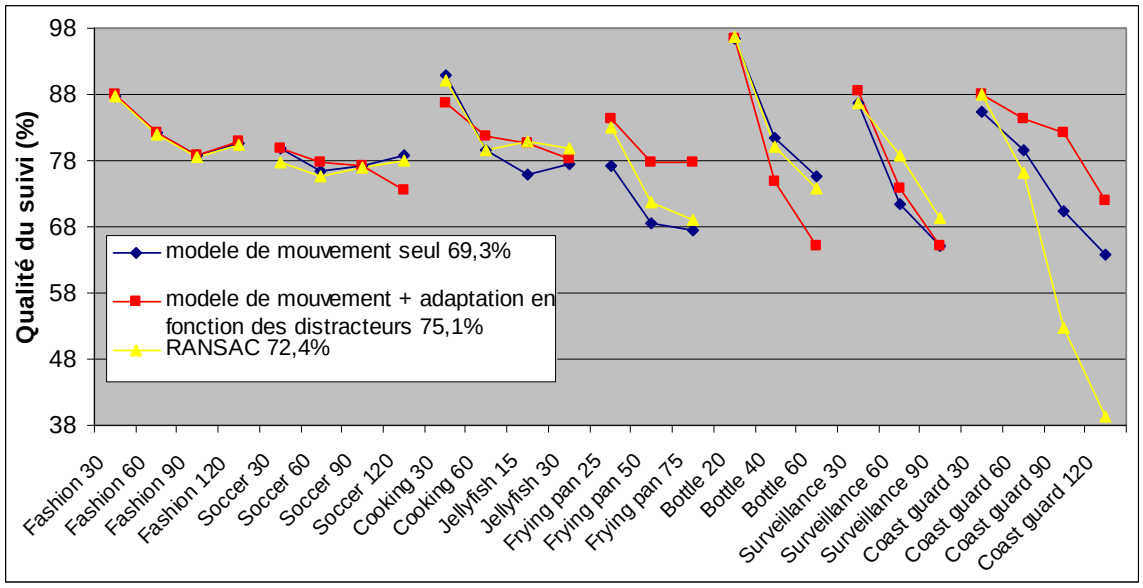
Le système d'optimisation de la position de la boîte englobante présenté ici est donc une réussite mais nécessite une densité de points suffisante pour pouvoir fonctionner efficacement. Il est également évident que son efficacité repose entièrement sur une estimation efficace du label des points.

5.2.2.4 Adaptation du mouvement en fonction du taux de *distracteurs*

Le taux de *distracteurs tauxDist* ($0 \leq tauxDist \leq 1$) défini en **4.2.1.4** à tendance à minimiser l'estimation du mouvement dans la mesure où les vecteurs mouvements qui leur sont associés sont nuls ou contraire au mouvement global de l'objet. Nous avons contrebalancé cet effet en **5.2.2.1**, en multipliant la magnitude du mouvement par un facteur *d* constant représentant l'influence des *distracteurs*. Toutefois le taux de *distracteurs* varie d'une vidéo à l'autre. Il convient donc, puisque l'on est capable d'estimer leur importance d'adapter notre estimation du mouvement en conséquence. Nous avons donc expérimentalement établi :

$$d = (tauxDist/3)+1 \qquad avec\ 1 \leq d \leq 1.2$$

Un autre algorithme connu donnant de bons résultats mais nécessitant une connaissance à priori du taux de *distracteurs* pour être efficace est l'algorithme RANSAC **[Fis81]**, décrit en **5.1.2.5**. Nous avons expérimentalement fixé le nombre de points indésirable à *tauxDist/2* et avons restreint le nombre d'itération à 10, un nombre plus élevé ne donnant pas de meilleurs résultats. Nous avons donc comparé les résultats de notre modèle de mouvement adaptant son estimation en fonction du taux de *distracteurs* avec RANSAC. Les résultats sont présentés dans la Figure 72.



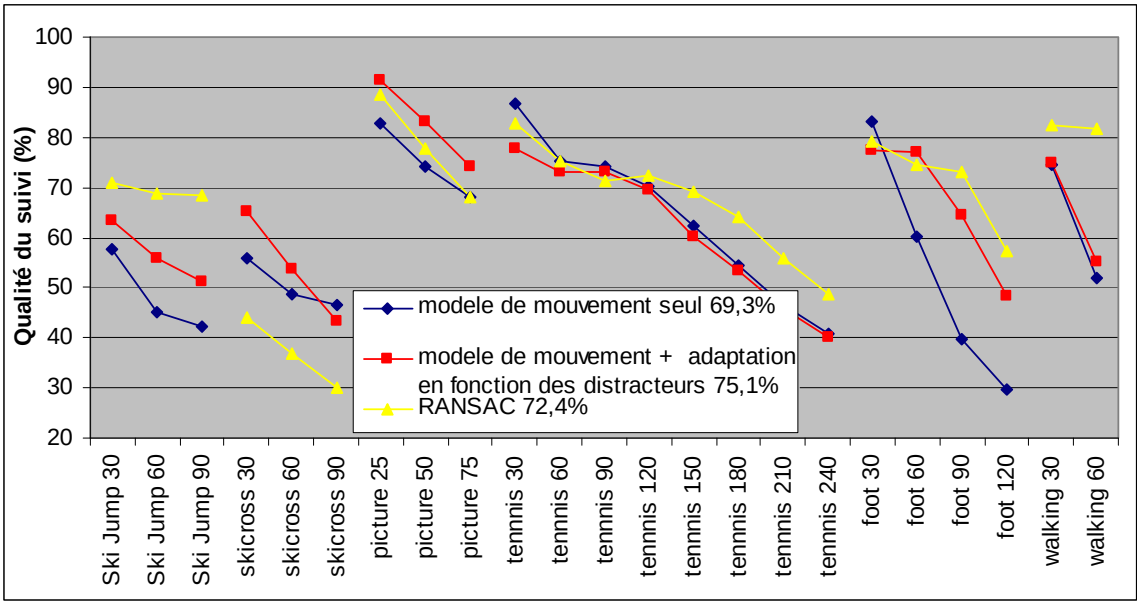


Figure 72: Performances du système de suivi comparées pour un appariement avec l'algorithme RANSAC et un appariement utilisant notre modèle de mouvement s'adaptant en fonction du taux de *distracteurs*. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des 13 tests est donnée dans l'intitulé des courbes.

Il est indéniable que l'intégration de l'estimation du taux de *distracteurs* accroît la précision des algorithmes. Toutefois, départager RANSAC de notre algorithme d'appariement s'adaptant en fonction du taux de *distracteurs* n'est pas aussi aisé. Si notre algorithme donne en moyenne de meilleurs résultats que RANSAC, les deux algorithmes ont presque toujours des performances différentes quelque soit le jeu d'essai testé. Cependant, et contrairement a RANSAC, le suivi avec notre système d'appariement n'est jamais un échec complet. Ce point, crucial dans le cadre d'un suivi générique fait pencher la balance en faveur de notre système d'appariement.

6 Performances de notre approche

Ce chapitre s'emploi à évaluer les performances de notre approche en termes de précision du suivi et de temps de calcul sur notre corpus de 17 séquences vidéo. Cet ensemble recouvre un vaste panel de difficultés et d'applications possibles afin de tester de manière appropriée un système de suivi générique. En conséquence, le paramétrage de notre algorithme est donc un paramétrage générique supposé offrir des résultats satisfaisant quelque soit la séquence testée. La preuve en est que ce paramétrage n'a été effectué que sur 13 des 17 séquences vidéo, sans que la qualité du suivi sur les autres séquences en soit affectée pour autant. Les vidéos n'ayant pas servies au paramétrage du système de suivi sont « domatica », « traffic in Bombay », « Cognac », et « Picture ».

6.1 Algorithme hybride utilisant des points-clés et un descripteur global

Malgré toutes les améliorations présentées, notre système de suivi souffre de limitations inhérentes aux points-clés. La mise à jour des points-clés à chaque image, indispensable pour la robustesse du processus crée un modèle de l'objet extrêmement changeant. De plus, ce modèle est constitué d'un grand nombre de descripteurs locaux (issus du voisinage des points-clés), sans relations les uns avec les autres, n'offrant pas une description efficace de l'objet dans son ensemble. Ce modèle est donc, d'une part, très mal adapté à la détection d'occultations et, d'autre part, favorise une déviation de l'objet par rapport à sa position optimale. Afin de permettre à notre système de suivi de s'affranchir de ces problèmes, nous nous sommes proposés de l'enrichir d'un descripteur global pour créer un système de suivi hybride se basant à la fois sur une information locale issue des points-clés et sur un modèle global de l'objet résumé par ce descripteur d'ensemble.

La première question soulevée par la création de cet algorithme hybride est celle de la fusion des données. Les deux suivis doivent-ils être combinés en série ou en parallèle ? Quelle importance relative doit-elle leur être accordée ? Dans la mesure où les défaillances du suivi basé sur des points-clés sont minimes et ne se traduisent uniquement que par une légère déviation du suivi, il est logique que le suivi basé sur un descripteur global ne prenne place qu'une fois celui basé sur les points-clés effectué et qu'il cherche à en peaufiner les résultats. Cette approche se traduit par une recherche d'une éventuelle meilleure position de l'objet dans un proche voisinage de la position indiquée par le premier algorithme de suivi, la taille de ce voisinage représentant son erreur possible. Cette erreur sera, bien sûr, fonction de l'amplitude du mouvement de l'objet suivi. Il résulte de cette exploration locale un coût algorithmique réduit au minimum. Afin de ne pas donner une trop grande importance au résultat du suivi basé sur le modèle global de l'objet, nous avons choisi comme position et échelle finale de l'objet la moyenne des deux suivis.

Nous avons choisi une zone de recherche de 4, 6 ou 8 pixels dans chaque direction fonction de l'importance du vecteur mouvement et un pas de recherche de 2 pixels. Le nombre de positions possibles à explorer varie donc, pour chaque image, entre 16 et 64. Pour chacune de ces possibilités, la zone correspondante est comparée avec le descripteur global et la meilleure est conservée.

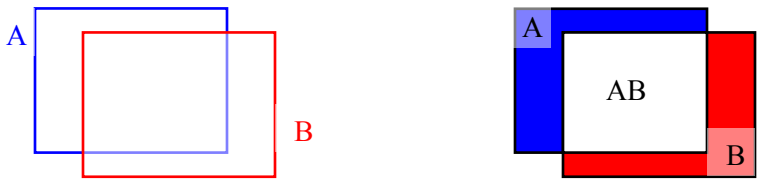
La principale inconnue de ce système de suivi hybride consiste à trouver le descripteur global idoine. En effet, afin de compenser l'extrême variabilité des descripteurs locaux associés aux points-clés, ce descripteur ne devra pas (ou peu) être mis à jour. En conséquence de quoi, il doit représenter les parties les plus discriminantes de l'objet de façon à ce qu'il caractérise l'objet par rapport à son environnement le plus longtemps possible. Trois hypothèses de travail ont été retenues :

1. Bâtir un descripteur qui soit la concaténation des descripteurs associés aux points-clés.

2. Utiliser un histogramme couleur des couleurs discriminantes.
3. Utiliser un masque des couleurs discriminantes.

La première hypothèse était la plus prometteuse. En effet, des travaux **[Bre99]** identifient le voisinage de points-clés comme des zones susceptibles de résumer visuellement un objet. On peut donc supposer que leur concaténation puisse former un descripteur global satisfaisant. De plus, dans le cadre d'une fusion avec un autre algorithme s'appuyant sur des points-clés, cette information à l'avantage d'être déjà disponible. Toutefois, lors de nos expérimentations, la surface recouverte s'est révélée trop restreinte et disparate pour former un masque convenable, et les valeurs des pixels concernés insuffisamment discriminante dans leur ensemble pour identifier l'objet. Nos tests se sont cependant limités aux points de Harris seuls, utilisés par notre système de suivi. Malgré les mauvais résultats obtenus, l'expérimentation de cette idée sur d'autres points, plus spécifiques au résumé d'image **[Bre99, Kad01, Itt99]**, reste une piste viable.

Pour la deuxième hypothèse, nous avons choisi un histogramme RGB de 256 colonnes construit par ajout de l'histogramme de l'objet (rappelons que l'objet est défini par une boîte englobante) et soustraction de l'histogramme représentant l'environnement extrait de la zone entourant l'objet. Afin de ne conserver que les valeurs discriminantes, toute colonne de l'histogramme représentant moins de 5% des pixels de l'objet est éliminée. Pour chaque possibilité, un histogramme est établi, puis comparé au modèle global de l'objet à l'aide de la distance de Battacharrya **[Bha43]**. Afin de gagner du temps de calcul, tous les histogrammes après le premier sont bâtis en fonction du précédent. En effet, puisque seuls deux pixels séparent la position des fenêtres d'où sont issus les deux histogrammes, la majorité des pixels les composant sont identiques. Comme le montre la Figure 73, créer un histogramme peut donc se faire à partir de l'histogramme précédent. Seuls quelques pixels doivent être ajoutés et ôtés. Ce procédé permet de gagner un temps de calcul considérable lors de la création des histogrammes.



(a) (b)

Figure 73: Calculs successifs d'histogramme. (a) Histogramme a calculer à partir des pixels de la fenêtre B, l'histogramme créé à partir de la fenêtre A étant disponible (b) AB : pixels à conserver. B : pixels à ajouter à l'histogramme. A : pixels à ôter.

Le masque de notre troisième hypothèse est, de la même manière, créé à partir des plages de couleur discriminantes uniquement. Dans cette dernière hypothèse sont comparés des patrons de l'objet. A chaque pixel du modèle est associé le numéro de la plage de couleur (i.e la colonne de l'histogramme) à laquelle il appartient. La comparaison de deux patrons s'effectue en termes de pourcentage de pixels dont la plage de couleur est la même. Il n'est ici pas nécessaire de créer d'histogramme pour comparer deux patrons.

Les résultats comparatifs de ces deux méthodes avec un suivi basé sur les points-clés seuls sont présentés dans la Figure 74. Dans cette série d'expérience, les éventuelles rectifications du suivi apportées par la comparaison au modèle global de l'objet ne sont pas intégrées au modèle de mouvement. De plus, le modèle global utilisé ici s'adapte à l'échelle détectée par l'algorithme précédent.

Ces résultats montrent que malgré le caractère déformable de la plupart des objets suivis, le masque des couleurs discriminantes de l'objet se comporte, dans l'ensemble mieux que l'histogramme RGB des mêmes couleurs discriminantes de l'objet. Nous avons donc opté pour ce descripteur global.

De nombreuses variations de cet algorithme sont envisageables. Il est tout d'abord possible de rectifier le vecteur mouvement de l'objet en fonction des résultats du modèle global de l'objet. Dans le

cadre d'importantes déformations de l'objet, le modèle de mouvement échoue généralement à détecter correctement les changements d'échelles. Afin de compenser ces erreurs, il est donc également envisageable d'interdire au modèle global la gestion des changements d'échelles. Enfin, l'étude d'une éventuelle mise à jour du modèle a attiré notre attention. Ces différentes variations ont été étudiées successivement sur les mêmes jeux d'essai que précédemment. La Figure 75 en détaille les résultats comparés à l'algorithme de base présenté précédemment.

