الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – Biskra
Faculté des Sciences et de la Technologie
Département : Génie Electrique
Ref :……

جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم: الهندسة الكهربائية
المرجع:.....

Thèse présentée en vue de l'obtention
du diplôme de

Doctorat en sciences en : Automatique

Détection et suivi d'objets

Présentée par :

MEDOUAKH Saadia

Soutenue publiquement le 23 Juin 2019

Devant le jury composé de :

Pr. OUAFI Abdelkrim	**Professeur**	**Président**	**Université de Biskra**
Pr. BOUMEHRAZ Mohamed	**Professeur**	**Rapporteur**	**Université de Biskra**
Pr. BENOUDJIT Nabil	**Professeur**	**Examinateur**	**Université de Batna**
Pr. BENZID Redha	**Professeur**	**Examinateur**	**Université de Batna**
Pr. SAIGAA Djamel	**Professeur**	**Examinateur**	**Université de M'sila**
Pr. TERKI Nadjiba	**Professeur**	**Examinateur**	**Université de Biskra**

Remerciements

Tout d'abord je remercie ***Allah*** *de tout mon cœur de m'avoir donné le courage et la patience qui m'ont permis d'accomplir ce modeste travail.*

Je tiens à exprimer mes vifs remerciements au ***Prof. BOUMEHRAZ Mohamed*** *de m'avoir soutenu et fait confiance durant ces années avec une grande patience.*

Mes remerciements vont également aux membres de jury d'avoir accepté d'examiner et d'évaluer ce travail :

Mr. OUAFI Abdelkrim, *professeur à l'université de Biskra et président du jury.*

Mr. BENOUDJIT Nabil, *professeur à l'université de Batna.*

Mr. BENZID Redha, *professeur à l'université de Batna.*

Mr. DJAMEL Saigaa, *professeur à l'université de M'sila.*

Mme. TERKI Nadjiba, *professeur à l'université de Biskra.*

Je tiens à remercier spécialement ***Mme Nadjiba*** *pour ses conseils et ses encouragements qui m'ont aidé beaucoup.*

Finalement, je remercie ***ma famille*** *et mes amis* ***Khiera, Meriem*** *pour leur patience, leur encouragement, leur soutien qui m'a été bien utile durant ma thèse.*

Dédicaces

Je dédie ce modeste travail :
A mes chers parents
A mes frères et sœurs
A toutes mes amies
A vous

Saâdia

Résumé

Au cours de ces dernières décennies, la détection et le suivi d'objets ont attiré beaucoup d'intérêt en raison de leurs diverses applications dans la vie humaine, en particulier la vidéosurveillance et la robotique. Le suivi d'objet est l'estimation de la localisation d'un objet et la détermination de sa trajectoire au cours du temps dans une séquence vidéo. Bien que de nombreux algorithmes de suivi aient été développés ces dernières années pour sa résolution, il demeure un problème non résolu à cause du nombre élevé de facteurs environnementaux. Dans ce travail de thèse, nous nous intéressons spécifiquement à l'étude et à l'amélioration de l'algorithme de suivi Mean shift, qui est l'un des algorithmes de suivi les plus efficaces pour les applications en temps réel, en raison de sa simplicité et de sa robustesse. Bien qu'il soit robuste à l'occultation partielle, la rotation, le mouvement de fond et la déformation non-rigide de la cible, il est très sensible aux changements d'échelles, aux occultations importantes, et il peut échouer en présence d'un autre objet de couleurs similaires, ou de fond de couleurs similaires ou dans le cas de grands déplacements, parce qu'il est basé sur l'histogramme de couleur pour représenter le modèle de l'objet cible. La première partie de cette thèse est consacrée à l'étude et à l'analyse des effets de l'utilisation de différentes configurations d'espace de couleurs, sur l'efficacité et la qualité du suivi en utilisant le tracker Mean shift qui se base sur l'information de couleur pour construire le modèle d'apparence de l'objet cible. Dans notre étude, nous avons utilisé les informations intrinsèques de tracker Mean shift : le coefficient de Bhattacharyya et la carte de rétroprojection pour comprendre l'influence des espaces de couleurs sur la performance de ce tracker. Les résultats obtenus sur les bases de données OTB 2013, OTB 2015, en utilisant les espaces de couleurs les plus utilisés dans la vision par ordinateur, et confirment que l'espace de couleur HSV donne une représentation robuste de l'objet cible dans la plupart des séquences vidéos. Dans la deuxième partie, nous proposons une nouvelle approche qui a pour but d'améliorer l'efficacité et la robustesse de tracker Mean shift par la combinaison des informations couleurs avec les informations spatiales, afin de construire le modèle d'apparence de l'objet cible. Le tracker Mean shift utilise seulement l'histogramme de couleur RGB pour modéliser l'objet cible, ceci rend ce tracker incapable de détecter l'information spatiale de chaque pixel dans l'image, et ne peut pas distinguer entre la cible et le fond lorsqu'ils sont similaires. Pour surmonter ce problème nous proposons une nouvelle représentation de modèle d'apparence de cible et qui se base sur une mixture des caractéristiques de couleur HSV et de texture LPQ, LBP ou BSIF pour construire l'histogramme pondéré conjoint couleur-texture. Nous avons évalué nos histogrammes conjoints couleur-texture, de manière étendue sur deux bases de donnés citées précédemment. Les résultats obtenus et leurs comparaison avec le traditionnel algorithme Mean shift et les trackers de l'état de l'art montrent la robustesse et la précision de ces histogrammes conjoints, ainsi ils sont capables de gérer certains défis liés au suivi, à savoir le flou de mouvement, la similarité entre l'objet cible et le fond et les changements d'illuminations.

Mots clés : Suivi d'objet visuel, L'algorithme Mean shift, Histogramme conjoint couleur-texture, Espaces de couleurs, Descripteurs locaux LPQ, LBP et BSIF, Vision par ordinateur.

Abstract

In recent decades, object detection and tracking has attracted a lot of interest because of its diverse applications in human life, especially video surveillance and robotics. Object tracking is the estimation of the location of an object and the determination of its trajectory over time in a video sequence. Although, many object tracking algorithms have been developed in recent years, it remains an unresolved problem because of the high number of environmental factors. In this thesis work, we are specifically interested in the study and improvement of the Mean shift tracking algorithm, which is one of the most effective tracking algorithms for real-time applications, due to its simplicity and robustness. Although it is robust to partial occlusion, rotation, background motion and non-rigid deformation of the target, it is very sensitive to scale changes, full occlusion, and it can fail in the presence another object of similar colors, or background of similar colors or in the case of large displacements, because it is based on the color histogram to represent the model of the target object. The first part of this thesis is devoted to study and to analysis the effects of using different color space configurations, on the efficiency and the quality of the tracking using the Mean shift tracker which is based on color information to build the appearance model of the target object. In our study, we used the intrinsic information of tracker Mean shift: the Bhattacharyya coefficient and the back projection map to understand the influence of the color spaces on the performance of this tracker. The results obtained on the OTB 2013, OTB 2015 databases, using the most used color spaces in computer vision, confirm that the HSV color space gives a robust representation of the target object in most video sequences. In the second part, we propose a new approach that aims to improve the efficiency and robustness of the Mean shift tracker by combining color and spatial information, in order to build the object's appearance model. The Mean shift tracker uses only the RGB color histogram to model the target object; this makes this tracker unable to detect the spatial information of each pixel in the image, and cannot distinguish the target from the background when the target has similar appearance to the background. To overcome this problem we propose a new target appearance model representation based on a mixture of HSV color characteristics and LPQ, LBP or BSIF texture to construct the joint color-texture weighted histogram. We evaluated our joint color-texture histograms, extensively on two databases previously mentioned. The results obtained and their comparisons with the traditional Mean shift algorithm and the state of the art trackers show the robustness and the precision of these joint histograms, as well as they are able to handle some challenges related to tracking as the motion blur, the similarity between the target object and the background and changes in illuminations.

Keywords: Visual object tracking, Mean shift algorithm, Joint color-texture histogram, Color spaces, Local descriptors LPQ, LBP and BSIF, Computer vision.

