REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DES SCIENCES ET DE LA THECHNOLOGIE HOUARI BOUMEDIENE

FACULTE D'ELECTRONIQUE ET INFORMATIQUE

Mémoire présenté par

REBBACHE Rabah

Pour l'obtention du diplôme

MAGISTERE EN ELECTRONIQUE

Spécialité : Traitement du Signal et des Images

THEME

APPROCHE GEOMETRIQUE POUR LE SUIVI DU MOUVEMENT DU VISAGE

Soutenu publiquement le 08/07/2010, devant le jury composé de :

Mr. Y. SMARA	Professeur à l'USTSB	Président
Mr. A. HOUACINE	Professeur à l'USTSB	Directeur de mémoire
Mr. S. LARABI	Professeur à l'USTHB	Examinateur
Mme. N. ACHOUR	Maître de conférences à l'USTHB	Examinatrice

Remerciements

Je tiens à remercier tous ceux qui m'ont aidé plus spécialement Mr. A. Houacine pour m'avoir encadré avec un intérêt constant.

Je remercie les membres du jury qui ont accepté de juger ce travail.

TABLE DES MATIERES

Table des figures

Introduction générale

La vision par ordinateur se situe au carrefour de nombreuses sciences de l'Ingénieur, telles que les mathématiques fondamentales et appliquées, l'intelligence artificielle, le traitement du signal, l'automatique et l'informatique. De plus, l'expansion des technologies de l'information va amener de nouvelles problématiques, comme par exemple, les problèmes liés à la vidéo-conférence via Internet. Le suivi de visage par vision artificielle est une tâche indispensable dans la conception d'un grand nombre de systèmes et applications dans ce domaine. On assiste aujourd'hui à une explosion des domaines applicatifs dont les plus connus du grand public sont, bien sûr, la création audiovisuelle. Il en existe bien d'autres, comme par exemple, l'aide à la conduite ou la conduite automatique de véhicule, la construction de modèle tridimensionnel comme dans le cas d'images géologiques ou médicales, la détection et la caractérisation du mouvement d'objets ou de véhicules.

Le problème traité dans ce mémoire est le suivi de visages dans des séquences vidéo ainsi que l'estimation de la pose de la tête ou du visage par rapport à un référentiel. L'objectif pratique est de fournir à la machine les données de position nécessaire pour assurer une extraction efficace des informations véhiculées par le visage qui seront exploitées pour la reconstruction 3D, la reconnaissance ou l'authentification…, pour cela différentes approches et techniques utilisées dans le suivi et qui sont jugées intéressantes (en se basant sur la bibliographie) sont étudiées et développées dans ce présent travail. Le suivi dans des séquences d'images au cours du temps inclut typiquement l'association d'objets sur des images consécutives en utilisant des caractéristiques telles que les points, les lignes, les jonctions T, L, X, les blobs, voir des modèles plus complexes tels que des squelettes 3D, des volumes 3D…etc.

Les points d'intérêt sont des points dans l'image qui se distinguent par leurs saillance et qui peuvent être localisés facilement dans des images successives et ne se perdent pas facilement ce qui les rend faciles à détecter et à suivre dans le temps. Une telle caractéristique est très importante dans le suivi du visage puisque ce dernier contient beaucoup de points texturés. La méthode de Lucas, Kanade et Tomasi ou KLT a connu un grand succès et reste parmi les plus robuste jusqu'à nos jours pour cela elle est utilisée dans ce présent travail. Elle est présentée pour la première fois comme une nouvelle technique de mise en correspondance entre deux images (registration). Ils utilisent l'intensité spatiale et le gradient de l'image pour chercher directement la position de l'objet caractéristique. Un modèle de translation est supposé entre deux images successives. Par suite la méthode dite KLT est appliquée pour le suivi d'objets en 1991 par C. Tomasi et T. Kanade.

Dans le but d'utiliser le KLT dans le suivi de visage pour des applications de suivi des composantes faciales ainsi que pour des applications de l'estimation de la pose, nous avons étudié et implémenté cette méthode. Les résultats obtenus dans ce travail sont intéressants et peuvent être exploités dans plusieurs applications telles que le suivi de composantes faciales ou la reconstruction 3D de visages humain.

Ce mémoire se compose de cinq chapitres organisés comme suit :

Le premier chapitre introduit le d'une manière générale le travail de ce mémoire. Il traite les différentes méthodes et approches utilisées dans la vision par ordinateur. Au second chapitre est traité l'estimation de mouvement dans les images vidéo. Les principales méthodes d'estimation de mouvement sont présentées à commencer par le filtrage de Kalman puis le flot optique qui constituent des outils très importants. Cette étape introduit la troisième dimension dans les images qui est le temps. Le troisième chapitre constitue le cœur de ce travail. On en a étudié les différents détecteurs des points caractéristiques parmi eux les plus importants tels que le détecteur de Harris et le KLT. L'algorithme de Lucas et Kanade est étudié dans ce chapitre et implémenté pour le suivi de visage. Dans une application visant à suivre un visage d'une personne se présentant devant une caméra l'algorithme est testé sur un nombre de séquence vidéo puis comparé à une autre méthode qui est le suivi par corrélation. Les résultats sont présentés et commentés. L'estimation de la pose et la calibration de la caméra constituent le sujet du quatrième chapitre. C'est une partie qui se base sur l'estimation de la pose par le calcul de l'homographie entre deux images consécutives d'une séquence vidéo. Avec un certain nombre de couples de points pris sur les deux images on a pu estimer la pose du visage. Les résultats de cette application sont complémentaires à ceux du chapitre précédent. Au dernier chapitre nous avons présenté une méthode qui améliore les résultats du suivi. Basée sur le KLT combinée avec le filtrage de Kalman, elle présente des avantages et des inconvénients. Les différents résultats sont illustrés dans ce chapitre.

Chapitre I

Généralités et état de l’art

I.1 La vision par ordinateur :

La VAO (Vision Assistée par Ordinateur) est un ensemble d'outils qui permet à l'ordinateur d'imiter la perception de l'être humain afin d'extraire des informations d'une image brute pour pouvoir prendre des décisions d'appartenance d'un objet à une certaine classe ou une forme. C'est une branche de l'intelligence artificielle dont le but est de permettre à une machine de comprendre ce qu'elle «voit » lorsqu'on la connecte à une ou plusieurs caméras. La VAO constitue une chaîne de traitements allant de l'acquisition de l'image brute jusqu'à son interprétation par machine. Le schéma de la figure (I.1) montre les étapes essentielles d'un système de VAO :

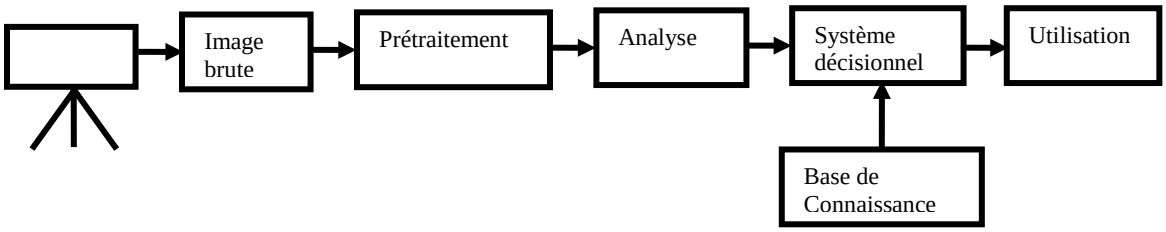


Figure (I.1) : Chaîne de la VAO [17].

Comme on le voit sur la figure ci-dessus, dans la chaîne de la VAO, le traitement d'une image passe par plusieurs étapes avant d'arriver au but final qui consiste à prendre des décisions selon le contexte de l'image.

Le traitement artificiel d'une image passe essentiellement par les étapes suivantes :

Acquisition, Prétraitement, Analyse et Interprétation. Les deux premières étapes sont appelées [17]: Traitement de bas niveau et les deux dernières étapes : Traitement de haut niveau.

Ces traitements se font par ordinateur en essayant de faire une description des objets physiques qui sont dans la réalité.

I.2 Domaines d'application de la VAO :

Les domaines d'application de la VAO se regroupent essentiellement suivant trois catégories : L'imagerie aérienne et spatiale : dans laquelle les traitements concernent l'amélioration des images satellites, l'analyse des ressources terrestres, la cartographie, les analyses météorologiques. Le type d'image utilisé est issu de caméras dans le visible ou l'infrarouge ainsi que le radar. Les technologies biomédicales : dont l'exemple le plus connu utilisant le traitement d'image est le scanner mais on trouve des utilisations de cette technique dans l'échographie, la résonance magnétique nucléaire ainsi que dans le domaine de la

reconnaissance automatique de cellules ou chromosomes. La robotique et la vidéosurveillance: Ces domaines connaissent actuellement un grand développement et les principales taches utilisant de l'imagerie sont l'assemblage (pièces mécaniques, composants électroniques, …), le contrôle de la qualité ainsi que la robotique mobile. Dans la vidéosurveillance on a la détection de mouvement, le suivi et l'identification de personnes.

I.3 Analyse du mouvement de l'homme :

Dans les recherches en biométrie on s'intéresse à la compréhension et à l'interprétation du comportement d'une personne dans des environnements complexes. Dans beaucoup d'applications il est important d'identifier les actions de certaines parties du corps, par exemple ; l'analyse des expressions faciales ou les gestes de la main. De telles applications sont importantes dans les domaines qui se rapportent à la communication entre l'homme et la machine, la sécurité et la biométrie visant l'identification des individus à travers leurs actions. Dans d'autres applications il est nécessaire d'analyser le mouvement de tout le corps. Une telle analyse dite de haut niveau est nécessaire dans l'interprétation des comportements dans des séquences vidéo et a aussi comme application la classification des actions passives et actives. Finalement la modélisation du comportement de l'être humain peut être utile pour les animations et les graphiques 3D ainsi que pour la distinction entre le comportement normal et anormal [1]. La figure (I.2) montre les différents domaines qui se rapportent à l'analyse de l'activité de l'homme.

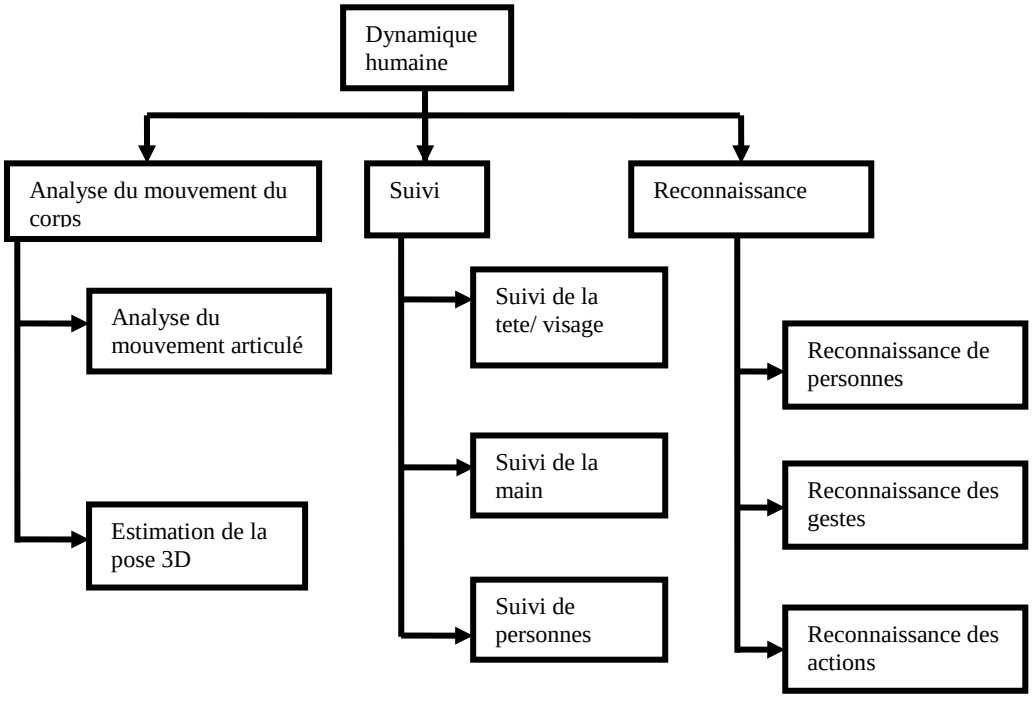


Figure (I.2) : Analyse du mouvement de l'homme [1].

En premier la détection de la présence de personnes sur la scène constitue une étape qui permet de passer à d'autres étapes de plus haut niveau comme l'analyse de la structure, le mouvement du corps et l'estimation de sa vitesse. Par la suite viennent le suivi, la reconnaissance ou l'identification. Toutes ces étapes nécessitent beaucoup d'outils, modélisations mathématiques et différentes méthodes de traitement des images pour arriver à des applications de haut niveau comme cité précédemment la communication entre l'homme et la machine et les différentes applications en biométrie. Du coté matériel le nombre de caméras utilisées est généralement une ou deux caméras. Pour la stéréoscopie, deux Caméras sont utilisées. Les différentes méthodes utilisées pour traiter les images acquises dépendent de l'objectif visé. Dans la stéréoscopie les données acquises sont traitées pour la reconstruction en 3D de l'objet, la reconnaissance ou l'estimation de la pose 3D.

I.3.1 Modélisation du corps humain :

Le corps est non rigide et fortement déformable. La modélisation exacte de sa forme est difficile dans ce contexte. Les approches de suivi temporel basées sur des modèles du corps humain utilisent des connaissances a priori sur la structure du corps humain pour le suivre au cours du temps. Le corps humain est considéré comme un ensemble de parties (tête, avant-bras, bras, mains, torse, cuisses, jambes, pieds) qui peuvent être définies de façon plus ou moins précise. De façon courante, la structure du corps humain est associée aux mouvements du squelette, qui constitue un ensemble de segments reliés par des articulations. Les segments peuvent être estimés en tant que tels, au sens géométrique (1D), en utilisant un modèle simplifié de squelette humain. C'est une estimation linéaire, à une dimension, des segments, où ils sont approchés par des lignes. Il est aussi possible de les estimer par des paramètres 2D, en se basant sur des informations de contour ou de silhouette. C'est alors une estimation surfacique, à deux dimensions, des segments. Dans une dernière approche, il est aussi possible de considérer ces segments au sens volumique, à trois dimensions, et en les approchant par des parallélépipèdes, des cylindres, des blobs 3D ou un autre modèle volumique [26]. Ainsi, les segments du corps humain sont approchés respectivement en tant que lignes, surfaces et volumes qui seront respectivement appelées approches 1D, 2D, ou 3D.

I.3.1.1 approche 1D :

L'essence du mouvement humain est contenue dans les mouvements de la tête, du torse et des quatre membres. La représentation la plus simple d'un corps humain, figure (I.3), consiste en un squelette de personne formé de segments de ligne (stick figure) reliés par des articulations.

Le mouvement des articulations et des segments permet d'estimer le mouvement d'un corps humain dans son ensemble.

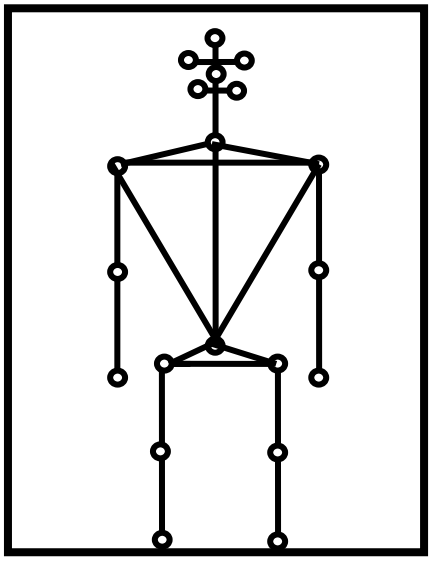


Figure (I.3) : Modélisation 1D du corps humain.

I.3.1.2 Approche 2D :

Ce type de représentation du corps humain est directement lié à la projection du corps dans le plan de l'image. Dans une telle représentation, figure (I.4), les différentes parties du corps peuvent être représentées par des régions définies par des rectangles ou des parallélogrammes (aussi appelés rubans) 2D, par des contours 2D ou par des gabarits (templates). L'avantage de ces approches par rapport aux approches 1D est que l'utilisation de surfaces par rapport à des lignes peut réduire la probabilité de mauvaise association. Il est aussi possible généralement d'extraire un squelette de personne d'un modèle par une approche 2D.

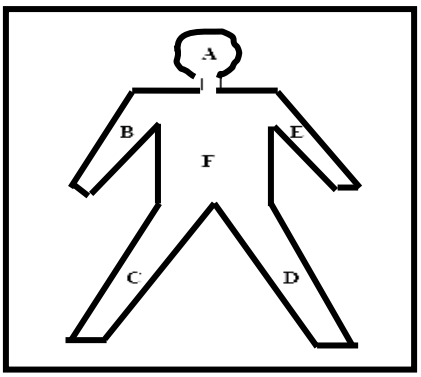


Figure (I.4) : Modélisation 2D du corps humain.

I.3.1.3 Approche 3D :

Le principal inconvénient des modèles 2D est leur restriction suivant l'angle de vue de la caméra, alors de nombreux chercheurs essaient de trouver une structure géométrique du corps humain plus détaillée en utilisant des modèles 3D, figure (I.5), comme des cylindres ou des ellipsoïdes, des cônes tronqués, des sphères, appelées aussi balles. Plus les modèles 3D sont complexes, meilleurs sont les résultats mais ils requièrent plus de paramètres et conduisent souvent à des calculs beaucoup plus coûteux durant le processus d'association du suivi temporel. Il est possible d'utiliser des modèles 3D avec une seule caméra, il faut alors faire correspondre la projection 2D de la personne dans l'image à une configuration du modèle 3D. Mais différentes vues permettent d'améliorer l'analyse, problème bien connu en stéréovision.

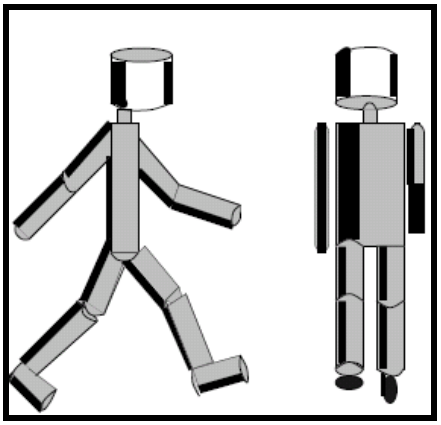


Figure (I.5) : Modélisation 3D du corps humain.

I.4 Introduction au suivi :

Le suivi d'objets en mouvement dans les séquences vidéo est un thème de recherche très étudié en vision par ordinateur. Il consiste en la localisation de cet objet dans chaque image de la vidéo sachant qu'il est en mouvement et que la caméra possède aussi un mouvement, et aussi de réaliser des liens temporels entre les objets détectés à l'instant $t-1$ et les objets détectés à l'instant t. Le suivi dans les séquences d'images au cours du temps inclut typiquement l'association d'objets sur des images consécutives en utilisant des caractéristiques telles que les points, les lignes, les jonctions T, L, X, les blob, voir des modèles plus complexes tels que des squelettes 3D, des volumes 3D etc. Plusieurs facteurs comme le changement du fond de l'image au cours du temps dû par exemple aux mouvements de la caméra, changement de l'illumination de la scène ou l'occultation de l'objet, rendent le problème difficile et lui donnent une certaine variabilité. Le besoin important de systèmes de

vidéosurveillance avancés provient de l'existence même d'endroits nécessitant une sécurité par rapport aux biens ou aux personnes comme les banques, les magasins, les parkings, les frontières etc. Les sorties vidéo des caméras de surveillance dans la plupart de ces endroits sont souvent enregistrées sur bandes vidéo et archivées puis utilisées, si besoin est, "après coup", principalement comme outil d'identification. Le fait que les caméras soient des moyens de traitement temps réel est donc en général peu utilisé. Afin d'avertir aussi vite que possible les personnes ou services concernés (police, pompiers etc.) et faciliter / guider le travail du personnel de surveillance, il est donc nécessaire de développer des systèmes de traitement temps réel [2].

I.4.1 Domaines d'application du suivi :

Dans le domaine de la vidéosurveillance, des algorithmes qui analysent en temps réel les séquences vidéos filmées par une caméra permettent la détection de mouvement, le comptage et le suivi de personnes, l'identification de personnes suspectes (sur base de comportements), l'identification de personnes suspectes (sur base d'une base de données de suspects), ... etc. et d'autres applications comme l'analyse du mouvement de l'être humain et l'analyse des mouvement des athlètes pour l'amélioration de la technique utilisée et pour le meilleur entraînement. La vision par ordinateur a trouvé plusieurs applications dans la réalité virtuelle par exemple le projet de Télémondes [41] lancé en 2001 entre les écoles de GET Son but était d'étudier de nouveaux services de communication interpersonnelle à distance médiatisée par des représentations virtuelles ou ce qu'on appelle avatars. Le tableau (I.1) nous résume quelques applications dans différents domaines :

Vidéosurveillance	- Contrôle d'accès. - Parking, magasins, distributeurs de billets etc. - Personnes âgées.
Interfaces homme-machine	- Reconnaissance des gestes / Langage des signes. - Contrôle guidé par les gestes.
Réalité mixte	- Animation d'avatars. - Jeux vidéo interactifs. - Vidéoconférence.

Tableau (I.1) : Domaine d'application du suivi.

I.4.2 Modélisation de l'arrière plan d'une scène :

Par l'arrière plan on sous-entend la partie statique de la scène qui ne change pas ou qui ne bouge pas à travers la séquence. Seuls les objets en mouvement qui change de position dans la scène. Cependant il existe plusieurs facteurs qui affectent cet arrière plan et le rendent variable dans le temps. La modélisation de l'arrière plan est nécessaire dans plusieurs applications telles que le contrôle continu d'une scène par caméras de surveillance où l'arrière plan doit être séparé des objets à surveiller ou à suivre tels que les êtres humain ou une partie du corps comme la tête et les mains [17]. La modélisation de l'arrière plan d'une scène dépend de l'application visée. Dans le cas où l'arrière plan est fortement affecté par le changement de l'illumination du lieu par exemple les coins non enfermés comme les carrefours ou l'extérieur d'un immeuble, la modélisation statistique doit être utilisée. Cette technique modélise l'arrière plan en séparant la luminance et la chrominance des pixels de l'image. Cela nous mène à classifier les pixels selon l'appartenance à l'arrière plan ou aux objets en mouvement. Avec un seuillage qui peut être obtenu par l'apprentissage statistique, l'arrière plan de la scène peut être séparé efficacement du reste de la scène. Pour les milieux où l'arrière plan est pratiquement le même tout le temps, comme l'intérieur des immeubles qui a des conditions d'illumination inchangées, la méthode la plus simple pour séparer l'arrière plan consiste à soustraire la scène de la scène référence (arrière plan fixe) [13].