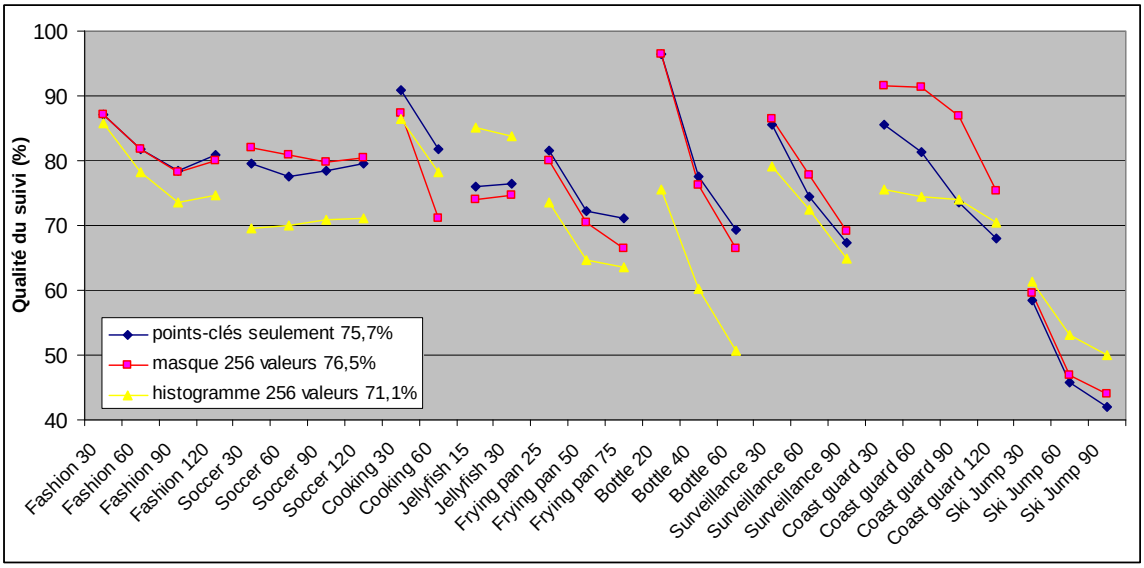


Figure 74: Performances du système de suivi avec les points-clés seuls comparé avec deux suivis hybrides utilisant les points-clés en association avec, respectivement, un histogramme et un masque des couleurs discriminantes de l'objet. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des 9 tests est donnée dans l'intitulé des courbes.

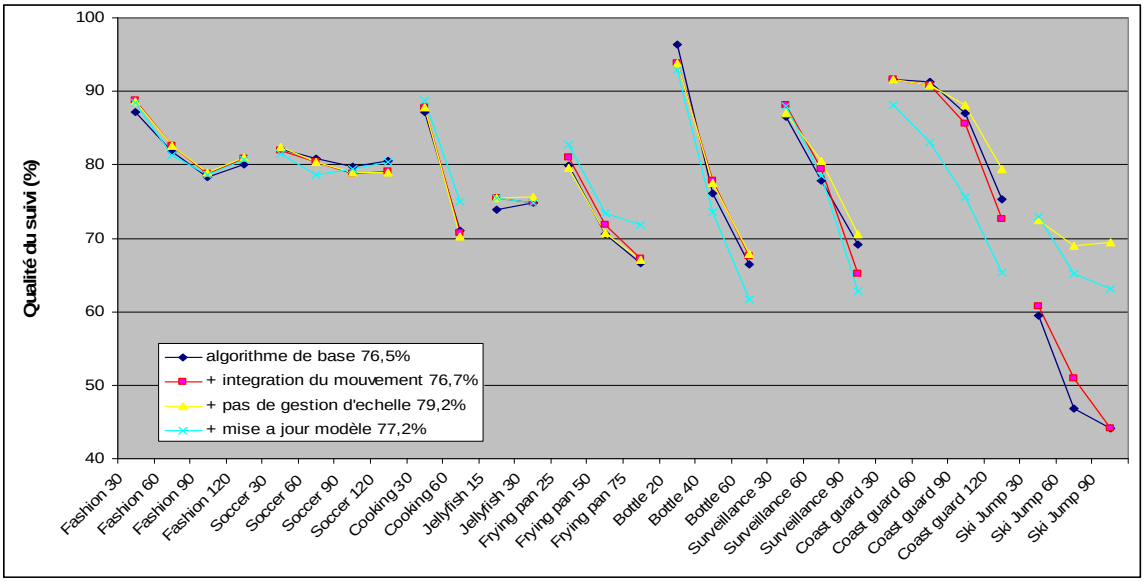


Figure 75: Performances de différentes variantes du système de suivi hybride basé sur des points-clés et le masque des régions discriminantes. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des 9 tests est donnée dans l'intitulé des courbes.

A part pour la vidéo « Cooking », l'algorithme hybride ne gérant pas les changements d'échelle et intégrant les rectifications du modèle global de l'objet dans le modèle de mouvement donne des résultats visuellement très satisfaisant pour tous les jeux d'essais. Ces résultats comparés à l'algorithme exploitant uniquement notre modèle de points-clés sont montrés en Figure 76.

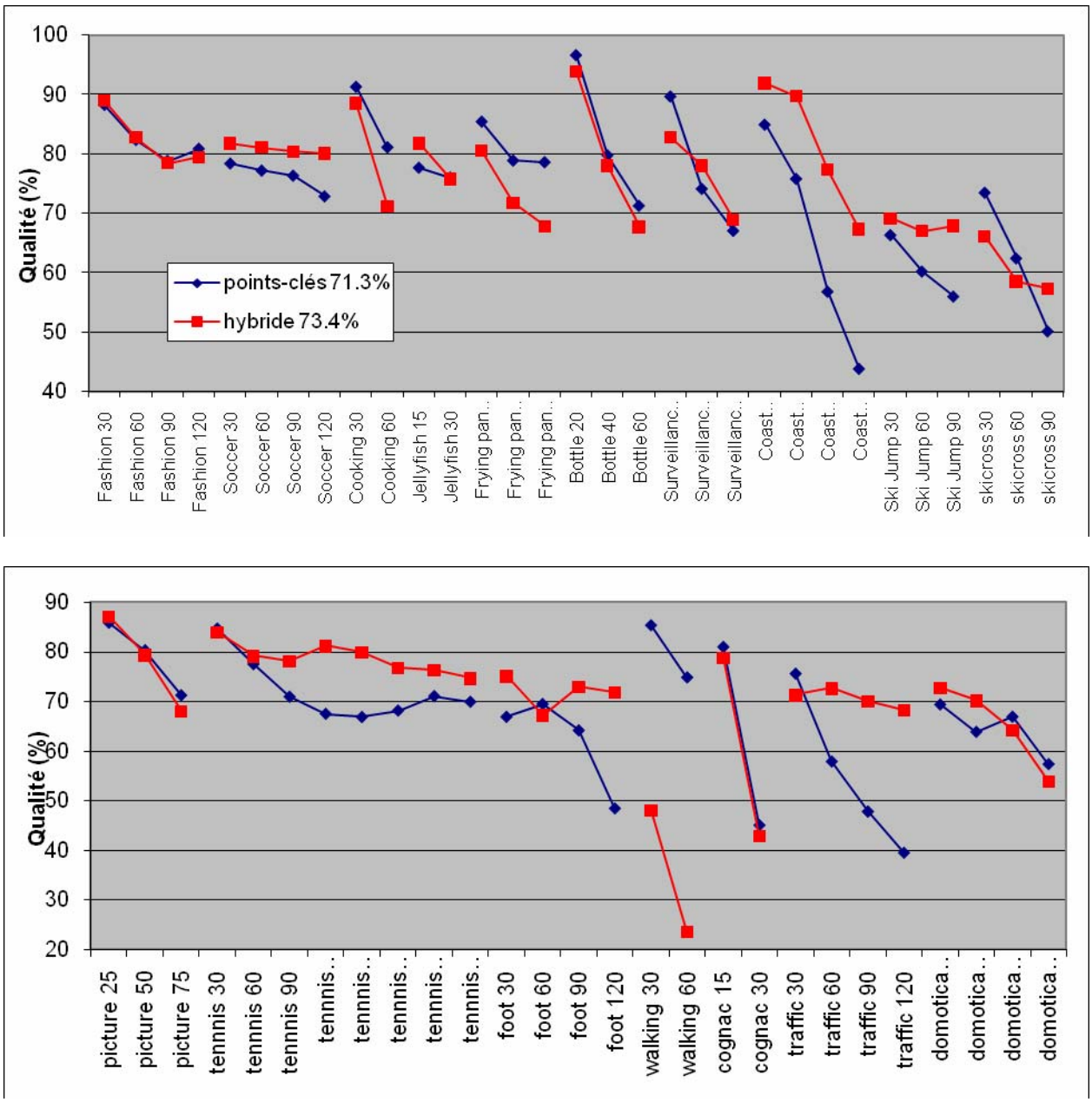


Figure 76: Performances comparées du suivi hybride avec un suivi uniquement basé sur notre modèle de points-clés. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des 17 tests est donnée dans l'intitulé des courbes.

6.2 Performances comparées du système de suivi et d'algorithmes de références

Cette section est consacrée à la comparaison de notre algorithme final se basant uniquement sur le modèle de points-clés ainsi que l'algorithme hybride utilisant également un masque de couleurs discriminantes avec le Basic Meanshift **[Com02]** et l'algorithme de Gabriel **[Gab05]** (voir **2.4.3**). Les résultats présentés en Figure 77 montrent une prédominance de nos méthodes pour toutes les séquences testées exceptées « Coast guard », « skicross » et « traffic in bombay» où le Meanshift prévaut. Il s'agit ici de cas de figures où le Meanshift est réputé donner d'excellent résultats : des objets petits et dont la couleur dominante se détache bien du décor. Il faut aussi reconnaître que la métrique utilisée (voir **2.4.1**) dévalorise un peu (d'environ 2 à 5% suivant la séquence testée) le Meanshift à cause des effets de tremblements.

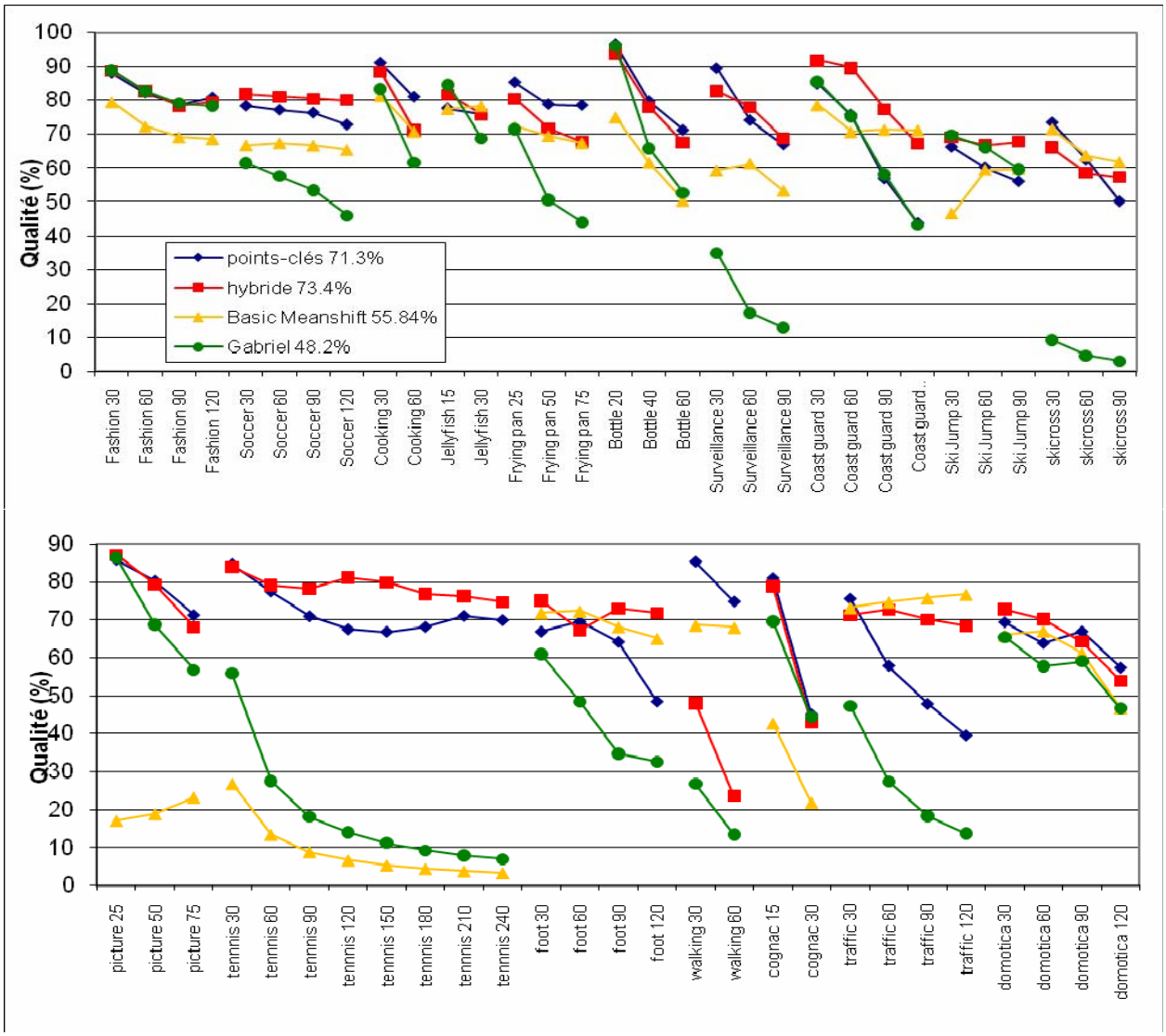


Figure 77: Performances comparées du suivi à base de points-clés, du suivi hybride, du basic Meanshift **[Com02]** et de l'algorithme de Gabriel **[Gab05]**. Pour chaque séquence vidéo, les résultats donnés mesurent la qualité moyenne du suivi sur le nombre d'images indiquées. La qualité moyenne sur l'ensemble des tests est donnée dans l'intitulé des courbes.

6.3 Optimisation du temps d'exécution

6.3.1 Optimisation algorithmique

Au même titre que la qualité des résultats le temps d'exécution est une attente primordiale dans la majorité des systèmes de suivi qui doivent s'exécuter en temps-réel. Si, dans un premier temps, nos efforts se sont orientés vers l'amélioration des performances, nous avons, dans un second temps, effectué des expériences en vue d'accélérer le processus sans susciter de répercutions notables sur les résultats.

Il est important de rappeler ici que ce travail est dédié à l'annotation vidéo. Dans ce contexte de travail particulier, la majorité du traitement, à savoir l'extraction des points-clés et le calcul des descripteurs associés, est effectué lors d'une étape de prétraitement soulageant ainsi le suivi de la majorité des calculs. Nous obtenons ainsi un suivi dont l'exécution temps-réel ne se fait pas au prix d'une dégradation des résultats. Toutefois notre système de suivi a la particularité d'être générique : il peut gérer tout type de vidéos et de difficultés. Il est donc transposable à tout type d'application. Mais le prétraitement ne sera alors plus réalisable. Etudier l'accélération éventuelle de notre suivi dans ce cas de figure tombe alors sous le sens.

Trois expériences ont été tentées dans ce sens. Les programmes utilisés sont en C++ optimisé. Les temps d'exécution calculés comprennent le temps d'extraction des points-clés, de calcul des descripteurs, et le temps de calcul du suivi. Le temps de lecture de la vidéo n'est pas pris en compte. La quantité de calculs effectués sera proportionnelle aux nombres de points-clés extraits à chaque image. Les deux facteurs déteminant le nombre de points seront la taille de la boîte englobante de l'objet ainsi que la quantité d'information présente dans celle-ci. Le temps de calcul sera bien évidemment dépendant à la taille de l'objet, un objet recouvrant un plus grand nombre de pixels impliquant davantage de traitement possible. Les points seront finalement extraits sur les coins dont la quantité est liée à l'encombrement de l'arrière-plan et aux variations locales de la l'objet. L'ordinateur utilisé est un pentium 4, 2.0Ghz avec 768Mb de RAM. Les deux premières expériences visent à éviter des opérations inutiles alors que la dernière applique les points de Harris rapide présentés en **4.2.1.7**.

Notre première tentative se base sur le principe qu'une quantité excessive de points-clés n'accroît en rien la qualité du suivi. En d'autres termes, n points-clés sont suffisants pour estimer le mouvement de l'objet. Passé ce nombre, l'ajout d'information sera superflu. Il est important ici de remarquer que c'est le nombre et non la densité de points-clés qui influe sur la précision du suivi. La taille de l'objet ne sera donc en rien déterminante. Nous avons donc limité le nombre de points ajoutés au modèle lors de sa mise à jour à n. Le modèle de l'objet est initialisé avec tous les points-clés détectés à la première image. Par la suite, si, après élimination des points-clés n'apparaissant plus depuis k images (voir **4.2.1.3**), le modèle comporte plus de n points, aucun nouveau point n'est ajouté. Dans le cas contraire, des points sont ajoutés à concurrence de n. Nous avons essayé de répartir les points ajoutés de façon uniforme ou aléatoire sans aucune incidence sur les résultats. Les tests pour n égal à 100, 80, 60 ou 40 sont présentés dans les Figure 78.