ملخص

في العقود الأخيرة، اجتذب اكتشاف وتتبع الأهداف الكثير من الاهتمام بسبب تطبيقاتها المختلفة في حياة الإنسان، خاصة المراقبة بالفيديو والروبوتات. تتبع الهدف هو تقدير موقع الهدف وتحديد مساره مع مرور الوقت في تسلسل الفيديو.على الرغم من أن العديد من خوارزميات التتبع قد تم تطويرها في السنوات الأخيرة من أجل حلها، إلا أنها تظل مشكلة غير محسومة بسبب العدد الكبير من العوامل البيئية المؤثرة. في هذه الأطروحة، نحن مهتمون بشكل خاص بدراسة وتحسين خوارزمية متوسط التتبع Mean shift، التي تعد من أكثر خوارزميات التتبع كفاءة في تطبيقات الوقت الحقيقي،و ذلك بسبب بساطتها ومتانتها. على الرغم من أنها قوية للانسداد الجزئي، والدوران، وحركة الخلفية وتشوه الهدف الغير جامد، إلا أنها حساسة للغاية لتغيير الحجم، والانسداد الكبير، وقد تفشل في وجود كائن آخر من ألوان متشابهة، أو خلفية من ألوان متشابهة أو في حالة نزوح كبير، و هذا بسبب اعتمادها على الرسم البياني للون لتمثيل نموذج الكائن المستهدف. الجزء الأول من هذه الأطروحة مخصص لدراسة وتحليل تأثيرات استخدام مختلف تكوينات فضاء اللون على كفاءة وجودة المتابعة باستخدام المتعقب Mean shift الذي يعتمد على معلومات اللون لبناء نموذج المظهر للكائن الهدف. في دراستنا استخدمنا المعلومات الجوهرية للمتعقب Mean shift : معامل Bhattacharyya وخريطة الإسقاط العلوي لفهم تأثير مساحات اللون على أداء هذا المتعقب. النتائج التي تم الحصول عليها في قواعد البيانات OTB باستخدام مساحات اللون الأكثر استخدامًا في رؤية الكمبيوتر و المقارنة فيما بينها تؤكد أن فضاء اللون HSV تعطي تمثيلًا قويًا للكائن المستهدف في معظم تسلسلات الفيديو. في الجزء الثاني، نقترح نهجا جديدا يهدف إلى تحسين كفاءة ومتانة المتعقب Mean shift من خلال الجمع بين المعلومات اللونية و المكانية، من أجل بناء نموذج الكائن الهدف. في الجزء الثاني، نقترح نهجا جديدا يهدف إلى تحسين كفاءة ومتانة المتعقب Mean shift من خلال الجمع بين المعلومات اللونية و المكانية، من أجل بناء نموذج المظهر للكائن المستهدف. يستخدم المتعقب Mean shift الرسم البياني للألوان RGB فقط لنمذجة الكائن المستهدف، مما يجعل هذا المتعقب غير قادر على اكتشاف المعلومات المكانية لكل بكسل في الصورة، ولا يمكنه التمييز بين الهدف والخلفية عند التشابه. للتغلب على هذه المشكلة، نقترح تمثيلًا جديدًا لنموذج المظهر المستهدف والذي يعتمد على مزيج من خصائص ألوان HSV ونسيج LPQ و LBP أو BSIF لبناء الرسم البياني المشترك لون-نسيج. قمنا بتقييم رسومنا البيانية المشتركة لون-نسيج، على نطاق واسع على قاعدتي البيانات المذكورتان سابقا. تظهر النتائج التي تم الحصول عليها ومقارنتها مع Mean shift التقليدي وأحدث التقنيات متانة ودقة هذه الرسوم البيانية المشتركة، فضلا عن أنها قادرة على التعامل مع بعض التحديات المتعلقة بالتتبع وهي طمس الحركة، والتشابه بين الكائن المستهدف والخلفية والتغيرات في الإضاءة.

الكلمات المفتاحية : تتبع الكائن المرئي، وخوارزمية Mean shift، الرسم البياني المشترك لون-نسيج، مساحات اللون، الواصفات المحلية LPQ، LBP و BSIF ، رؤية الكمبيوتر.

Table de matière

Chapitre 2 Suivi d'objet par l'algorithme Mean shift

Chapitre 3 Etude de l'influence de l'espace couleur sur la performance du tracker Mean shift

Liste des figures

Liste des tableaux

Abréviations

SVM	Support Vector Machines
ACP	Analyse de Composantes Principales
HOG	Histogramme d'orientations de gradient
LBP	Local Binary Patterns
LTP	Motifs locaux ternaires
MB-LBP	Multi-Block LBP
KLT	Kanade-Lucas-Tomasi
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
MoG	Mixture of Gaussian
JPDAF	Joint Probability Data Association Filtering
MHT	Multiple Hypothesis Tracking
SSD	Somme des Carrés des Différences
NCC	Normalized Cross-Correlation
FoT	Flock of Trackers
Camshift	Continuously Adaptive Mean Shift
KaMS	Mean shift avec Kalman
IVT	Incremental Learning for Robust Visual Tracking
WTS	Weighted Tensor Subspace
MIL	Multiple Instance Learning Tracking
Struck	Structured Output Tracking with Kernels
TLD	Tracking Learning and Detection
FFT	Transformée de Fourier Rapide
MOSSE	Minimum Output Sum of Squared Error
DSST	Discriminative Scale Space Tracker
LCT	Long-term Correlation Tracking
TSV	Temporal Spatio-Velocity
DGT	Dynamic Graph based Tracker
HMM	Modèles de Markov cachés
SLIC	Simple Linear Iterative Clustering
SOAMST	Mean shift adapté à l'échelle et à l'orientation
ASMS	Scale Adaptive MeanShift
CBWH	Corrected Background-Weighted Histogram
KMS	L'algorithme Mean shift original
LPQ	Local Phase Quantization
BISF	Binarized Statistical Image Features
OTB	Base de données
VOT	Base de données
CLE	Center Location Error
VOR	Pascal VOC Overlap Ratio
OPP	Opponent Color Space

Introduction

La vision par ordinateur est à la base de tout système de vision artificielle. C'est une branche de l'intelligence artificielle dont l'objectif est de permettre à une machine d'analyser, traiter et comprendre une ou plusieurs images prises par un système d'acquisition (caméras, etc.). Le suivi d'objets est l'un des sujets de recherche les plus importants et les plus difficiles en vision par ordinateur et a été appliqué avec succès dans un large éventail d'applications réelles, comme la vidéosurveillance et la robotique. Le suivi d'objet est un domaine de recherche dont les premiers travaux remontent à la fin des années 80 et dont les grands progrès ont été réalisés ces dernières années. De nos jours, le suivi d'objets dans une séquence vidéo est classé parmi les sujets de recherche les plus actifs. Le but central du suivi est d'estimer au fil du temps la localisation de l'objet cible dans chacune des images d'une séquence vidéo.

Le suivi d'objets reste une tâche complexe et difficile en raison de nombreux défis liés aux limitations des capteurs de vision (faible cadence d'image, basse résolution, faible dynamique par pixel, distorsions des couleurs, bruit, etc), aux objets (objets non rigides, nombre d'objets variant au fil de temps, occultations entre objets, petites tailles d'objets, etc), aux exigences des scénarios applicatifs (fonctionnement en temps réel, haute fiabilité du système, etc) et à l'environnement (variation d'éclairage, occultations causées par l'environnement, etc). De plus, la prolifération de données vidéo et de nouveaux dispositifs d'acquisition de données ont suscité un grand intérêt pour la construction d'algorithmes de suivi plus intelligents. De nombreux algorithmes de suivi d'objet ont été proposés dans la littérature pour faire face à ces défis et pour assurer une bonne qualité de suivi, certains utilisent des modèles générateurs [1]-[5] tandis que d'autres utilisent des modèles discriminants [6]-[11]. Cependant, Il n'existe aucune méthode de suivi unique qui peut être appliquée avec succès à tous les scénarios.

Le processus du suivi d'objets peut s'exprimer en fonction de la détection des objets. Le suivi s'attache à détecter l'objet cible à chaque trame de la séquence vidéo puis à mettre en correspondance l'objet détecté à l'instant courant avec des objets détectés aux trames précédentes de façon à construire la trajectoire de l'objet. Les tâches de détection de l'objet et

d'établissement de la correspondance entre les instances d'objet entre les trames peuvent être effectuées séparément ou conjointement [12]. Le suivi d'objets fait intervenir trois étapes principales: La détection des objets candidats, la construction du modèle d'apparence et l'association des données, qui permet de déterminer la position et l'état de l'objet cible à chaque instant (pour sélectionner le meilleur objet candidat pour chaque objet cible). L'étape de construction du modèle d'apparence est une étape cruciale et affecte d'une manière directe la performance d'un système de suivi d'objets. Elle Consiste à associer à chaque objet détecté des descripteurs qui permettent de caractériser son modèle d'apparence afin de le comparer avec d'autres objets dans les trames suivantes. La validité et la robustesse des opérations de suivi dépendent de la qualité représentative de caractéristiques visuelles extraites à partir des objets cibles.