I.4.3 Schéma général de suivi d'objets dans les images vidéo :

Le schéma général pour le suivi est donné sous forme d'un organigramme possédant quatre étapes essentielles qui sont : détection- mise en correspondance- mise à jour- prédiction. L'organigramme de la figure (I.6) illustre les différentes étapes du suivi dans le cas général

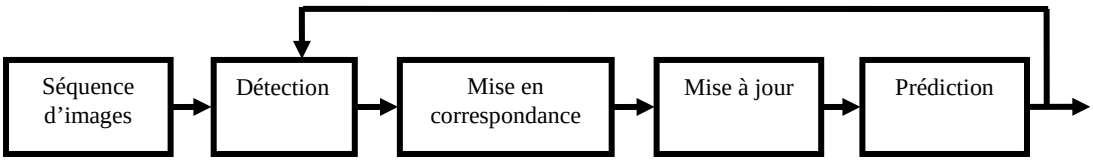


Figure (I.6) : Schéma général du suivi [14].

Dans la première étape, la position de l'objet caractéristique suivi est prédite pour l'image suivante en se basant sur ses positions dans les images précédentes et sur quelques modèles de mouvement. Puis un nombre d'objets caractéristiques sont détectés dans la nouvelle image.

Par la mise en correspondance avec l'objet caractéristique original le meilleur candidat est sélectionné en optimisant certains critères de mise en correspondance [14].

Le bloc 'détection' : Dans cette étape une ou plusieurs parties qui appartiennent à cet objet sont détectées. Cette étape est généralement appelée étape d'initialisation pour la première image de la séquence. Dans les algorithmes qui utilisent une initialisation manuelle la détection implique l'intervention de l'être humain pour la précision dans la localisation des points caractéristiques. Les méthodes utilisées pour la détection sont généralement celles citées ci-dessus comme les détecteurs de points caractéristiques de Harris ou le KLT pour les coins et le Laplacien pour les blobs.

Le bloc 'mise en correspondance' : c'est une étape de recherche du meilleur correspondant à l'objet original. Plusieurs méthodes existent pour mettre en correspondances deux éléments. La mise en correspondance par blocs est parmi les plus connues qui utilisent une recherche de l'élément correspondant à l'original sur une fenêtre de recherche en minimisant une mesure [15]. Différentes mesures sont utilisées telle que la somme des différences absolues ou la distance de Mahalanobis. On a aussi les méthodes de classification comme le maximum de vraisemblance et les K plus proches voisins.

Le bloc 'prédiction' : la prédiction de la prochaine position de l'objet est faite à base des connaissances a priori sur la position et la vitesse de l'objet dans l'image précédente ou dans les deux images précédentes. Une simple extrapolation des vitesses de l'objet dans les deux avant dernières images peut nous aider à prédire la position et la vitesse de l'objet dans l'image courante. La prédiction nous aide énormément à réduire l'espace de recherche et par conséquent réduire le temps de traitement.

I.5 Techniques de suivi :

Plusieurs travaux de recherche ont été fait dans ce domaine. Selon l'objet qu'on veut suivre, (rigide ou flexible, mouvement dans un plan ou 3D, mouvements de rotations), plusieurs approches en étés développées pour le suivi basées sur des connaissances a priori sur l'objet et sur son mouvement. En général il existe deux approches de suivi [3] :

1. Suivi basé sur l'estimation du mouvement de l'objet.
2. Suivi basé sur la reconnaissance de l'objet.

Pour le suivi de cibles qui se déplacent très rapidement (ex : missile, avions de chasse etc.) des systèmes de la prédiction et l'estimation du mouvement de la cible ont étés développés comme le filtre de Kalman, le filtre de Kalman étendu, les filtres particulaires… etc.

Le suivi du corps humain et ses différentes parties qui le composent dans une séquence d'images suscite beaucoup d'intérêt actuellement. Dans les applications qui nécessitent une interface homme-machine par le biais du traitement d'une séquence d'images, nous cherchons à obtenir des informations sur les personnes présentes dans la scène. Ces informations peuvent décrire chaque individu dans son ensemble: silhouette, trajectoire, posture, etc. mais peuvent aussi décrire des parties plus précises du corps : localisations et trajectoires du visage et des mains, estimation de la direction du regard etc. Détecter et suivre le visage et les mains d'un individu placé devant une caméra sont des fonctionnalités indispensables des interfaces homme-machine avancées. C'est en effet une première étape pour l'analyse et l'interprétation des gestes et des actions d'un être humain [2].

On relève de nombreuses techniques de suivi de visage en vision par ordinateur, chacune adoptant une stratégie différente.

I.5.1 Suivi par modèle de couleurs (suivi basé couleurs) :

De nombreux travaux utilisent la couleur de la peau de l'être humain comme l'information discriminante pour effectuer le suivi et la détection de visage [4], [5], [6] de main ou de personnes [2], [7]. La couleur de la peau forme une distribution compacte dans certains espaces couleur [2]. L'utilisation de la couleur est efficace lorsque l'espace couleur réalise une bonne séparation entre la chrominance et la luminance des couleurs de l'image originale [6]. La couleur de la peau se regroupe dans de petits volumes dans l'espace RGB. Une façon pour exploiter la couleur dans le suivi est l'analyse de l'histogramme des images couleurs qui représente la distribution des pixels de la peau de visage. Certaines transformations donne des espaces qui enlèvent la composante de la luminance comme l'espace normalisé NNC (Normalised Color Components). L'histogramme couleur des objets reste invariable à l'occultation [8]. Les statistiques faites sur la couleur de peau montrent que la meilleure modélisation de la couleur de la peau c'est bien la distribution Gaussienne ou par mélange de Gaussiennes. Les avantages que possède l'utilisation de la couleur sont la rapidité des calculs et la non sensibilité de la couleur à certaines orientations et à l'échelle. D'un autre coté le suivi de visages humains en utilisant la couleur rencontre de nombreuses difficultés. La couleur de la peau de visage humain est très variable est possède une gamme très large et diffère d'une personne à une autre. De même que la couleur obtenue par la caméra est sensible aux conditions d'illumination. Ajoutons à cela que la couleur obtenue pour un même objet et sous les mêmes conditions d'illumination est différente d'une caméra à une autre [8].

I.5.2 Suivi par contours actifs :

Les contours actifs ou snackes ont été proposés pour la première fois par par Kass, Witkins et Terzopoulos en 1987 pour la segmentation et le suivi de régions par la modélisation de l'objet par ses contours extérieurs qui sont relativement insensibles aux variations de la luminance. Ce sont des contours fermés dont la forme et la position peuvent évoluer d'un état initial vers un état final. Ils ont été utilisés à des fins de segmentation spatiale d'objets, mais aussi pour effectuer un suivi temporel. Leur principe consiste à définir une courbe paramétrique fermée qui représentera le contour de l'objet à segmenter. Cette courbe est définie par ses coordonnées cartésiennes x et y en fonction de l'abscisse curviligne s qui évolue suivant la minimisation d'une fonctionnelle F faite de deux termes d'énergie. L'initialisation se fait souvent manuellement par l'utilisateur en traçant une courbe au voisinage de l'objet. Les contours actifs ont été utilisés pour suivre un visage dans une séquence d'image vidéo [9].

I.5.3 Suivi basé modèle de forme :

L'analyse de forme des objets est un élément fondamental dans la perception et la reconnaissance dans la vision par ordinateur. La forme d'un objet est un ensemble d'informations géométriques qui reste invariant par rapport à la translation, rotation ou l'échelle de l'objet [3].

I.5.3.1 Modèles actifs de forme (ASM) :

Les modèles actifs de forme sont proposés pour la première fois par Cootes [10]. La technique de l'ASM (Active Shape Modèle) classique consiste à modéliser une classe d'objets à l'aide d'un ensemble de points de repère constituant des caractéristiques communes à toutes les instances de la classe. Pour modéliser un visage humain, on place généralement des points de repère au niveau des yeux, du nez, de la bouche et du contour du visage (le long de la mâchoire inférieure). On associe à cet ensemble de points de repère un vecteur de paramètres de forme représentant les variations de chaque individu par rapport au modèle. Le fait d'agir sur ces paramètres a pour effet de déformer le modèle original et permet de traduire les différentes formes qu'il peut prendre dans la réalité. Lors de l'analyse d'une image pour identifier la forme d'un visage, on commence par estimer la position des points de repère sur l'image. On déforme ensuite le modèle en faisant varier les paramètres de forme pour tenter de faire coïncider ses points de repère avec ceux de l'image analysée.

I.5.3.2 Modèle actif d'apparence (AAM) :

Un autre modèle similaire à l'ASM a été développé par Cootes qui est le Modèle active apparence (AAM) qui décrit un objet d'une classe prédéfinie comme étant une forme et une texture. Chaque objet, pour une classe donnée, peut être représenté par sa forme, à savoir un ensemble de coordonnées 2D d'un nombre fixé de points d'intérêt, et une texture, à savoir l'ensemble des pixels inclus dans l'enveloppe convexe de la forme. Les approches basées modèle de forme pour le suivi de visage peuvent utiliser le visage ou la tête complètement ou seulement les caractéristiques faciales [16].

Récemment des modèles 3D de visage sont construits à partir d'un faible nombre de couples de points 2D-3D en correspondance. Il existe deux grandes classes de techniques d'extraction de modèle 3D : la première classe nécessite un minimum de deux images pour faire un calcul de triangulation (maillage) et retrouver les positions 3D des points utilisés lors de la triangulation la seconde classe nécessite la connaissance d'un modèle 3D qu'il faut positionner. Le modèle d'apparence du visage comprend deux composantes : Le modèle de forme contenant un modèle paramétrique 3D du visage et le modèle de texture faciale. A partir de ces deux composantes, une apparence du visage peut être reconstituée. Ces deux approches sont coûteuses en temps de calcul et nécessitent pour être robustes de disposer d'un grand nombre de points 2D.

I.5.4 Suivi par modèle paramétrique de mouvement :

Le mouvement d'une large classe d'objets peut être décrit par un modèle paramétrique de mouvement [27]. Plusieurs modèles peuvent être posés dans le but d'estimer le mouvement et suivre un objet. Parmi les modèles qu'on trouve on a ; modèle de translation pure, translation et rotation, modèle affine et un modèle quadratique qui peut inclure tous les modèles cités précédemment. Le problème de suivi est équivalent à l'estimation des paramètres du modèle posé. Les paramètres de ce mouvement sont calculés sur la base de tous les pixels inclus dans la région à suivre. Dans le travail de [27] on considère un ensemble de point inclus dans la région dans l'image référence on considère les valeurs radiométrique des pixels puis une minimisation d'une somme de différences carrées (SSD) entre deux régions, l'une étant la région de l'image référence et l'autre étant la même région mais dans l'image suivante. Un travail similaire est celui de [38]. Un modèle quadratique de mouvement puis le suivi se fait sur les contours de la région à suivre.

Cette approche permet le suivi global d'un objet déterminé sur la première image, de manière assez robuste, grâce au nombre peu élevé de paramètres. Les objets déformables sont par

contre mal décrits par le modèle trop simple, ce qui conduit à un suivi approximatif de l'étendue de l'objet.

I.4.4.4 Mise en correspondance par la corrélation :

La mise en correspondance est une idée très simple. On cherche à mettre en correspondance une sous-image modèle contenant la forme de l'objet qui est supposé présent sur notre image. On centre le modèle sur un point de l'image puis on calcule une différence entre les deux images. On répète le processus pour tous les points de l'image, la position de l'objet est donnée pour une différence minimale.

La mise en correspondance peut être considérée comme une méthode d'estimation de paramètres. Les différents paramètres définissent la position et la pose de l'objet. On définit une fenêtre contenant l'objet. Supposant que chaque pixel dans l'image est affecté par un bruit additif gaussien. La probabilité pour que les pixels d'une fenêtre correspondent aux pixels de l'image est donnée par une distribution normale. On a alors à choisir une fenêtre pour laquelle l'erreur carrée entre ses pixels et leurs correspondants dans l'image est minimale. Une autre formulation consiste à considérer la maximisation de l'intercorrélation.

Conclusion

Nous avons présenté dans ce chapitre les différentes méthodes rencontrées dans la littérature relative au domaine du suivi. Ces méthodes utilisent différentes techniques utilisant les contours, la couleur ou les points saillants de l'image. Les points saillants sont étudiés dans le chapitre suivant, ils constituent l'élément de base de la méthode développée dans ce travail.

Chapitre II

Techniques d’estimation du mouvement

II.1 Le filtrage de Kalman :

Le filtrage consiste à estimer l'état d'un système dynamique, c'est-à-dire évoluant au cours du temps, à partir d'observations partielles, généralement bruitées. Les applications du filtre de Kalman sont nombreuses dans les métiers de l'ingénieur. Le filtre de Kalman permettant de donner un estimé de l'état de système à partir d'une information a priori sur l'évolution de cet état (modèle) et de mesures réelles.

II.1.1 Introduction

Une idée centrale dans le filtre de Kalman est de modéliser le système étudié comme un système dynamique linéaire affecté par des bruits, les capteurs du système sont également soumis à des bruits. En disposant d'une information statistique sur la nature du bruit (ses premiers ordres statistiques), il est possible de construire une estimation optimale de l'état du système bien que les capteurs soient imparfaits. C'est l'idée fondamentale de la théorie de l'estimation. Sans connaître les erreurs elles-mêmes, la connaissance de leurs statistiques permet la construction des estimateurs utiles en se basant seulement sur cette information. La méthode de Kalman est une procédure d'estimation dynamique des paramètres qui sont fonctions du temps.

Les applications du filtre de Kalman sont nombreuses dans les métiers de l'ingénieur. Le filtre de Kalman permet de donner un estimé de l'état de système à partir d'une information a priori sur l'évolution de cet état (modèle) et de mesures réelles, il est utilisé pour estimer des conditions initiales inconnues (balistique), prédire des trajectoires de mobiles trajectographie), localiser un engin (navigation, radar,...) et également pour implanter des lois de commande fondées sur un estimateur de l'état et un retour d'état (Commande Linéaire Quadratique Gaussienne). Les bases de traitement de signal sur lesquelles repose le filtre de Kalman seront également utiles à tout ingénieur confronté à des problèmes de définition de protocoles d'essais, de dépouillements d'essais et également d'identification paramétrique, c'est-à-dire la détermination expérimentale de certains paramètres du modèle.

II.1.2 Le filtre de Kalman linéaire discret :

Le système d'équations utilisé dans le filtre de Kalman repose sur la définition de deux modèles que sont le processus et la mesure (les modèles sont la représentation d'états d'un système dynamique). Il est utilisé pour estimer l'état $x \in \Re^n$ d'un système dynamique observé. Le modèle

du processus qui décrit l'évolution de ce système dynamique est défini par l'équation d'état linéaire récursive suivante:

$$x_k = A_{k-1}.x_{k-1} + B_{k-1}.u_{k-1} + w_{k-1} \quad (2.1)$$

où,

$x_k, x_{k-1} \in \Re^n$, les vecteurs d'état aux instants k et $k-1$

$A_k \in M_{(n\times n)}(\Re)$, est la matrice dynamique du système (la matrice de transition de k à $k+1$), c'est une matrice qui fait le lien entre les paramètres du système à deux étapes successives.

$U_k \in \Re^m$, est le vecteur de commande, (vecteur d'entrée),

$B_k \in M_{(m\times m)}(\Re)$, est la matrice de commande (matrice d'entrée) qui représente la distribution de l'entrée (vecteur de commande) dans le vecteur d'état. Elle fait le lien entre les valeurs optionnelles de contrôle et l'état du système.

$W_k \in \Re^n$, est le bruit d'état.

Le modèle de mesure décrit l'information fournie par les capteurs en une équation liant les paramètres de l'état de la mesure et du bruit. L'équation de mesure ou d'observation est donnée par :

$$z_k = H.x_k + v_k \quad (2.2)$$

Avec,

$z_k \in \Re^p$, est la mesure à l'instant k,

$H_k \in M_{(p\times n)}(\Re)$, est la matrice d'observation, c'est la matrice qui fait le lien entre les paramètres du système et les mesures,

$v_k \in \Re^n$, est le bruit de mesure.

Les variables aléatoires w et v représente respectivement le bruit du processus et de la mesure. Ces deux bruits sont supposés être indépendants l'un de l'autre (non corrélés) et blancs [21] avec des distributions gaussiennes connues a priori (doivent être estimées à l'avance), et indépendantes de l'état initial du système. Cette indépendance des bruits permet de simplifier le formalisme des équations d'évolution et d'observation, donc :

$$p(w) \sim N(0, Q_k)$$
$$p(v) \sim N(0, R_k)$$
$$E\left[wv^T\right] = 0 \quad (2.3)$$

Où, Q_k et R_k sont les matrices de covariances de processus et des mesures respectivement à l'instant k.

Après l'initialisation du système il y a deux étapes principales pour l'estimation des paramètres du système : l'estimation a priori (la prédiction) et l'estimation a posteriori (la correction). Les paramètres du système sont estimés premièrement à partir des valeurs de l'étape précédente et ensuite ils sont corrigés par des mesures dans l'étape de correction.

On définit $\hat{x}_k^- \in \Re^n$ une estimation a priori des paramètres du système à l'instant k, c'est une prédiction basée sur le modèle déterministe d'évolution du système à partir de l'instant $k-1$, et $\hat{x}_k \in \Re^n$ l'estimation a posteriori de l'état du système à l'instant k sachant que la mesure z_k à l'instant k est disponible. Pour dériver les équations du filtre de Kalman, on commence à trouver une équation qui exprime l'estimation a posteriori $\hat{x}_k$ sous forme de combinaison linéaire de l'estimation à priori $\hat{x}_k^-$ et de la différence pondérée entre la mesure z_k et la mesure prédite $H\hat{x}_k^-$ comme suit ;

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.4}$$

La différence $(z_k - H\hat{x}_k^-)$ dans (2.4) est appelée innovation de la mesure ou résiduel. Le résiduel (innovation) reflète l'écart entre la mesure prédite et la mesure réelle. Un résiduel de zéro, signifie que les deux mesures sont équivalentes.

Le gain K_k est une matrice de $n \times m$ éléments choisi de façon à minimiser la covariance de l'erreur a posteriori P_k donnée par ;

$$P_k = E[e_k e_k^T] \tag{2.5}$$

$$P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \tag{2.6}$$

Pour trouver K_k qui minimise la covariance a posteriori de l'erreur a posteriori, on doit remplacer l'expression de $\hat{x}_k$ donner par (2.4) dans (2.6) puis considérer la dérivée de (2.6) par rapport à K_k égale à zéro et résoudre par rapport à K_k. L'expression de K_k est donnée par l'équation suivante :

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{2.7}$$

Où ; P_k^- est la covariance a priori de l'erreur a priori, elle est donnée par ;

$$P_k^- = AP_k A^T + Q \tag{2.8}$$

II.1.2.1 Algorithme du filtre de Kalman :

Le filtre de Kalman, après initialisation, possède deux étapes qui sont la prédiction et la correction.

- Prédiction temporelle

Les différentes équations de cette étape sont ;

$$\hat{x}_k^- = A\hat{x}_k + BU_{k-1}$$

$$P_k^- = AP_{k-1}A^T + Q$$

- Correction et mi à jour

Les différentes équations de cette étape sont ;

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H)P_k^-$$

Un schéma complet du filtre de Kalman qui tient est donné par la figure (II.1) ;

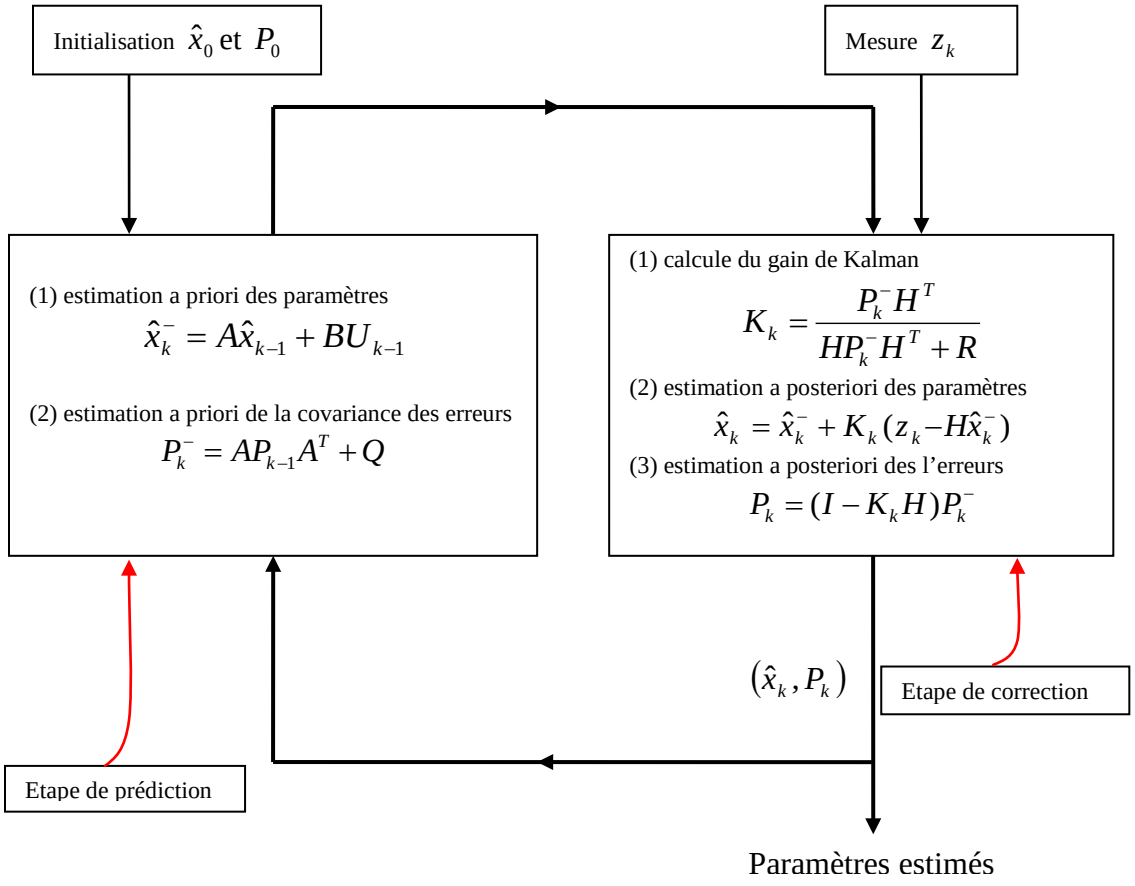


Figure (II.1) : Le cycle complet du filtre de Kalman. Les équations de prédiction, de mesure et correction du filtre de Kalman.