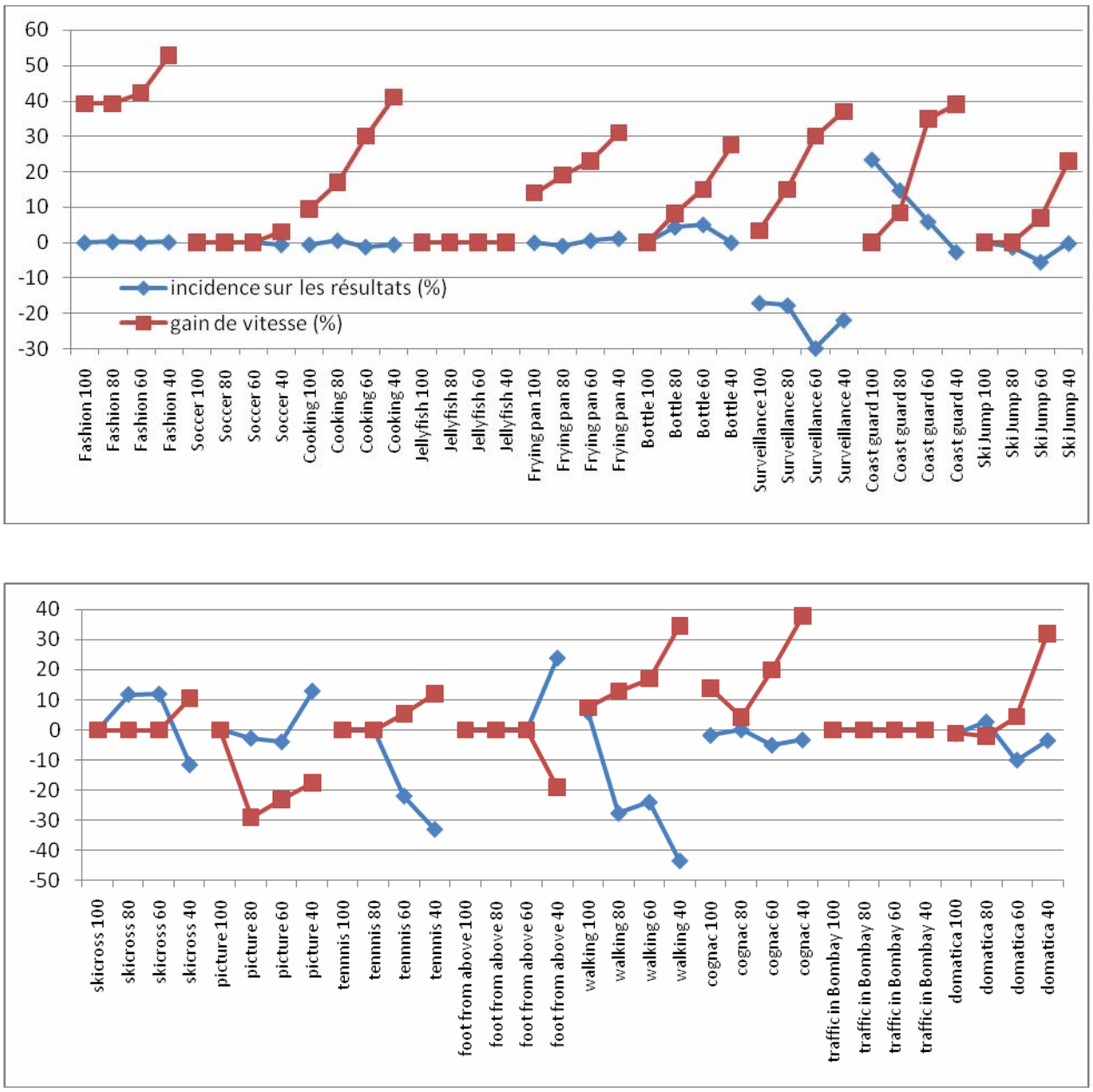


Figure 78: Incidence sur les résultats et gain de vitesse occasionné (en pourcentages) par la limitation du modèle de l'objet à *n* points-clés, pour n égal à 100, 80, 60, ou 40. La valeur en abscisse représente la séquence vidéo testée ainsi que la valeur de *n*.

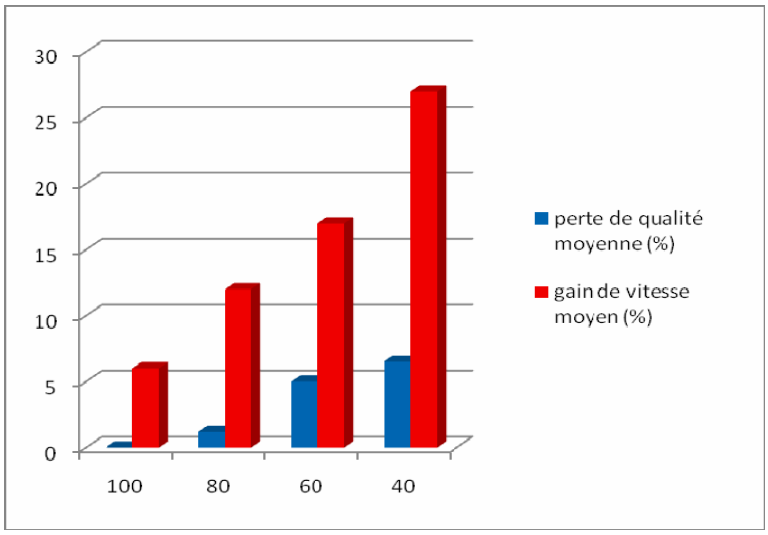
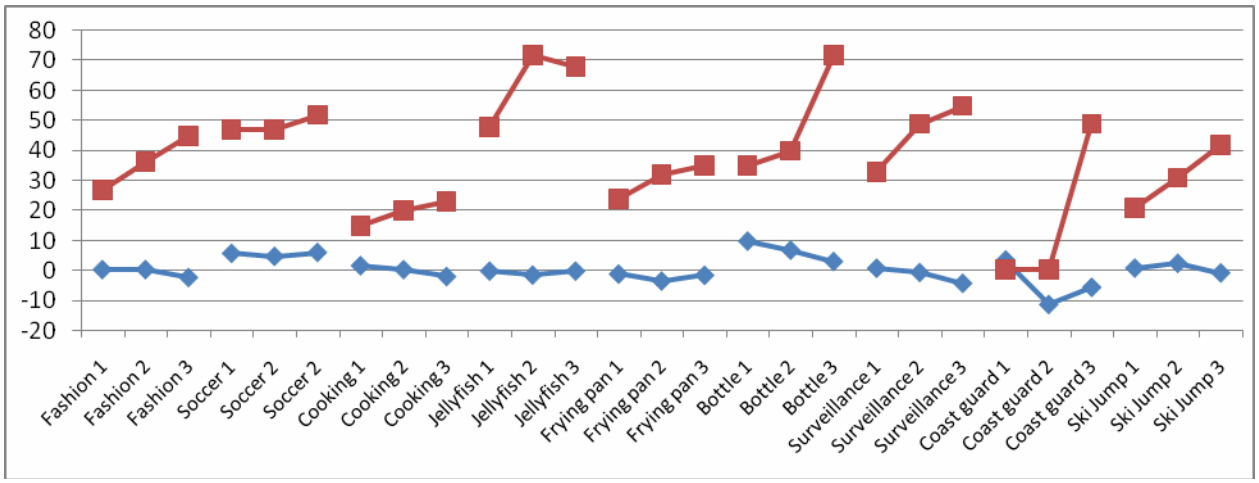


Figure 79: Incidence moyenne sur les résultats et gain de vitesse moyen occasionné par la limitation du modèle de l'objet à n points-clés, pour n égal à 100, 80, 60, ou 40. En abscisse est représentée la valeur de n.

Les résultats montrent un accroissement notable de la vitesse du suivi au prix d'une légère incidence sur les résultats. Un modèle de l'objet se limitant à 100 ou 80 points semble être un bon compromis. De plus, pour ces paramètres, la variabilité des résultats sur les 17 séquences testées est faible.

Notre deuxième expérience repose sur le principe que d'importants calculs ne sont pas nécessaires en permanence, mais uniquement lorsque le passage analysé présente des difficultés. En conséquence notre idée est de restreindre le suivi aux images où le déplacement de l'objet est rapide ou discontinu. Dans le cas contraire l'objet est déplacé en se fiant seulement au vecteur mouvement qui lui est associé pendant *n* images. Dans la pratique, ces images sont détectées d'après le mouvement de l'objet à l'image précédente. Sa magnitude doit être inférieure à *M* et sa variation par rapport à l'image précédente inférieure à *V*. Nous avons estimé *M=2* et *V=1* susceptibles de remplir les critères d'un suivi ne présentant aucunes difficultés. Les Figure 80 illustrent l'efficacité de ce procédé pour *n* égal à 1, 2, ou 3.



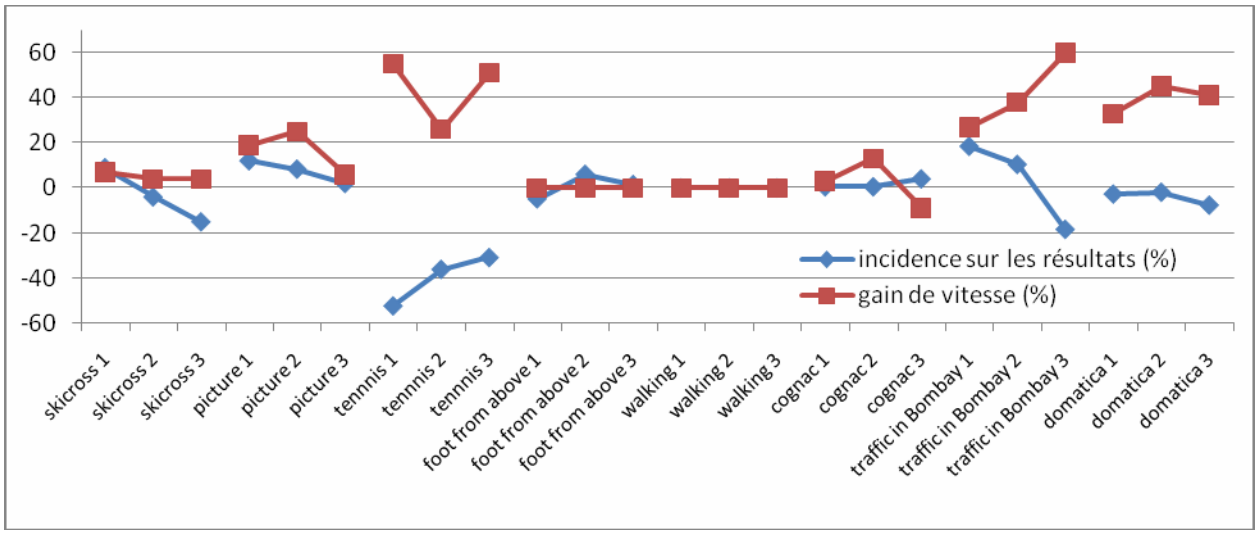


Figure 80: Incidence sur les résultats et gain de vitesse (en pourcentages) occasionné par l'absence de traitement pendant *n* images lorsque le mouvement de l'objet est faible et continu. Tests effectués pour *n* égal à 1, 2, ou 3. La valeur en abscisse représente la séquence vidéo testée ainsi que le nombre *n* d'images non traitées.

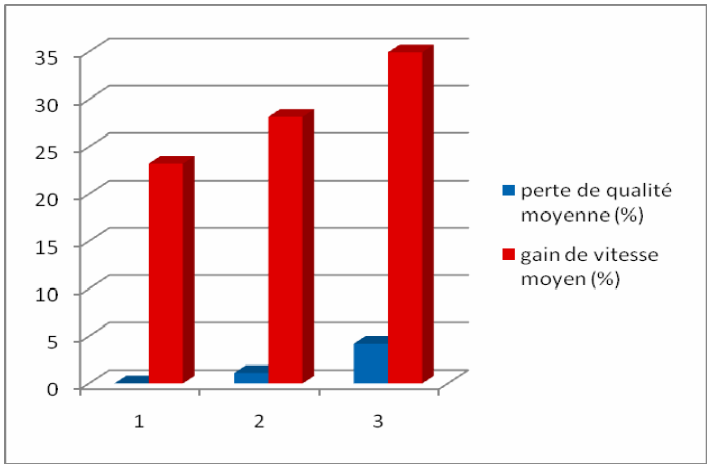


Figure 81: Incidence moyenne sur les résultats et gain de vitesse moyen occasionné par l'absence de traitement pendant *n* images lorsque le mouvement de l'objet est faible et continu. Tests effectués pour *n* égal à 1, 2, ou 3. La valeur en abscisse représente le nombre *n* d'images non traitées.

On constate que cette alternative offre un très bon compromis perte de qualité / gain de vitesse. Et tout particulièrement pour *n=1*. Dans ce cas de figure, tous les résultats sont bons sauf pour la séquence « Tennis ».

Notre troisième expérience applique l'extracteur de points-clés *FastHarris* détaillé en **4.2.1.7**. Les résultats présentés faisant preuve d'un fort écart-type, nous avons estimé son influence sur la qualité du suivi pour plusieurs paramétrages de la variable *S*. La Figure 82 montre les dégâts occasionnés. Les résultats les plus intéressants sont donnés pour le paramétre S = 25 pour lequel seule la séquence « skicross » est dégradée de façon significative.

La Table 3 présente enfin les résultats et les temps de calculs pour les combinaisons prometteuses des améliorations testées ci-dessus. Le temps réel est généralement atteint pour des objets recouvrant environ 300 pixels ou moins.

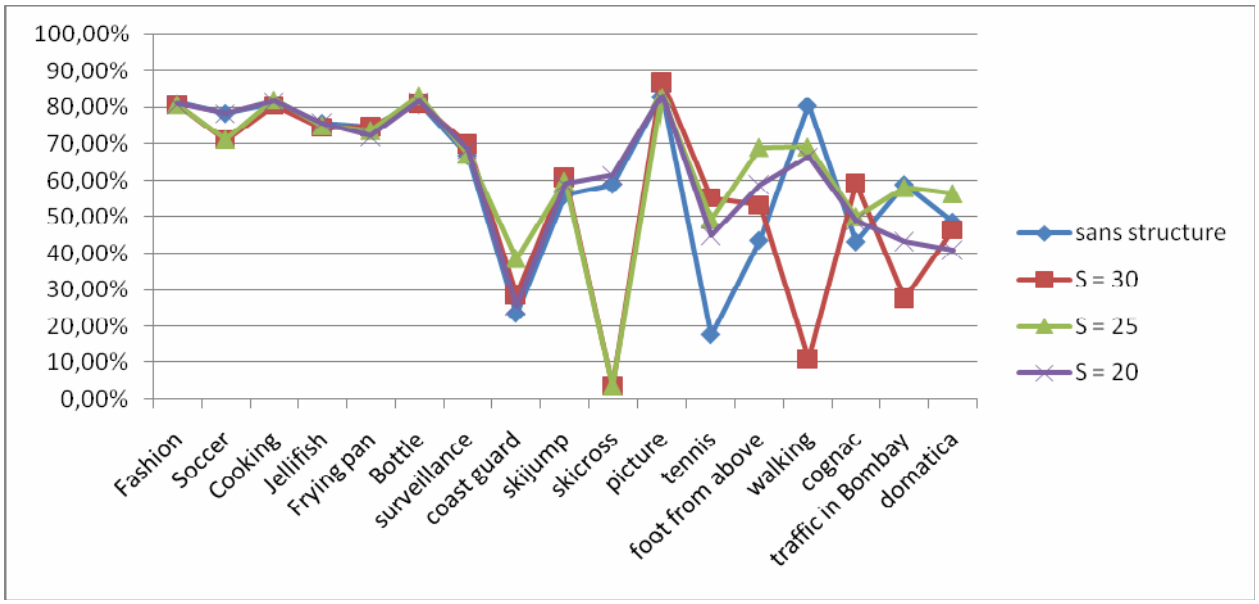


Figure 82: Qualité du suivi comparée (en pourcentages) avec et sans l'utilisation de la structure de FastHarris. La structure est testée pour S=20, 25, et 30. Les 17 séquences testées sont énumérées en abscisse.