Le suivi visuel utilisant plusieurs types de caractéristiques visuelles telles que l'intensité, couleur, information spatio-temporelle, gradient et texture, a été prouvé comme une approche robuste parce que les caractéristiques pourraient se compléter les unes les autres [13]. Dans la littérature, de nombreux algorithmes de suivi [13]-[18] tentent de combiner plusieurs caractéristiques pour augmenter la précision de la représentation contre les variations d'apparence et améliorer la discrimination entre l'objet cible et le fond. Mais le choix des caractéristiques et la manière de les combiner restent des problèmes au cœur de la recherche. C'est dans ce contexte que notre étude a été développée. L'objectif principal de ce travail est de proposer une méthode de représentation de l'objet cible en combinant les caractéristiques des couleurs et des textures de manière plus distinctive et efficace pour créer un histogramme conjoint couleur-texture dans le cadre de l'algorithme Mean shift. Ainsi, une étude de l'influence des espaces de représentation de la couleur sur la performance du tracker Mean shift.

Mean shift [1] est l'un des algorithmes de suivi les plus utilisés et les plus efficaces pour les applications en temps réel, en raison de sa simplicité et de sa robustesse. Il adopte à l'histogramme pondéré de couleur pour représenter son modèle d'objet cible. Le tracker Mean shift est robuste à l'occultation partielle, la rotation, le mouvement de fond et la déformation d'objets. Cependant, il existe également de nombreuses limitations telles que le manque de l'information spatiale et le déclin des performances pour le changement d'échelle de l'objet, l'occultation complexe et lorsque la couleur de l'objet est similaire à la couleur de fond. Cela est dû à l'utilisation des caractéristiques couleurs pour construire le modèle d'apparence de l'objet cible.

Pour cette raison, de nombreux chercheurs [14][16][19][20] ont proposé des méthodes qui peuvent améliorer la convergence de tracker Mean shift dans des conditions complexes en combinant des caractéristiques de couleur et de texture pour construire l'histogramme pondéré conjoint couleur-texture. Les motifs de la texture reflètent la structure spatiale de l'objet cible, et fournissent de nouvelles informations à l'histogramme de couleur, cela rend le modèle d'apparence de l'objet plus robuste. La plupart des chercheurs ont utilisé les motifs binaires locaux (LBP) pour extraire les caractéristiques de texture, car ces techniques sont les plus utilisées pour la classification des textures dans la vision par ordinateur. Cependant, le choix du meilleur descripteur de texture et de la méthode efficace de combiner les caractéristiques de l'intensité de la couleur et de la texture est encore un problème important.

Contributions

Dans cette thèse, nous avons étudié l'influence du choix de l'espace colorimétrique aux performances de l'algorithme de suivi Mean shift et nous avons proposé une nouvelle représentation de modèle d'apparence combinant les caractéristiques de couleur et de texture pour améliorer la robustesse et la précision du tracker Mean shift dans une scène vidéo complexe. Nos contributions principales dans cette thèse sont résumées comme suit:

- Nous avons fait une étude de l'influence des espaces colorimétriques sur le suivi d'objets à l'aide de l'algorithme Mean shift puisqu'il est basé uniquement sur des informations de couleur RGB (histogramme de couleur RGB). L'utilisation de nombreux espaces colorimétriques pour construire un modèle d'apparence de l'objet cible pour des séquences d'images couleurs, a demontré que l'espace de couleur HSV rend le tracker Mean shift plus robuste et plus précise, dans la plupart de ces séquences.
- Nous avons combiné les caractéristiques de couleur HSV et les caractéristiques de texture extraites par le descripteur LPQ, le descripteur LBP et le descripteur BSIF pour construire des histogrammes conjoints HSV couleur-LPQ texture, HSV couleur-LBP et HSV couleur-BSIF texture afin d'améliorer la robustesse du tracker Mean shift, en particulier au flou de mouvement, au fond clutter et aux changements d'apparence. Ils ont été démontré que l'opérateur LPQ est robuste pour flou et surpasse l'opérateur LBP et BSIF dans la classification de la texture. Il est insensible au flou central symétrique, ce qui inclut le mouvement et le flou de la turbulence atmosphérique. Aussi, nous avons proposé une nouvelle méthode combinant les caractéristiques de texture LPQ, LBP ou BSIF avec l'histogramme pondéré de couleur pour créer un histogramme conjoint

couleur-texture plus robuste de manière plus distinctive et efficace. Les modèles d'apparences proposées peuvent exploiter efficacement l'information structurelle de l'objet cible, ce qu'est plus discriminant et insensible au flou, en particulier LPQ. Ces modèles ont été atteints des performances très élevées par rapport au modèle de couleur dans le cadre du tracker Mean shift.

Structure (Organisation) de la thèse

Cette thèse est structurée comme suit :

- **Le chapitre 1** est introductif donne une vue globale sur dans le domaine de détection et suivi d'objets. Un état de l'art des techniques de détection et de suivi d'objet et une classification des approches de suivi d'objets sont présentés, afin de montrer la diversité de la conception des approches développées.
- **Le chapitre 2** présente d'abord, un état de l'art sur l'algorithme Mean shift. Ensuite, décrive le fonctionnement de ce tracker dans le but de comprendre les différentes étapes de l'algorithme de suivi qui utilise une fonction de densité des histogrammes de couleur pour représenter le modèle et le candidat cible. Enfin, on présente le tracker Camshift et le tracker KaMS qui combine le Mean shift avec le filtre de Kalman.
- **Le chapitre 3** introduit la première contribution de cette thèse. Dans ce chapitre, nous nous intéressons à la compréhension de la construction de modèle d'apparence de l'objet, ainsi que l'étude et l'analyse les différents effets de l'utilisation de différentes configurations d'espace de couleurs sur l'efficacité et la précision de suivi en utilisant le tracker Mean shift.
- **Le chapitre 4** introduit la contribution principale de cette thèse. Dans ce chapitre, nous proposons une nouvelle méthode efficace et robuste de suivi d'objets qui utilise l'histogramme conjoint HSV couleur-texture (texture LPQ, LBP ou BSIF) pour représenter l'objet cible au cadre de l'algorithme de suivi Mean shift. Ensuite, nous présentons l'analyse mathématique des descripteurs locaux LPQ, LBP et BSIF utilisés dans notre travail, ainsi que la méthode proposée de combinaison entre les caractéristiques de couleur et de texture.

- **Le chapitre 5** présente la base de données et métriques utilisées pour évaluer les performances de trackers individuels. Ainsi, les résultats du tracker Mean shift avec différents espaces couleurs et avec l'histogramme conjoint couleur-texture proposé, sont présentés et discutés. Dans ce chapitre, nous comparons l'efficacité de la méthode proposée avec celle de méthodes de l'état de l'art et les études expérimentales de suivi d'objets sont réalisées sur les bases de données OTB et VOT2013.
- Finalement, nous concluons le travail de cette thèse et nous présentons les différentes perspectives du travail réalisé.

Chapitre 1

Etat de l'art sur la détection et le suivi d'objet

Sommaire

1.1 Introduction

La détection et le suivi d'objets sont parmi les problèmes les plus étudiés ces dernières années. Ils sont des tâches importantes et difficiles dans de nombreuses applications de vision par ordinateur telles que la robotique, la vidéosurveillance [21]-[23]. La détection d'objet consiste à localiser l'objet dans chacune des trames d'une séquence vidéo. Le suivi d'objet est le processus de localisation spatiotemporelle d'un objet en mouvement au cours d'une séquence vidéo. Chaque méthode de suivi d'objet nécessite un mécanisme de détection d'objet, soit dans chaque trame ou lorsque l'objet apparaît d'abord dans la vidéo.