II.1.3 Le filtre de Kalman étendu (EKF) :

Le filtre de Kalman tel que présenté à la section précédente permet d'estimer dans le temps l'état d'un procédé défini par une équation linéaire. Cependant, dans la réalité, l'hypothèse de linéarité d'un procédé ne peut pas toujours être utilisée. Le filtre de Kalman ne peut donc être utilisé. Cependant, permet d'ajuster le filtre pour un procédé avec une équation non linéaire pour linéariser les variables dont le filtre a besoin. L'idée est de linéariser localement sur la moyenne et la variance de la sortie du procédé. Avec les séries de Taylor, une linéarisation de l'estimation peut être effectuée en utilisant les dérivées partielles des fonctions du procédé f et de la mesure h. Cette linéarisation utilise l'hypothèse que l'erreur sur la dérivée est petite, ainsi, la série de Taylor est tronquée dès le premier ordre [21]. Pour cela on commence d'abord par quelques modifications des formules présentées précédemment pour le filtre de Kalman linéaire.

On a toujours $x_k \in \Re^n$ mais dans ce cas le système est gouverné par une équation différentielle stochastique non linéaire :

$$x_k = f(x_{k-1}, u_{k-1,} w_{k-1}) \tag{2.9}$$

Avec une mesure $z_k \in \Re^m$ telle que

$$z_k = (x_k, v_k) \tag{2.10}$$

Où les variables aléatoires $W_k \in \Re^n$ et $v_k \in \Re^n$ représentent le bruit d'état du système et le bruit de la mesure. On peut faire une approximation de l'état du système et la mesure sans les bruits W et v à chaque étape par :

$$\hat{x}_k = f(\hat{x}_{k-1}^-, u_{k-1,} 0) \tag{2.11}$$

et

$$\hat{z}_k = h(\hat{x}_k, 0) \tag{2.12}$$

où $\hat{x}_k^-$ et est une estimation a posteriori de l'état du système.

Pour linéariser le procédé estimé selon les équations (2.11) et (2.12), de nouvelles équations sont développées à partir des deux dernières:

$$x_k = \hat{x}_k + A(x_{k-1} - \hat{x}_{k-1}^-) + Ww_{k-1} \tag{2.13}$$

$$z_k = \hat{z}_k + H(x_k - \hat{x}_k) + Vv_k \tag{2.14}$$

où

- x_k et z_k sont le vecteur d'état et la mesure,
- $\hat{x}_k$ et $\hat{z}_k$ sont le vecteur d'état et la mesure approximés,
- $\hat{x}_k^-$ est une estimation apostériori à l'étape k,

- A est une matrice jacobienne des dérivées partielles de f par rapport à x, telle que

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0),$$

- W est une matrice jacobienne des dérivées partielles de f par rapport à w, telle que

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0),$$

- H est une matrice jacobienne des dérivées partielles de h f par rapport à x telle que

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\hat{x}_k, 0),$$

- V est une matrice jacobienne des dérivées partielles de h f par rapport à v telle que

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\hat{x}_k, 0),$$

II.1.3.1 Algorithme du filtre de Kalman étendu :

Les étapes du filtre de Kalman étendu sont les mêmes que pour le filtre de Kalman linéaire. Deux étapes qui sont la prédiction et la correction après l'initialisation sont nécessaires.

- Prédiction temporelle

Les différentes équations de cette étape sont ;

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1,}, 0)$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_k W_k^T$$

- Correction et mise à jour

Les différentes équations de cette étape sont ;

$$K_k = \frac{P_k^- H_k^T}{H_k P_k^- H_k^T + V_k R_k V_k^T}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0))$$

$$P_k = (I - K_k H_k) P_k^-$$

Un schéma complet du filtre de Kalman est donné à la figure (II.2);

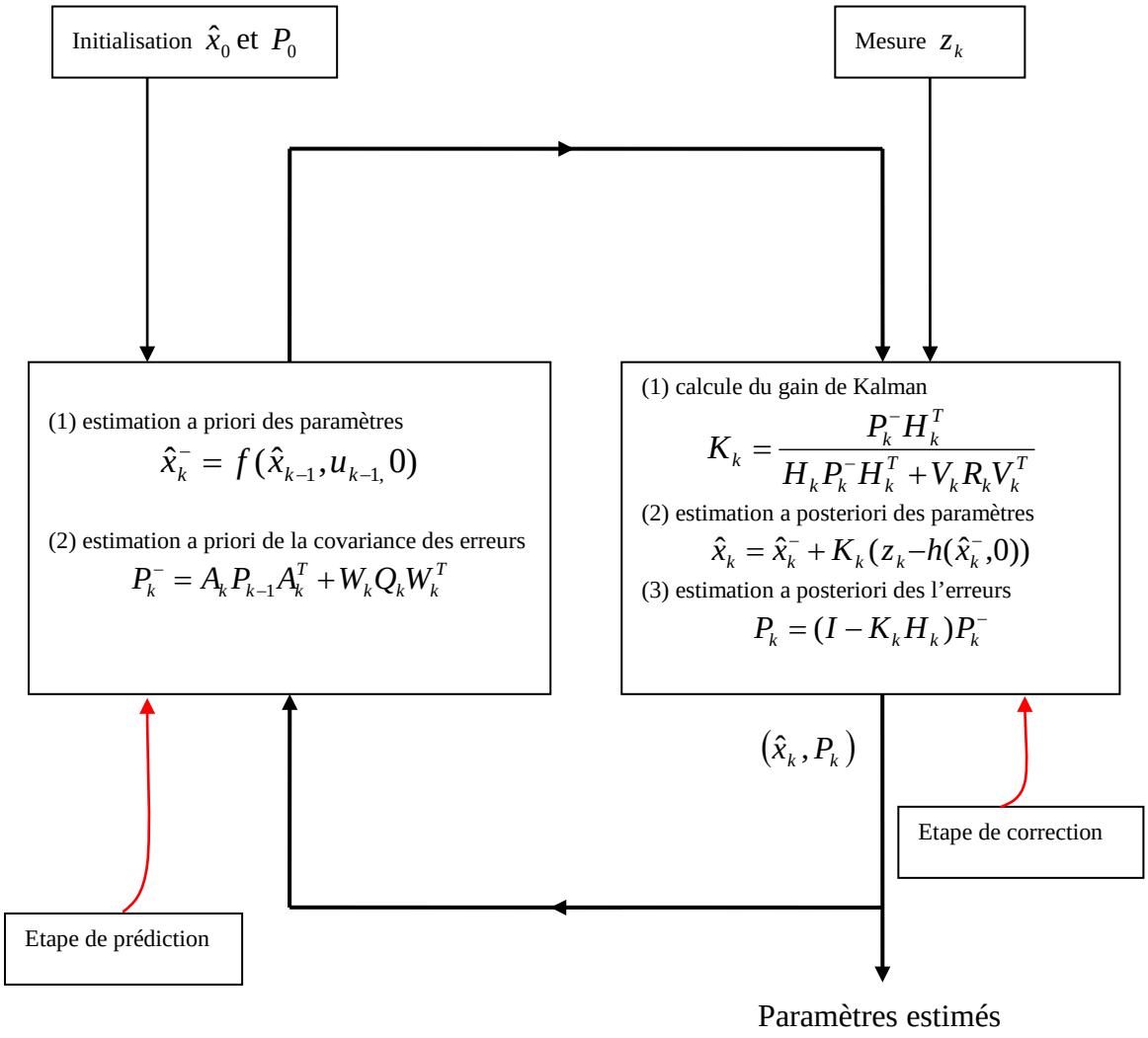


Figure (II.2) : Le cycle complet du filtre EKF. Les équations de prédiction, de mesure et correction du EKF.

II.2 Techniques d'estimation locale du mouvement :

L'estimation du mouvement est un problème fondamental pour l'analyse de séquences d'images. Il consiste à mesurer la projection 2D dans le plan de l'image d'un mouvement réel 3D, dû à la fois au mouvement des objets dans la scène et aux déplacements de la caméra. Une séquence d'images peut être représentée par sa fonction de luminance $I(x, y, t)$. L'hypothèse de conservation de la luminance stipule que la luminance d'un point physique de la séquence d'image ne varie pas au cours du temps [28], c'est à dire :

$$I(P,t) = I(P + v(P)\delta.t, t + \delta.t) \qquad (2.15)$$

Avec $P = (x, y)^T$ et $v(P) = (u, v)^T$ le vecteur vitesse associé au point p et au temps t. Les composantes u et v sont respectivement la vitesse selon x et selon y. Cette hypothèse n'est pas respectée dans le cas d'occultations, de transparences, de réflexions spéculaires, et plus généralement de tout ce qui peut produire des variations brutales de l'illumination de la scène (flash d'appareil photo, lumières clignotantes,. . .). Pour prendre en compte ces phénomènes, l'hypothèse de conservation de la luminance peut être enrichie de manière à autoriser les variations de luminances et de contrastes :

$$I(P,t) = c(P,t)I(P + v(P)\delta.t, t + \delta.t) + b(P,t) \qquad (2.16)$$

où $c(P,t)$ et $b(P,t)$ représentent les variations multiplicatives et additives de la luminance et sont deux nouvelles inconnues à déterminer. Black et. al. [7] ont généralisé l'équation (2.16) de manière à prendre en compte les variations dues à la fois au mouvement, aux variations de luminance et de contraste et à l'apparition soudaine de nouvelles régions dans la scène (par exemple sur un visage, les paupières qui s'ouvrent font apparaître les yeux). L'utilisation d'une modélisation complexe de la variation de luminance nécessite d'estimer un grand nombre de paramètres et conduit à des solutions peu stables. En pratique l'hypothèse de la conservation de la luminance sous sa forme simple (2.15) est la plus utilisée et permet d'obtenir les meilleurs résultats.

II.2.1 Le mouvement par différence d'images :

On se place dans le cas ou la caméra est statique et la scène contient un ou plusieurs objets en mouvement, figure (II.3). Ces méthodes ne donnent pas une mesure du mouvement au sens de vecteurs de vitesse et de déplacement des objets, mais permettent de recueillir des indices sur les endroits de l'image où il y a eu un déplacement : $D(t) = P(t) - P(t-1)$

(a) Image 1 **(b) Image 2** **(c) Image Différence**

Figure (II.3) : Détection de mouvement par différence d'image [17].

Le principe est de calculer la différence entre deux images. Lorsqu'il n'y a pas recouvrement entre les deux positions, l'information du signe de la différence permet de séparer les objets. Par contre, s'il y a recouvrement, seules les zones en périphérie de l'objet permettent de détecter le mouvement. Dans ce cas, la différence apporte un indice visuel qu'il faut coupler avec d'autres informations pour reconstruire la scène complète.

II.2.2 Flot optique :

Le calcul du flot optique consiste à extraire un champ de vitesse dense à partir d'une séquence d'images, typiquement en faisant l'hypothèse que l'intensité (ou la couleur) est conservée au cours du déplacement. De nombreuses techniques ont été proposées pour le calcul du flot optique parmi ces techniques :

1. Les méthodes basées sur la corrélation, qui consistent à chercher une correspondance entre des fenêtres d'images de deux instants consécutifs.
2. Les méthodes différentielles, qui calculent la vitesse à partir des dérivées de l'intensité dans l'image ou en utilisant des images préfiltrées.

II.2.2.1 Flot optique par corrélation :

Quand une scène est filmée à des instants différents, on obtient son image. Si une partie de cette image n'est pas occultée, elle pourrait être reliée entre les deux images et le mouvement peut être exprimé sous forme d'un déplacement dans l'image. Ce déplacement correspond à la projection du mouvement de l'objet sur la scène. Le flot optique est vu comme la mesure de la vitesse où le mouvement est donné en pixels/image. Le flot optique peut être retrouvé par la mise en correspondance entre les points caractéristiques dans les images. On peut prendre un point caractéristique comme un seul pixel ou une fenêtre de pixels comprenant des textures complexes. La mise en correspondance dans les images est développée par plusieurs techniques selon le point caractéristique considéré et selon aussi la méthode de recherche du point correspondant. Si le point caractéristique est pris comme un seul pixel, le pixel correspondant dans l'image suivante peut être retrouvé par la recherche dans son voisinage (dans l'image suivante) d'un pixel portant la même valeur radiométrique. En général, si les pixels ne sont pas dans une région occultée ils peuvent être reliés par la relation suivante :

$$P(t+1)_{x+\delta x, y+\delta y} = P(t)_{x,y} + H(t)_{x,y} \qquad (2.17)$$

où la fonction $H(t)_{x,y}$ représente une compensation de la différence d'intensité entre les deux images, et $(\delta x, \delta y)$ défini le déplacement des pixels entre les images au temps $(t+1)$. Cela se traduit par : l'intensité du pixel dans l'image $(t+1)$ à la position $(x+\delta x, y+\delta y)$ est égale à l'intensité de ce pixel dans l'image t à la position (x, y) plus un certain changement dû aux conditions d'illumination de la scène ainsi que d'autres facteurs physiques il est représenté par $H(t)_{x,y}$. En général il difficile de modéliser les différents bruits affectant le pixel donc l'équation (2.17) est simplifier en supposant que :

$H(t)_{x,y} \approx 0$. Ce qui donne :

$$P(t+1)_{x+\delta x, y+\delta y} = P(t)_{x,y} \tag{2.18}$$

Dans la pratique un seul pixel ne suffit pas pour exprimer le mouvement dans l'image puisqu'il est vulnérable aux bruits. Pour cela on supposera que le voisinage de ce pixel se déplace avec une même vélocité. Donc l'objectif est de trouver le vecteur $(\delta x, \delta y)$ qui minimise l'erreur suivante :

$$e_{x,y} = S\left(P(t+1)_{x+\delta x, y+\delta y}, P(t)_{x,y}\right) \tag{2.19}$$

où S est une fonction qui mesure la similarité entre les pixels. La minimisation de l'erreur quadratique est donnée par :

$$e_{x,y} = \left(P(t+1)_{x+\delta x, y+\delta y} - P(t)_{x,y}\right)^2 \tag{2.20}$$

ou pour une fenêtre de pixels par :

$$e_{x,y} = \sum_{(x',y') \in W} \left(P(t+1)_{x'+\delta x, y'+\delta y} - P(t)_{x',y'}\right)^2 \tag{2.21}$$

Le déplacement est donné pour un minimum de la fonction $e_{x,y}$. Pour le choix de la taille de fenêtre des points caractéristiques il faut réaliser un compromis entre une grande taille pour minimiser et filtrer les bruits des images et une petite taille pour réduire le temps de traitement.

Algorithme :

- **Flot optique par corrélation**
- **d** déplacement maximum
- **w** la taille de la fenêtre
- Charger deux images
- Pour **X1** allant de **w+d+1** jusqu'à **Nlines- w+d+1** faire
- Pour **X1** allant de **w+d+1** jusqu'à **NColonnes- w+d+1** faire
    - Pour **X2** allant de **X1-d** jusqu'à **X1+d** faire
    - Pour **Y2** allant de **Y1-d** jusqu'à **Y1+d** faire

  Chercher le minimum donné par équation (2.21)

  Fin Pour (**X2, Y2**)

Fin Pour (**X1, Y1**)

II.2.2.2 Flot optique par méthode différentielle :

Une autre manière d'estimer le mouvement dans les images est d'utiliser la méthode différentielle. En considère le développement de l'équation (2.17). On suppose que l'intensité lumineuse d'un point dans la nouvelle position est la même intensité que l'ancienne.

En utilisant le développement en série de Taylor de $P(t+\delta t)_{x+\delta x,y+\delta y}$ on obtient :

$$P(t+\delta t)_{x+\delta x,y,+\delta y} = P(t)_{x,y} + \delta x \frac{\partial P(t)_{x,y}}{\partial x} + \delta y \frac{\partial P(t)_{x,y}}{\partial y} + \delta t \frac{\partial P(t)_{x,y}}{\partial t} + \xi \qquad (2.22)$$

où ξ est un élément qui contient les termes d'ordre supérieur de la série. Si on prend la limite quand $\delta t \to 0$ on pourra ignorer ξ puisqu'il tend aussi vers zéro. Donc on aura :

$$P(t+\delta t)_{x+\delta x,y,+\delta y} = P(t)_{x,y} + \delta x \frac{\partial P(t)_{x,y}}{\partial x} + \delta y \frac{\partial P(t)_{x,y}}{\partial y} + \delta t \frac{\partial P(t)_{x,y}}{\partial t} + \xi \qquad (2.23)$$

En remplaçant $P(t+\delta t)_{x+\delta x,y,+\delta y}$ dans (2.18) et après arrangement des termes on obtient :

$$\frac{\delta x}{\delta t}\frac{\partial P}{\partial x} + \frac{\delta y}{\delta t}\frac{\partial P}{\partial y} = -\frac{\partial P}{\partial t} \qquad (2.24)$$

Dans cette formule on peut reconnaître certains termes ; $\partial P/\partial x$ et $\partial P/\partial x$ ont les premières dérivées par rapport à x et y. $\partial P/\partial t$ est la vitesse de variation dans le temps de l'intensité des pixels. Les deux termes $\partial x/\partial t$ et $\partial y/\partial t$ décrivent le mouvement le long des deux axes x et y donc le flot optique. En posant $u = \partial x/\partial t$ et $v = \partial y/\partial t$ en les remplaçant on obtient :

$$u\frac{\partial P}{\partial x} + v\frac{\partial P}{\partial y} = -\frac{\partial P}{\partial t} \tag{2.25}$$

Cette équation montre comment le flot optique (u,v) et la variation spatiale de l'intensité décrivent le changement temporel du contenu de l'image. A présent, nous avons les opérateurs qui peuvent calculer le changement spatial de l'intensité par les opérateurs de détection de contours. Nous pouvons aussi calculer le changement temporel de l'image par la différentiation. Reste le problème de la détermination des composantes u et v du flot optique. On ne peut pas déterminer ces deux inconnues à partir d'une seule équation puisqu'il existe plusieurs couples de (u,v) qui peuvent satisfaire l'équation (2.25). Le problème peut être résolu en cherchant une estimée de u et v qui minimise l'erreur suivante :

$$ec = \iint \left(u\nabla x + v\nabla y + \nabla t\right)^2 dxdy \tag{2.26}$$

On peut approcher la solution en tenant en compte de la deuxième contrainte qui est le voisinage du point d'intérêt qui ce déplace avec une même vélocité. A partir de cette deuxième contrainte on va chercher à minimiser les variations du flot optique sur les deux axes. L'erreur à minimiser est la suivante :

$$es = \iint \left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \right) dxdy \tag{2.27}$$

L'erreur totale est un compromis entre l'importance accordée aux deux erreurs précédentes par un certain paramètre régularisation λ. L'erreur totale est donc :

$$e = \lambda \times ec + es \tag{2.28}$$

$$e = \iint \left(\lambda \times \left(u\frac{\partial P}{\partial x} + v\frac{\partial P}{\partial y} + \frac{\partial P}{\partial t} \right)^2 + \left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \right) \right) dxdy \tag{2.29}$$

L'implémentation de cette équation est donnée par :

$$e=\sum_x\sum_y\left(\begin{array}{l}\lambda\times\left(u_{x,y}\nabla x_{x,y}+v_{x,y}\nabla y_{x,y}+\nabla t_{x,y}\right)^2\\+\frac{1}{4}\left(\left(u_{x+1,y}-u_{x,y}\right)^2+\left(u_{x,y+1}-u_{x,y}\right)^2+\left(v_{x+1,y}-v_{x,y}\right)^2+\left(v_{x,y+1}-v_{x,y}\right)^2\right)\end{array}\right) \quad (2.30)$$

La minimisation par rapport aux variables du flot optique nous donne un système de deux équations :

$$\frac{\partial e_{x,y}}{\partial u_{x,y}}=\left(\lambda\times 2\left(u_{x,y}\nabla x_{x,y}+v_{x,y}\nabla y_{x,y}+\nabla t_{x,y}\right)\nabla x_{x,y}+2\left(u_{x,y}-\bar{u}_{x,y}\right)\right)=0 \quad (2.31)$$

Et

$$\frac{\partial e_{x,y}}{\partial v_{x,y}}=\left(\lambda\times 2\left(u_{x,y}\nabla x_{x,y}+v_{x,y}\nabla y_{x,y}+\nabla t_{x,y}\right)\nabla y_{x,y}+2\left(v_{x,y}-\bar{v}_{x,y}\right)\right)=0 \quad (2.32)$$

La solution de ces deux équations est donnée sous forme itérative

$$u_{x,y}^{\langle n+1\rangle}=\bar{u}_{x,y}^{\langle n\rangle}-\lambda\left(\frac{\nabla x_{x,y}\bar{u}_{x,y}+\nabla y_{x,y}\bar{v}_{x,y}+\nabla t_{x,y}}{1+\lambda\left(\nabla x_{x,y}^2+\nabla y_{x,y}^2\right)}\right)\left(\nabla x_{x,y}\right) \quad (2.33)$$

$$v_{x,y}^{\langle n+1\rangle}=\bar{v}_{x,y}^{\langle n\rangle}-\lambda\left(\frac{\nabla x_{x,y}\bar{u}_{x,y}+\nabla y_{x,y}\bar{v}_{x,y}+\nabla t_{x,y}}{1+\lambda\left(\nabla x_{x,y}^2+\nabla y_{x,y}^2\right)}\right)\left(\nabla y_{x,y}\right) \quad (2.34)$$

Les valeurs moyennes du flot optique sont données par :

$$\bar{u}_{x,y}=\frac{u_{x-1,y}+u_{x,y-1}+u_{x+1,y}+u_{x,y+1}}{2}+\frac{u_{x-1,y-1}+u_{x-1,y+1}+u_{x+1,y-1}+u_{x+1,y+1}}{4} \quad (2.35)$$

$$\bar{v}_{x,y}=\frac{v_{x-1,y}+v_{x,y-1}+v_{x+1,y}+v_{x,y+1}}{2}+\frac{v_{x-1,y-1}+v_{x-1,y+1}+v_{x+1,y-1}+v_{x+1,y+1}}{4} \quad (2.36)$$

La différentiation temporelle est donnée par l'équation suivante entre deux images successives:

$$\nabla t_{x,y}=\frac{\left(P(1)_{x,y}+P(1)_{x+1,y}+P(1)_{x,y+1}+P(1)_{x+1,y+1}\right)-\left(P(0)_{x,y}+P(0)_{x+1,y}+P(0)_{x,y+1}+P(0)_{x+1,y+1}\right)}{8} \quad (2.30)$$

Algorithme :

Flot optique par méthode différentielle :

- **N** nombre maximum d'itérations
- **S** paramètre de régularisation
- Charger deux images
- initialiser les valeurs du flot optique à zéro
- Pour **k** allant de **1** jusqu'à **N** faire
  - Pour **i** allant de **2** jusqu'à **NLines** faire
  - Pour **j** allant de **2** jusqu'à **NColonnes** faire
- Calculer les différentes dérivées
  - Calculer la dérivée **Ex** suivant l'axe des x
  - Calculer la dérivée **Ey** suivant l'axe des y
  - Calculer la différence temporelle **Et**
- Calculer les valeurs moyennes du flot
  - Calculer **Au,** équation (2.33)
  - Calculer **Av,** équation (2.34)
- Calculer les valeurs estimées du flot
  - Calculer l'estimée **u** suivant l'axe des x, équation (2.35)
  - Calculer l'estimée **v** suivant l'axe des y, équation (2.36)
  - fin Pour (i, j)
- fin Pour itérations

Conclusion

Deux méthodes d'estimation du mouvement sont étudiées dans ce chapitre. Le filtrage de Kalman est une méthode très importante et son implémentation est simple. Elle sera utilisée dans notre application dans les chapitres suivants.