Table 3: Qualité du suivi et temps d'exécution par image sur les 14 séquences testées pour les combinaisons prometteuses d'accélération du suivi. En rouge les suivis dont la qualité est fortement dégradée.

Nom de la vidéo		Sans améliorations	Saut 1 image + max 100pts	Saut 1image + max 100pts + FastHarris
Fashion	**qualité**	80,78%	81,20%	80,88%
	temps / im	0,225s	0,118s	0,102s
Soccer	**qualité**	72,78%	78,42%	71,31%
	temps / im	0,034s	0,018s	0,016s
Cooking	**qualité**	81,03%	81,16%	82,13%
	temps / im	0,22s	0,157s	0,16s
Jellifish	**qualité**	75,95%	75,76%	75,07%
	temps / im	0,025	0,013s	0,009s
Frying pan	**qualité**	77,93%	74,55%	73,77%
	temps / im	0,193s	0,126s	0,129s
Bottle	**qualité**	71,19%	80,95%	83,27%
	temps / im	0,16s	0,105s	0,101s
surveillance	**qualité**	66,98%	66,92%	67,19%
	temps / im	0,206s	0,123s	0,102s
coast guard	**qualité**	43,78%	23,44%	38,69%
	temps / im	0,168s	0,12s	0,107s
skijump	**qualité**	55,93%	55,93%	59,88%
	temps / im	0,071s	0,071s	0,02s
skicross	**qualité**	50,08%	58,90%	3,71%
	temps / im	0,055s	0,051s	0,001s
picture	**qualité**	71,15%	83,02%	82,83%
	temps / im	0,033s	0,027s	0,035s
tennis	**qualité**	69,83%	17,68%	49,36%
	temps / im	0,073s	0,033s	0,053s
foot from above	**qualité**	48,39%	43,56%	69,14%
	temps / im	0,022s	0,022s	0,024s
walking	**qualité**	74,74%	80,56%	69,24%
	temps / im	0,109s	0,104s	0,016s
cognac	**qualité**	45,02%	43,17%	50,14%
	temps / im	0,093s	0,083s	0,115s
traffic in Bombay	**qualité**	39,48%	58,86%	58,12%
	temps / im	0,067s	0,05s	0,05s
domatica	**qualité**	57,30%	48,62%	56,50%
	temps / im	0,113s	0,082s	0,015s

6.3.2 Structure propre au projet porTiVity

Dans le cadre du projet porTiVity, tous les points-clés et leurs descripteurs sont calculés pour la totalité de la vidéo. A l'issue de cette étape de prétraitement (voir **2.3.4**), les descripteurs et leurs positions sont stockés dans un fichier. Pendant l'utilisation de l'outil d'annotation, lorsqu'un objet est suivi, les informations nécessaires sont lues dans le fichier de descripteurs et seules les étapes d'appariement et de mise à jour du modèle sont effectuées. Nous avons évalué dans cette sous-section la vitesse du suivi dans ce contexte particulier de l'annotation vidéo. Nous avons testé le système d'annotation pour des objets de grande et de petite taille (celle-ci variant en moyenne de 5% d'un objet à l'autre dans une même catégorie) au travers d'une série de 28 exécutions sur une vidéo de foot au format MPEG-2 (avec une résolution 720×576 pixels). L'algorithme intégré dans le système est notre suivi exploitant uniquement des points-clés (sans descripteur global) et sans aucune des optimisations du temps d'exécution présentées dans la section précédente. L'ordinateur utilisé est un Pentium 4, 2.66 Ghz, avec 768 Mo de RAM. Les résultats sont présentés en Table 4. On constate que la majorité du traitement concerne la lecture dans le fichier. La vitesse de l'outil peut donc être substantiellement améliorée si les informations sont, par exemple, stockées en mémoire.

Table 4: Temps d'execution du suivi dans la structure d'annotation du projet porTiVity.

Taille moyenne de l'objet	Temps d'execution moyen	Temps de lecture du fichier (%)
Petit (71×38)	0.02s	66%
Grand (425×283)	0.11s	93%

7 Conclusions

7.1 Résumé

Nous avons présenté ici un système de suivi d'objet générique apte à répondre efficacement à la grande variabilité de difficultés susceptibles de surgir dans les domaines d'application les plus variés. Le trait suivi est la couleur en raison de sa possible exploitation quelque soit l'application considérée. Dans cette optique, un prétraitement des canaux couleurs vise à en harmoniser l'information pour une meilleure extraction de caractéristiques. Ce système modélise un objet grâce à un nuage de points-clés. Ce modèle a été étendu pour gérer, dans une certaine mesure, l'instabilité temporelle des points-clés et les occultations. De plus, nous avons mis au point un système de labellisation des points évolutif se basant sur quatre caractéristiques : la couleur, le mouvement, la position du point par rapport à la boîte englobante ainsi que le label du point auquel il est associé dans l'image précédente. Cette labellisation nous permet de différencier efficacement l'objet de son environnement lorsque celui-ci n'est pas occulté.

A chaque point est associé un descripteur de 18 moments couleurs. Ce descripteur est plus adapté au suivi d'objet que d'autres descripteurs renommés comme par exemple les SIFTs grâce à sa compacité. Afin d'en optimiser l'utilisation, nous en avons étudié le comportement dans notre application pour diverses bases de moments et des supports spatiaux variables.

L'appariement des points d'une image à l'autre est effectué grâce à un algorithme normalisé qui considère le descripteur associé aux points-clés aussi bien que leurs relations spatiales modélisées par une triangulation de Delaunay. Enfin, les variables du modèle de mouvement sont évaluées par la méthode des moindres carrés sur le nuage de points de l'objet et la boîte englobante est replacée en conséquence. Sa bordure est ensuite peaufinée en fonction du label des points l'environnant pour mieux s'ajuster aux déformations externes de l'objet.

Ce système, considérant tout objet comme hautement déformable, donne des résultats globalement satisfaisant dans la majorité des cas, remplissant ainsi les prérequis pour un système de suivi d'objet générique. Toutefois, il présente certaines limitations. Il reste sensible aux occultations de plus de 6 à 8 images, nécessite un nombre minimal de points-clés pour être efficace, et peut dériver de l'optimal lors de certains cas difficiles. Pour compenser ces faiblesses, nous avons expérimenté une approche hybride couplant les descripteurs locaux (des points-clés) avec un descripteur global (l'histogramme des régions aux couleurs discriminantes de l'objet). Cette approche nous a donné des résultats encourageants.

Enfin la dernière faiblesse de ce système est son coût algorithmique relativement élevé. Pour accélérer le suivi, nous avons expérimenté certaines variantes limitant le nombre de points-clés ou les images traitées et nous avons développé les points de Harris rapides. Ces derniers exploitent une structure inspirée des ondelettes de Haar pour une extraction accélérée des points de Harris.

7.2 Perspectives

De nombreuses pistes peuvent être envisagées pour continuer ces travaux. Tout d'abord, la couleur étant le trait caractéristique choisi à toutes les étapes du suivi, des tests peuvent être conduits en vue d'en accroître le potentiel. Par exemple, le seul espace couleur utilisé fut RGB. D'autres espaces colorimétriques tels que HSV ou xyY peuvent être envisagés.

L'utilisation d'un algorithme hybride utilisant notre modèle de descripteurs locaux et un descripteur global a montré des résultats prometteurs et mérite d'être exploré plus avant. La prise en charge des changements d'échelle par l'histogramme ou la combinaison de notre algorithme avec le Meanshift sont des exemples de futurs travaux possibles.

L'algorithme de labellisation des points-clés est également une réussite. Toutefois son efficacité peut être étendue par un système de paramétrisation automatique des poids. La comparaison de cette approche singulière à des techniques plus classiques se basant sur le flot optique est également une piste à explorer.

De plus, en se basant sur l'hypothèse qu'en moyenne, les variations entre des images consécutives d'une vidéo sont faibles, notre structure d'extraction accélérée des points-clés « Fast Harris » peut être étendue à la dimension temporelle.

Enfin, nos travaux ne sont pas isolés. Ils soulignent la tendance actuelle de la communauté scientifique à utiliser des points-clés dans les vidéos. Mais si cet outil est parfaitement adapté pour la reconnaissance d'objet, il présente encore certaines lacunes dans le cadre du suivi d'objet, tel que l'*instabilité temporelle*, et il existe un réel besoin de développement de points-clés dédiés à cette application. Ces limitations ont été abordées très tôt, avec la tentative pour créer des objets vidéo dans la norme MPEG-4. Plus, récemment, certains travaux ont été effectués pour créer des points spatiaux temporels, notamment avec l'extension les points de Harris à la vidéo. Une autre approche consiste à extraire des blobs couleurs 3D, en utilisant deux dimensions spatiales et une temporelle. Ces avancées sont peut-être les prémices de points-clés spécifiques au suivi d'objet.

Une autre lacune importante dans ce domaine de recherche concerne l'évaluation des résultats.Il existe plusieurs paramètres objectifs d'estimation des performances d'un suivi et, comme nous l'avons démontré, les métriques disponibles ne les recouvrent jamais tous. De plus, les résultats exhibés dans la littérature ne sont généralement constitués que d'un faible corpus de vidéos à même de montrer les avantages de la méthode. En conséquence, juger de la qualité d'un suivi est toujours une tâche ardue dans la mesure où les résultats présentés peuvent être effectués pour une mesure et des vidéos complaisantes. Il existe donc un réel besoin de normes pour mesurer la qualité d'un suivi ainsi qu'une base de données de référence accessible a tous.

8 Annexes : rappels mathématiques

Dans ce chapitre, nous rappellerons quelques définitions et propriétés mathématiques utilisées par les méthodes ou les algorithmes décrits dans la suite de ce manuscrit. Nous ne présenterons pas une description exhaustive des sujets abordés, mais n'expliquerons que les concepts nécessaires à la compréhension de ce qui va suivre.

Dans la première section, nous nous intéresserons aux moments cartésiens 2D. Ensuite, nous décrirons les bases de l'analyse d'ondelette, puis la triangulation de Delaunay incrémentale.

8.1 Annexe1 : Moments 2D

Définition : Les moments 2D m_{pq} d'ordre $p+q$ d'une fonction de densité f(x,y) sont définis par :

$$m_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x^p y^q f(x,y)dxdy$$

Dans le cas d'une image, f(x,y) est le niveau de gris du pixel aux coordonnées (x,y).

Moments centrés : les moments centrés μ_{pq} sont calculés par rapport au centre de gravité de l'objet (x_g,y_g), et sont définis par :

$$\mu_{pq} = \int_{-\infty-\infty}^{+\infty+\infty}\int (x-x_g)^p (y-y_g)^q f(x,y)dxdy$$

Les moments centrés peuvent être calculés directement à partir des moments d'ordre inférieurs à p+q+1 par :

$$\mu_{pq} = \sum_{k=0}^{p}\sum_{l=0}^{q}\binom{p}{k}\binom{q}{l}(-x_g)^{(p-k)}(-y_g)^{(q-l)} m_{kl} \qquad \text{où } \binom{p}{k} = \frac{p!}{k!(p-k)!}.$$

Réciproquement, il est possible de déterminer les moments à partir des moments centrés. On a en effet :

$$m_{pq} = \sum_{k=0}^{p}\sum_{l=0}^{q}\binom{p}{k}\binom{q}{l}(x_g)^{(p-k)}(y_g)^{(q-l)} \mu_{kl}$$

Il est important de noter que les moments centrés sont **invariants aux translations.**

Moments normalisés : On définit les moments normalisés par :

$$v_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\omega}} \qquad \text{avec } \omega = (p+q+2)/2$$

Les moments normalisés sont **invariants aux changements d'échelles.**

Moments complexes : Les moments complexes sont définis par :

$$c_{pq} = \int_{-\infty-\infty}^{+\infty+\infty}\int (x+iy)^p (x-iy)^q f(x,y)dxdy$$

Leur utilité vient de leur comportement face aux rotations. En effet, en coordonnées polaires, cette formule devient :

$$c_{pq} = \int_0^{+\infty} \int_0^{2\pi} r^{p+q+1} e^{i(p-q)\theta} f(r,\theta) dr d\theta$$

Il s'en suit que $c_{pq} = \overline{c_{pq}}$, autrement dit que la magnitude d'un moment complexe est **invariante à la rotation.**

Image numériques : Dans le cas des images, la double intégrale doit être remplacée par une sommation. On a alors :

$$m_{pq} = \sum_V x^p y^q I(x,y)$$

avec V l'ensemble de pixels sur lequel le calcul du moment est appliqué et I(x,y) l'intensité du pixel (x,y). Les moments centraux et complexes sont également définis par :

$$\mu_{pq} = \sum_V (x - x_g)^p (y - y_g)^q I(x,y)$$

$$c_{pq} = \sum_V (x + iy)^p (x - iy)^q I(x,y)$$

où (x_g, y_g) est le centre de gravité ou *centroïd* de l'objet étudié. Notons que le moment centré du second ordre n'est alors autre que la variance. Dans la suite de cette section, seul le cas des images numériques sera considéré.

Moment d'ordre 0 : surface. Le moment d'ordre zéro m_{00} d'une distribution de points (ou de pixels) n'est autre que l'aire recouverte par l'objet, soit son nombre de pixels.

Moments d'ordre 1 : centre de gravité. Les moments d'ordre 1 m_{10} et m_{01} permettent de définir le centre de gravité de l'objet. Plus précisément, ses coordonnées x_g et y_g sont données par :

$$x_g = \frac{m_{10}}{m_{00}} \quad et \quad y_g = \frac{m_{01}}{m_{00}}$$

Moments d'ordre 2 : axes de symétrie. Les moments d'ordre 2, m_{02}, m_{11}, et m_{20} permettent, avec les coordonnées du centre de gravité de définir un *rectangle englobant* ayant les mêmes moments d'ordre 0, 1, et 2 que l'objet assimilé. Le centroïde $G = (x_g, y_g)$, l'angle θ entre l'ordonnée et l'axe principal, et les longueurs des axes (w,l) sont donnés par les équations suivantes :

$$x_g = \frac{m_{10}}{m_{00}}$$

$$y_g = \frac{m_{01}}{m_{00}}$$

$$\theta = \frac{1}{2} \arctan\left(\frac{b}{a-c}\right)$$

$$w = \sqrt{6(a + c - \sqrt{b^2 + (a-c)^2})}$$

$$l = \sqrt{6(a + c + \sqrt{b^2 + (a-c)^2})}$$

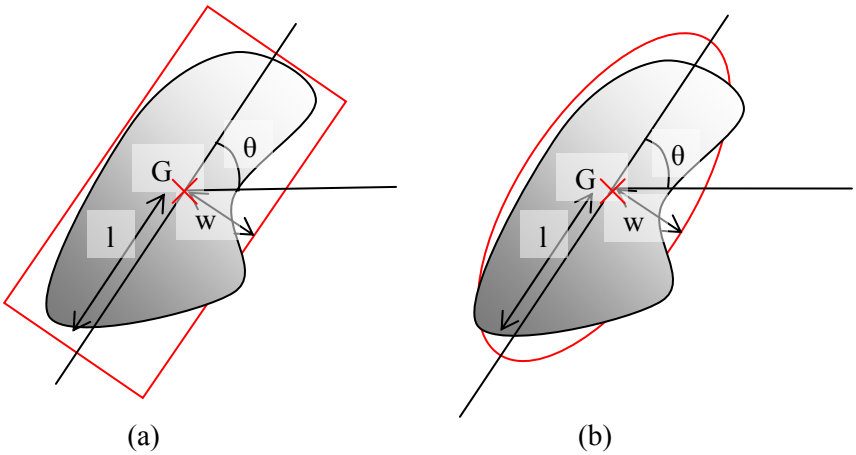
où a,b,c sont définis par :

$$a = \frac{m_{20}}{m_{00}} - x_g^2$$

$$b = 2\left(\frac{m_{11}}{m_{00}} - x_g y_g\right)$$

$$c = \frac{m_{02}}{m_{00}} - y_g^2$$

De même, il est possible de construire une *ellipse englobante*, ayant les mêmes propriétés que le rectangle englobant avec G le centre, et w et l les rayons principaux et secondaires. La Figure 83 illustre ce concept.



(a) (b)

Figure 83: Approximation basée sur les moments. (a) Rectangle englobant (b) Ellipse englobante.