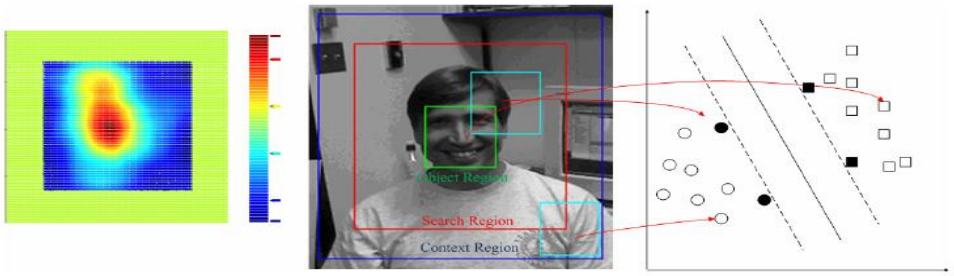
Le suivi d'objets dans une séquence d'images vidéo est un problème qui demande une extraction et un traitement d'informations provenant d'images complexes. Ce problème devient de plus en plus difficile si la contrainte temps réel est exigée. Il existe dans la littérature un grand nombre important de méthodes de suivi. Ce nombre est dû, d'une part au nombre important de problèmes à régler et d'autre part à la diversité des types d'applications concernées par le suivi. Ainsi, chacune des méthodes peut traiter certains aspects et échoue sur d'autres.

Ce chapitre introductif est consacré à la description des difficultés que les algorithmes de suivi d'objet sont susceptibles de rencontrer. Ensuite, une brève présentation sur la représentation de la forme d'objet et les primitives de suivi d'objet. Enfin, nous décrivons les principales techniques de détection et suivi d'objet dans une séquence d'images.

1.2 Suivi d'objet

Le suivi d'objet, dans sa forme la plus simple, est l'estimation de la trajectoire d'un objet en mouvement dans le plan image [21]. En d'autres termes, le suivi d'objet est l'estimation de la localisation de l'objet dans chacune des images d'une séquence vidéo et initialement détecté sur la première image [21][23]-[26]. Le procédé de localisation se base sur la reconnaissance de l'objet d'intérêt à partir d'un ensemble de caractéristiques visuelles telles que la couleur, la forme, la vitesse, où un objet est une zone de l'image qui peut être modélisée par des contours, silhouettes, primitives géométriques (rectangle englobant l'objet d'intérêt) ou encore par les points d'intérêt. Selon [21], toutes les méthodes de suivi comportent deux couches techniques : la première permet de détecter l'ensemble des candidats potentiels (objets similaires à la cible suivie) dans chaque image de la séquence et la seconde effectue la mise en correspondance d'une image à l'autre d'un de ces candidats avec la cible afin de maintenir la cohérence du suivi au fil du temps.

Plusieurs méthodes de suivi d'objet ont été proposées. La plupart des méthodes qui permettent le suivi d'objets visent à identifier l'objet d'intérêt en utilisant la technique de la soustraction du fond pour identifier les objets en mouvement dans les séquences vidéo [27][28]. Ces méthodes ne peuvent pas être appliquées au cas dans lequel la caméra se déplace à cause du changement de contexte. Les autres méthodes basées sur les paramètres de l'objet consistent à reconnaitre l'objet par son modèle et estimer sa position par quelques techniques. Les paramètres à estimer peuvent être divers, mais comprennent principalement une composante géométrique, indiquant la position dans l'image du centre de l'objet [1][29]. Ces méthodes de suivi nécessitent la détection seulement à l'initialisation du suivi quand un objet apparaît pour



(a) (b) (c)

Figure 1.1 – Illustration du suivi par la détection basée sur la classification des SVM [24]. (a) montre la carte de score de visage / non-visage classification; (b) affiche la zone de recherche et de la région de contexte pour; (c) trace l'hyperplan de classification.

la première fois. Après initialisation, la détection et le suivi s'effectuent conjointement. Récemment, le suivi d'objets visuels a été posé comme un problème de suivi par détection, comme illustré dans la figure 1.1, où la modélisation statistique est effectuée dynamiquement pour supporter la détection d'objet. Selon le mécanisme de construction de modèle, la modélisation statistique est classé en trois catégories, y compris générative, discriminative et hybride générative-discriminative [24]. Les méthodes de suivi par détection sont devenues l'un des paradigmes les plus efficaces pour le suivi d'objets visuels, et ont obtenu des performances à la pointe de la technologie. Cela est dû, en partie, au succès des composants des algorithmes de détection d'objets [30]-[32].

1.2.1 Domain d'application

Le suivi d'objets dans des séquences vidéo a suscité un grand intérêt dans la dernière décennie en raison de la variété de ses domaines d'applications [21]-[23][25][26], tels que :

- La vidéosurveillance (détection, suivi, reconnaissance du comportement de personnes, d'intrus),
- Le militaire (suivi des cibles ou guidage de missiles),
- La vidéoconférence (suivi des interlocuteurs),
- La gestion et l'analyse du trafic (le suivi d'une voiture ou des bords d'une route depuis une caméra embarquée sur un véhicule),
- La robotique : suivie d'obstacles pendant une phase d'évitement,
- Suivi d'indices visuels dans une tâche asservie par vision,
- Suivi d'un opérateur (corps, visage, main…) pour définir des modes d'interaction évolués entre l'Homme et la Machine,...etc.
- Imagerie médicale.

La figure 1.2 illustre quelques exemples d'applications.

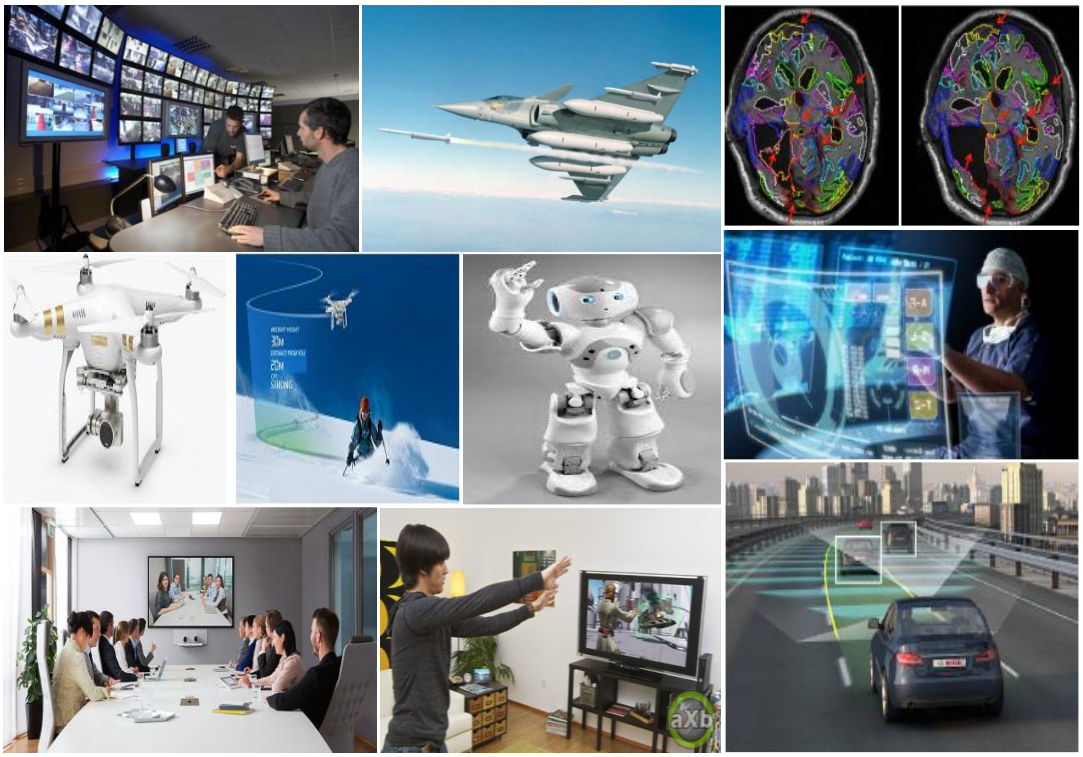


Figure 1.2 – Quelques applications de suivi d'objets.

1.2.2 Les défis du suivi d'objet

Bien que le suivi d'objets a été étudié depuis plusieurs décennies, et beaucoup de progrès ont été réalisés au cours des dernières années [1]-[3][6][33], il reste un problème très difficile [23]. Il n'existe aucune approche de suivi unique qui peut gérer avec succès tous les scénarios. De nombreux facteurs influençant la performance de l'algorithme de suivi d'objets sont les suivantes [23][24] (Figure 1.3) :

- Changements d'illumination ;
- Changements d'échelle ;
- Occultations partielles ou totales ;
- Similarité de couleur entre l'objet cible et le fond ;
- Mouvement de caméra ;
- Rotation dans le plan de l'image ;
- Objets non-rigides et/ou articulés ;
- Objet de petite taille ;
- Objet en mouvement rapide ;
- Présence de bruit dans les images ;
- Nécessité d'un suivi en temps réel.