Chapitre III

Détection et suivi des points d'intérêt

III.1 Définition des points d'intérêt :

Les points d'intérêt sont des points dans l'image qui se distinguent par leurs saillance et qui peuvent être localisés facilement dans des images successives et ne se perdent pas facilement ce qui les rend facile à détecter et à suivre dans le temps [11]. Les points d'intérêt peuvent être sélectionnés en se basant sur certaines mesures de la texture, coins, contours ou la couleur. La détection de coins par exemple est très importante parce que ce point d'intérêt apporte une information très utilisée dans la vision par ordinateur.

III.2 détection des coins :

Pour le système visuel humain, l'essentiel de l'information pertinente dans une image est contenue dans ses singularités. Il nous est en effet très difficile de reconnaître une image dont on aurait coupé les hautes fréquences. Les singularités les plus fréquentes et les plus pertinentes sont les coins, c'est-à-dire les extrémités de secteurs angulaires homogènes. Dans le cas des images réelles, ceux-ci peuvent provenir de trois origines différentes :

- des singularités dans la teinte des textures appliquées sur les objets (les coins des carreaux d'une chemise par exemple)
- de la projection des coins des volumes contenus dans la scène
- des obstructions entre les différents objets, ou encore des ombres portées, qui induisent des secteurs angulaires plus sombres.

Les coins sont souvent utilisés pour identifier des objets dans une scène ou utilisés dans la mesure de déplacement d'objets ou aussi dans la stéréoscopie. Donc une bonne détection et bonne localisation des coins sont d'un intérêt important.

Un certain nombre d'algorithmes ont été élaborés ces dernières années. Ces algorithmes peuvent être divisés en deux groupes [11]. Les algorithmes du premier groupe travaillent sur l'extraction des contours de l'image d'abord puis cherchent les points qui possèdent une courbure importante ou les intersections entre les segments. Le deuxième groupe est le plus vaste. Il consiste en la recherche des coins directement sur le niveau de gris des images.

III.2.1 Le détecteur de Kitchen-Rosenfeld :

Parmi les plus anciens détecteurs, on note celui présenté par Kitchen et Rosenfeld [12]. Ceux-ci ont proposé un détecteur de coins basé sur la mesure du changement de l'orientation du gradient de l'image le long d'un contour multiplier par la magnitude du gradient local. La mesure suivante est alors définie:

$$k = \frac{I_{xx}.I_y^2 + I_{yy}.I_x^2 - 2I_{xy}.I_x I_y}{I_x^2 + I_y^2} \quad (3.1)$$

où :

I_{xx}, I_{yy} Et I_{xy} : la deuxième dérivée de l'image (niveau de gris).

I_x Et I_y : le gradient de l'image.

Donc la mesure de k nous donne directement une idée sur le point. Un point de l'image est déclaré comme un coin si la valeur de mesure k coïncide avec un certain seuil. Généralement la courbure du point est inversement proportionnelle à k. Cet algorithme a servi comme référence pour évaluer les performances de nouveaux algorithme qui l'on suivi comme le détecteur de Harris et celui de KLT [11].

III.2.2 Le détecteur de Harris

L'algorithme connu sous le nom de Harris [35] est une version modifiée du détecteur Plessey [12]. C'est un algorithme moins sensible aux bruits et n'a besoin de calculer que la première dérivée de l'image. Le détecteur de Harris est capable de détecter les jonctions L. Il est utilisé avec succès dans beaucoup d'applications pour la détection des points caractéristiques. Harris a défini une mesure du coin par l'opérateur suivant :

$$H(x,y) = \det(C) - \alpha Trace^2(C) \quad (3.2)$$

Avec $C = w_G(r,\sigma) \times \begin{bmatrix} I_X^2 & I_X I_Y \\ I_X I_y & I_Y^2 \end{bmatrix}$ Et $w_G(r,\sigma)$ est un filtre gaussien.

α est un coefficient dont la valeur préconisée par Harris est 0.04.

Ce qui donne :

$$H(x,y) = (I_x^2.I_y^2 - I_{xy}^2) - \alpha(I_x^2 + I_y^2)^2 \quad (3.3)$$

Le détecteur de Harris attribue à chaque point de l'image une certaine courbure. Un point est déclaré coin si la mesure donnée par l'opérateur de Harris est supérieure à un certain seuil fixé. Une autre version de cet opérateur qui peut nous éviter le choix du coefficient α est la suivante :

$$H(x,y) = \frac{\langle I_X^2 \rangle + \langle I_Y^2 \rangle}{\langle I_X^2 \rangle \langle I_Y^2 \rangle - \langle I_X I_Y \rangle^2} \quad (3.4)$$

Le calcul de H a besoin des opérations suivantes :

- le calcul des premières dérivées I_X et I_Y et calculer par suite $I_X.I_Y$ (pixel par pixel).
- le calcul du carré de I_X et I_Y (pixel par pixel).
- en utilisant un noyau gaussien de 3×3 on calcule les versions filtrées sur un voisinage de 3×3 de I_X , I_Y et $I_X.I_Y$.
- enfin calculer la courbure au point (x, y) à l'aide de l'équation (3.3).

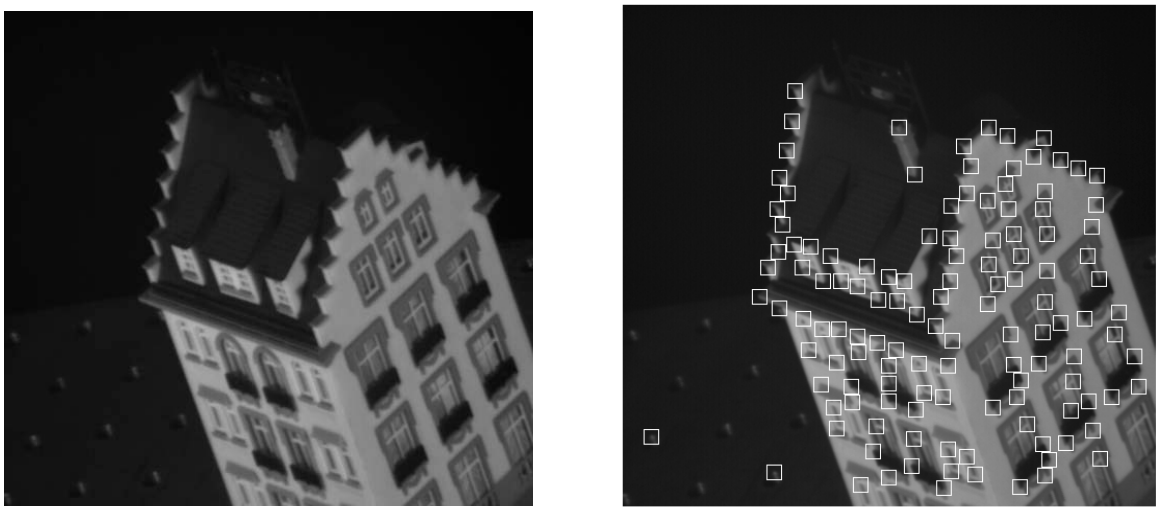


Figure (III.1) : Exemple de détection sur une image par Harris.

III.2.3 Le détecteur KLT

Le détecteur de Lucas-Kanade (KLT) [31] est un algorithme qui peut sélectionner des point caractéristiques notamment des régions texturées, les L-jonctions et les coins qui sont stables dans le temps et qui peuvent être suivi automatiquement par le KLT [30]. Le KLT est donc le nom d'un algorithme qui peut sélectioner des points et assuer le suivi dans le temps en travaillant sur deux images consécutives d'une même séquence. Lucas et Kanade utilise la même matrice (matrice de la structure locale) utilisée dans l'algorithme de Harris, mais cette fois c'est les valeurs propres de cette matrice qui sont prises en considération en chaque point de l'image.

On a :

$$C = \begin{bmatrix} \overline{I_X^2} & \overline{I_X I_y} \\ \overline{I_X I_y} & \overline{I_Y^2} \end{bmatrix}$$

$\bar{I}$ Représente le moyennage sur une fenêtre avec une gaussienne.

C'est une matrice symétrique donc diagonalisable avec les éléments de la diagonale qui sont les valeurs propres de C.

$$C = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Le point est accepté si les valeurs propres de C vérifient certaines conditions ($> \lambda_{seuil}$) qui sont définies dans la section suivante.

Relation entre le KLT et Harris :

On a : $\lambda_1 \geq \lambda_2 \geq 0$

Introduisons les relations suivantes : $\lambda_1 = \lambda$, $\lambda_2 = \theta\lambda$ avec $0 \leq \theta \leq 1$

En utilisant les relations entre les valeurs propres, déterminant et trace de C on a :

$$\det(C) = \prod_{i=1}^{2} \lambda_i \quad \text{Et} \quad trace(C) = \sum_{i=1}^{2} \lambda_i \tag{3.5}$$

On va obtenir :

$$H = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 = \lambda^2(\theta - \alpha(1+\theta)^2) \tag{3.6}$$

Pour de petites valeurs de θ on aura : $H = \lambda^2(\theta - \alpha)$ avec $\alpha \leq \theta$

Donc dans l'opérateur de Harris α joue le rôle de λ_{seuil} dans le KLT.

Les deux algorithmes sont basés sur une matrice de structure locale et la recherche des points pour lesquels la variation dans les deux directions orthogonales est importante.

La différence entre ces deux méthodes est que le Harris utilise un seuillage implicite et le KLT utilise un seuillage explicite. Le KLT détecte souvent des coins d'une façon similaire à la vision de l'être humain. L'algorithme de Lucas-Kanade est utilisé avec succès dans suivi d'objets à travers une séquence d'image et aussi utilisé dans plusieurs autres applications. Le Harris est robuste aux changements de luminance et la rotation. Il est utilisé beaucoup plus dans la vision stéréo.

III.2.4 Le détecteur de Smith (SUSAN) :

SUSAN (Smallest Univalue Segment Assimilating Nucleus) est une méthode présentée par Smith [32] pour l'extraction des points caractéristiques dans une image en niveau de gris tels que les contours et les coins. SUSAN est aussi une méthode applicable au filtrage. Le principe de cette méthode est complètement différent des précédentes. Elle se base sur la réponse d'une partie de l'image à un masque circulaire (figure (III.2)). Ce masque est approché dans la pratique par un masque de 7×7 pixels et 3 pixels dans chaque coté

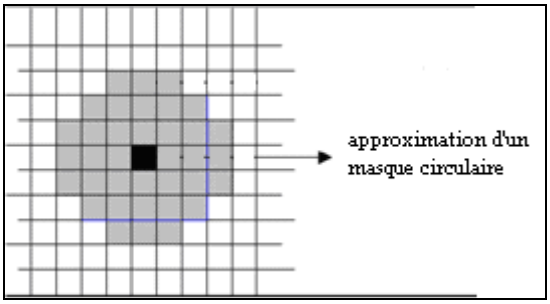


Figure (III.2) : Masque circulaire.

L'intensité du pixel central est comparée à tous les pixels inclus dans le masque en utilisant la fonction de comparaison suivante :

$$C(r,r_0)=100\exp\left(-\left(\frac{\mathrm{I(r)}-\mathrm{I(r_0)}}{t}\right)^6\right) \tag{3.7}$$

où r_0 est la position du pixel centrale, r est la position des autres pixels à l'intérieur du masque. $\mathrm{I(r)}$ est l'intensité de chaque pixel, t est un paramètre appelé seuil de différence d'intensité.

III.3 Méthode de Lucas et Kanade pour le suivi :

III.3.1 Introduction

Cette technique a été introduite pour la première fois en 1981 par Bruce D. Lucas et Takeo Kanade [29]. Elle est présentée comme une nouvelle technique de mise en correspondance entre deux images (registration). Les auteurs utilisent l'intensité spatiale et le gradient de l'image pour chercher directement la position de l'objet caractéristique. Ils supposent un modèle de translation entre deux images successives. Par suite la méthode dite KLT est appliquée pour le suivi d'objets en 1991 par C. Tomasi et T. Kanade [30]. C'est une méthode basée sur le suivi de points par la minimisation d'une SSD (somme de différences carrées) en utilisant un modèle de translation. Puis, Shi et Tomasi [31] ont proposé un modèle affine.

Principe :

Etant donné deux images i et i+1 d'une séquence d'images qui contient un objet à suivre, le problème est de trouver le déplacement **d** de l'objet d'une image à une autre sachant qu'on ne peut pas suivre un seul pixel à cause du bruit qui peut affecter ce pixel (figure (III.3)). Le suivi s'effectue alors sur une fenêtre de pixels de taille généralement allant de 3×3 à 25×25. Ces fenêtres sont bien choisies sur l'objet à suivre et correspondent aux zones texturées pour ne pas perdre rapidement le point suivi. Les fenêtres utilisées contiennent généralement les coins en L ou les jonctions T ce qui les rend suffisamment différentes de leurs voisinage. L'étape la plus essentielle dans le KLT est la sélection de ces points caractéristiques qu'on peut suivre durant une séquence d'image.

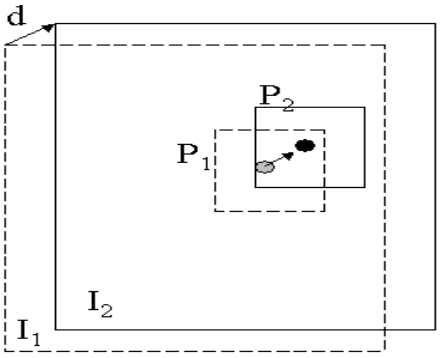


Figure (III.3) : Principe de suivi de points par KLT.

Donc quand la caméra ou l'objet est en mouvement, la fenêtre choisie sur l'objet à suivre va être affectée par un bruit dû aux changements de l'éclairage de la scène ou des déformations dues aux mouvements de l'objet. Le déplacement **d** est choisi de façon à minimiser une erreur résiduelle.

On défini :

$$J(\mathrm{x},t) = I(x, y, t+\tau) \tag{3.8}$$

Et

$$I(\mathrm{x}-\mathrm{d},t) = I(x-\Delta x, y-\Delta y, t) \tag{3.9}$$

Le point à suivre au temps $t+\tau$ est déterminé par les déplacements Δx et Δy dans un plan sachant que

$$J(\mathrm{x},t+\tau) = I(\mathrm{x}-\mathrm{d},t)+\eta(x,y) \tag{3.10}$$

Où η est un bruit.

Le déplacement $\mathrm{d} = (\Delta x, \Delta y)$ est choisi de façon à minimiser une erreur résiduelle qui est définie par la double intégrale suivante :

$$\varepsilon = \int_W [I(x-d) - J(x)]^2 \, w dx \tag{3.11}$$

Dans l'expression, w est fonction de pondération dans le cas le plus simple w est égale à 1. W est une fenêtre centrée sur le point d'intérêt, (x, y) est une position sur l'image et **d** est un déplacement entre deux images d'une séquence. L'intégrale est remplacée dans la pratique par une sommation sur les pixels de la fenêtre. Dans le cas où le déplacement de la fenêtre est petit par rapport à la taille de la fenêtre le changement local de l'intensité à l'intérieur de la fenêtre est modélisé par une translation plus un bruit. Puis le problème est de minimiser l'erreur résiduelle définie par l'équation (3.11). On réécrit les équations en cachant la variable temporelle t :

$$J(\mathrm{x}) = J(\mathrm{x}, t) \quad \text{Et} \quad I(\mathrm{x}-\mathrm{d}) = I(\mathrm{x}-\mathrm{d}, t)$$

Pour un petit déplacement d la fonction de l'intensité $I(\mathrm{x}-\mathrm{d})$ peut être approchée par un développement en série de Taylor de premier ordre :

$$I(\mathrm{x}-\mathrm{d}) = I(\mathrm{x}) - \mathrm{g}.\mathrm{d} \tag{3.12}$$

En introduisant le développement de Taylor dans (3.11) on peut écrire l'erreur résiduelle de la façon suivante :

$$\varepsilon = \int_W [I(\mathrm{x}) - \mathrm{g}.\mathrm{d} - J(\mathrm{x})]^2 \, w dx \tag{3.13}$$

On pose $I(\mathrm{x}) - J(\mathrm{x}) = h$

Ce qui donne

$$\varepsilon = \int_W [h - \mathrm{g}.\mathrm{d}]^2 \, w dx \tag{3.14}$$

La minimisation se fait en différenciant le résidu ε par rapport au déplacement d et mettant le résultat à zéro, d'où:

$$\int_W [h - \mathrm{g}.\mathrm{d}]\mathrm{g}\, w dA = 0 \tag{3.15}$$

On a : $(\mathrm{g}.\mathrm{d})g = (\mathrm{g}\mathrm{g}^{\mathrm{T}})\mathrm{d}$ et en supposant que d est constant à l'intérieur de la fenêtre W on a :

$$\left(\int_W (\mathrm{g}\mathrm{g}^{\mathrm{T}}\, \omega.dA)\right)\mathrm{d} = \int_W h\,\mathrm{g}\,\omega.dA \tag{3.16}$$

C'est un système de deux équations à deux inconnues qui peut être écrit sous forme matricielle suivante :

$$z\vec{d} = \vec{e} \tag{3.17}$$

où Z est une matrice symétrique de 2×2

$$\mathrm{z} = \int_W \mathrm{g}\mathrm{g}^{\mathrm{T}}\omega.dA \tag{3.18}$$

$$Z = \iint_W \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} \omega.dA$$

$$e = \iint_W (I - J)\begin{bmatrix} g_x & g_y \end{bmatrix}^{\mathrm{T}} dA \tag{3.19}$$

g_x Et g_y représente les dérivées de l'intensité I suivant respectivement x et y.

La méthode Lucas-Kanade minimise (3.11) itérativement. On peut écrire cela sous la forme suivante :

$$\vec{d}(k+1) = \vec{d}(k) + Z^{-1}\vec{e}(k) \text{ Avec } \vec{d}(0) = 0 \tag{3.20}$$

Où $\vec{d}(k)$ représente l'estimation du déplacement à la k$^{\text{ième}}$ itération.

La méthode KLT proposée par Kanade, Lucas et Tomasi repose sur une extraction de points dans la première image, et le suivi de ces points dans les images suivantes. Seuls les

voisinages des points caractéristiques sont traités d'une image à la suivante [18]. Donc la sélection des points à suivre est une étape très importante dans le KLT.

III.3.2 Sélection des points :

Les points et leurs voisinages qu'on va utiliser pour le suivi doivent décrire traduire le mouvement de cet objet. Par exemple les régions homogènes ne contiennent aucune information de mouvement, de même qu'un contour horizontal ou vertical ne peut pas fournir une information complète sur le mouvement il y a qu'une seule composante qu'on peut décrire. Pour cela on utilise généralement des régions texturées ou celles qui contiennent des hautes fréquences spatiales. Lucas et Kanade ont proposé une méthode pour sélectionner les régions d'une image. Une fenêtre est bonne pour le suivi d'une image à une autre si le système (3.17) donne une bonne mesure et est bien conditionné. C'est-à-dire la matrice Z des coefficients 2*2 est à la fois non sensible aux bruits et bien conditionnée. Cela peut être traduit par les valeurs propres λ_1, λ_2 de la matrice Z par le fait qu'une fenêtre est acceptable si ses valeurs propres sont suffisamment grandes par rapport à un certain seuil pour l'immunité aux bruits et avec une légère différence entre elles pour le bon conditionnement [11].

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

Où λ est un seuil prédéfini.

L'algorithme suivant résume les différentes étapes à suive pour sélectionner les meilleures fenêtres avec le KLT :

1. Calculer les dérivées g_x et g_y sur toute l'image I_0.
2. Pour chaque point P de l'image I_0 :
   a. Former la matrice Z sur un voisinage de taille N×N autour du point P.
   b. Calculer les valeurs propres de Z.
   c. Si la valeur propre la plus petite λ_2 est supérieure au seuil λ sauvegarder le point P dans une liste L.
3. Ordonner les points de la liste avec un ordre.
4. Supprimer les points qui sont très proches sur l'image.

En général une fenêtre est acceptable si $IRHarris = \min(\lambda_1, \lambda_2) > \lambda$

Le choix du seuil se fait généralement par l'analyse de l'histogramme de λ_2.

Le KLT n'est pas le seul algorithme qui permet de sélectionner des fenêtres texturées mais y a aussi le détecteur de Harris, le détecteur SUSAN ou la sélection par le IPAN [33], qui sont des méthodes non itératives. Les fenêtres sélectionnées permettent d'assurer le suivi pour une longue durée temporelle par rapport à des fenêtres quelconques. Par suite on va voir l'application du KLT au suivi de visage dans une séquence vidéo et on va considérer aussi les différentes régions sélectionnées pour le suivi par le KLT et Harris.