L'équation de l'ellipse est la suivante :

$$\frac{(x - x_g + t(y - y_g))^2}{l^2(1+t^2)} + \frac{(y - y_g - t(x - x_g))^2}{w^2(1+t^2)} = 1$$

Il est également possible d'inférer l'ellipse englobante ou le rectangle englobant à partir des moments centraux d'ordre 2 (qui sont appelés moments d'inertie) et inférieur.

Moments centrés d'ordre 3 : asymétrie. Les moments centrés d'ordre 3 évaluent le degré d'asymétrie par rapport aux axes principaux de la forme étudiée. Ils sont définis de la manière suivante :

$$S_x = \frac{\mu_{30}}{\mu_{20}^{3/2}} \quad et \quad S_y = \frac{\mu_{03}}{\mu_{02}^{3/2}}$$

Un coefficient d'asymétrie est égal à zéro pour une distribution répartie de façon symétrique, alors qu'un coefficient d'asymétrie supérieur ou inférieure à zéro représentera une distribution déséquilibrée respectivement vers la gauche ou vers la droite (voir Figure 84)

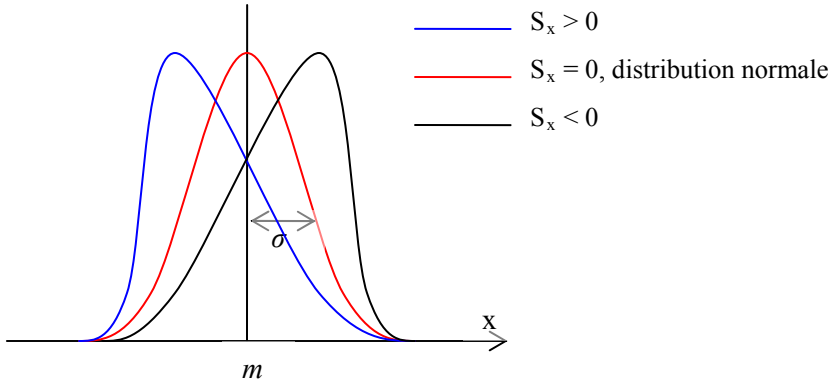


Figure 84: Relation entre la distribution et l'asymétrie.

De plus, les moments d'ordre 3 permettent de résoudre le problème du choix de la valeur θ. En effet, comme on peut le remarquer sur la Figure 84, il existe deux orientations possibles pour l'objet : θ et θ+л. Les moments d'ordre 3, lorsqu'ils ne sont pas nuls, offrent un moyen de résoudre cette ambiguïté. En effet, le signe des moments d'ordre 3 changent pour une rotation de l'objet de plus de 180 degrés. Il suffit donc de vérifier que leurs signes sont les mêmes que ceux d'une image de référence. Plus formellement, soient μ_{ij} les moments d'ordre 3 calculés pour l'image de référence dans un repère dont le centre coïncide avec le centre de l'objet et θ_r la valeur de l'orientation de l'objet fixée pour cette image. Les moments après un mouvement de rotation peuvent être calculés par la formule suivante :

$$\mu'_{ij} = \sum_{L1=0}^{i}\sum_{L2=0}^{j}\binom{i}{L1}\binom{j}{L2} r_{11}^{L1} r_{21}^{L2} r_{12}^{i-L1} r_{22}^{j-L2} \mu_{L1+L2,i+j-(L1+L2)}$$

où r_{mn} sont les éléments de la matrice de rotation dans le plan d'angle $(\theta-\theta_r)$:

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos(\theta-\theta_r) & \sin(\theta-\theta_r) \\ -\sin(\theta-\theta_r) & \cos(\theta-\theta_r) \end{bmatrix}$$

En effet, l'ensemble de pixels V de l'objet considéré étant le même après rotation, on a :

$$\mu'_{ij} = \sum_{V} (r_{11}x + r_{12}y)^i (r_{21}x + r_{22}y)^j \mathrm{I}(x,y)$$

Ce qui, après avoir développé $(r_{11}x+r_{12}y)^i(r_{21}x+r_{22}y)^j$ nous donne l'équation précédente.

Moments centrés d'ordre 4 : Kurtosis. Les moments centrés d'ordre 4 quant à eux expriment le Kurtosis de l'objet, soit le degré d'aplatissement par rapport à une distribution normale selon une direction donnée (l'axe x ou l'axe y). Ils sont calculés grâce aux formules suivantes :

$$K_x = \frac{\mu_{40}}{{\mu_{20}}^2} - 3 \quad et \quad K_y = \frac{\mu_{04}}{{\mu_{02}}^2} - 3$$

Un Kurtosis inférieur à zéro indique une distribution plus plate, qu'une distribution normale alors qu'un kurtosis supérieur à zéro indiquera une distribution plus prononcée selon la direction considérée (voir Figure 85). Rappelons que l'équation de la courbe normale est donnée par :

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-m)^2}{2\sigma^2}\right]$$

avec m la moyenne et σ l'écart type pour la distribution étudiée.

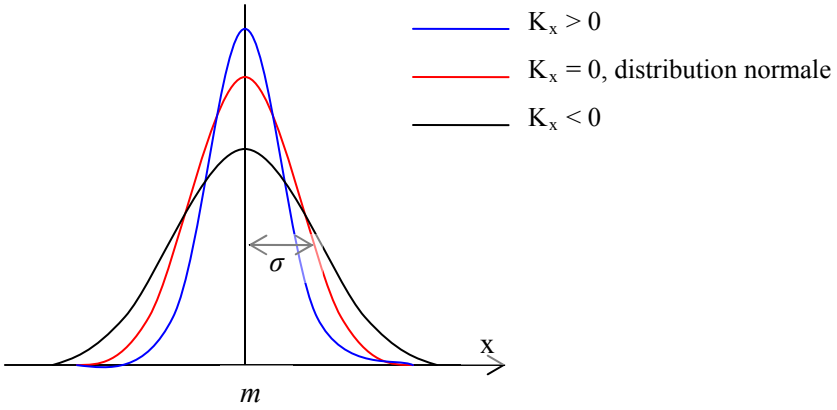


Figure 85: Relation entre la distribution et le Kurtosis.

Moments d'ordre supérieur : Les moments d'ordre supérieur n'ont pas d'interprétation géométrique aisée.

8.2 Annexe2 : Ondelettes

La transformée de Fourier et ces limites

L'analyse de Fourier est un outil qui révolutionna les mathématiques au 19eme siècle. Partant du principe qu'un signal est une somme de fonctions périodiques, elle permet de le décomposer en une série de sinus et cosinus correspondant chacun à une fréquence donnée. Un signal étant une représentation numérique d'un phénomène physique (son, image, onde,…), leur omniprésence dans notre vie moderne fait de l'analyse de Fourier un outil aujourd'hui utilisé par de nombreuses méthodes et algorithmes.

La transformée de Fourier d'un signal est donnée par la formule suivante :

$$f(t) = \frac{a_0}{2} + \sum_{n=0}^{\infty} a_n \cos nt + \sum_{n=0}^{\infty} b_n \sin nt$$

avec a_n et b_n les coefficients de la transformée codant l'**importance de la fréquence** qu'ils pondèrent pour le signal.

Dans le cadre de l'analyse d'image, la transformée de Fourier est très riche en information. Elle permet de caractériser la **régularité** d'un signal qui est directement dépendante du nombre de coefficients non nuls. En effet, pour un signal régulier, peu de coefficients sont nécessaires pour reproduire le signal les coefficients devenant rapidement négligeables pour les fréquences élevées. Par contre, un signal irrégulier ou bruité se traduira par la présence de hautes fréquences dans le signal, plus de coefficients seront donc nécessaires pour caractériser l'image.

De plus, dans l'espace de Fourier, on peut réaliser des opérations de filtrages moins coûteuses que par des convolutions spatiales notamment grâce à l'algorithme de la FFT (*Fast Fourier Transform*). La convolution dans le domaine spatial se traduit en effet par une simple multiplication dans le domaine fréquentiel.

Mais malgré ces avantages, la transformation de Fourier ne permet **aucune localisation spatiale** des informations obtenues. Autrement dit, la transformation de Fourier permet de déterminer le nombre et l'importance des fréquences d'un signal, mais pas de localiser celui-ci dans le temps.

L'analyse de Fourier à fenêtres apporta une réponse à ce problème. Cette méthode découpe un signal en fenêtres sur lesquelles une transformée de Fourier est effectuée séparément. Son inconvénient réside dans le choix de la taille de la fenêtre. Plus la fenêtre est petite, plus la localisation sera précise,

mais plus le nombre de basses fréquences observé sera faible. Cette solution à l'absence de localisation spatiale de l'information n'est donc que partielle.

Ce n'est qu'en 1909, que Haar apporta une solution satisfaisante, en développant les premières ondelettes.

8.2.1 Principe des ondelettes

Les ondelettes sont des fonctions **oscillantes** comme sinus et cosinus qui permettent, tout comme la transformée de Fourier, de décomposer un signal. Par contre, elles sont non seulement localisées en fréquence mais aussi **dans le temps**. Le caractère localisé de l'ondelette s'exprime par le fait que la fonction est non nulle sur un intervalle fini et nulle partout ailleurs (on parle d'ondelette à **support compact**).

La transformée en ondelettes est une représentation **en fréquences**. Le signal est codé comme étant une somme d'ondelettes qui sont issues d'une seule et même ondelette appelée ***ondelette mère***. Selon la résolution, cette ondelette sera **dilatée** pour représenter un signal plus grossier, ou compressée pour modéliser les hautes fréquences. De plus, l'ondelette sera **translatée** suivant la localisation dans le temps du signal à représenter. Enfin, tout comme dans la transformée de Fourier, l'importance du signal sera codée grâce à des coefficients appelés **coefficients d'ondelettes** (voir Figure 86). Le choix de l'ondelette mère dépend bien sûr de ce que l'on désire réaliser.

La transformée en ondelettes d'une image consiste à coder les **variations de l'image** à différentes résolutions. A chaque résolution, le signal est approximé par une fonction d'échelle φ aussi appelée ***signal d'échelle***, et une ondelette ψ, aussi appelée ***signal de détail,*** qui code la différence entre le signal d'échelle et le signal d'origine.

L'image est ainsi codée à différentes échelles ½, ¼,…,2^j avec $j \in Z \quad et \quad j \leq -1$. A chaque étape, le nombre de coefficients est donc divisé par 2 (on parle d'analyse **dyadique**). La **transformée en ondelettes discrète** est obtenue en effectuant ce calcul pour E_{max} échelles avec $E_{max} = \log_2 N$, N étant le nombre d'échantillons du signal.

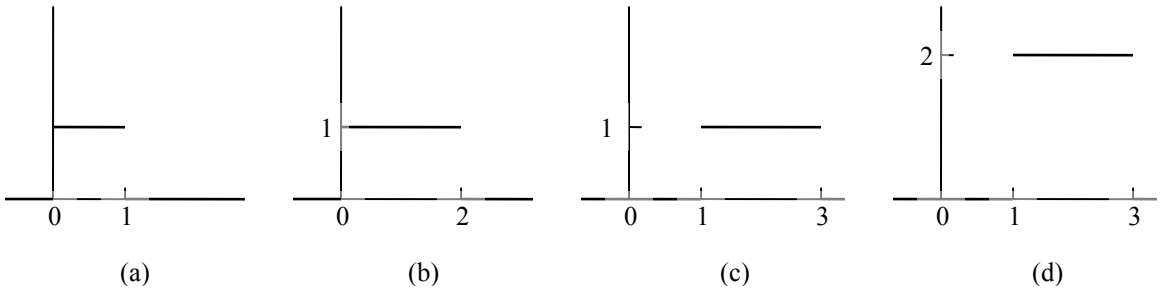


Figure 86: Exemple avec la fonction d'échelle de Haar φ (a) φ[0,1] (b) φ[0,2] : dilatation d'un rapport 2 (c) φ[1,3] : dilatation d'un rapport 2 et translation de 1 (d) 2.φ[1,3] :fonction dilatée, translatée, et pondérée du cœfficient d'ondelette 2.

8.2.2 Exemple de l'ondelette de Haar

Afin de bien comprendre le fonctionnement des ondelettes, nous allons ici étudier le cas de la transformée de Haar au travers de l'exemple simple présenté en Figure 87.

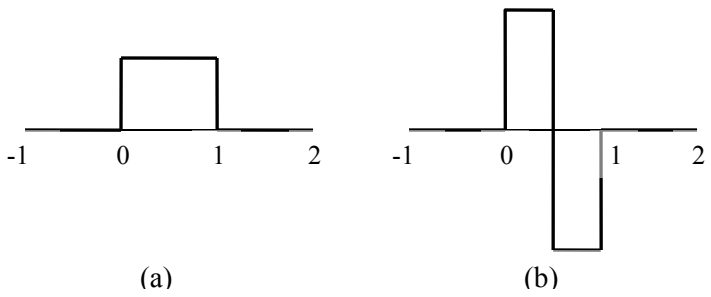


Figure 87: (a) Fonction d'échelle de Haar φ (b) Ondelette de Haar ψ.

Le filtre d'échelle a donc une valeur de [1,1] et le filtre d'ondelette [1,-1].
Soit un signal mono-dimentionnel composé de quatre échantillons :

[7 5 4 10]

La première étape consiste à approximer le signal à partir de la fonction d'échelle. Ici, cette approximation s'effectue en calculant la moyenne deux à deux des coefficients. Ensuite, afin de conserver l'information perdue, on calcule la différence entre le signal et l'approximation (le signal de détail). On répète cette procédure récursivement pour chaque résolution en représentant le signal par un coefficient de moyenne du signal et ses signaux de détails successifs (Figure 88).

Echelle	Approximation du signal	Signal de détail
1	[4 2 3 7]	
½	[3 5]	[1 -2]
1/4	[4]	[-1] [1 -2]

Signal d'origine
↓
Transformée de Haar

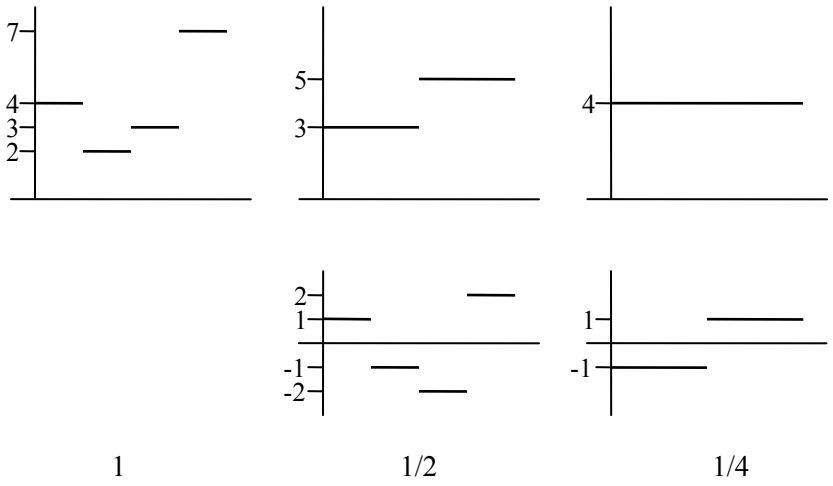


Figure 88: Calcul de la transformée de Haar d'un signal.

ns d'échantillons, conduit à un algorithme très efficace dit algorithme du papillon (Figure 89) qui se généralise en FWT (*Fast Wavelet Transform*).