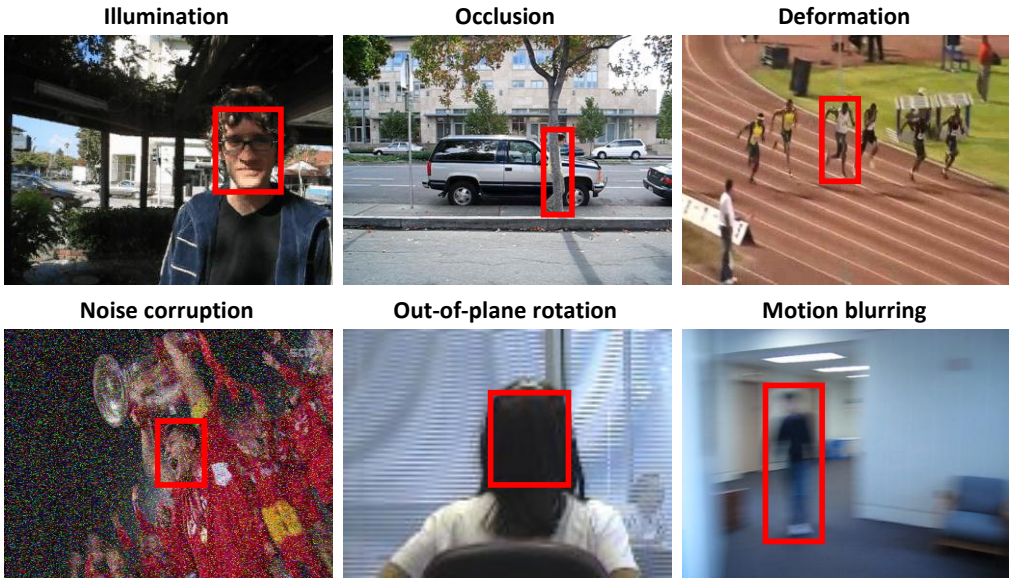


Figure 1.3 – Quelques difficultés de suivi d'objets.

1.3 Représentation d'objets

Comme d'autres tâches en vision par ordinateur, la représentation visuelle joue un rôle fondamental dans le suivi d'objets. La plupart des méthodes de suivi d'objets se basent sur l'apparence d'un objet. L'utilisation de ces méthodes nécessite une représentation pertinente de l'objet possédant des primitives fiables pour décrire son contenu. Les objets peuvent être représentés de nombreuses façons en termes de leurs formes et leurs apparences. Certaines approches utilisent uniquement la forme de l'objet pour le représenter, mais certaines combinent aussi la forme et l'apparence. Le choix de la représentation d'un objet dépend fortement du domaine d'application. Yilmaz et al [21] ont été les premiers à proposer une classification de la représentation d'objets. Nous reprenons cette classification dans cette section.

1.3.1 Représentation de la forme d'un objet

Les représentations basées sur la forme d'un objet sont nombreuses (voir Figure 1.4) et sont classées en plusieurs catégories, comme suit: représentation par points, représentation par formes géométriques (représentation par fenêtres englobantes), représentation par silhouettes et contours et représentation par modèle articulé et squelette [21][24].

- **Points :** L'objet est représenté par un point (centre de l'objet) (Figure 1.4 (a)) ou par un ensemble de points (Figure 1.4 (b)). Ce type de représentation est généralement souhaitable pour les objets qui occupent une petite partie de l'image.

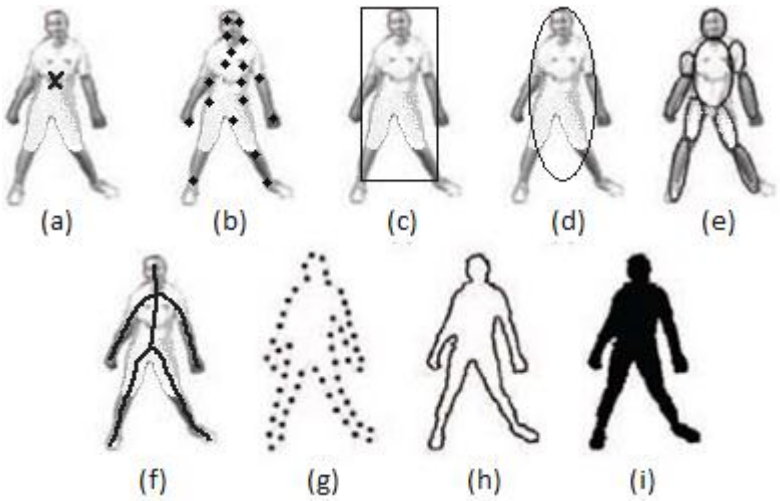


Figure 1.4 – Représentation d'un objet dans un système de suivi [21] : (a) centroïde, (b) ensemble de points, (c) rectangle, (d) ellipse (e) blocs d'articulation (f) squelette, (g)-(h) contour (i) silhouette

- **Formes géométriques primitives :** L'objet est représenté par une forme géométrique simple telle qu'un rectangle (Figure 1.4 (c)), une ellipse (Figure 1.4 (d)), etc. permettant une description de la dimension de l'objet. Cette représentation est particulièrement adaptée au suivi d'objets rigides (véhicules, ...) mais peut également convenir pour des objets non-rigides.
- **Contour et silhouette :** La représentation d'un objet par son contour permet de définir les limites exactes de l'objet (Figure 1.4 (g), (h)). La région interne du contour est appelée silhouette de l'objet (Figure 1.4 (i)) et peut être utilisée conjointement à l'information de contour pour le suivi d'objets. L'utilisation de cette représentation convient au suivi d'objets complexes non-rigides.
- **Modèle articulé :** Cette représentation se base sur la cinématique de l'objet. Objet articulé est composé de plusieurs parties du corps qui sont modélisées par une primitive géométrique (par exemple une ellipse) et sont reliées entre elles par des connexions (Figure 1.4 (e)). Cette représentation est adaptée au suivi d'objets articulés (suivi de personne).
- **Squelette :** Le squelette d'un objet peut être extrait en utilisant la transformation de l'axe central (Figure 1.4 (f)). Ce modèle est couramment utilisé comme représentation de forme dans la reconnaissance d'objet. Cette représentation est souhaitable pour les objets rigides et articulés.

1.3.2 Représentation de l'apparence d'un objet

Les caractéristiques d'apparence sont généralement utilisées conjointement aux caractéristiques de forme pour compléter la représentation de l'objet à suivre. Il existe plusieurs façons de représenter les caractéristiques d'apparence des objets. Certaines représentations d'apparence dans le contexte du suivi d'objets sont :

- **Densité de probabilité d'apparence** : L'apparence d'un objet est modélisée par une variable aléatoire dans un espace des caractéristiques (couleur, texture, information spatiale, ...) avec une fonction de densité de probabilité associée. Les estimations de la densité de probabilité de l'apparence d'un objet peuvent être paramétriques, par exemple des gaussiennes [35] ou un mélange de gaussiennes [36], ou non-paramétriques, telles les fenêtres de Parzen [37] et les histogrammes [1].
- **Template :** Templates (modèles) sont formés en utilisant des formes géométriques simples ou des silhouettes. L'utilisation des templates a pour but de représenter des objets avec un ensemble de modèles prédéfinis. L'avantage principal des templates est qu'ils comportent des informations spatiales et des informations d'apparence, mais ils ont tendance à être sensibles au changement de pose [21].
- **Modèle à apparence active:** Les modèles à apparence active sont une extension des modèles à forme active de Cootes et al. [38]. Ils ont pour objectif de prendre en compte les variations d'apparence d'un objet. Les modèles à apparence active sont générés en modélisant simultanément les caractéristiques de formes et d'apparence des objets. Les formes possibles ainsi que de texture interne des modèles sont représentées statistiquement à l'aide d'un ensemble d'apprentissages.
- **Modèles d'apparences multi-vues :** Contrairement aux templates ces modèles encodent différentes vues d'un objet. Il existe différentes approches pour faire cela, par exemple, nous pouvons représenter toutes les vues d'un objet comme un sous-espace en utilisant les méthodes d'analyse telle que l'analyse de composantes principales (ACP) et l'analyse en composantes indépendantes (ACI), qui ont été utilisées pour la représentation de la forme et de l'apparence [33][39]. Une autre approche pour modéliser l'apparence est d'entraîner un ensemble de classifieurs pour représenter les différentes vues de l'objet par exemple Machine à Support de Vecteurs (Support Vector Machines (SVM)) [40].