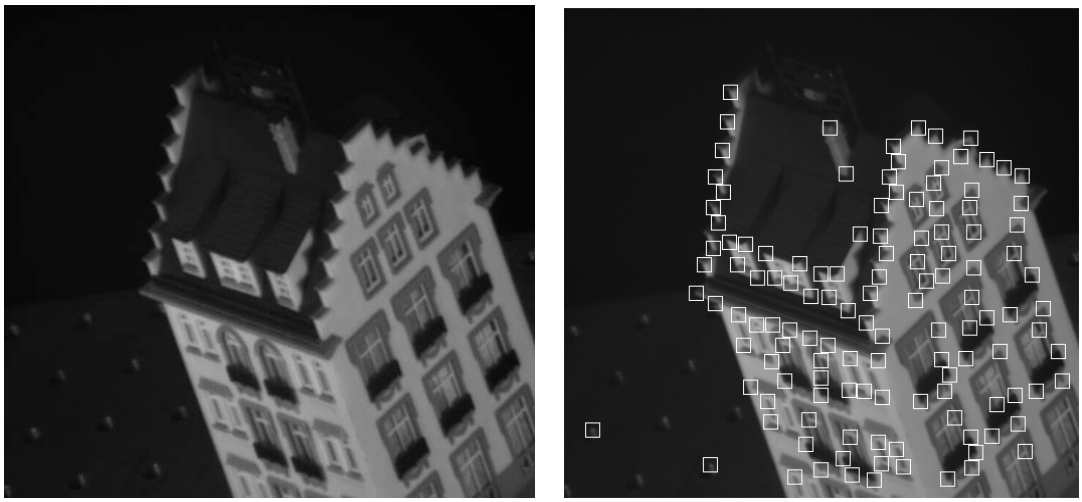


Figure (III.4) : 125 points détectés par le KLT.

Limitations :

L'algorithme KLT est très robuste dans le cas de petits déplacements entre deux images consécutives. Le problème est dans le cas où le déplacement du point suivi dépasse la taille de la fenêtre sélectionnée la perte du point est la conséquence. Le KLT est aussi sensible aux déformations importantes du point suivi.

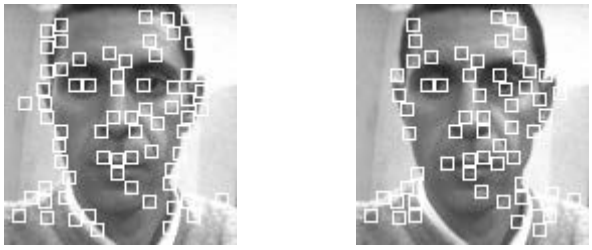
III.3.3 Application au suivi de visage :

Le suivi de visage par une machine est une tache très difficile. Pour cela plusieurs algorithmes ont été mis au point pour le suivi. Nous considérons ici l'apport du KLT appliqué aux fenêtres sélectionnées sur l'objet à suivre qui est dans notre cas un visage.

III.3.3.1 Détection des points caractéristiques sur le visage :

L'application du KLT et Harris pour la sélection des meilleurs points est illustrée par les images de la figure (III.5). On remarque que plusieurs points sélectionnés par KLT et Harris en même temps. Le voisinage utilisé ici est de taille 3×3 pixels lissé avec une gaussienne. Les

points caractéristiques KLT sont répartis de manière non homogène dans l'image. Ils sont généralement concentrés dans les zones texturées. Il faut en choisir suffisamment pour avoir un recouvrement correct de tous les objets dans l'image. Un nombre trop élevé de points caractéristiques KLT entraîne par contre la sélection de points pour lesquels l'estimation du déplacement n'est pas fiable. Cela peut conduire à l'apparition d'artefacts comme des déplacements le long de contours fixes. Pour recouvrir tout le visage la sélection des points ne se fait pas directement par un seuillage sur tout le visage mais par des seuils locaux de toutes les parties du visage de 15×15 pixels. Donc le nombre de points qui seront sélectionnés est fixé d'avance par la taille de la zone locale et la taille du visage en pixel. Si on utilise un seuillage sur toute la fenêtre contenant le visage la majorité des points qui seront sélectionnés se concentre sur et autour des yeux, sur les narines, la bouche et les différents contours du visage. Durant le suivi plusieurs de ces points sélectionnés seront perdus à cause du bruit, l'occultation ou par la déformation du voisinage du point.



1. KLT **2.** Harris

Figure (III.5) : Exemple de points détectés sur Le visage.

III.3.3.2 Suivi des points sélectionnés :

La durée de vie moyenne d'un point suivi par le KLT, définie par le nombre d'images successives dont le point est apparu, est de 115 images pour une fenêtre de 3×3 pixels, 349 images pour une fenêtre de 5×5 pixels et 479 images pour une fenêtre de 7×7 pixels [19]. Cette durée de vie pour différentes tailles de fenêtres n'est valable que pour des vitesses de déplacement faibles du point d'intérêt. Pour parer aux bruits une fonction de pondération du voisinage du point considéré peut réduire l'effet du bruit. Généralement c'est une fonction gaussienne pour des objets réels qui possèdent des contours non abrupts comme le cas de visage de l'être humain. Dans le cas d'images d'objets fabriqués par l'homme ou des images synthétiques qui contiennent des contours aigues, la meilleure fonction pour réduire les bruits est la fonction LoG [20].

III.3.4 Autres algorithmes basés sur le principe de KLT :

L'algorithme de suivi des composantes faciales proposé par [34] est basé sur le KLT. Ce nouvel algorithme travaille sur les composantes faciales du visage de l'être humain. Les narines sont utilisées comme points de référence et constituent un repère pour les autres points. La configuration des composantes faciales est exploitée durant le suivi. La recherche des points perdus durant le suivi se fait par une fenêtre de recherche autour du point suivi. La figure (III.6) montre l'ensemble des points pris sur le visage. Les douze points sont localisés manuellement sur la première image de la séquence.

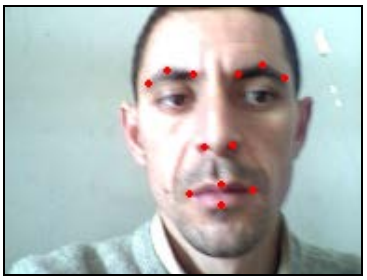


Figure (III.6) : Les meilleurs points pour le suivi de visage.

L'algorithme utilise un nombre de 12 points situés sur le visage :

- Un point sur chacune des narines
- Un point au milieu des deux lèvres
- Un point sur chaque coin de la bouche
- Trois points pour chaque sourcil

Les douze points sont localisés manuellement sur la première image de la séquence.

Les narines sont considérées comme des points qu'on peut suivre aisément dans une séquence d'images par KLT seul. Si les narines sont perdues, une fenêtre de recherche est créée. Son centre est la moyenne des positions des narines à l'image précédente. Le milieu des narines est utilisé comme référence pour créer des fenêtres de recherche pour les autres points qui sont perdues durant le suivi par KLT. La figure (III.7) illustre la méthode proposée par Bourel et al. [34].

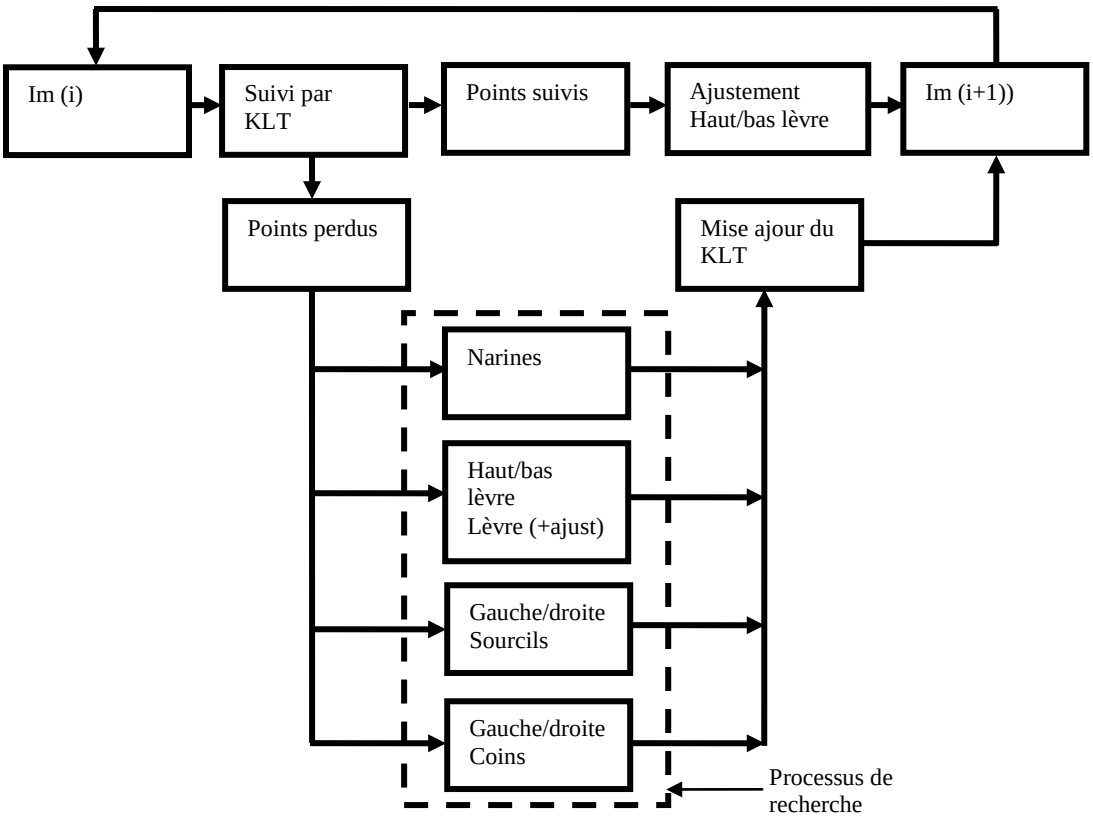


Figure (III.7) : Schémas de suivi par EKLT [34].

III.3.5 Résultats et discussion :

Dans le but d'utiliser le KLT pour le suivi de visage pour des applications de suivi des composantes faciales ainsi que pour des applications de l'homographie, on utilise une fenêtre de 25×25 pixels qui est la taille maximale qui peut être prise comme indiqué dans le travail de Lucas et Kanade. Dans notre travail on impose la taille de la tête telle que les deux yeux ainsi que le nez soient à l'intérieur de cette fenêtre de 25×25 pixels comme montré sur la figure (III.8) :

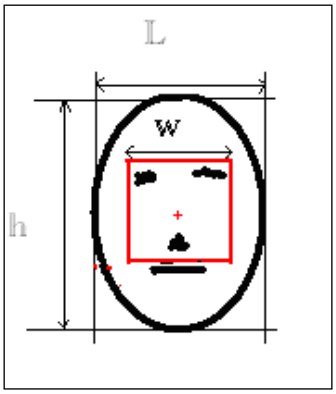


Figure (III.8) : Taille idéale du visage pour le suivi par KLT.

On a $W = 25$ pixels, pour retrouver la taille en pixels de la tête de la personne sur l'image sachant que les composantes faciales doivent être incluses dans une fenêtre de 25×25 pixels on considère comme suit :

On considère la largeur de la tête $L = 2 \times W$ ce qui donne une largeur $L = 50$ pixels.

Pour calculer la hauteur h on prend les proportions entre la hauteur et la largeur utilisée dans plusieurs travaux [13] : $L = \frac{3}{4} \times h$, dans ce cas $h = \frac{4}{3} \times L \rightarrow h \approx 66 \pm quelques$ pixels.

Pour l'évaluation des résultats de l'algorithme de KLT pour une fenêtre de 25×25 pixels nous utilisons deux séquences vidéo qui contiennent des mouvements de translation, de rotation ainsi que des gestes brusques de la tête. Les résultats sont comparés avec ceux obtenus par la méthode de suivi par corrélation. L'évaluation se fait sur trois différentes tailles de fenêtre. Les séquences vidéo sont prises par une WebCam. Le fond de ces images n'est pas important puisque ces deux algorithmes travaillent sur des fenêtres locales contenant l'objet d'intérêt (ROI). Les résultats sont présentés dans ce qui suit.

Première séquence :

Dans cette étude nous utilisons l'algorithme de KLT avec une fenêtre de 25×25 pixels. La séquence échantillon d'images sur la figure (III.9) nous montre comment le visage est localisé sur chaque image.

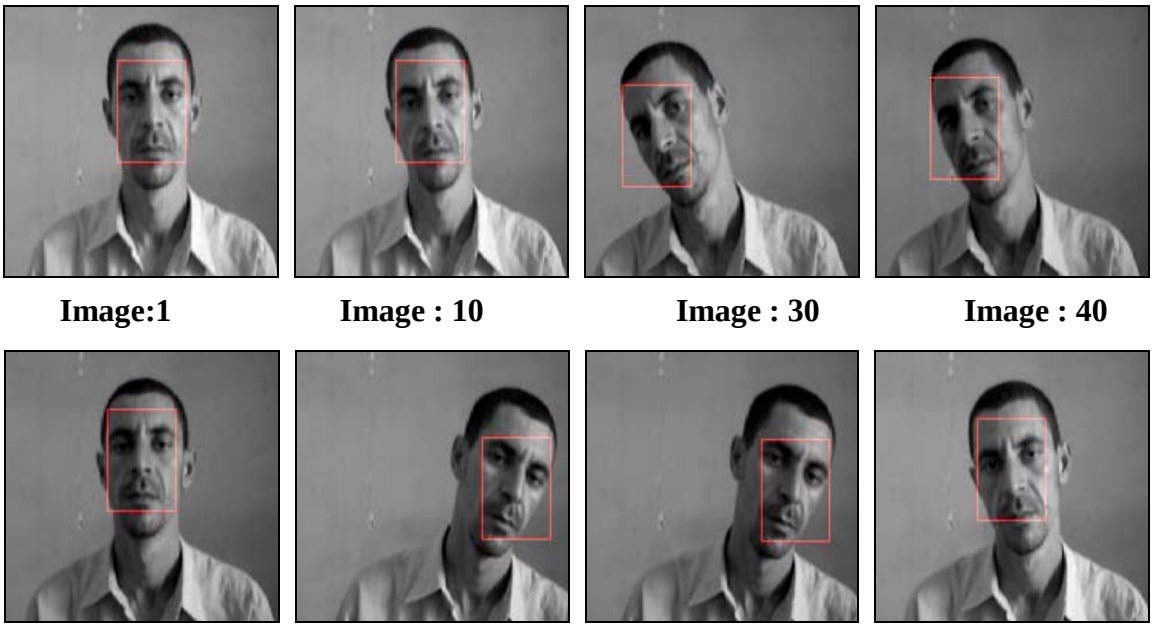


Figure (III.9) : localisation du visage dans une séquence par KLT.

La figure (III.10) montre l'erreur de position commise sur chaque image de la séquence et cela pour différentes tailles de fenêtre. Les fenêtres les plus utilisées sont 25×25, 15×15 , 9×9 et 7×7 . Les principaux paramètres qui influencent la localisation sont le type de mouvement c'est-à-dire un mouvement de translation pure, un mouvement de translation + rotation ou des mouvements brusque, il y á aussi l'ombre qui peut occulter certaines parties de la ROI.

Sur la figure (III.10) nous notons que le meilleur résultat de suivi est obtenu par la plus grande fenêtre qui est de 25×25. Pour les 30 premières images l'erreur de suivi est inférieure à 4 pixels. À partir de la 35 image l'erreur dépasse les 5pixels cela est dû au mouvement de la rotation de tête qui conduit à l'occultation de la partie droite du visage et par mauvaise illumination de celle-ci comme on peut le constater sur l'image 30 de la séquence. Pour la taille de fenêtre de 15×15 les performances sont moindres que 25×25. La divergence dès le début de la fenêtre de 7×7 s'explique le fait que celle-ci n'est une fenêtre texturée puisque elle n'inclut pas les yeux et le nez.

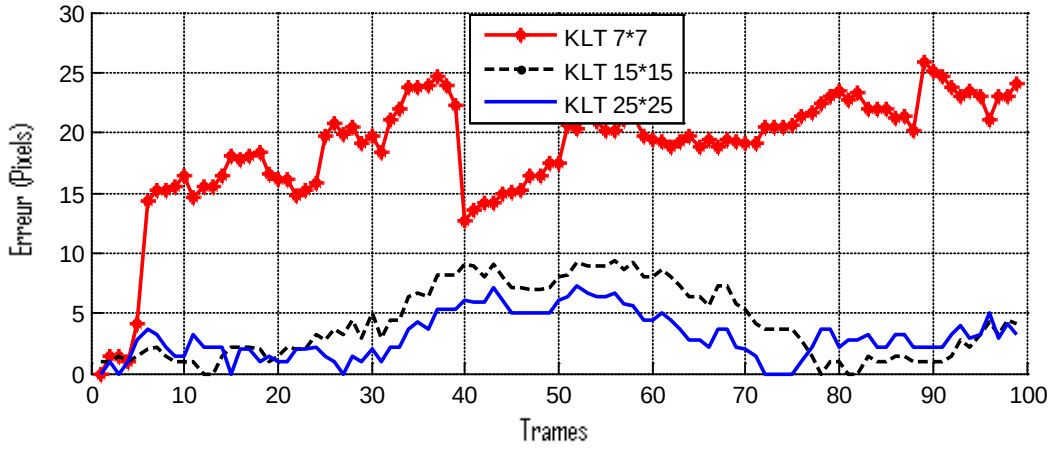


Figure (III.10) : Erreurs de position pour différentes tailles de fenêtres par KLT.

Les performances enregistrées avec la corrélation, figure (III.11), sont nettement moindres que celles du KLT. Sur les images suivantes on voit que pour les 45 premières images la localisation est faite comme le KLT. A partir de cette image de la séquence le visage n'est pas bien localisé.

Figure (III.11) : Localisation du visage dans une séquence par corrélation.

Sur le graphe de la figure (III.12) relatif aux erreurs de suivi par corrélation on peut dire que le visage est perdu à partir de la $50^{ième}$ image de la séquence. Pour différentes tailles de la fenêtre on a choisi trois fenêtres de 15×15, 25×25 et 35×35 pixels. Le meilleur résultat est obtenu par la fenêtre de 35×35 pixels. Une divergence dès le début pour la fenêtre de 15×15 pixels. Pour la fenêtre de 15×15 réalise des performances proches à la première fenêtre.

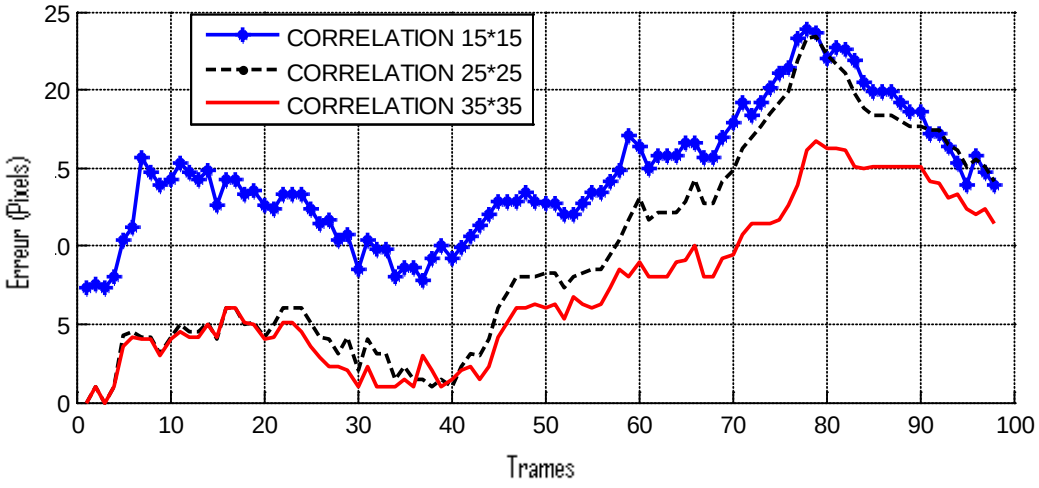


Figure (III.12) : Erreurs de position pour différentes tailles de fenêtres par corrélation.

III.3.5.1 Comparaison entre les deux méthodes :

Pour la plupart des images de cette séquence le KLT réalise des erreurs contenues dans un intervalle inférieur à 5 pixels excepté pour les images où se présentent des mouvements de rotation et effet de l'ombre. Même chose pour la corrélation mais celle-ci diverge pour perdre carrément le visage à cause de ces conditions de rotation et de l'ombre. A la figure (III.13) nous présentons une comparaison pour une même taille de fenêtre (25×25).On peut dire que le KLT est meilleur pour le suivi. Dans ce qui suit on va utiliser une autre séquence d'image pour encore évaluer les performances des deux méthodes sous différentes rotations de la tête de l'être humain à suivre.

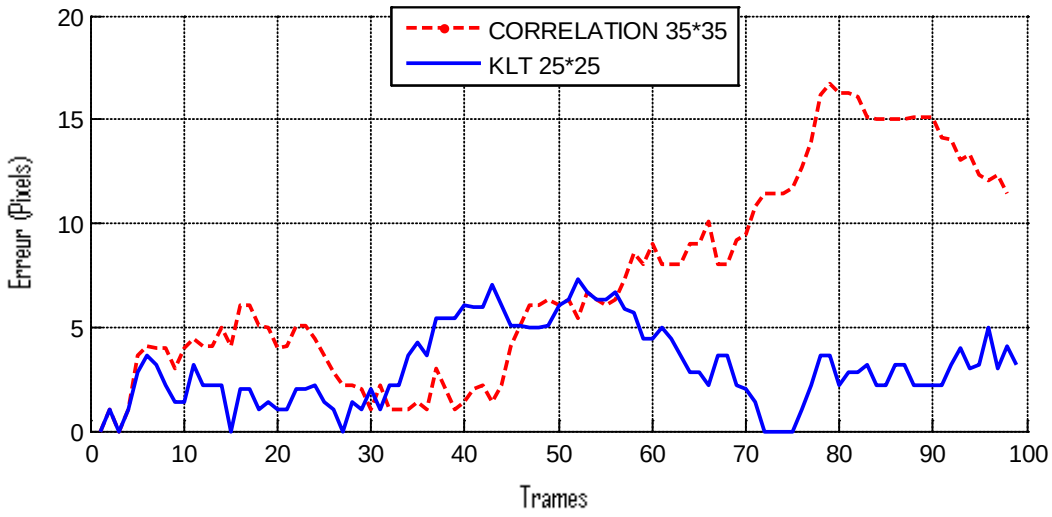


Figure (III.13) : Comparaison entre KLT et corrélation.