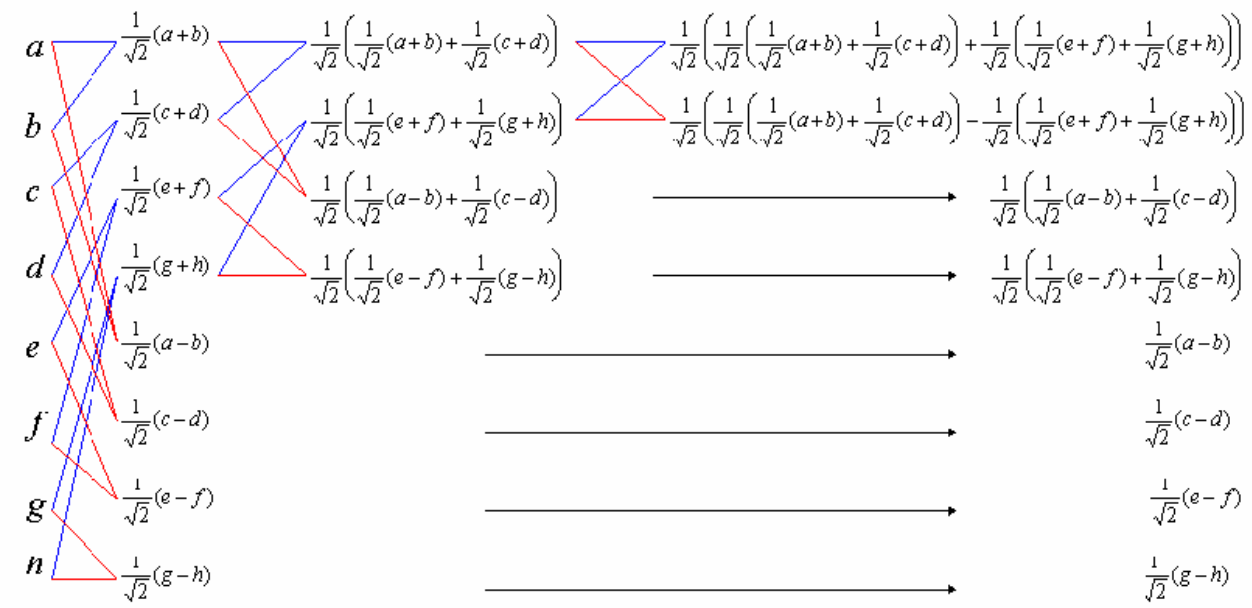


Figure 89: La transformée de Haar par l'algorithme du papillon **[LOU00].**

8.2.3 Ondelettes de Daubechie

Les ondelettes dilatées et translatées conduisent à une représentation complète et non redondante du signal si elles forment une base orthonormale de l'espace fonctionnel (ce qui est toujours le cas pour un signal échantillonné). Plus précisément, il s'agit de $L^2(R)$, l'espace des fonctions mesurables et d'énergie finie. Les ondellettes orthogonales permettent donc un codage exact et non redondant du signal.

A part l'ondelette de Haar, une seule famille d'ondelettes est à la fois orthogonale et à support compact : les ondelettes de Daubechies. Contrairement aux ondelettes de Haar qui analyse l'image en blocs disjoints, les ondelettes de Daubechies sont continues. En effet, le support spatial de leur filtre étant toujours de taille supérieur ou égal à 2, elles se recouvrent toujours à une échelle donnée. En conséquence, elles approximent les signaux continus plus précisément et sont moins sensibles aux translations que les ondelettes de Haar (*effet de bloc*). En revanche, leur calcul est beaucoup plus coûteux.

L'ondelette de Daubechie la plus connue, aussi appelée Daubechie 4 à cause du nombre de coefficients de ses filtres est représentée en Figure 90.

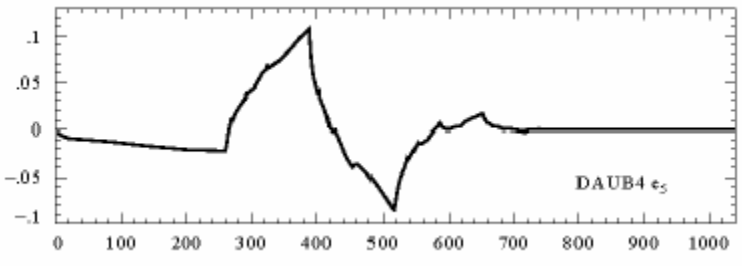


Figure 90: Ondelette de Daubechie 4.

Les filtres d'échelle de Daubéchie 4 sont constitué des quatre coefficients [h_0 h_1 h_2 h_3] suivants :

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \qquad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}$$
$$h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \qquad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

8.2.4 Application aux images

Le modèle des ondelettes étant généralisable à une dimension quelconque, le cas des signaux monodimensionnels peuvent être étendus aux signaux bidimensionnels que sont les images. La fonction d'ondelette ψ est alors remplacée par 3 fonctions (ψ_1, ψ_2, ψ_3) tel que :

$$\psi_1(x,y) = \varphi(x) \,.\, \psi(y)$$
$$\psi_2(x,y) = \psi(x) \,.\, \varphi(y)$$
$$\psi_3(x,y) = \psi(x) \,.\, \psi(y)$$

Les fonctions ψ_1 et ψ_2 expriment les variations horizontales et verticales de l'image alors que la fonction ψ_3 exprime une combinaison des deux (voir Figure 91). Ces ondelettes sont toutes à support carré. Il existe une autre décomposition qui consiste à appliquer les filtres en découpant selon les lignes et les colonnes de l'image. Toutefois, les ondelettes résultantes sont à support rectangulaire, ce qui les rend plus difficile à interpréter. Cette technique est dite **transformée standard** par opposition à la précédente dite **transformée non standard**. Leur application est présentée dans la Figure 92.

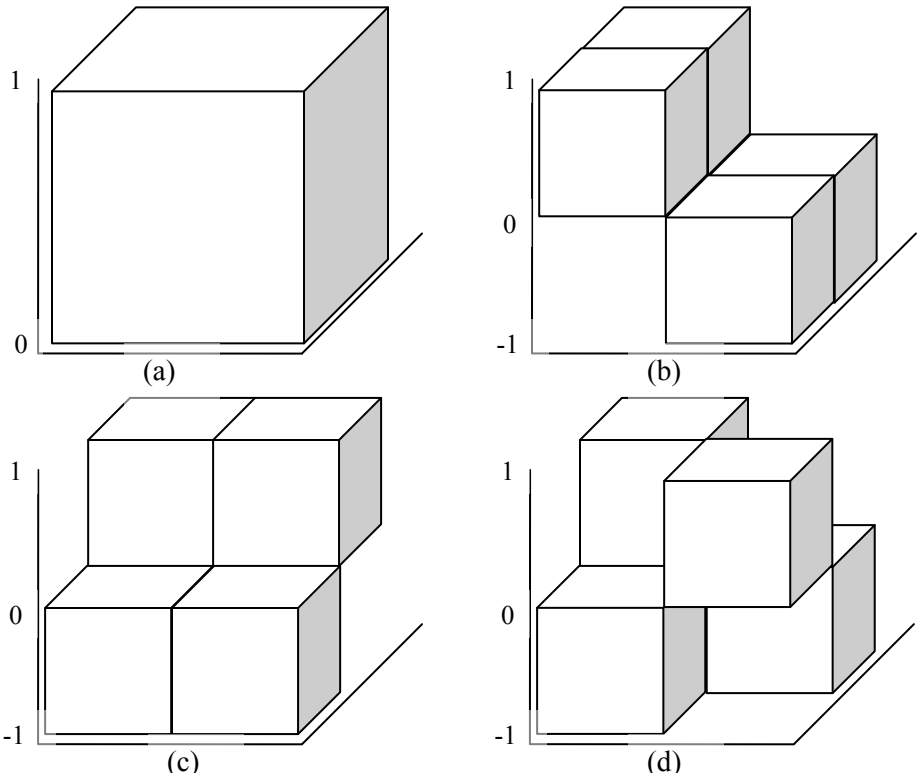


Figure 91: Ondelettes de Haar 2D (a) φ(x,y) (b) ψ1(x,y) (c) ψ2(x,y) (d) ψ3(x,y).

Image Transformée de Haar Filtres

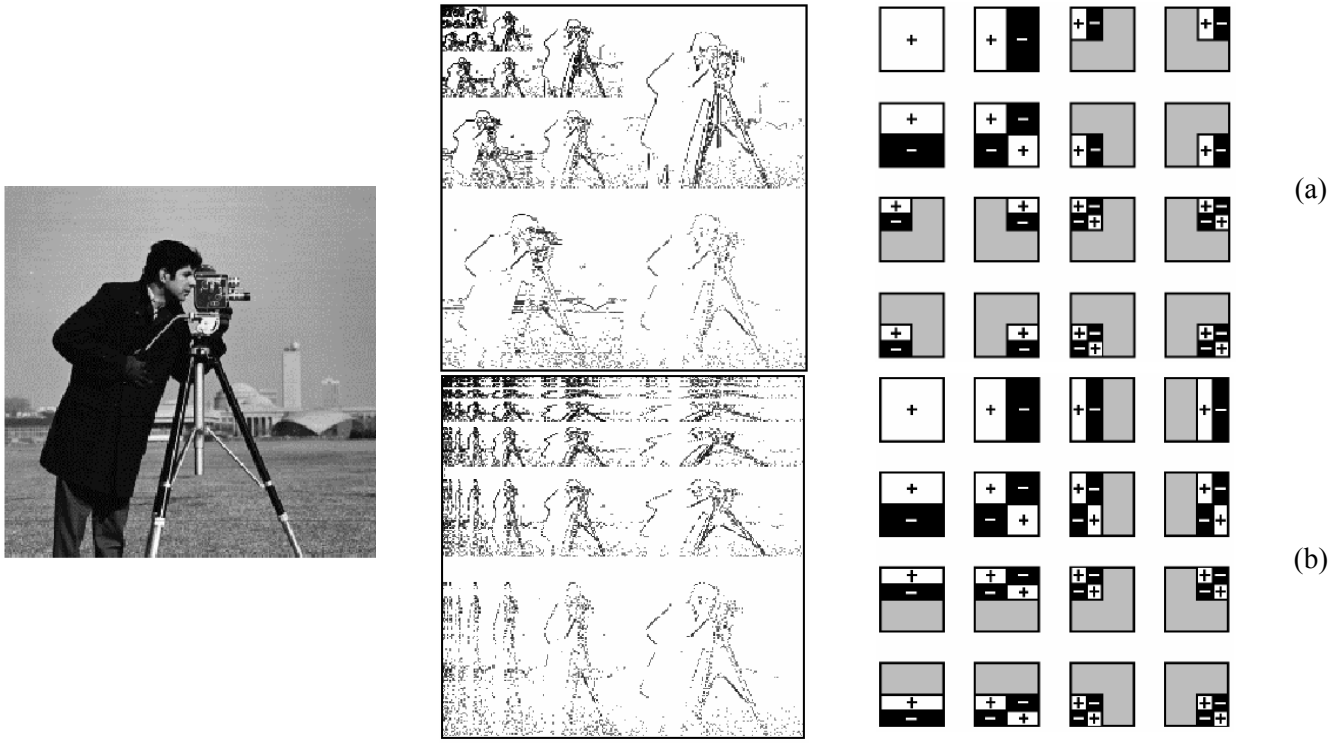


Figure 92: Transformée de Haar pour l'image du cameraman (a) transformée non standard (b) transformée standard.

8.3 Annexe3 : Triangulation de Delaunay incrémentale

Trianguler un ensemble de points dans un espace à n dimensions consiste à mailler ce semis de points à l'aide de triangles ayant pour sommets les points du semis. Ce maillage doit répondre à certaines propriétés, à savoir que l'intersection de deux triangles est soit l'ensemble vide (les deux triangles sont distincts), soit un point, soit une arête (aucun croisement ou superposition de triangle n'est autorisé). Bien que le problème puisse se poser pour un nombre de dimensions quelconque, nous nous limiterons ici à un espace à deux dimensions, qui est le cas d'école.

8.3.1 Triangulation de Delaunay

Une méthode de triangulation de Delaunay s'appuie sur le critère du même nom suivant :

Un triangle est dit « de Delaunay » si son cercle circonscrit ne contient aucun sommet en son intérieur.

(Figure 93)

De même une triangulation est dite « de Delaunay » si le cercle circonscrit de chacun de ses triangles ne contient aucun sommet en son intérieur.

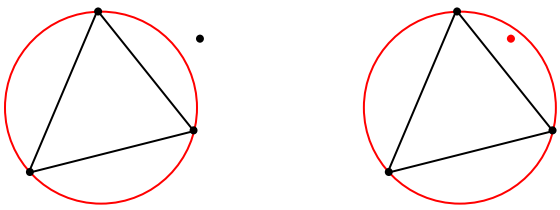


Figure 93: Triangle de Delaunay Triangle non Delaunay.

Une triangulation de Delaunay offre plusieurs avantages. Tout d'abord, elle a la particularité d'être unique. Ensuite, elle optimise la qualité de la triangulation selon le critère de l'angle minimal (le plus petit angle de la triangulation est maximisé).

Il existe de nombreux algorithmes pour créer une triangulation de Delaunay. Nous ne décrirons ici que la triangulation incrémentale aussi connue sous le nom de triangulation itérative. Après avoir présenté les opérations de base utilisées pour mener à bien une triangulation de Delaunay, nous décrirons l'insertion et la suppression incrémentale de points. Pour une plus grande documentation sur les algorithmes existants, consulter l'état de l'art de Kumar **[Kum96]**.

8.3.2 Opérateurs de triangulation

Nous allons introduire dans cette partie la plupart des opérations élémentaires effectuées lors d'une triangulation de Delaunay :

- **Recherche du triangle englobant un point** : Il est possible de déterminer de quel coté d'une arête se trouve un point. Donc, pour un triangle donné et pour une arête de ce triangle, il est possible de savoir si un point se trouve à l'intérieur ou à l'extérieur du triangle par rapport a cette arête suivant qu'il se trouve ou non du même coté que le point du triangle opposé à l'arête (Figure 94). Si un point est vers l'intérieur du triangle par rapport à chacune des arêtes de ce triangle, alors il est inscrit dans le triangle. Etant donné les coordonnées d'un point, on peut donc trouver le triangle dans lequel il est inscrit avec l'algorithme suivant initialisé sur un triangle sélectionné au hasard :

Tant que le triangle inscrit au point ≠ trouvé
Si le point est à l'extérieur du triangle par rapport à A
Triangle = triangle voisin par rapport à A
Sinon
Si le point est vers l'intérieur du triangle pour toutes les arêtes
triangle inscrit au point = trouvé
Sinon
Itérer sur les arêtes du triangle

La Figure 95 illustre ce procédé

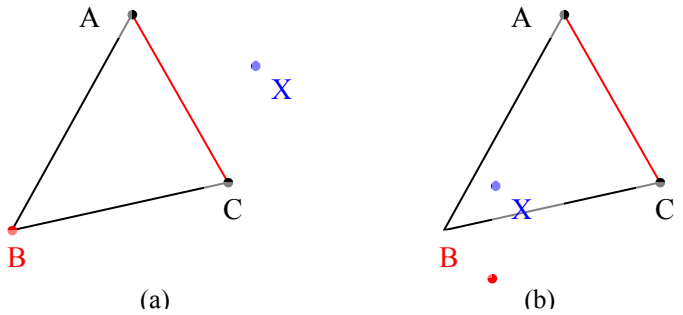
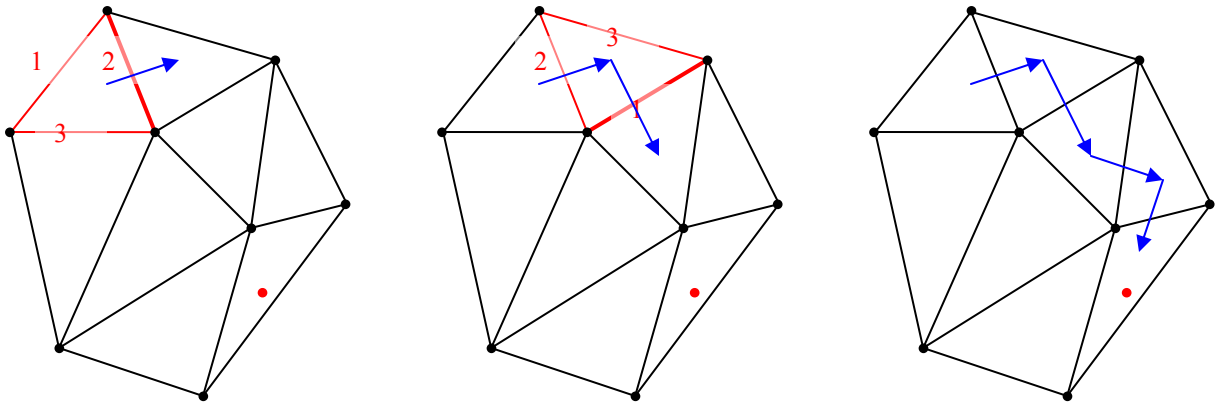


Figure 94: (a) X n'est pas du même coté que B par rapport à l'arête, il est donc vers l'extérieur du triangle (b) X est du même coté que B par rapport à l'arête, il est donc vers l'intérieur du triangle.