1.4 Caractéristiques visuelles pour le suivi d'objets

La sélection des bonnes caractéristiques joue un rôle essentiel dans le suivi visuel. Généralement, les caractéristiques qui distinguent mieux entre les objets multiples et entre l'objet et l'arrière-plan sont les meilleures pour le suivi de l'objet. Ces caractéristiques ont pour objectif de décrire les propriétés visuelles de l'objet dans l'image. Certaines méthodes se basent sur un seul type de caractéristiques, et d'autres utilisent une combinaison pondérée de plusieurs types de caractéristiques pour améliorer les performances. Les détails des caractéristiques visuelles les plus connues dans la littérature, et surtout, les plus utilisées sont les suivants.

1.4.1 Caractéristique de couleur

Les couleurs sont une des caractéristiques les plus utilisées en représentation d'objets, et les plus intuitives pour décrire l'apparence d'un objet, car ils sont le repère le plus évident pour l'œil humain. La perception de la couleur par l'œil humain diffère de la perception par un ordinateur. Généralement, l'information couleur est représentée dans l'espace de couleur RGB. L'inconvénient majeur d'une telle représentation est sa forte corrélation entre les composantes couleurs et sa non-uniformité dans la perception faite par l'humain [41]. La transformation de l'information de l'espace RGB à un espace couleur différent a pour objectif de séparer l'information dans les canaux pertinents. Les espaces YUV, YIQ, et YCbCr, visent à isoler la composante de luminance dans le signal. Les espaces HSI, HLS et HSV, mettent l'accent sur la teinte et la saturation et les espaces de couleurs Luv, Lab offrent l'avantage de représenter la couleur dans un espace perceptif linéaire, au prix de transformations non-linéaires. De nombreuses évaluations ont été faites pour évaluer les performances entre les différents espaces de couleur. Les espaces Lab et Luv sont des espaces de couleurs perceptuellement uniformes [42]. Tandis que, l'espace HSV est un espace de couleur approximativement uniforme. Cependant, ces espaces de couleurs sont sensibles au bruit. En résumé, il n'y a pas d'espace couleur idéal pour la représentation [21].

1.4.2 Caractéristique de gradient

Récemment, les caractéristiques de gradient ont été prouvées utiles dans la détection humaine. De nombreuses techniques consacrées à la recherche connexe ont été proposées. En générale, il existe deux catégories de caractéristiques de gradient [42]. Une catégorie principale des méthodes basées sur l'information de gradient consiste à utiliser la forme et le contour pour représenter les objets, tels que le corps humain. Cette information est extraite à partir de l'analyse spatiale de l'intensité lumineuse de l'image. Une propriété importante du gradient

est sa sensibilité plus faible aux changements d'illumination comparée aux caractéristiques couleurs. Les contours issus du gradient sont exploités dans de nombreuses approches de suivi d'objet. L'algorithme Condensation [44] (Conditionnal density propagation) consiste à initialiser une courbe spline sur les contours, et un filtre à particules est utilisé pour mettre à jour les paramètres de la courbe paramétrée. Des techniques de minimisation d'énergie le long des contours des objets ont également été proposées pour suivre les objets sous certaines contraintes de régularisation (snakes et contours actifs [45]). Une autre catégorie principale est d'utiliser la statistique des gradients. Par exemple, dans [46], Lowe a introduit le descripteur SIFT (Scale Invariant Feature Transform) pour la reconnaissance d'objet, qui combine un détecteur et un descripteur invariants à l'échelle basés sur la distribution du gradient. Dalal et Triggs [31]. ont utilisé le descripteur histogramme d'orientations de gradient (HOG) pour entraîner le classifieur SVM, pour la détection des piétons. Le descripteur HOG est fondé sur le calcul des histogrammes locaux de chaque orientation des gradients normalisés d'une image sur une grille. Il a été utilisé en tant que caractéristique pour la construction de certains descripteurs [47].

1.4.3 Caractéristique de texture

La texture d'un objet est également une caractéristique utilisée pour modéliser les objets. La texture est une mesure de la variation d'intensité d'une surface, en décrivant des propriétés comme la douceur et la régularité [21]. Comparée à la couleur, la texture nécessite une étape de traitement pour générer les descripteurs. Il existe différents descripteurs de texture, y compris matrices de co-occurence [48], filtre de Gabor [49], transformée en ondelettes. Filtre de Gabor est probablement la caractéristique de texture la plus étudiée, il est efficace mais difficile à utiliser dans des applications temps-réel. Récemment, des méthodes plus économiques et tout aussi discriminantes ont été proposées. Dans [50], Ojala et al ont développé un descripteur de texture très efficace, appelé Motifs Binaires Locaux LBP (Local Binary Patterns). L'opérateur d'analyse de texture LBP est défini comme une mesure de texture invariante en niveaux de gris. De nombreuses variantes du LBP existent à ce jour, telles que le Multi-Block LBP (MB-LBP) [51] et les motifs locaux ternaires (LTP) [52]. Les caractéristiques de texture sont moins sensibles aux changements d'illumination par rapport à la couleur.

1.4.4 Caractéristique de flot optique

Le flot optique est un champ dense de vecteurs de déplacement qui décrivent le mouvement de chaque pixel dans une région d'image. Le calcul du flot optique consiste à extraire un champ de vitesses dense à partir d'une séquence d'image, en supposant que le même pixel

conserve la même luminosité entre les images consécutives. La théorie de flot optique est fondée sur l'hypothèse de conservation spatio-temporelle de la luminance de la scène. Le flot optique est couramment utilisé comme une caractéristique dans les applications de détection de mouvement ou de segmentation spatio-temporelle, de suivi d'objet. De nombreuses techniques ont été proposées pour estimer le flot optique, on peut trouver une comparaison des techniques populaires de flot optique dans [53].

1.5 Détection d'objets

La première étape dans un processus de suivi d'objets consiste en la détection des objets à suivre. Chaque méthode de suivi nécessite un mécanisme de détection d'objet, soit sur toutes les trames, soit lorsque l'objet apparaît pour la première fois dans la vidéo. L'approche commune de détection d'objets à suivre est d'utiliser l'information à partir d'une seule image initiale. Cependant, certaines méthodes de détection d'objet utilisent l'information temporelle calculée à partir d'une séquence d'images afin de réduire le nombre de fausses détections. Cette information temporelle est généralement sous la forme de différenciation de trame, qui met en évidence les régions qui changent entre les images. On peut distinguer quatre ensembles d'algorithmes de détection communs décrits dans [21] : la détection de points d'intérêt, la soustraction de fond, la segmentation d'image et enfin la classification supervisée.

1.5.1 Détection des points d'intérêt

La détection des points d'intérêt représente une étape importante pour plusieurs processus de vision par ordinateur. Elle permet de localiser dans l'image l'ensemble des points d'intérêt, ceux-ci étant définis comme points de l'image possédant une information localement discriminante. Les principaux avantages des détecteurs de points sont l'insensibilité à la variation de l'éclairage. L'extraction de points d'intérêt dans des images est devenue un traitement standard depuis une quinzaine d'années. Dans cette section, nous évoquerons les méthodes de détection les plus utilisées dans la littérature.

1.5.1.1 Le détecteur de Moravec

Le détecteur de Moravec [54] est l'un des plus anciens algorithmes de détection de points d'intérêt. Il est développe à la base pour des applications robotiques. Son principe est de calculer la variation des intensités d'image dans un patch 4x4 dans les directions horizontale, verticale, diagonale et anti-diagonale, et de sélectionner le minimum des quatre variations comme valeur représentative pour la fenêtre. Un point est considéré intéressant si la variation d'intensité est un maximum local dans un patch 12x12. L'opérateur de Moravec possède des

limitations : les calculs sont effectués pour un nombre limité de directions, ayant une réponse bruitée du fait de l'utilisation d'une fenêtre binaire et rectangulaire et très sensible au bruit du fait de l'utilisation d'images en niveaux de gris.