Deuxième séquence :

Les deux images suivantes d'une autre séquence contiennent des mouvements de rotation gauche et droite ainsi qu'un mouvement de translation pour voir grossièrement quelles sont les limites de ces deux méthodes.

Image : 37 **Image : 58**

Le graphe relatif à la deuxième séquence nous montre que pour cette rotation de la tête gauche et droite la corrélation est plus sensible. A l'image 37 de la séquence la corrélation diverge complètement alors que le KLT reste toujours robuste jusqu'à l'image 58 de la séquence (figure (III.14)) où la fenêtre utilisée accroche l'arrière plan de l'image.

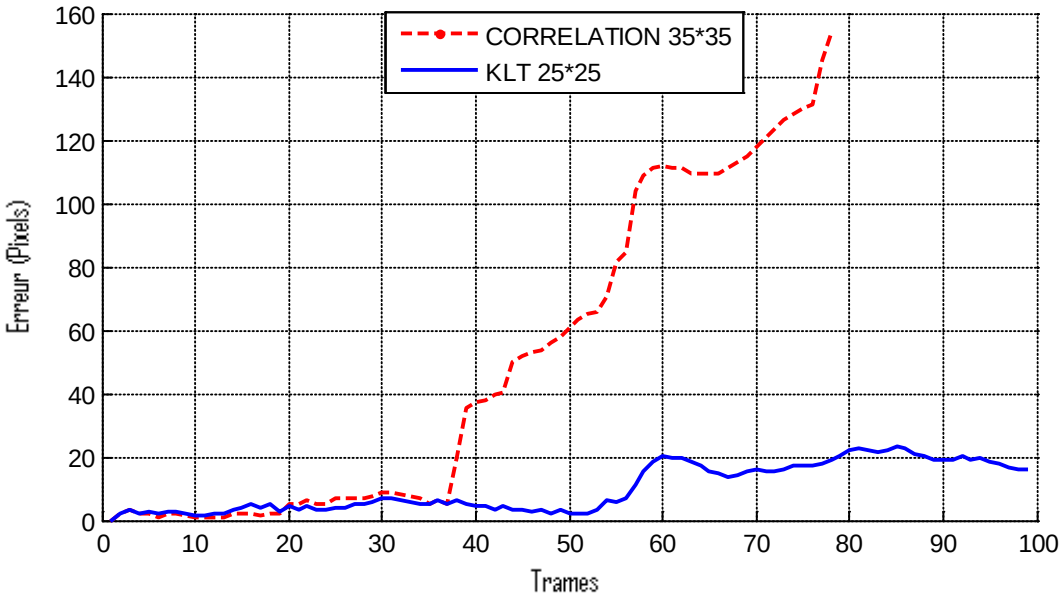


Figure (III.14) : Comparaison entre KLT et corrélation.

III.3.5.2 Temps d'exécution :

Le temps d'exécution d'un algorithme est un paramètre très important, c'est un facteur qui dépend de la complexité de l'algorithme ainsi que des caractéristiques de la machine Le tableau suivant nous donne les temps d'exécution des algorithmes étudiés ici sur un PC avec

processeur Pentium IV 1.5 GHz et une RAM de 512 Mo. Les algorithmes sont exécutés avec le logiciel MATLAB 7.1.

Sur le tableau (III.1) on peut lire les fréquences d'exécution en nombre d'images par seconde (i/s). Le traitement en temps réel est ici pratiquement atteint par la méthode KLT.

	7×7	15×15	25×25	35×35
KLT	33 i/s	31 i/s	28 i/s	Non utilisée
Corrélation	Non utilisée	20 i/s	13 i/s	3 i/s

Tableau (III.1) : Vitesse d'exécution en images par second.

Conclusion

L'algorithme de Lucas et Kanade a constitué l'objet d'étude de ce chapitre. Jusqu'à nos jours il reste l'une des méthodes les plus robustes dans le suivi basé sur les points caractéristiques. Avant d'entamer tout travail de suivi basé points caractéristiques il faut étudier les différents détecteurs de points caractéristiques. Plusieurs détecteurs de coins ou de fenêtres texturées sont présentés et étudiés dans ce chapitre ; le Kitchen-Rosenfeld est parmi les premiers détecteurs de courbure utilisé, par suite vient le détecteur de Harris qui a connu beaucoup de succès dans le domaine de traitement des images et de la vision par ordinateur. Un autre algorithme qui sélectionne les points et très proche du détecteur de Harris est le KLT et qui présente des meilleurs performances dans le contexte du suivi.

Dans une application qui vise à suivre le visage d'une personne par une caméra ou une WebCam, on a étudié l'algorithme de KLT sur plusieurs séquences vidéo dans lesquelles se présente une personne avec différents mouvements de la tête. Une comparaison avec la méthode de suivi par la corrélation est faite et illustrée sur des séquences d'images et des courbes d'erreurs montrent que le KLT est meilleur et présente des performances supérieures que ce soit dans la précision de localisation ou dans le temps d'exécution.

Cette application constitue une étape de suivi des constituants du visage comme les yeux, le nez et la bouche. Elle permet la localisation exacte de ces composantes pour utilisation dans d'autres d'applications comme l'estimation de la pose ou d'autres applications en 3D.

Chapitre IV

Estimation de la pose

IV.1 Modèle de caméra et calibration :

Le modèle de caméra le plus utilisé en vision par ordinateur est le modèle sténopé (pin-hole), basé sur une projection perspective, il permet une modélisation simple et linéaire en coordonnées homogènes du processus de formation des images [23]. Mathématiquement, la formation de l'image peut être définie comme une projection de l'espace 3D d'un corps sur le plan de l'image illustrée dans la figure (IV.1).

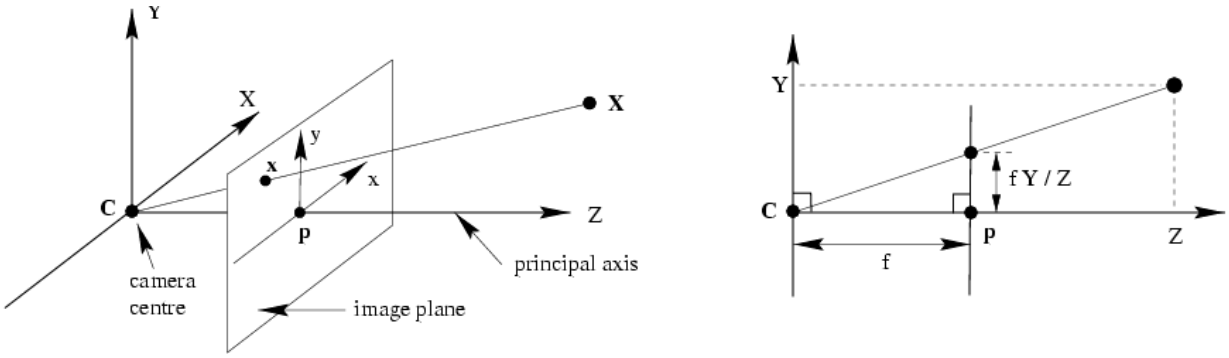


Figure (IV.1) : Modélisation de la caméra.

Le modèle sténopé associe à la caméra un repère $F_c = \left(C, \vec{i}_c, \vec{j}_c, \vec{k}_c\right)$ dont l'origine correspond au centre de projection. Les coordonnées 3D d'un point $M = [X, Y, Z]^T$ exprimées dans un repère cartésien et $m = [x, y]^T$ définit les coordonnées de la projection du point M sur le plan de l'image.

Ces coordonnées sont reliées par l'équation suivante [24]:

$$s\tilde{m} = P\tilde{M} \tag{4.1}$$

où s est un facteur d'échelle, $\tilde{m} = [x, y, 1]^T$ et $\tilde{M} = [X, Y, Z, 1]^T$ sont les coordonnées homogènes des points m et M, et P est appelée matrice de projection de 3×4 éléments. Cette matrice peut être décomposée en deux autres matrices :

$$P = K[R \mid t] \tag{4.2}$$

où :

1 K est la matrice de calibration de la caméra elle est de taille de 3×3. Elle dépend des paramètres intrinsèques de la caméra telle que la focale.

2 $[R|t]$ est une matrice de 3×4 éléments, elle correspond à la transformation des coordonnées du monde (figure (IV.2)) au système de coordonnées de la caméra. R est une matrice de rotation et t une matrice de translation.

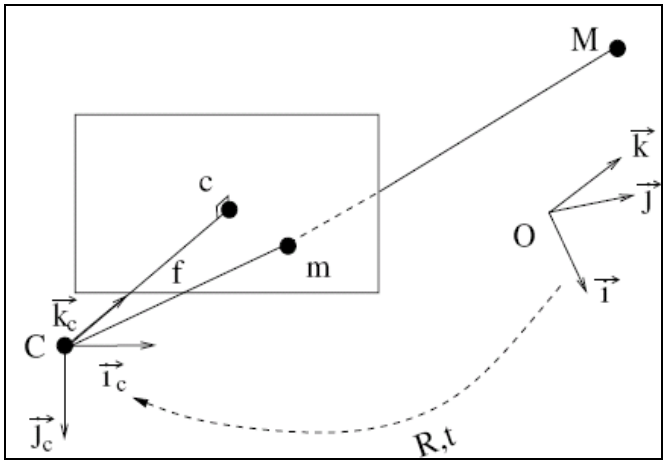


Figure (IV.2): projection d'un point sur le plan de l'image.

Dès lors que l'on souhaite utiliser une caméra pour obtenir des informations métriques, il est nécessaire de la calibrer. Le calibrage d'une seule caméra (pour les applications monoculaires) tente d'estimer ses paramètres intrinsèques ainsi que sa position par rapport au référentiel du monde.

De nombreux travaux ont été menés concernant le calibrage qui peut être hors ligne ou en ligne (auto-calibration) d'un capteur [36]. La première méthode nécessite une mire de calibrage de géométrie parfaitement connue pour calibrer hors-ligne le système de vision. La mire sous forme d'échiquier est fréquemment utilisée. La figure (IV.3) nous montre une mire de calibration 3D.

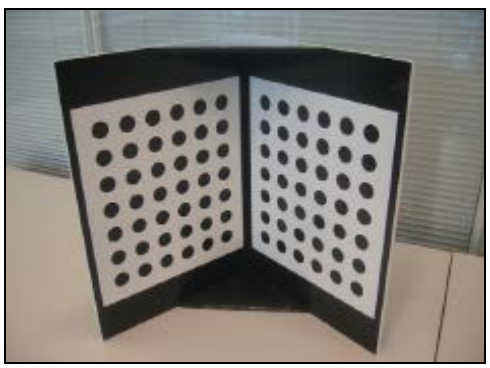


Figure (IV.3) : Mire de calibration 3D**.**

La connaissance des paramètres de calibrage permet de calculer les coordonnées 3D d'un point à partir de ses deux projections dans les deux images par une simple triangulation.
La matrice K contient les paramètres internes suivants :

$$K = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

où,

α_u Et α_v sont respectivement les facteurs d'échelle suivant les directions u et v. Ils sont proportionnels à la focale f de la caméra : $\alpha_u = k_u f$ et $\alpha_v = k_v f$ où k_u et k_v sont le nombre de pixels par unité de distance dans les directions u et v.

$[u_0, v_0]^T$ Représente le point d'intersection de l'axe optique avec le plan de l'image il représenté par l'axe z sur la figure (IV.1).

$[u_0, v_0]^T$ sont les coordonnées de la projection du centre optique de la caméra sur le plan image. Aussi, si on suppose que les pixels sont carrés, α_u et α_v peuvent être mis égaux. On peut dire qu'une caméra est calibrée si ses paramètres internes sont connus. Les 3×4 paramètres externes de la matrice $[R \mid t]$définissent la position et l'orientation de la caméra. Elle se compose de la matrice de rotation Ret de la matrice de translation t. Dans les applications de suivi on s'intéresse à l'estimation de R et t, autrement dit à la position et l'orientation de l'objet par rapport à la caméra, ceci est connu sous l'appellation de calcul de pose.

IV.2 Homographie :

L'analyse du contenu de plusieurs images d'une même scène fournit de nombreuses informations sur la géométrie de cette scène. Il existe plusieurs configurations multi vues. Le cas le plus simple est celui de deux images prises par une même caméra mobile. Elle permet d'estimer la structure 3D de la scène en se basant sur le mouvement de la caméra entre les deux prises de vue. Les relations qui existent entre les images d'une même scène peuvent être décrites par la géométrie épipolaire ou encore par l'homographie. Celles-ci définissent des transformations d'une image à l'autre.

Ce terme (homographie) désigne toute transformation projective de P^n dans P^n plus spécialement de P^2 dans P^2. Cette notion a été introduite par Faugeras [36]. La définition est

valable pour tout mouvement de caméra observé. L'idée générale est d'utiliser les coordonnées homogènes des projections d'un point X sur deux plans de projections (images sur la caméra).

La figure (IV.4) illustre les projections x_1 et x_2 du point X sur les deux plans.

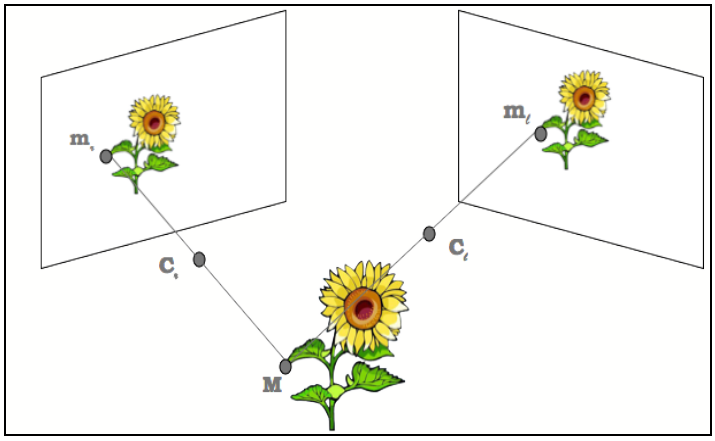


Figure (IV.4) : Projection d'une image sur deux plans.

Soient le référentiel 3D et deux plans auxquels appartiennent les deux projections du même point X **.**

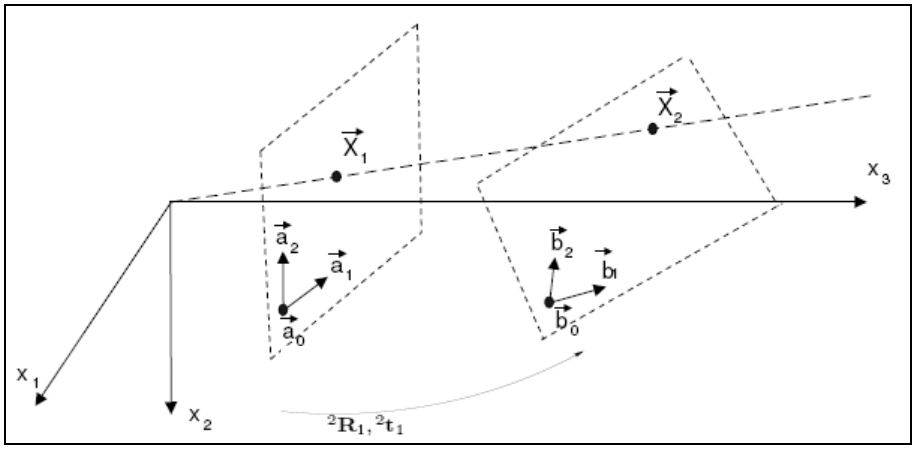


Figure (IV.5) : deux plans auxquels appartiennent les deux projections du même point.

Le premier plan est défini par le point a_0 et deux vecteurs a_1, a_2 linéairement indépendants appartenant à ce plan. Le vecteur x_1 peut être écrit de la manière suivante **:** puisque a_1 et a_2 forment une base dans ce plan,

$$\vec{x}_1 = p_1\vec{a}_1 + p_2\vec{a}_2 + \vec{a}_0 = A\vec{p} \qquad (4.3)$$

où :

$A = (\vec{a}_1, \vec{a}_2, \vec{a}_0) \in \Re^3$ définit le plan.

$\vec{p} = (p_1, p_2, 1)^T$ définit les coordonnées 2D de $\vec{x}_1$ par rapport à la base $(\vec{a}_1, \vec{a}_2)$.

D'une façon similaire on peut définir pour le deuxième plan :

$$\vec{x}_2 = B.\vec{q} \tag{4.4}$$

où :

$B = (\vec{b}_1, \vec{b}_2, \vec{b}_0) \in \Re^3$ définit le plan.

$\vec{q} = (q_1, q_2, 1)^T$ définit les coordonnées 2D de $\vec{x}_2$ par rapport à la base $(\vec{b}_1, \vec{b}_2)$.

On impose la contrainte que le point $\vec{x}_1$ correspond au point $\vec{x}_2$ sous une projection perspective dont le centre est l'origine $X = 0$, donc :

$$\vec{x}_1 = \alpha(\vec{q})\vec{x}_2 \tag{4.5}$$

Avec $\alpha(\vec{q})$ Est un scalaire qui dépend de $\vec{x}_2$ est par conséquence de $\vec{q}$.

En combinant les relations précédentes on obtient la relation entre les coordonnées 2D de ces deux points:

$$\vec{p} = \alpha(\vec{q})A^{-1}B\vec{q} \tag{4.6}$$

On sait que la matrice A est inversible puisque les vecteurs $\vec{a}_0, \vec{a}_1$ et $\vec{a}_2$ sont linéairement indépendants et différents de zéro. Aussi on remarque que p et q ont la troisième cordonnées égale à l'unité.

Le rôle de $\alpha(\vec{q})$ est de ramener le troisième élément du terme $\alpha(\vec{q})A^{-1}B\vec{q}$ à l'unité. On peut ramener p et q aux coordonnées homogènes puis on écrit :

$$\vec{p}^{\,h} = H\vec{q}^{\,h} \tag{4.7}$$

Où :

$\vec{p}^{\,h}, \vec{q}^{\,h}$ des vecteurs 3D homogènes.

$H \in \Re^{3\times3}$ est appelée matrice de l'homographie elle possède huit degrés de liberté.

D'autre part la matrice de l'homographie peut s'écrire de la manière suivante [40]:

$$H = R + \frac{tn}{d} \tag{4.8}$$

Composée d'une rotation R par rapport à l'origine et d'une translation t la relation (précédente) définit une transformation entre deux images acquises par une caméra dans deux poses où n et d sont respectivement la normale et la distance par rapport au plan de la caméra à la pose 1. Ces deux paramètres sont supposés connus à la première image et sont mis a jour en utilisant l'estimation des déplacements de la caméra [25].

IV.3 Estimation de l'homographie :

Pour estimer la matrice de l'homographie H, on commence par l'équation $x_2 \approx Hx_1$ écrite élément par élément [40]:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{4.9}$$

IV.3.1 Estimation de l'homographie par la DLT :

En coordonnées homogènes on va écrire :

$$x'_2 = x_2 / z_2 \text{ Et } y'_2 = y_2 / z_2$$

On pose $z_1 = 1$ on obtient :

$$x'_2 = \frac{h_1 x_1 + h_2 y_1 + h_3}{h_7 x_1 + h_8 y_1 + h_9} \tag{4.10}$$

$$y'_2 = \frac{h_4 x_1 + h_5 y_1 + h_6}{h_7 x_1 + h_8 y_1 + h_9} \tag{4.11}$$

Avec un réarrangement :

$$x'_2 \left(h_7 x_1 + h_8 y_1 + h_9\right) = h_1 x_1 + h_2 y_1 + h_3$$
$$y'_2 \left(h_7 x_1 + h_8 y_1 + h_9\right) = h_4 x_1 + h_5 y_1 + h_6$$

Notre objectif maintenant est de résoudre le système d'équations précédent par rapport aux éléments de H. Les deux équations apparaissent linéaires par rapport aux éléments de H avec un arrangement on obtient :

$$a_x^T h = 0 \tag{4.12}$$

$$a_y^T h = 0 \tag{4.13}$$

Où :

$$h = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)^T \quad (4.14)$$

$$a_x = (-x_1, -y_1, -1, 0, 0, 0, x_2' x_1, x_2' y_1, x_2')^T \quad (4.15)$$

$$a_Y = (0, 0, 0, -x_1, -y_1, -1, y_2' x_1, y_2' y_1, y_2')^T \quad (4.16)$$

Soit un ensemble de n couples de points qui sont en correspondance on peut alors former le système d'équations linéaires suivant:

$$Ah = 0 \quad (4.17)$$

Avec : $A = \begin{bmatrix} a_{x1}^T & a_{y1}^T & . & . & . & a_{xN}^T & a_{yN}^T \end{bmatrix}^T$

A est une matrice de 2N×9 éléments et h est formée des éléments inconnus de la matrice de l'homographie H. Le nombre de points nécessaires pour résoudre ce problème est N≥4.

Algorithme DLT :

Problème : soient $N \geq 4$ de couples de points en correspondance $x_i \leftrightarrow x_i'$. On cherche à déterminer la matrice de l'homographie H tel que $x_i' = Hx_i$.

Algorithme :

1. pour chaque correspondance $x_i \leftrightarrow x_i'$ construire la matrice $L = \begin{bmatrix} a_x & a_y \end{bmatrix}$
2. construire la matrice A de dimension $2N \times 9$ à partir des matrices L de 2×9.
3. décomposer en SVD (single values decomposition)
4. déduire H à partir de h.

IV.3.2 Estimation de l'homographie par RANSAC :

RANSAC est une méthode d'estimation robuste, qui peut être considérée comme un algorithme d'optimisation très simple à l'implémenter. Il est développé dans le contexte d'estimation de la pose de la caméra. Son avantage est qu'il n'a pas besoin de paramètres initiaux à estimer.

Principe :

A partir d'un ensemble de données, RANSAC extrait aléatoirement le plus petit sous-ensemble possible nécessaire pour générer les paramètres du modèle qui soient compatibles avec le plus grand ensemble de données.

Dans le cas de l'homographie on veut estimer les paramètres de la matrice H à partir d'un sous-ensemble de quatre mesures ou quatre points en correspondance entre deux images $x_i \leftrightarrow x_i'$.

Algorithme RANSAC :

Problème : calculer l'homographie entre deux images étant donné un ensemble de points en correspondance.