(a) (b) (c)

Figure 95: Déroulement d'un algorithme de recherche du triangle englobant d'un point. En rouge le triangle à l'étude, l'ordre de ses arêtes ainsi que le point dont on cherche le triangle englobant (a) 1ere itération (b) 2eme itération (c) Dernière itération.

- **Ajout d'un sommet dans un triangle** : Cette opération consiste, étant donné un point et le triangle dans lequel il est inscrit, à éliminer le triangle dans un premier temps, pour ensuite le remplacer par trois triangles issus du point inséré et de chacune de ces arêtes (Figure 96).

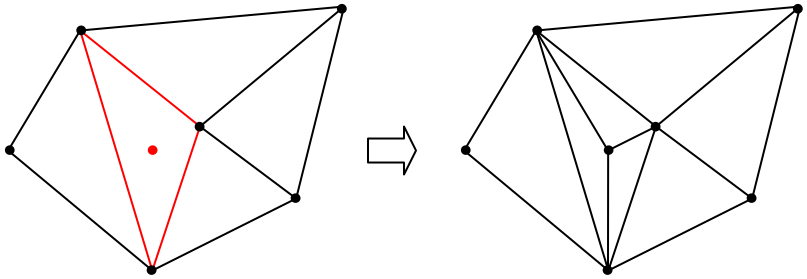


Figure 96: Ajout d'un sommet dans un triangle.

- **Renversement d'arête** : Cette opération survient lorsque un triangle et le point d'un de ces triangles voisins sont en conflit, i.e. lorsque le critère de Delaunay n'est pas respecté pour un triangle. Cet opérateur prend en entrée les deux triangles en conflit et donne en sortie deux triangles comportant les mêmes points mais dont l'arête commune diffère (Figure 97).

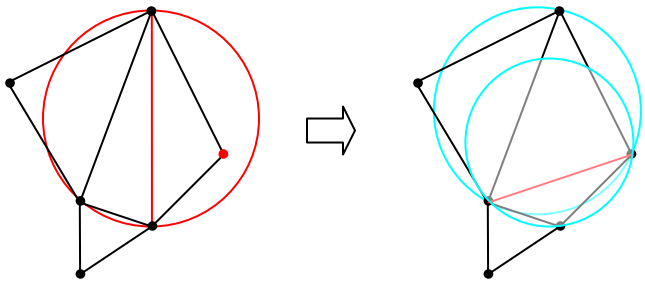


Figure 97: Renversement d'arête suite à un conflit entre deux triangles.

8.3.3 Triangulation et suppression incrémentale

Le principe de la triangulation de Delaunay incrémentale est le suivant : Les points sont insérés un par un. Pour chaque point inséré, on identifie le triangle dans lequel il est inscrit. On supprime alors ce triangle et on le remplace par trois nouveaux triangles ayant pour sommets ce point et deux des points du triangle englobant. On vérifie ensuite le critère de Delaunay pour ce point, c'est à dire que le point inséré ne se trouve pas dans le cercle circonscrit d'un des triangles voisins. Si c'est le cas, on fait pivoter l'arête commune entre les deux triangles. Dans la mesure où la triangulation était une triangulation de Delaunay avant l'insertion du point, tous nouveau cas de conflit aura nécessairement pour origine le point inséré. Il y a donc au maximum trois conflits (un pour chacun des nouveaux triangles) pouvant être résolus indépendamment par un renversement d'arête. Pour que cet algorithme soit applicable, il faut initialiser le maillage de façon à se que tous les points à insérer soit englobés par au moins un triangle. Une fois tous les points insérés, on retire ce (ou ces) triangle(s) du maillage et on corrige éventuellement en rajoutant des triangles sur le bord du maillage de façon à former une enveloppe convexe.

L'insertion de points n'est pas la seule opération incrémentale qui peut être effectuée. On peut également être amené à supprimer un point du maillage. Le principe consiste à éliminer tous les triangles et arêtes issus du point p, puis à retrianguler le trou. Mais, avant de supprimer les triangles et arêtes, les points et triangle voisins concernés par la retriangulation qui va suivre sont enregistrés dans l'ordre défini en suivant la bordure du trou. On crée ensuite les triangles de façon à ne pas superposer un triangle déjà existant, i.e. à ne pas dépasser de la bordure du trou. Cette étape est la plus délicate dans la mesure où le trou à retrianguler n'est pas nécessairement convexe. Enfin, on met à jour le voisinage des triangles grâce aux valeurs sauvegardées, et on vérifie que les triangles créés satisfont le critère de Delaunay (tournant les arêtes des triangles en conflit, le cas échéant).

9 Références

[Agg99] J. K. Aggarwal, and Q. Cai, **Human motion analysis: A review**. Computer Vision Image Understanding, 73, 3, pp. 428–440, 1999.

[Bad02] W. Badawy, M. A. Bayoumi, **A Low Power VLSI Architecture for Mesh-Based Video Motion Tracking**, *IEEE Transactions on Cicuits and Systems*, vol. 49, no. 7, july 2002.

[Bar88] Y. Bar-Shalom, T.E.FortMann, **Tracking and Data Association**, *New-York, Academic Press*, 1988.

[Bau00] A. Baumberg**. Reliable feature matching across widely separated views,** *In IEEE Conference on Computer Vision and Pattern Recognition*, pp 774-781, 2000.

[Bey00] D. Beymer, **Person counting using stereo**; *Workshop on Human Motion*, Dec. 2000.

[Bez81] J.C. Bezdek, **Pattern Recognition with Fuzzy Objective Function Algorithms**, *Plenum Press*, New York, 1981.

[Bha43] A. Bhattacharyya, **On a measure of divergence between two statiscal populations defined by their probability distribution**, *Bulletin of the Calcutta Mathematical society*, vol 35, pp. 99-110, 1943.

[Bha99] S. Bhattacharjee, **Image retrieval based on structural content**, *Workshop on image analysis for multimedia interactive services*, Heinrich-Hertz-Institut(HHI) Berlin, Germany, 31 May, 1er June 1999.

[Bid02] R. Bidoggia, S. Gentili, **A basis of invariant moments for color images**, *IWSSIP'02*, pp. 527-531, Royaume-Uni, 2002.

[Bla03] J. Black, T. Ellis, and P. Rosin. **A novel method for video tracking performance evaluation**, In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 125-132, 2003.

[Bre99] S. Bres, J.-M. Jolion, **Detection of Interest Points for Image Indexation** , *Third International Conference, VISUAL'99*, Amsterdam, The Netherlands, pp. 427-434, June 1999.

[Bre05] L.Brethes, P.Danes, F.Lerasle, **Stratégies de filtrage particulaire pour le suivi visuel de personnes: description et évaluation**, Rapport LAAS N°05359, 10p, Juin 2005.

[Bro02] M. Brown and D. G. Lowe. **Invariant Features from Interest Point Groups**. *13th British Machine Vision Conference (BMVC2002)*, pages 253-262, 2002.

[Bro05] M. Brown, R. Szeliski and S. Winder. **Multi-Image Matching using Multi-Scale Oriented Patches,** *International Conference on Computer Vision and Pattern Recognition (CVPR2005)*, pp 510-517, 2005.

[Bru81] Bruce D. Lucas and Takeo Kanade. **An Iterative Image Registration Technique with an Application to Stereo Vision**. *International Joint Conference on Artificial Intelligence*, pages 674-679, 1981.

[Cav05] A. Cavallaro, O.Steiger, T. Ebrahimi, **Tracking video objects in cluttered background**, *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4), pp.575- 584, April 2005.

[Che98] D. Chetverikov, J. Verestoy, **Tracking feature points: a new algorithm**, *Proc. Int. Conf. on Pattern Recognition*, pp. 1436-1438, 1998.

[Chu05] O. Chum, J. Matas, **Matching with PROSAC - progressive sample consensus,** *Computer Vision and Pattern Recognition*, CVPR 2005. IEEE Computer Society Conference on, Volume 1, 20-25, pp 220-226 vol. 1, June 2005.

[Col05] R.T. Collins, X. Zhou, and S.K. Teh, **An Open Source Tracking Testbed and Evaluation Web Site,** *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005)*, January, 2005.

[Com02] Comaniciu D., Meer P, **Mean Shift: A Robust Approach Toward Feature Space Analysis**, IEEE Trans. Pattern Anal. Mach. Intell. 24(5): 603-619, 2002.

[Cox96] I. Cox and S. Hingorani, **An efficient implementation of Reid's Multiple Hypothesis Tracking algorithm and its evaluation for the purpose of visual tracking**, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, Sept. 1996.

[Dal05] N. Dalal, B. Triggs, **Histograms of Oriented Gradients for Human Detection**, *International Conference on Computer Vision & Pattern Recognition* - June 2005.

[Doe00] D. Doermann and D. Mihalcik, **Tools and Techniques for Video Performances Evaluation**, *ICPR*, pp. 167-170, 2000.

[Don06] M. Donoser, H. Bischof: **Efficient Maximally Stable Extremal Region (MSER) Tracking**. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, Vol. 1, pp. 553-560, 2006.

[Don06-2]] M. Donoser, H. Bischof, M. Wiltsche, **Color Blob Segmentation by MSER analysis**, *Proc. of International Conference on Image Processing*, 2006

[Duf00] Y. Dufournaud, Cordelia Schmid, Radu Horaud, **Matching Images with Different Resolutions**, *International Conference on Computer Vision & Pattern Recognition*, June 2000.

[DVB-H] DVB-H: ETSI EN 302 304 **Transmisión for handheld terminals (DVB-H).**

[Eti02] E. Etievent, **Assistance à l'indexation vidéo par l'analyse du mouvement**, Thèse lyon 1, 2002.

[Fis81] M. A. Fischler and R. C. Bolles, **Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography**. *Comm*. ACM 24 (6), pp 381-395, 1981.

[Flu06] J. Flusser, **Moment invariants in image analysis**. *Proceedings of the International Conference on Computer Science*. ICCS'06. Computica, pp. 196-201, Prague 2006.

[For07] Per-Erik Forssén, **Maximally Stable Colour Regions for Recognition and Matching**, *IEEE Conference on Computer Vision and Pattern Recognition*, 2007

[Fra04] F. Fraundorfer, H. Bischof, **Evaluation of Local Detectors on Non-Planar Scene**, *28th OAGM/AAPR Workshop*, pp. 125-132, June 2004.

[Fra05] F. Fraundorfer, M. Winter, H. Bischof, **MSCC: Maximally Stable Corner Clusters**. *In Proc. 14th Scandinavian Conference on Image Analysis (SCIA 2005)*, pp. 45-54, 2005.

[Fre91] W. T. Freeman, E. H. Adelson, **The Design and Use of Steerable Filters**, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1991.

[Gab05] P. Gabriel, J.-B. Hayet, J. Piater, J. Verly. **Object Tracking Using Color Interest Points,** *in Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance* (AVSS'05), 2005.

[Gar07] V. Garcia and E. Debreuve and M. Barlaud. **Region-of-interest tracking based on keypoint trajectories on a group of pictures**. *In Proceedings of the IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, Bordeaux, France, June 2007.

[Gau97] H. Gauvrit, **extraction multi-pistes: approche probabiliste et approche combinatoire**, Thèse université Rennes 1, 1997.

[GMF4iTV] B. Cardoso and al, **Hyperlinked Video with Moving Objects in Digital Television**, *ICME* 2005.

[Gou00] V. Gouet **Mise en correspondance d'images en couleur - Application à la synthèse de vues intermédiaires**, *Thèse de doctorat*, Université de Montpellier II, Oct. 2000.

[Gou04] V. Gouet and B. Lameyre,**SAP: A robust approach to track objects in video streams with Snakes And Points**, *British Machine Vision Conference (BMVC'04)*, pp. 737-746, Londres, Angleterre, Sept. 2004.

[Gra06] Grabner Michael, Grabner Helmut, Bischof Horst, Fast Approximated SIFT, *in Proc. ACCV 2006, Springer*, LNCS 3851, pp. 918-927, Hyderabad, India, 2006.

[Gro97] P. Gros, G. Mclean, R. Delon, Roger Mohr, C. Schmid, G. Mistler, **Utilisation de la couleur pour l'appariement et l'indexation d'images**, *Technical Report 3269*, INRIA, Sep. 1997.

[Gru00] P. Grünwald, **Model Selection based on Minimum Description Length**, *Journal of mathematical psychology*, Vol. 44-1, pp. 133-152, Orlando, Florida, USA, 2000.

[Gya03] Gyaourova A., Kamath C., and Cheung S.-C., **Block matching for object tracking**, LLNL Technical report. UCRL-TR-200271, October 2003.

[Hal02] D. Hall, B. Leibe, B. Schiele, **Saliency of interest points under scale changes**, *British Machine Vision Conference*, Sep. 2002.

[Har88] C. Harris and M.J. Stephens, **A combined corner and edge detector**, In *Alvey vision conference*, pp.147-152, 1988.

[Har05] K. Hariharakrishnan, D. Schonfeld, P. Raffy, F. Yassa, **Video tracking using block matching**, *ICIP* (3), pp.945-948, 2003.

[Hoe06] J. Hoey**, Tracking using flocks of features, with application to assisted handwashing**, *British Machine Vision Conference 2006*, pp.367-376, Edinburgh, Scotland, August, 2006.

[Hou59] P. V. C. Hough, **Machine analysis of bubble chamber pictures**. *In Int. Conf. on high Energy Accelerators and Instrumentation*, pp. 554-556, CERN, 1959.

[Hu62] M. K. Hu, **Visual pattern recognition by moment invariants**, *IRE Trans. Information Theory*, vol 8, pp179-187, 1962.

[Ill88] J. Illingworth and H. Kittler, **A survey of the Hough transform**, *Computer Vision, Graphics, and Image Processing*, vol.44(1) pp. 87-116, 1988.

[Isa98] M. Isard and A. Blake, **CONDENSATION conditional density propagation for visual tracking,** *International Journal of Computer Vision*, Vol. 29, Issue 1, pp. 5—28, 1998.

[Isa00] M. Isard and J. MacCormick. **BraMBLe: A Bayesian multiple-blob tracker**, ICCV, 2:3--19, 2000.

[Isl05-1] Md. S. Islam, A. Sluzek and L. Zhu, **Towards invariant interest point detection of an object**, *Proc. 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 101-104, Czech Republic, 2005.

[Isl05-2] Md. S. Islam, Z. Lin, **Matching Interest Points of an Object,** *IEEE International Conference on Image Processing*, Volume 1, Page(s):373 – 376, 11-14 Sept. 2005.

[Itt98] L. Itti, C. Koch, E. Niebur, **A Model of Saliency-Based Visual Attention for Rapid Scene Analysis**, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 20, No. 11, pp. 1254-1259, Nov 1998.

[Iva99] Y. Ivanov, C. Stauffer, A. Bobick and W. E. L. Grimson, **"Video surveillance of Interactions"**, *Second IEEE Workshop on Visual Surveillance*, Fort Collins, Colorado, pp. 82-90, 1999.

[Jaf03] Jaffré G., Crouzil A, **Non-rigid object localization from color model using mean shift**, ICIP (3), 317-320, 2003.

[Jay02] C. Jaynes, S. Webb, M. Steele, and Q. Xiong, **An Open Development Environment for Evaluation of Video Surveillance Systems**, *IEEE Workshop on Performance Analysis of Video Surveillance and Tracking (PETS'2002),* In conjunction with ECCV, June 2002.

[Kad01] T. Kadir and M. Brady, **Scale, Saliency and Image Description**, *International Journal of Computer Vision,* 45 (2):83-105, November 2001.

[Kad04] T. Kadir, A. Zisserman and M. Brady, **An Affine Invariant Salient Region Detector**, *European Conference on Computer Vision*, pp 228–241, 2004.

[Ke04] by Y. Ke and R. Sukthankar, **PCA-SIFT: A More Distinctive Representation for Local Image Descriptors**, *Computer Vision and Pattern Recognition*, 2004.

[Koe87] J. J. Koenderink and A. J. van Doom, **Representation of local geometry in the visual system**, Biol. Cybern., vol. 55, no. 6, pp. 367-375, 1987.