1.5.1.2 Le détecteur de Harris

En 1988, Harris et Stephens [55] ont proposé un détecteur de coins connu sous le nom de détecteur de Harris, afin de remédier à certaines limitations de détecteur de Moravec. Le détecteur de Harris est basé sur la matrice des moments du second ordre *M* (matrice d'auto-corrélation). Étant donnée une image *I* et *p* un pixel de *I* de coordonnées (*x, y*), la matrice de second moment en *p* est définie ainsi :

$$M(p) = \begin{bmatrix} I_x^2(p) & I_x I_y(p) \\ I_x I_y(p) & I_y^2(p) \end{bmatrix} \quad (1.1)$$

Où I_x et I_y sont les dérivées partielles de I dans respectivement la direction horizontale et la direction verticale.

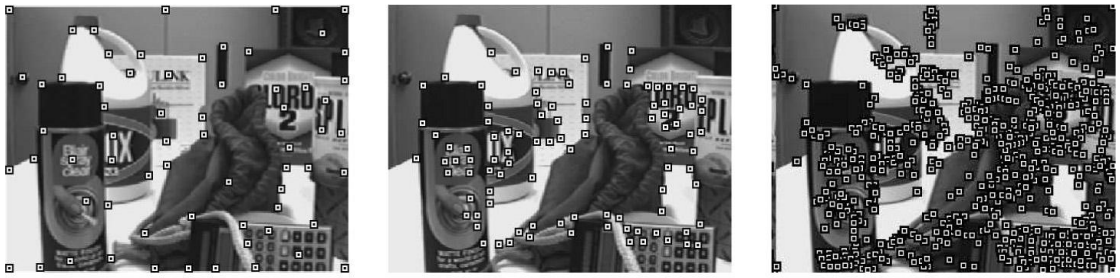
La fenêtre rectangulaire utilisée par le détecteur de Moravec est remplacée par une fenêtre circulaire de type gaussienne pour réduire le bruit au niveau de la réponse du détecteur. L'équation régissant la variation d'intensité subie par un pixel est analytiquement développée au voisinage de l'origine du déplacement afin de couvrir tous les petits déplacements possibles et ainsi de rendre le détecteur invariant en rotation. Le détecteur de Harris est plus efficace que l'opérateur de Moravec. Cependant, il est sensible au bruit à cause de l'utilisation d'informations de gradient, et le temps de calcul augmente.

1.5.1.3 Le détecteur de KLT

Similaire au détecteur de Harris, le détecteur KLT (Kanade-Lucas-Tomasi) [56] est lui aussi, basé sur la matrice des moments du second ordre *M*. La différence entre les deux algorithmes réside dans leur utilisation de cette matrice pour construire une mesure de confiance, et ainsi repérer les pixels intéressants. Le principe de ce détecteur est de calculer les valeurs propres de la matrice M. Lorsque la plus faible des valeurs propres est supérieure à un seuil, le point est conservé. Le détecteur KLT produit une liste de pixels de coin séparés les uns des autres. Les détecteurs de Harris et KLT sont très semblables, et possèdent les mêmes limitations.

1.5.1.4 Le détecteur de SIFT

Le détecteur SIFT (Scale Invariant Feature Transform) introduit par Lowe [57], pour pallier les problèmes d'invariance des autres détecteurs. SIFT est un détecteur permettant de décrire un point à partir des orientations locales du gradient. Les vecteurs de description construits



(a) (b) (c)

Figure 1.5 – Détection de points d'intérêts par le détecteur [21]: (a) Harris, (b) KLT, (c) SIFT.

par ce détecteur sont spatialement discriminants et robustes aux déformations géométriques usuelles, changement d'échelle, apparition du bruit et aux variations d'intensité. Le détecteur SIFT génère un nombre de pixels de coin plus important que les détecteurs Harris et KLT (figure 1.5). Ceci est dû à l'accumulation des points d'intérêt à différentes échelles et différentes résolutions.

1.5.1.5 Le détecteur de SURF

Bay et al [58], ont proposé le détecteur SURF (Speeded Up Robust Features), pour améliorer la performance de détecteur SIFT, en termes de rapidité. En effet, il pallie le manque d'efficience de détecteur SIFT en modifiant certains calculs longs à effectuer. Le détecteur SURF repose sur l'approximation du déterminant de la matrice Hessienne d'un noyau Gaussien et l'utilisation de l'image intégrale. Ceci permet de rendre le détecteur invariant en rotation, et permet en plus un calcul en temps réel.

1.5.2 Détection par soustraction de fond

La soustraction de fond est une méthode populaire pour détecter un objet en le segmentant à partir d'une scène. L'approche de base lors de l'utilisation de cette méthode consiste à construire un modèle de fond qui représente la scène, puis à trouver tout changement ou déviation par rapport à ce modèle dans chaque image. Tout changement significatif dans une région d'image par rapport au modèle de fond, signifie un objet mobile. Les pixels de la région subissant ce changement sont marqués pour un traitement ultérieur. Il existe plusieurs méthodes de soustraction de fond construisant des modèles plus ou moins complexes. Nous présentons, ici, un aperçu général sur les méthodes les plus importantes.

1.5.2.1 Modèles Gaussiens

La soustraction de fond est devenue populaire récemment dans la communauté de la vision par ordinateur, depuis le travail de Wren et al de [59]. Wren et al, ont proposé de modéliser l'intensité de la couleur de chaque pixel par une seule distribution gaussienne. Chaque pixel

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

Figure 1.6 – Soustraction de fond en utilisant des mixtures de Gaussiennes [21] *:* (a) Image courante, (b) Modèle de fond utilisant les moyennes des gaussiennes de poids, (c) Moyenne des gaussiennes de poids minimal représentant l'objet en mouvement, (d) Résultat de la soustraction de fond

du modèle est représenté par une densité de probabilité Gaussienne définie par la couleur moyenne du pixel et une covariance liée à cette couleur. Bien que très performante en besoins mémoire et en temps de traitement, cette technique est peu adaptée aux scènes extérieures. Une amélioration dans la modélisation du fond est effectuée par Stauffer et Grimson [60], en utilisant des modèles statistiques multimodaux. Ils ont utilisé un mélange de Gaussiennes (mixture of Gaussian MoG) au lieu d'une seule Gaussienne, pour modéliser la couleur des pixels. Les régions en mouvement, qui sont détectées à l'aide de cette approche, avec les modèles de base sont illustrées dans la figure 1.6. Cette technique est sensible aux changements dans les scènes dynamiques dérivées des changements d'éclairage, etc. Différentes variantes de la modélisation en mélange de Gaussiennes ont été présentées dans ces dernières années, [61]-[63].

1.5.2.2 Modèles basés sur l'apprentissage de sous-espaces

Une autre manière pour la modélisation du fond, emploie les méthodes d'apprentissage de sous-espaces. L'idée est de considérer les pixels comme des dimensions d'un espace de représentation, et les images successives comme des individus dans cet espace. Les méthodes d'analyse de données permettent alors de considérer tous les pixels de l'image dans une approche globale pour définir de nouvelles caractéristiques que l'on pourra appliquer en tout point pour y détecter d'éventuels mouvements. Oliver et al. [64] utilisent une modélisation à base de vecteurs propres, en appliquant l'analyse en composantes principales (PCA). Cette dernière est appliquée sur N images d'apprentissage prises à des instants non consécutifs afin de générer l'image moyenne et la matrice de projection comprenant les p premiers vecteurs propres significatifs de la PCA. Cette approche est moins sensible à l'éclairage. Dans [65] les auteurs utilisent un modèle de représentation multimodal pour améliorer la gestion des changements d'éclairage soudains. Ils ont proposé l'apprentissage de multiples sous-espaces

représentant différentes conditions d'éclairage à l'aide d'une PCA locales (LPCA). Ainsi, à chaque nouvelle image, l'algorithme sélectionne le sous-espace partageant les mêmes caractéristiques d'éclairage.

1.5.3 Détection par segmentation

La segmentation d'images est une technique importante et très utilisée en imagerie informatique. Elle vise à détecter les objets en partitionnant l'image en régions perceptuellement similaires, selon des critères prédéfinis. Chaque algorithme de segmentation concerne deux problèmes, les critères d'une bonne partition et la méthode pour réaliser le partitionnement efficace. Les techniques de segmentation les plus populaires, et qui sont pertinentes au suivi des objets sont : les algorithmes basés sur l'approche Mean Shift, les algorithmes utilisant des coupes de graphes et les algorithmes basés sur des contours actifs.