1. sélectionner aléatoirement quatre points en correspondance et calculer la matrice de l'homographie H.
2. sélectionner tous les couples (x, x') qui vérifient le modèle $x' = Hx$ avec une certaine erreur < à erreur seuil : si $d(Hx, x') < t$ où t est un seuil et $d(.)$ est une distance euclidienne entre les deux points (x, x').
3. répéter les étapes 1 et 2 jusqu'à ce qu'un nombre suffisant de couples vérifient la condition de l'étape 2.
4. recalculer la matrice de l'homographie avec tous les points issus de l'étape 3.

Une fois les paramètres de calibration interne sont connus, les paramètres de la matrice externe $[R \mid \mathrm{t}]$ peuvent être extrait à partir de H à un facteur près :

$$[R \mid \mathrm{t}] \approx K^{-1} P \tag{4.18}$$

IV.4 Calcul des paramètres de R et t :

La matrice de l'homographie H peut être décomposée en une matrice de rotation et un vecteur représentant la translation de la caméra. La représentation de la translation ne pose pas de problèmes, la représentation de la rotation en 3D quand à elle est plus difficile. On sait bien que la rotation en 3D possède seulement 3 degrés de liberté. Six éléments additionnels

sont ajoutés ; trois éléments pour forcer les différentes colonnes à avoir une norme égale à l'unité, trois autres éléments pour les tenir mutuellement orthogonaux.

Les différentes représentations efficaces de la matrice de rotation 3D sont : les angles d'Euler, représentation en exponentielle (exponentiel maps). Ces deux représentations sont équivalentes pour de petites rotations parce qu'elles mènent vers la même approximation du premier ordre.

IV.4.1 Angles de Euler :

La matrice de rotation R peut s'écrire toujours sous la forme d'un produit de trois matrices représentant les rotations autour des axes x, y et z. il existe plusieurs façons de faire l'ordre de produit entres ces matrices. Si on prend par exemple α, β, et γ des angles de rotation autour des axes x, y et z respectivement, la matrice R s'écrit :

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \tag{4.19}$$

L'opération inverse c'est-à-dire extraire les angles suivants les différents axe pour une matrice de rotation donnée peut être réalisée facilement par identification des coefficients de la matrice avec leur expression analytique. Une singularité se présente dans le cas où deux des trois axes coïncident, à ce moment une rotation n'aura aucun effet. La formulation suivante nommée 'exponentiel maps' va éviter ce problème.

IV.4.2 Formule de Rodriguez :

La représentation de la matrice R par l'exponentielle (exponentiel maps) a besoin seulement de trois paramètres. Cette représentation possède une singularité qui peut être évité.

Soit $\vec{w} = \left[w_x, w_y, w_z\right]^T$ un vecteur 3D et $\theta = \left\|\vec{w}\right\|$ sa norme. Une rotation angulaire de θ autour d'un axe de direction $\vec{w}$ peut s'écrire sous forme d'un développement en séries infinie :

$$\exp(\Omega) = \mathrm{I} + \Omega + \frac{1}{2!}\Omega^2 + \frac{1}{3!}\Omega^3 + \ldots \tag{4.20}$$

où :

$$\Omega = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

L'équation (4.20) c'est la représentation exponentielle, qui vient du fait que la représentation à la même forme que le développement en séries infinie d'une exponentielle. Elle peut être évaluée en utilisant la formule de Rodriguez :

$$R(\Omega) = \exp(\Omega) = \mathrm{I} + \sin\theta.\hat{\Omega} + (1 - \cos\theta)\hat{\Omega}^2 \tag{4.21}$$

où $\hat{\Omega}$ est la matrice qui correspond au vecteur unitaire $\frac{\vec{w}}{\|\vec{w}\|}$:

$$\hat{\Omega} = \frac{1}{\theta}\begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

A première vue une singularité apparaît quand $\|\vec{w}\|$ tend vers zéro. Mais ce n'est pas le cas parce que l'équation (4.21) s'écrit:

$$R(\Omega) = \exp(\Omega) = \mathrm{I} + \frac{\sin\theta}{\theta}\Omega + \frac{(1 - \cos\theta)}{\theta^2}\Omega^2 \tag{4.22}$$

Par remplacement des thermes $\frac{\sin\theta}{\theta}$ et $\frac{(1-\cos\theta)}{\theta^2}$ par les premiers termes de leur développement de Taylor la singularité autour de zéro est évitée.

Cas des petites rotations :

Dans les applications du suivi 3D, le mouvement de la caméra entre deux images consécutives est souvent petit. Dans ce cas il est commode d'utiliser l'approximation du premier ordre de la rotation ce qui va simplifier les calculs comme suit :

Soit M' la position en 3D du point M après une rotation R autour de l'origine avec un angle petit. La simplification donne :

$$M' = RM \approx (\mathrm{I} + \Omega)M = M + \Omega M \tag{4.23}$$

IV.4.3 Transformation affine

Le suivi de petites régions de l'image (patchs) où une estimation locale du mouvement peut engendrer des simplifications du modèle de mouvement. Le modèle affine de mouvement suppose que le mouvement inter image soit assez faible pour que l'on puisse négliger les effets de perspective entre deux images pour une région de l'image, alors la transformation suivante est valable pour un mouvement de rotation plus une translation.

$$\bar{x}_2 = A\bar{x}_1 + t \tag{4.24}$$

La matrice A de dimension 2×2 exprime les termes de divergence, de rotation et de distorsion du mouvement. Il existe plusieurs modélisations de la matrice A [37], certaines permettant une décomposition en champs de mouvements divergents et rotationnels, ou plus simplement une rotation pure. Enfin la simplification peut continuer en allant jusqu' à prendre en compte uniquement le terme de translation.

$$X_2 = X_1 + t \tag{4.25}$$

IV.4 Résultats :

Pour une application de l'estimation de la pose du visage ainsi que le suivi des composantes faciales, les résultats suivants illustrent l'utilisation du KLT pour un suivi de points entourant les deux yeux, le nez et la bouche comme montré sur les figures suivantes. Les déplacements de ces points sont calculés par KLT. A partir de ces déplacements nous calculons les coordonnées homogènes de ces points. La matrice de l'homographie est calculée par la décomposition en valeurs singulières (SVD) de la matrice A du système (4.17). Le mouvement des différents points de contrôle sur la séquence suivante est composé d'une rotation autour de l'axe z plus une translation.

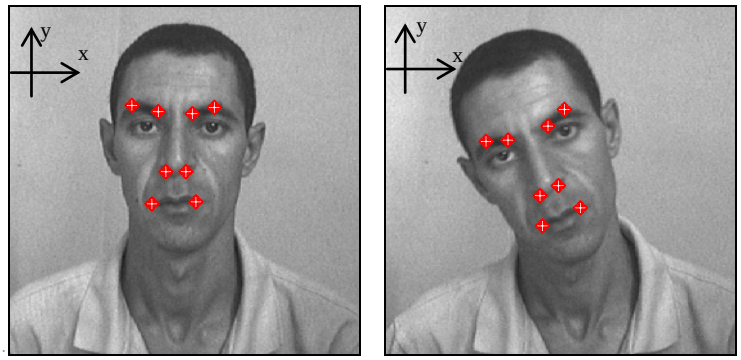


Figure (IV.6) : Deux poses du visage le long d'une séquence vidéo.

Les mesures faites manuellement et celles calculées par l'algorithme de l'homographie sont comparées sur le tableau (IV.1). Sur ce tableau nous remarquons l'effet du nombre de points sur la précision du calcul de l'angle de la pose. Avec un nombre de huit points on obtient une meilleure précision par rapport à quatre points.

Angle mesuré (degrés)	Erreur sur l'angle estimé (4 points)	Erreur sur l'angle estimé (8 points))
4	1.6	1.2
11	1.2	0.6
13	2.5	0.5
17	0.5	0.1
20	2.9	1.7

Tableau (IV.1) : erreurs commises pour chaque nombre de points pris sur le visage.

Les erreurs de l'estimation des paramètres de la pose sont aussi directement liées à la précision de suivi automatique des points de contrôle, pour cela on cherche à améliorer la précision du suivi en introduisant le filtrage de Kalman dans ce qui suit.

Conclusion

Dans ce chapitre nous avons introduit les différentes notions sur l'estimation de pose ainsi que les différentes techniques utilisées dans ce contexte. L'utilisation des points caractéristiques pour l'estimation de la pose par homographie est la méthode qui convient pour les objets

texturés. Les différents paramètres de la pose sont extraits à partir de la matrice de l'homographie. RANSAC est une méthode d'optimisation utilisée pour garder les points les plus robustes au calcul de la matrice H. Les erreurs de l'estimation des paramètres de la pose sont aussi directement liées à la précision de suivi automatique des points de contrôle, pour cela on cherche à améliorer la précision du suivi en introduisant le filtrage de Kalman dans ce qui suit.

Chapitre V

Méthode de suivi basée sur le KLT combiné avec le filtrage de Kalman

V.1 Principe de la technique :

Introduction

Le KLT comme on a dit est sensible aux changements de l'éclairage de l'objet et aux déformations des points d'intérêts au cours du temps d'où donc une perte de ces derniers. L'introduction de la fonction de pondération Gaussienne a pour effet le filtrage (dans le domaine spatial) et l'élimination des bruits dans l'image. Les points d'intérêts peuvent aussi perdre leur saillance par cause de déformation ou l'occlusion. Le filtre de Kalman est un outil qui permet le filtrage et la prédiction comme on a vu dans la section précédente. Donc c'est un outil qui peut assister le KLT pour un suivi plus robuste des points choisis. Dans ce chapitre on va présenter un schéma de suivi en combinant entre le KLT et le filtre de Kalman.

Principe :

Dans un premier temps on va utiliser le filtre de Kalman standard pour le filtrage de la trajectoire des points. Dans la phase initialisation, les points à suivre sont sélectionnés automatiquement ou manuellement. Les positions initiales des points sont utilisées pour l'initialisation du filtre de Kalman standard. Par suite le KLT nous retourne le déplacement des points entre deux images successives. Ce déplacement est supposé être entaché par un bruit blanc. Pour filtrer le bruit, en ajoute en aval un filtre de Kalman Standard comme indiqué sur la figure (V.1).

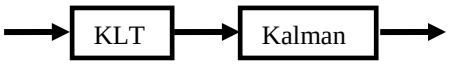


Figure (V.1) : Principe de la combinaison KLT avec Kalman.

Dans nos essais on a constaté une divergence des points suivis, pour cela on a modifié le schéma de suivi en rajoutant un bloc qui va choisir entre les positions calculées directement par KLT et les positions à la sortie du filtre de Kalman et ceci par le calcul de la différence carrée $(I_t - I_{t-1})^2$ sur la fenêtre du point suivi pour les deux positions. Le choix se fait selon l'erreur la plus petite.

Le schéma complet de l'algorithme se présente comme par la figure (V.2) ;

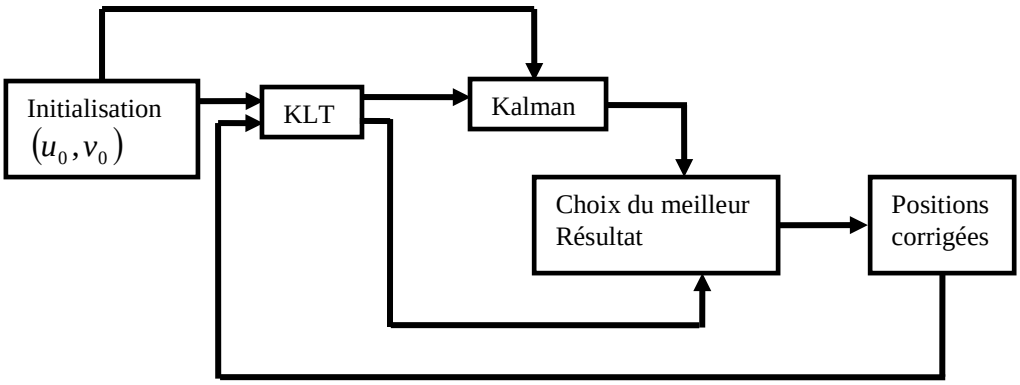


Figure (V.2) : Introduction du filtre de Kalman dans l'algorithme KLT (schéma complet).

V.2 Modélisation du mouvement du visage :

Le mouvement du visage humain est considéré comme un système dynamique avec une accélération constante. Cette modélisation est utilisée dans beaucoup de travaux et approuvée dans plusieurs articles [22]. La vélocité peut être obtenue dans ce cas par les étapes de correction et de prédiction du filtre de Kalman. La position P du point suivi est bruitée par un bruit blanc elle est donnée sous forme : $P = P' + \eta$ où P' et la position réelle et η représente le bruit de la mesure il est supposé être blanc et une moyenne nulle.

Les équations du modèle du mouvement sont comme suit :

$$x_k = A.x_{k-1} + B.u_{k-1} + w_k$$

Le modèle de mesure décrit l'information fournie par le ou les capteurs en une équation liant les paramètres de l'état de la mesure et du bruit. L'équation de mesure ou d'observation est donnée par :

$$z_k = H.x_k + v_k$$

La position P contient les coordonnées spatiales du point ; $P = (U, V)$ sur l'image. Le vecteur d'états du système et donné comme suit ;

$$x = \left[U \quad V \quad \dot{U} \quad \dot{V} \quad \ddot{U} \quad \ddot{V}\right]^T$$

où

$\dot{U}$ Et $\dot{V}$ sont les vélocités du point suivant les directions respectivement U et V .

$\ddot{U}$ Et $\ddot{V}$ sont les accélérations du point suivant les directions respectivement U et V .

La matrice A du modèle de mouvement est donnée ;

$$A = \begin{bmatrix} 1 & 0 & dt & 0 & \frac{1}{2}dt^2 & 0 \\ 0 & 1 & 0 & dt & 0 & \frac{1}{2}dt^2 \\ 0 & 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 0 & 1 & 0 & dt \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A partir de cette matrice et de l'équation d'état en remarque que c'est un modèle linéaire et à une accélération constante. La commande B ici est nulle. C'est un système sont commande.

$$x_k = A.x_{k-1} + w_k$$

La mesure est linéaire. Elle est liée à l'état du système par la matrice H suivante :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Le bruit w qui affecte la mesure est supposé être Gaussien avec une moyenne nulle.

$$\Gamma = \begin{bmatrix} \frac{1}{2} & 0 & 1 & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 & 0 & 1 \end{bmatrix}^T$$

L'algorithme suivant nous résume le KLT combiné avec le filtre de Kalman pour réduire les erreurs commises par le KLT seul.

1. Initialisation du filtre de Kalman ;
2. Initialisation du KLT ;
3. Mesure de la position actuelle par KLT ;
4. Correction de la position actuelle par Kalman ;
5. Choisir le meilleur résultat entre les étapes 3 et 4.
6. Mise ajour du KLT ;

L'initialisation se fait manuellement mais l'initialisation automatique est possible [39]. Les points sont choisis parmi les meilleurs qui peuvent être suivis dans le temps.

V.3 Réglage du filtre de Kalman :

Les paramètres initiaux du filtre sont un facteur important. Les valeurs utilisées dans notre expérimentation sont choisies pour des résultats considérés comme meilleurs. Les valeurs suivantes sont utilisées pour obtenir les résultats présentés dans ce qui suit.

$$Q = 0.5 \times \begin{bmatrix} 0.25 & 0.25 & 0.50 & 0.50 & 0.50 & 0.50 \\ 0.25 & 0.25 & 0.50 & 0.50 & 0.50 & 0.50 \\ 0.50 & 0.50 & 1.00 & 1.00 & 1.00 & 1.00 \\ 0.50 & 0.50 & 1.00 & 1.00 & 1.00 & 1.00 \\ 0.50 & 0.50 & 1.00 & 1.00 & 1.00 & 1.00 \\ 0.50 & 0.50 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix} , \quad R = \begin{bmatrix} 1.0 & 0.1 \\ 0.1 & 1.0 \end{bmatrix}$$

Bien que ces valeurs donnent de bons résultats sur les séquences suivantes, pour d'autres points d'autres séquences les paramètres ne seront pas les même. C'est l'inconvénient majeur de cet algorithme.

V.4 Résultats :

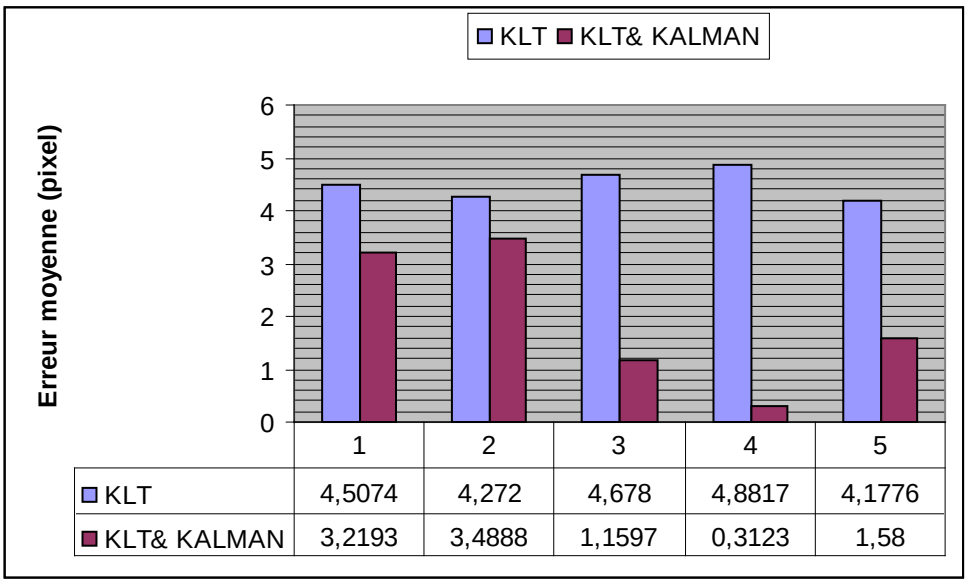
Les résultats de l'application de l'algorithme de KLT combiné avec Kalman sont montrés dans ce qui suit. Dans un premier temps on teste l'algorithme sur la séquence Hôtel puis sur la séquence du visage animé. Chacune de ces deux séquences possède un mouvement global spécifique. La séquence de l'hôtel est composée essentiellement d'un mouvement de rotation sur 3D. Des déformations importantes autour des points dues au mouvement de rotation de l'hôtel. Pour la dernière séquence qui est le visage modèle le mouvement est variable et plein d'expressions faciales avec différentes déformation est différentes illuminations autour des point suivis.

Expérience 1 :

Dans la séquence Hôtel, figure (V.3), en prend cinq points parmi les meilleurs sélectionnés par l'algorithme de Harris.

Figure (V.3): Séquence de l'hôtel

L'histogramme de la figure (V.4) nous montre l'erreur moyenne en pixel commise pour chaque point durant la séquence. Sur l'histogramme on voit bien que les erreurs moyennes données par KLT&KALMAN sont inférieures par rapport aux erreurs moyennes données par KLT seul.



	1	2	3	4	5
KLT	4,5074	4,272	4,678	4,8817	4,1776
KLT& KALMAN	3,2193	3,4888	1,1597	0,3123	1,58

Figure (V.4): Graphe des erreurs de suivi.

Les résultats de l'application des deux algorithmes sont illustrés ci-dessous. Sur l'histogramme ci-dessous on remarque que pour tous les points l'erreur moyenne ne dépasse pas la largeur de la fenêtre qui est de 5 pixels. Cette largeur de 5 pixels est prise comme limite pour déclarer la perte du point.

Séquence du visage animé (Expérience 2) :

Les douze points sont choisis comme décrit dans ce qui suit. Cette configuration des points nous permet de suivre le visage ainsi que le suivi des composantes faciales ; les yeux, le nez et la bouche. Le suivi des composantes faciales est une étape essentielle pour d'autres disciplines comme l'analyse des expressions faciales, le lip-reading, le suivi de l'œil ...etc. sur la figure (V.5) suivante on voit les douze points répartis sur le visage comme suit :

- chaque groupe de trois points couvre les sourcils gauche et droit ce qui permet le suivi des deux yeux.

- deux points un pour chaque narine généralement ce sont les points les plus stable et plus robuste pour le suivi.

- Un groupe de quatre points cadre la bouche. Un sur chacun des deux coins, un est au milieu de la lèvre supérieure et un sur le milieu de la lèvre inférieure.

Cette configuration présente une symétrie selon la composition et l'emplacement des composantes du visage ce qui permettra établir et de prévoir d'autres stratégies (analyse de constellation) pour performer le suivi.

Figure (V.5) : Répartition des 12 points avec Symétrie sur le visage

Le tableau suivant nous renseigne sur les erreurs moyennes (en pixels) commises par les deux algorithmes pour chaque point durant toute la séquence. On voit bien que la majorité des points sont bien suivi par le KLT combiné avec KALMAN. Le premier groupe (point 2, 3 et 4) qui couvre le sourcil droit marque un meilleur résultat et une meilleure précision avec le KLT&KALMAN. Même chose pour le deuxième sourcil qui est couvert par les trois points (1, 5 et 6) une différence nette sur les erreurs de localisation.

point	KLT	KLT&KALMAN
1	4,9654	4,2969
2	4,5324	3,6195
3	6,3896	2,2988
4	4,1970	2,7516
5	6,1162	2,1122
6	3,7739	2,2590
7	2,9755	1,7886
8	3,4636	2,4760
9	3,0097	3,3551
10	1,6121	2,2327
11	6,4895	3,3002
12	2,9902	2,3339

Tableau (V.1) : Erreurs (en pixels) commise pour chacun des deux algorithmes.

Les deux narines sont bien suivit aussi par le KLT&KALMAN. Les erreurs moyennes sont réduites de 2,9755 pixel à 1,7886 pixel pour le point 7 (narine gauche) et de 3,4636 à 2,4760 pixel pour le point 8 (narine droite).

Pour les quatre points (points : 9, 10, 11 et 12) qui suivent la bouche il y a une réduction de l'erreur moyenne amélioration pour les deux point de la mi-lèvre haut et bas. Pour les deux points des coins (points 9 et 10) les résultats donnés par l'algorithme de KLT standard sont meilleurs que ceux donnés par le KLT&KALMAN à cause des déformations importantes de ces deux points.

Les erreurs montrées sur le tableau (V.1) sont rapportées sur l'histogramme de la figure (V.6) pour beaucoup plus d'illustration. Si on considère que le seuil de perte d'un point et la taille de la fenêtre qui est dans notre cas 5 pixels, les points 3, 5 et 11 sont perdus par le KLT standard alors qu'aucun point n'est perdu par l'algorithme KLT&KALMAN.