[Kol05] M. Kölsch and M. Turk, **Hand Tracking with Flocks of Features**, *In Video Proc. CVPR IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[Kum96] S. Kumar, **Surface triangulation: a survey**, Tech. report TR-96-011, Univ. of North Carolina, Chapel Hill, Dept. of Computer Science, 1996.

[Kum04] S. Kumar and M. Hebert, **Discriminative Fields for Modeling Spatial Dependencies in Natural Images**, *Advances in Neural Information Processing Systems, NIPS 16*, 2004.

[Kum05] S. Kumar and M. Hebert, **A Hierarchical Field Framework for Unified Context-Based Classification**, *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[LASeR] MPEG-4: ISO/IEC FDIS 14496-20:2006(E).

[Laz05] S. Lazebnik, C. Schmid, J. Ponce, **A sparse texture representation using local affine regions**, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 27, No 8, 2005.

[Leo05] M. Leordeanu and M. Hebert, **A Spectral Technique for Correspondence Problems using Pairwise Constraints,** *International Conference of Computer Vision (ICCV)*, October, 2005.

[Lin96] T. lindeberg, **Feature detection with automatic scale selection**, *Technical report ISRN KTH NA/P--96/18--SE. Department of Numerical Analysis and Computing Science*, Royal Institute of Technology, S-100 44 Stockholm, Sweden, May 1996.

[Lou00] E. Loupias, **Indexation d'images: aide au télé-enseignement et similarités pré-attentives**, *Thèse*, 2000.

[Low99] D G. Lowe, **Object recognition from local scale-invariant features,** *International Conference on Computer Vision,* Corfu, Greece, pp. 1150-1157, September 1999.

[Low04] D G. Lowe, **Distinctive image features from scale-invariant keypoints,** *International Journal of Computer Vision,* 60, 2, pp. 91-110, 2004.

[Mal06] E. Malis, E. Marchand, **Experiments with robust estimation techniques in real-time robot vision**, *In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IROS'06, pp. 223-228, Pekin, Chine, Octobre 2006.

[Mat02-1] J. Matas, D. Koubaroulis, and J. Kittler, **The multimodal neighborhood signature for modeling object color appearance and applications in object recognition and image retrieval,** *Computer Vision and Image Understanding*, 88(1):1-23, October 2002.

[Mat02-2] J. Matas, O. Chum, U. Martin, and T. Pajdla, **Robust wide baseline stereo from maximally stable extremal regions,** In Paul L. Rosin and David Marshall, editors, *Proceedings of the British Machine Vision Conference (BMVA)*, volume 1, pages 384-393, London, UK, September 2002.

[May79] PS. Maybeck, **Stochastic models, estimation, and control**, vol. 1, Academic, NewYork, 1979.
Available at: http://www.cs.unc.edu/#welch/kalman/maybeck.html.

[McI00] A. M. McIvor, **Background Subtraction Techniques**, *IVCNZ00*, Hamilton, New Zealand, November 2000.

[Meg01] R. Megret, J.-M. Jolion. **Le suivi de blobs comme base pour la caractérisation du mouvement dans des séquences audiovisuelles**, *CORESA 2001*, Dijon, novembre 2001.

[Meg02] R. Megret, J.-M. Jolion. **Suivi de blobs de niveaux de gris pour la représentation du contenu dynamique d'une vidéo**. *RFIA'02* (Angers), vol. 2, pp. 397--406, Jan. 2002.

[Mey92] F. Meyer, **Colour image segmentation**, *Proc. IEE Int. Conf. on Image Processing and its Applications*, Maastricht, Netherlands, pp. 303-306, 1992.

[Mik01] K. Mikolajczyk, C. Schmid, Indexation à l'aide de points d'intérêt invariants à l'échelle Journées ORASIS GDR-PRC Communication Homme-Machine - May 2001.

[Mik02] K. Mikolajczyk, **Detection of local features invariant to affine transformation**, Ph.D thesis, institut national polytechnique de Grenoble, 2002.

[Mik03] K. Mikolajczyk, C. Schmid, **A performance evaluation of local descriptors**, *International Conference on Computer Vision & Pattern Recognition*, June 2003.

[Mik05-1] K. Mikolajczyk, C. Schmid, **A performance evaluation of local descriptors**, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Volume 27, Number 10, 2005.

[Mik05-2] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F.Schaffalitzky, F. Kadir, L. Van Gool, **A comparison of affine region detectors**, *International Journal of Computer* Vision, Volume 65, Number ½, 2005.

[Min99] F. Mindru, T. Moons, Luc J. Van Gool, **Recognizing Color Patterns Irrespective of Viewpoint and Illumination**, *CVPR,* pp. 1368-1373, 1999.

[Min03] Florica Mindru, Tinne Tuytelaars, Luc Van Gool, Theo Moons, **Moment Invariants for Recognition under Changing Viewpoint and Illumination**, ACM, Jul. 2003.

[Mon98] P. Montesinos, V. Gouet, and R. Deriche, **Differential invariants for color images,** *International conference on pattern recognition*, 1998.

[Mor80] H.P. Moravec, **Obstacle avoidance and navigation in the real world by a seeing robot rover**, Tech. Rept, CMU-RI-TR-3, The Robotic Institute, Carnegie-Mellon University, Pittsburgh, PA, 1980.

[MPEG-7] MPEG-7: ISO/IEC 15938:2001.

[Mur06] E. Murphy-Chutorian and M. Trivedi, **N-tree Disjoint-Set Forests for Maximally Stable Extremal Regions**, *Proc. British Machine Vision Conference (BMVC 2006),* Edinburgh, Scotland, UK, Sept. 2006.

[MXF] Material eXchange Format (MXF): SMPTE 377M.

[Nee03] Chris J. Needham, Roger D. Boyle, **Performance Evaluation Metrics and Statistics for Positional Tracker Evaluation**, *ICVS*, pp. 278-289, 2003.

[porTiVity] Gerhard Stoll and al, **porTiVity: New Rich Media iTV Services for Handheld TV**, *2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies,* London, 2005.

[Pup05] Pupilli, M., and Calway, A., **Real-Time Camera Tracking Using a Particle Filter**, In Proceedings of the British Machine Vision Conference, BMVA Press, 2005.

[Qin05] L. Qin, W. Gao, **Image Matching Based on A Local Invariant Descriptor,** *IEEE International Conference on Image Processing*, Volume 3, 11-14, Page(s):377 – 380, Sept. 2005.

[Ran91] K. Rangarajan, Shah M., **Establishing motion correspondence**, *CVGIP: Image Understanding*, 54:56-73, 1991.

[Rei79] D. B. Reid, **An algorithm for tracking multiple targets**. *IEEE Transactions on Automatic Control*, AC-24(6):843-854, December 1979.

[Ris04] Ristic, Branko; Arulampalam, Sanjeev; Gordon, Neil: **Beyond the Kalman Filter. Particle Filters for Tracking Applications**, *Artech House, London/Boston*, 2004.

[Rou84] P. J. Rousseeuw, **Least Median squares of regression,** *Journal American statistics Association*, vol. 79, pp.871-880, 1984.

[Rou87] P. J. Rousseeuw and A. M. Leroy, **Robust Regression and Outlier Detection**. *John Wiley and Sons*, New York, 1987.

[Sal90] V. Salari, and I.K. Sethi, **Feature Point Correspondence in the Presence of Occlusion**, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp.87-91, Jan. 1990.

[SAMBITS] Healey, P. and al., **Integrating Internet and Digital Video Broadcast Data**, *4th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2000)*, volume I, pages 624– 627, Orlando (USA), July 2000.

[SAVANT] U. Rauschenbach and al, **A Scalable Interactive TV Service Supporting Synchronized Delivery over Broadcast and Broadband Networks**, *Proc. IBC 2004 conference*, Sept. 9-13, 2004, Amsterdam, The Netherlands.

[Sch97] C. Schmid and R. Mohr, **Local Greyvalue Invariants for Image Retrieval,** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[Sch98] C. Schmid, and R. Mohr, Comparing and evaluating interest points, in proceedings of the 6th International Conference on Computer Vision, Bombay, India, pp230-235, janvier 1998.

[Sch02] F. Schaffalitzky, A. Zisserman: **Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?"**, *ECCV* (1), pp 414-431, 2002.

[Sch04] T. Schlögl, C. Beleznai, M. Winter, H. Bischof, **Performances evaluation metrics for motion detection and tracking**, *ICPR04*(IV: 519-522), 2004.

[Seb01] N. Sebe, M.S. Lew, **Salient Points for Content-based Retrieval**, *British Machine Vision Conference* (BMVC'01), pp. 401-410, Manchester, UK, 2001.

[Seb02] N. Sebe, Q. Tian, E. Loupias, M.S. Lew, T.S. Huang, **Evaluation of Salient Point Techniques**, *International Conference on Image and Video Retrieval* (CIVR'02), pp. 367-377, London, UK, July 2002.

[Set87] I.K. Sethi, and R. Jain, **Finding Trajectories of Feature Points in a Monocular Image Sequence**, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 56-73, Jan 1987.

[Shi94] Jianbo Shi and Carlo Tomasi. **Good Features to Track**. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593-600, 1994.

[Shi00] J. Shi, J. Malik, **Normalized Cuts and Image Segmentation**, *IEEE Trans. Pattern Anal. Mach. Intell*. 22(8), pp. 888-905, 2000.

[Sie04] Siebel, N., Maybank, S., In Markus Clabian, Vladimir Smutny and Gerd Stanke, Ruiz-del-Solar, J.; Shats, A.; Verschae, R., **The ADVISOR Visual Surveillance System**, *Proceedings of the ECCV 2004 workshop Applications of Computer Vision (ACV'04)*, Prague, Czech Republic, pp. 103-111, May 2004.

[Ska94] W. Skarbe, and A. Koschan, **Colour Image Segmentation- A Survey**, *Report of Technisher Bericht 94-32, Technical University of Berlin*, October 1994.

[Smi95] S. M. Smith and J.M. Brady, **SUSAN, A new approach to low level image processing**, tech. rept TR95SMS1c, Chertsey, surrey, UK, 1995.

[Sta00] C. Stauffer, E. Grimson, **Learning Patterns of Activity Using Real-Time Tracking**, *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, 22(8):747-757, 2000.

[Ste99] Charles V. Stewart, **Robust Parameter Estimation in Computer Vision,** *SIAM review,* 1999.

[Tec01] Techmer A., **Contour-based motion estimation and object tracking for real-time applications**, In International Conference on Image Processing, volume 3, pages 648--651, Thessaloniki, Greece, 2001.

[Tis04] P. Tissainayagam and D. Suter, **Object tracking in image sequences using point features**, *Pattern Recognition*, vol. 38, Issue 1, pp 105-113, January 2004.

[Tom91] Carlo Tomasi and Takeo Kanade, **Detection and Tracking of Point Features**. *Carnegie Mellon University Technical Report* CMU-CS-91-132, April 1991.

[Tor00] P. H. S. Torr and A. Zisserman, **MLESAC: A new robust estimator with application to estimating image geometry**, *Computer Vision and Image Understanding*, 78, pp 138-156, 2000.

[Tor02] B. Tordoff and D.W. Murray, **Guided sampling and consensus for motion estimation**, *In 7th European Conference on Computer Vision*, vol. I, pp 82-96, Copenhagen, Denmark, May 2002.

[Toy99] K. Toyama, J. Krumm, B. Brumitt and B. Meyers, **Wallower: Principles and practice of background maintenance**, *International Conference on Computer Vision*, pp. 255-261, 1999.

[Tra98] M. Trajkovic and H. Hedley, **Fast corner detection**, *image and vision computing*, 1998, vol. 16, No. 2, pp.75-87.

[Tse03] G. Tsechpenakis, K. Rapatzikos, N. Tsapatsoulis and S. Kollias, **Object tracking in clutter and partial occlusion through rule-driven utilization of Snakes**, Proc of the 2003 International Conference on Multimedia and Expo (ICME '03), Vol. 3, pp. 69-72, 2003.

[Tuy04] T. Tuytelaars and L. Van Gool, ***Matching Widely Separated Views based on Affine Invariant Regions,*** *Int. Journal on Computer Vision*, 59(1), pp. 61-85, 2004.

[Val04] Valette S, Magnin I, Prost R, **Mesh-based video objects tracking combining motion and luminance discontinuities criteria**, *Signal Processing archive*, Vol 84 , pp. 1213-1224, Issue 7, July 2004.

[Vee01] C. J. Veenman, M. J. T. Reinders, and E. Backer, **Resolving motion correspondence for densely moving points**, *IEEE Trans. on PAMI*, vol. 23, no 1, pp 54-72, Jan 2001.

[Vez08] R. Vezzani, R. Cucchiara, ViSOR: **Video Surveillance On-line Repository for Annotation Retrieval**, in press on *Proceedings of IEEE International Conference on Multimedia & Expo (IEEE ICME 2008)*, Hannover, 2008.

[VGal95] L. J. Van Gool, T. Moons, E. J. Pauwels, A. Oosterlinck: **Vision and Lie's approach to invariance**, *Image and Vision Computing*, vol. 13, no 4, pp. 259-277, 1995.

[Wan04] H.Wang and D. Suter, **MDPE: A Very Robust Estimator for Model Fitting and Range Image Segmentation**, *International Journal of Computer Vision (IJCV)*, Vol. 59, No.2, pp. 139-166, 2004.

[Wel04] G. Welch and G. Bishop, **An Introduction to the Kalman filter**. *Technical Report TR 95-041*, University of North Carolina at Chapel Hill, 1995, adapted 2004.

[Wre97] C. Wren, A. Azabayejani, T. Darrell and A. Pentland, **Pfinder: Real-time tracking of the human body**, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 19, pp.780-785, 1997.

[Yil06] Yilmaz, A., Javed, O., and Shah, M, **Object tracking: A survey,** *ACM Comput. Surv,* Vol. 38, Issue 4, Article 13, 2006.

[Zha95] Zhengyou Zhang, **Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting**, *RR-267, Projet ROBOTVIS*, Sophia Antipolis, Octobre 1995.

[Zha02] Hung-Xin Zhao; Yea-Shuan Huang, **Real-time multiple-person tracking system**, *International Conference on Pattern Recognition*, Aug. 2002.

[Zho03] J. Zhong, S. Sclaroff, **Segmenting Foreground Objects from a Dynamic Textured Background via a Robust Kalman Filter**, *ICCV*, pp. 44-50, 2003.

[Zis01] A. Zisserman, F. Schaffalitzky, **Viewpoint Invariant Texture Matching and Wide Baseline Stereo,** Proc. 8th International Conference on Computer Vision, Vancouver, Canada, 2001.

[Zit99] B. Zitova, J. Kautsky, G Peters and J Flusser, **Robust detection of significant points in multiframe images**, *pattern recognition letters*, vol 20, pp199-206, 1999.

10 Publications

Trichet Rémi; Mérialdo Bernard, **Keypoints Labeling for Background Substraction in Tracking Applications**, ICME 2008, IEEE International Conference on Multimedia & Expo, June 2008, Hannover, Germany.

Trichet Rémi; Mérialdo Bernard, **Accelerated Keypoint Extraction**, WIAMIS 2008, 9th International Workshop on Image Analysis for Multimedia Interative Services, May 2008, Klagenfurt, Autria.

Trichet Rémi; Mérialdo Bernard, Tracking **Repositioning Algorithm using Keypoint Labeling**, CBMI 2008, Sixth International Conference on Content-Based Multimedia Indexing, June 2008, London, England.

Neuschmied Helmut; Trichet Rémi, Mérialdo Bernard, **Fast Annotation of Video Objects for Interactive TV,** MM 2007, 15th international ACM conference on multimedia, September 24-29, 2007, Augsburg, Germany.

Trichet Rémi; Mérialdo Bernard, **Probabilistic Matching Algorithm for Keypoint Based Object Tracking using a Delaunay Triangulation,** WIAMIS 2007, 8th International Workshop on Image Analysis for Multimedia Interactive Services, June 6-8, 2007, Santorin, Greece.

Trichet Rémi, Mérialdo Bernard, **Generic Object Tracking for Fast Video Annotation,** VISAPP 2007, 2nd International Conference on Computer Vision Theory and Applications, 8 - 11 March, 2007 Barcelona, Spain.

Trichet Rémi, Mérialdo Bernard, **Fast Video Object Selection for Interactive Television,** ICME 2006, IEEE International Conference on Multimedia & Expo, July 9-16, 2006, Toronto, Canada.