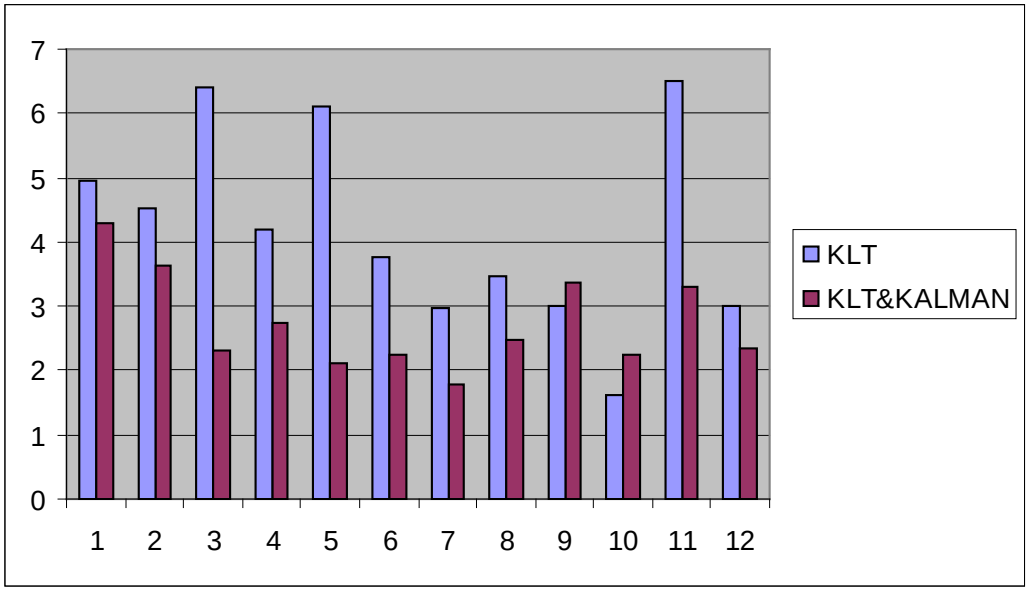


Figure (V.6): Graphe des erreurs de suivi (en pixels).

Les figures suivantes montrent les erreurs de localisation pour chacun des douze points à chaque image le long de la séquence.

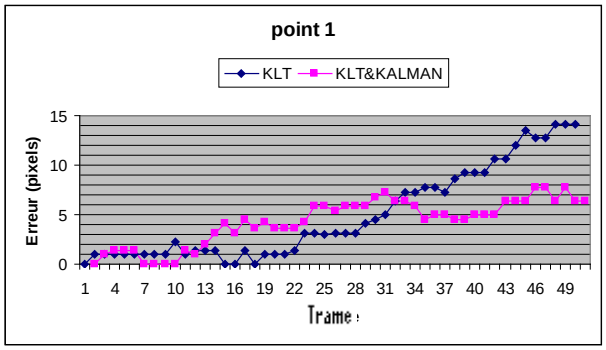


Figure (V.7a)

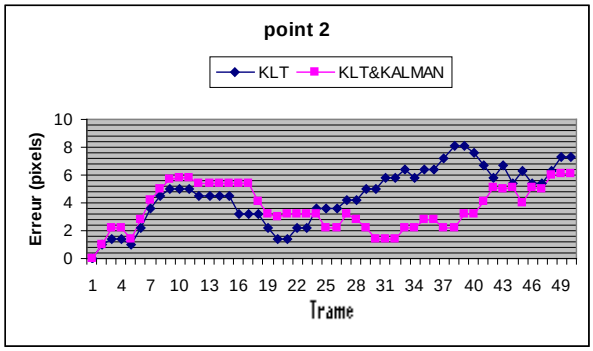


Figure (V.7b)

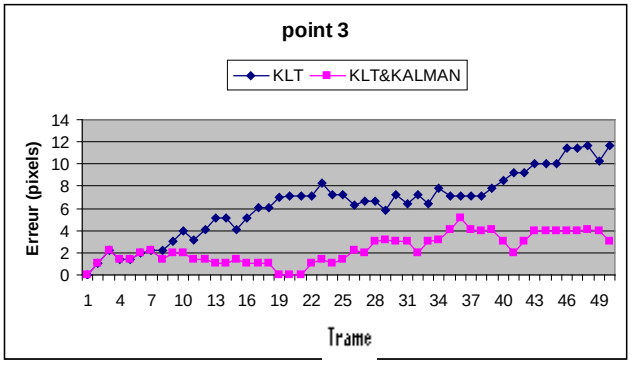


Figure (V.7c)

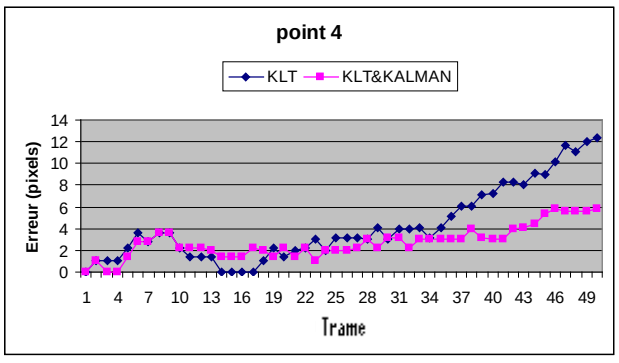


Figure (V.7d)

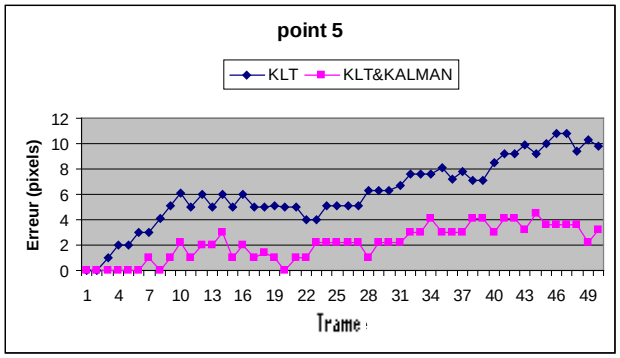


Figure (V.7e)

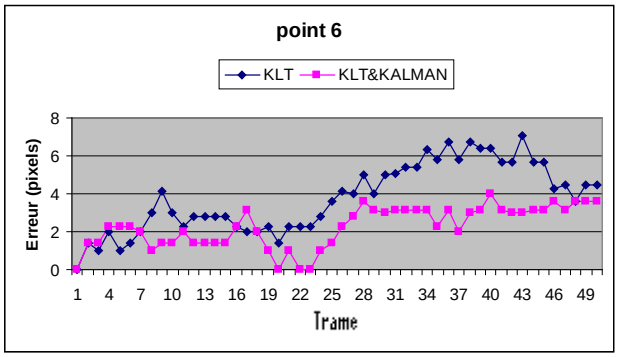


Figure (V.7f)

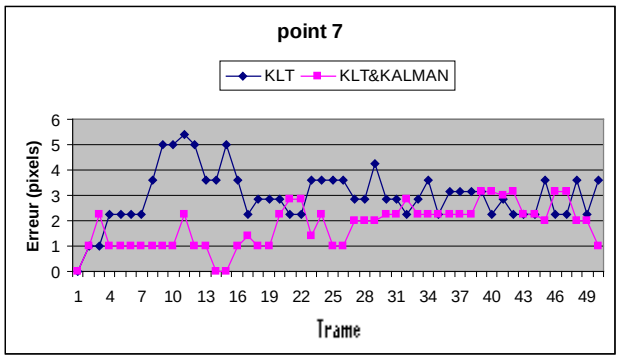


Figure (V.7g)

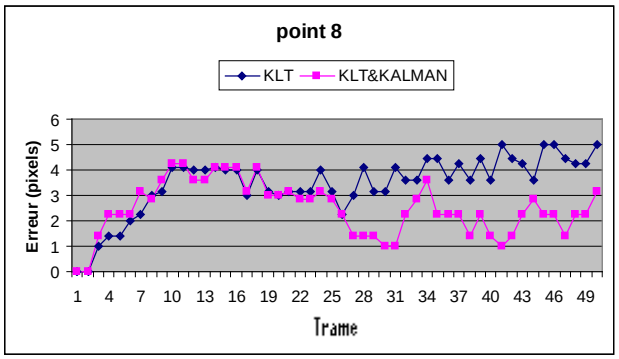


Figure (V.7h)

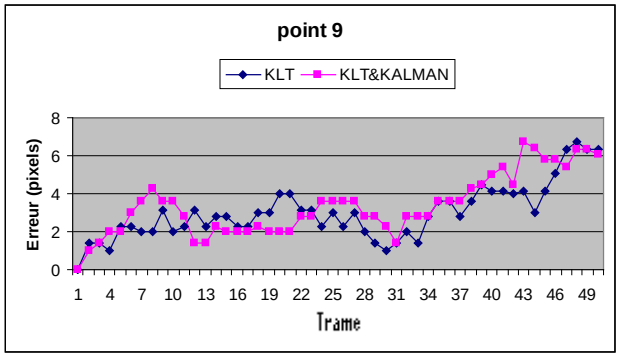


Figure (V.7i)

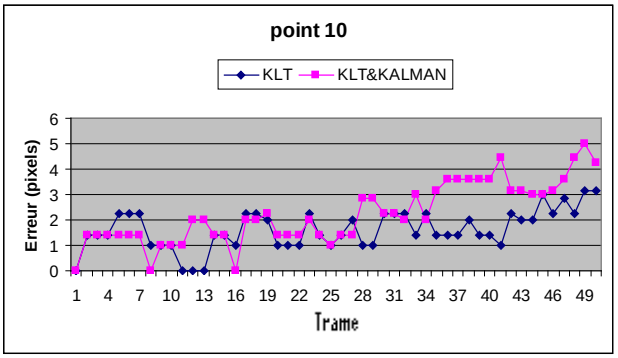


Figure (V.7j)

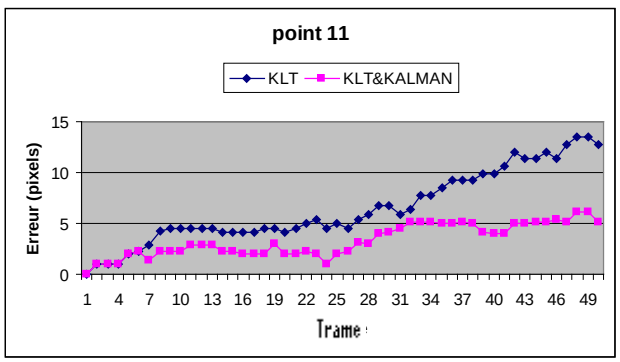


Figure (V.7k)

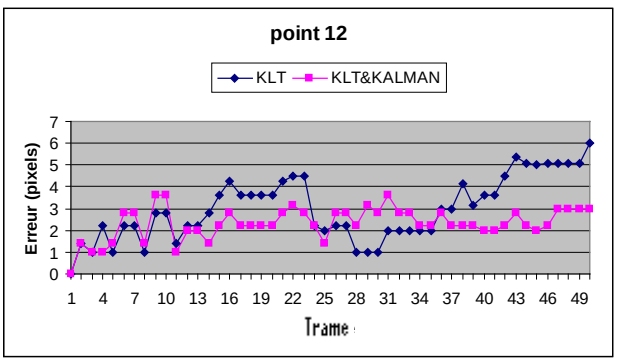


Figure (V.7L)

Dans une première analyse on remarque que pour la plupart du temps les courbes données par l'algorithme de KLT&KALMAN viennent au dessous des courbes réalisées par l'algorithme de KLT. En générale les points sont perdus rapidement par le KLT avant l'algorithme KLT&KALMAN ce qui améliore la durée de suivi d'un point. L'algorithme KLT&KALMAN possède des avantages par rapport au KLT mais des inconvénients sont à prendre en considération. Les avantages sont la précision et la stabilité. L'introduction du filtre de Kalman implique un temps de calcul en plus et va influencer sur le temps d'exécution ce qui est un inconvénient surtout dans le cas où le nombre de points pris et important. En plus de ça les paramètres du filtre de Kalman sont difficiles à définir, pour deux différents points les paramètres de réglage du filtre de Kalman sont différents.

Conclusion

Après cette initiative d'améliorer les résultats de suivi des points en introduisant le filtrage de Kalman, nous avons mis au point un algorithme qui corrige les positions courantes des points suivis. A partir des différents graphes on remarque que les résultats sont nettement améliorés par rapport à l'utilisation du KLT seul. On constate que sur les images de la séquence de l'hôtel les résultats sont meilleurs que ceux obtenus sur la séquence du visage. Cela est dû à la structure (le voisinage) des points choisis. Sur la séquence de l'hôtel les fenêtres correspondent pratiquement à des jonctions en L et X, alors que sur la séquence du visage animé le voisinage des points suivis est très variable. Les paramètres du filtre de Kalman sont difficiles à fixés et variable d'un point à un autre, dans nos essais nous avons abouti à des paramètres d'initialisation du filtre mais qui donnent des résultats positifs que pour notre séquence. Pour cela nous proposons d'introduire le filtre de Kalman étendu ou d'autres techniques pour aboutir à de meilleurs résultats.

Conclusion générale

La vision par ordinateur occupe un espace de recherche de plus en plus important dans les Sciences de l'Ingénieur car elle devient un carrefour incontournable tant ses domaines d'utilisation sont variés. Les recherches se portent de plus en plus sur des problématiques concrètes et dans des cadres naturels. Le travail présenté dans ce mémoire s'inscrit dans cette perspective. Nous avons présenté dans ce mémoire un système de suivi de points caractéristiques basé sur l'algorithme de Lucas et Kanade. Pour impliquer cette méthode dans le suivi des caractéristiques faciales et du mouvement du visage, on s'est intéressé dans la première partie de l'application au suivi du mouvement de la tête d'une personne se présentant devant une caméra. Pour cela nous avons étudié, implémenté cet algorithme puis nous avons choisi les paramètres adéquats pour son bon fonctionnement. Les paramètres les plus importants sont le choix de la fenêtre à suivre ainsi que sa dimension pour parer au problème de la déformation du point suivi, aussi le choix de la fonction de filtrage des bruits qui affectent l'image, noyau Gaussienne ou LoG. Nous avons présenté les résultats de suivi de visage sur différentes séquences qui contiennent des mouvements de translation et de rotation de la tête. Les résultats obtenus par la méthode KLT sont comparés à une autre méthode qui est le suivi par corrélation. La comparaison est faite sur des graphes pour différentes tailles de la fenêtre du point caractéristique. La comparaison montre que les résultats obtenus par KLT ont la meilleure précision et meilleure durée de suivi. Dans le contexte d'une application pour l'estimation de pose et le suivi des composantes faciales une constellation de points est choisie de façon à couvrir leur mouvement ainsi que le calcul de l'homographie pour la détermination de la pose du visage. Les résultats obtenus avec des écarts par rapport à ceux calculés manuellement conduisent à des erreurs acceptables. La précision dans le calcul des positions des points d'intérêt est le facteur qui détermine le meilleur suivi du visage ainsi que la meilleure estimation des paramètres de la pose. Dans ce travail nous avons présenté une technique de suivi basée sur la combinaison du KLT et le filtrage de Kalman, on a apporté une amélioration dans la précision de calcul des déplacements des points d'intérêt par rapport au KLT seul. Plusieurs paramètres sont pris en considération tels que la taille de la fenêtre du point d'intérêt et les paramètres d'initialisation du filtre de Kalman. L'inconvénient de cette méthode est la difficulté rencontrée lors de l'initialisation du filtre de Kalman. Pour cela on a choisi de mettre dans nos perspectives la combinaison entre la méthode de Lucas-Kanade et le filtrage de Kalman étendu pour améliorer le suivi.

Références

Références

[1] J.J.L Wang, S.Singh, “Video analysis of human dynamics”, Real-Time Imaging 9, pp. 321–346, UK 2003.

[2] V. Girondel, “contribution à l’analyse et à l’interprétation du mouvement humain : application à la reconnaissance de postures”, Thèse doctorat, INPG, Grenoble 2006.

[3] T.Acharia, A.K.Ray, “Image Processing: Principals and Applications”, Wiley-Interscience publishing company, 2005.

[4] R.J. Qian, M. Ibrahim, S.Kristine, E. Matthews, “A Robust Real-Time Face Tracking Algorithm”, IEEE International Conference on Image Processing (ICIP'98) - Volume 1
pp. 195, 1998.

[5] K. Schwerdt, J.L. Crowley, “Robust Face Tracking using Color”, pp. 90 Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), 2000.

[6] J.C. Terrillon, S. Akamatsu, “Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images”, Vision ’99, Canada, pp. 19-21 may 1999.

[7] Y. Wu, Q. Liu, T. S. Huang, “Robust Real-time Human Hand Localization by Self-Organizing Color Segmentation”, IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999.

[8] J. Yang, W. Lu, A. Waibel, “Skin-Color Modeling and Adaptation”, Technical Report CMU-CS-97-146, Carnegie Mellon University, May 1997.

[9] K. Sobottka, I. Pitas, “A novel method for automatic face segmentation facial feature extraction and tracking”, published in Signal Processing: Image Communication, Vol. 12, No. 3, pp. 263-281, University of Thessaloniki, Greece 1998.

[10] T.F. Cootes and C.J. Taylor, “Active shape models – ‘Smart Snakes’”, Proc. BMVC, Leeds, U.K., pp. 266-275, 1992.

[11] P. Tissainayagama, D. Suter, “Assessing the performance of corner detectors for point feature tracking applications”, Image and Vision Computing 22 pp. 663–679, Australia February 2004.

[12] R. Deriche, G. Giraudon,. Accurate corner detection: An analytical study, Proc. 3rd Int. Conf. on Computer Vision, Osaka, pp. 66–70, Japan 1990.

[13] R. Ceccarelli, “Pedestrian head detection using automatic scale selection for feature selection and statistical edge curvature analysis”, Doctorat Thesis, Lausanne, September 2004.

[14] L. Bretzner, “Multi-Scale Feature Tracking and Motion Estimation”, Doctorat Thesis, University of Stockhholms, October 1999.

[15] Y.S. Chen, Y.P. Hung, C.S. Fuh, “Fast Block Matching Algorithm Based on the Winner-Update Strategy ”, IEEE transactions on image processing, vol. 10, no. 8, august 2001.

[16] F. Dornaika, J. Ahlberg,” Fitting 3D face models for tracking and active appearance model training”, Image and Vision Computing 24, pp. 1010–1024, 2006.

[17] M. S. Nixon, A. S. Aguado, “Feature Extraction And Image Processing”, Newnes publishing company, 2002.

[18] M. Borgetto, “contribution à la construction de mosaïques d’images sous-marines géo-référencées par l’introduction de méthodes de localisation », thèse doctorat, Université du Sud Toulon-Var, Avril 2005.

[19] M. Krinidis, N. Nikolaidis and I. Pitas, “feature-based tracking using 3D physics-based deformable surfaces”, University of Thessaloniki, Greece 2005.

[20] Meghna Singh, Mrinal Mandal, Anup Basu, ” Robust KLT Tracking with Gaussian and Laplacian of Gaussian Weighting Functions”, University of Alberta, Canada 2003.

[21] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” Technical Report, TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, December 1995.

[22] Mathias Kolsch and Matthew Turk, “Flocks of Features for Tracking Articulated Objects”, In Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, May 2004.

[23] Ayman ZUREIKI, « Fusion de Données Multi-Capteurs pour la construction Incrémentale du Modèle Tridimensionnel Texturé d’un Environnement Intérieur par un Robot Mobile”, Thèse Doctorat, Université de Toulouse, Septembre 2008.

[24] Vincent Lepetit and Pascal Fua, “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey », Foundations and Trends in Computer Graphics and Vision Vol. 1, No 1 (2005) pp. 1–89.

[25] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, « Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework», IEEE Transactions on Visualization and Computer Graphics, Vol. 12, no. 4, July/August 2006.

[26] J. Aggarwal, Q. Cai. “Human motion analysis: A review”, Computer Vision and Image Understanding: CVIU, 73(3):428–440, 1999.

[27] G.D.Hager and P.N.Belhumeur, “Efficient Region Tracking with Parametric Models of Gemotry and Illumination,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 10, pp. 1,025-1,039, Nov. 1998.

[28] Eric BRUNO, « De l’estimation locale à l’estimation globale de mouvement dans les séquences d’images », thèse doctorat, Université Joseph Fourier Grenoble, novembre 2001.

[29] Bruce D. Lucas, T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision”, Proceedings of Imaging Understanding Workshop, pp. 121-130, Carnegie-Mellon University, 1981.

[30] C. Tomasi T. Kanade, « Detection and Tracking of Point Features”, Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[31] C. Tomasi T. Kanade, “Good features to track”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Seattle, June 1994.

[32] S.M. Smith, « SUSAN _ A New Approach to Low Level Image Processing”, Technical Report TR95SMS1c, Université d’Oxford UK, 1995.

[33] Dmitry Chetverikov, « Applying Feature Tracking to Particle Image Velocimetry », Computer and Automation Research Institute Budapest, Kende u.13-17, H-1111 Hungary, November 2002.

[34] F. Bourel, C. C. Chibelushi, A. A. Low, ‘Robust Facial Feature Tracking’, Proceedings of the Eleventh British Machine Vision Conference, pp. 232 – 241, Vol. 1, Sept. 2000.

[35] C. Harris, M. Stephens “A Combined Corner Detection and detector” Proceeding of the Alvey Vision conference, pp. 147-151, 1988.

[36] O. Faugeras “Three-Dimensional Computer Vision: A Geometric Viewpoint”, MIT Press, 1993.

[37] E. Francois, P. Bouthemy, “multiframe-based identification of mobile components of scene with a moving Camera”, IEEE conference on computer Vision and Pattern Recognition, CVPR’91, pp. 166-172, 1991.

[38] G. Zhu, Q. Zeng, C.Wang, “Efficient edge-based object tracking”, Communication, Pattern Recognition April 2006.

[39] P. Gejguš, M. Šperka, “Face tracking for expressions simulations”, International Conference on Computer Systems and Technologies - CompSysTech’2003.

[40] M. Pressiout, “Approches hybrides pour le suivi temps-réel d’objets complexes dans des séquences vidéo”, thèse doctorat, Université de Rennes I, Décembre 2006.

[41] http://www-eph.int-evry.fr, mars 2007.