1.5.3.1 Mean Shift

L'une des techniques de segmentation les plus robustes dans le domaine de la vision par ordinateur est l'algorithme Mean shift [66]. À l'origine, cet algorithme a d'abord été proposé par Fukunaga en 1975 [67], adapté plus tard par Cheng [68], dans le but d'analyse d'images. L'algorithme Mean shift est une procédure itérative (non-paramétrique) d'ascension de gradient, utilisée pour estimer les modes d'une densité de probabilité associée à une distribution de points. Comanicu et Meer ont été utilisés cet algorithme en 2002, dans le cadre de la segmentation d'images. Ils ont appliqué le principe général du Mean shift à un vecteur de caractéristiques plus grand auquel on ajoute les informations spatiales (x, y) des points. La méthode de la segmentation par Mean shift mélange les informations spatiales (position des pixels) et les informations colorimétriques (valeurs des pixels) pour segmenter les images dans des délais raisonnables. Les régions segmentées remplissent à la fois les critères de proximité spatiale et d'homogénéité colorimétrique des pixels qui les composent. Les approches basées sur l'algorithme Mean Shift se révèlent en général efficaces en termes de ressources de traitement. Cependant elles s'avèrent difficiles à "calibrer" au vu du nombre de paramètres à fixer.

1.5.3.2 Segmentation par Coupe-graphe

La segmentation peut également être formulée sous la forme d'un problème de partitionnement de graphes, dans lequel les sommets (pixels) d'un graphe (image) sont partitionnés en plusieurs sous-graphes (régions) disjoints par élagage des arêtes pondérées. Le poids total des arêtes élaguées entre deux sous-graphes est appelé "coupe". Le poids est généralement calculé relativement à la similarité de couleur, la texture ou même la similitude

(a) (b) (c)

Figure 1.7 – Segmentation d'une image [21] : (a) Image originale, (b) Image segmentée par Mean shift, (c) Image segmentée par coupe normalisée.

entre les nœuds. Wu et Leahy [69] utilisent le critère de coupe minimale, où l'objectif est de trouver les partitions qui minimisent une coupe. Dans leur approche, les poids sont définis en fonction de la similarité des couleurs. La limitation majeure de cette approche est son biais envers sur-segmentation de l'image. Shi et Malik [70] proposent la méthode de coupe minimale normalisée du graphe, pour surmonter ce problème. Dans leur méthode, la coupe ne dépend pas seulement de la somme des poids des arêtes, mais aussi du rapport entre les poids total de connexion des nœuds dans chaque partition à tous les nœuds du graphe. Les poids correspondent ici au produit de la similarité couleur et de la proximité spatiale entre les nœuds. Cette méthode nécessite moins de paramètres sélectionnés manuellement, par rapport à la segmentation par Mean shift. Dans la figure 1.7, nous montrons les résultats de segmentation obtenus par les approches Mean shift et coupe normalisée.

1.5.3.3 Contours actifs

La segmentation par contours actifs permet de détecter les contours des régions en faisant évoluer un contour vers les limites des objets, de manière à ce que le contour entoure la région de l'objet. L'évolution du contour est dirigée par une fonction d'énergie qui définit l'alignement du contour sur la région de l'objet hypothétique. La fonction d'énergie du contour a la forme suivante [21]:

$$E(C) = \int_0^1 E_{int}(v) + E_{im}(v) + E_{ext}(v)\, ds \qquad (1.2)$$

Où s est la longueur du contour C, E_{int} inclut les contraintes de régularisation internes, E_{im} exprime l'énergie à partir de l'apparence (l'image) et E_{ext} spécifie des contraintes additionnelles. E_{int} contient généralement les termes définissant la courbure du contour, soit un terme de continuité du premier ou du second ordre de manière à trouver le contour le plus court. L'énergie de l'apparence E_{im} est habituellement calculée à partir des gradients de l'image définie par le contour courant [71][72], ou bien à partir de la couleur [35][73][74] ou de la texture [36] évaluée à l'intérieur et à l'extérieur de l'objet.

La limitation principale des méthodes basées contours actifs est qu'elles nécessitent une information a priori sur la position de l'objet. En effet, le contour doit être initialisé, selon la méthode choisie, à l'intérieur ou à l'extérieur de l'objet à segmenter. Le temps de traitement nécessaire à l'utilisation de ces méthodes est très variable puisqu'il dépend de la fonctionnelle d'évolution choisie.

1.5.4 Détection par apprentissage supervisé

La détection d'objets peut s'effectuer par l'apprentissage automatique de différentes vues de l'objet contenues dans un ensemble d'exemples au moyen d'un mécanisme d'apprentissage supervisé. L'apprentissage des différentes vues de l'objet renonce à l'exigence de stocker un ensemble complet de modèles. Étant donné un ensemble d'exemples d'apprentissage, les méthodes d'apprentissage supervisé génèrent une fonction qui fait correspondre les entrées aux sorties désirées. Une formulation standard de l'apprentissage supervisé est le problème de classification où l'apprentissage établit une approximation du comportement d'une fonction en générant une sortie sous la forme soit d'une valeur continue, qui est appelé régression, ou une étiquette de classe, qui est appelée classification. Dans le cas de la détection d'objet, les exemples d'apprentissage sont composés de paires de caractéristiques d'objet et d'une étiquette de classe associée, où ces deux quantités sont définies manuellement [21].

Nous discuterons dans cette section les deux techniques les plus courantes, le boosting adaptatif (Adaboost) et les machines à vecteurs supports (SVM), en raison de leur applicabilité au suivi d'objets.

1.5.4.1. Boosting adaptatif

Freund et Schapire.[75] ont proposé la méthode AdaBoost (Adaptative Boosting), qui est basée sur le principe du boosting. Adaboost est une méthode itérative de combinaison permettant de construire un classifieur très efficace en combinant plusieurs classifieurs de base, dont chacun peut être modérément efficace. Cette méthode repose sur le principe de sélection itérative de classifieurs faibles en fonction des exemples d'apprentissage.

Dans la phase d'apprentissage de l'algorithme Adaboost, la première étape consiste à construire une distribution de poids pour l'ensemble d'apprentissage. Le mécanisme de boosting sélectionne ensuite un classifieur de base qui présente la plus petite erreur, où l'erreur est proportionnelle au poids des exemples mal classés. Ainsi les poids associés aux exemples mal classés par le classifieur de base sélectionné sont augmentés, tandis que les exemples bien classés sont diminués. Enfin, le processus est répété jusqu'à atteindre un certain seuil d'erreur d'apprentissage. De cette façon, à chaque itération, un exemple mal

classé aura plus de chance de servir à l'apprentissage du nouveau classifieur. La décision de classification de l'ensemble est donnée par la somme pondérée des décisions de classification de chacun des classifieurs de base construits.

Dans le contexte de la détection d'objet, en 2001, Viola et Jones [30] ont appliqué Adaboost dans la détection de visages. Puis en 2003, ils ont amélioré cette méthode [76] pour la détection de piéton, combinant les informations de mouvement et d'apparence.

1.5.4.2 Machines à vecteurs supports

Les machines à vecteurs supports sont l'une des techniques d'apprentissage supervisée les plus utilisées à ce jour. Elles sont notamment reconnues pour leurs bonnes performances vis-à-vis du traitement de problèmes non linéairement séparables. La technique SVM a été introduite par Vapnik en 1995 [77], est utilisée pour séparer les données en classes en trouvant l'hyperplan marginal maximal qui sépare une classe des autres. L'idée principale est de trouver l'hyperplan optimal qui sépare les données en deux classes (figure 1.8), en utilisant le principe de marge maximale : considérons des points d'apprentissage appartenant aux classes -1 et +1. La marge de l'hyperplan, qui est maximisée, est définie comme la distance entre l'hyperplan et les points de données les plus proches de celui-ci. Ce sont ces points qui sont appelés "vecteurs supports". Le problème de trouver les vecteurs supports est résolu en le formulant comme un problème d'optimisation quadratique.

Dans le contexte de la détection d'objets, plusieurs méthodes ont été proposées dans la littérature. Papageorgiou et al. [78] ont utilisé SVM pour détecter les piétons et les visages dans les images. Les caractéristiques utilisées pour discriminer les classes sont extraites en appliquant des ondelettes Haar. Et il y a d'autres travaux [31][32][79] ont utilisé les caractéristiques de HOG pour discriminer les classes.

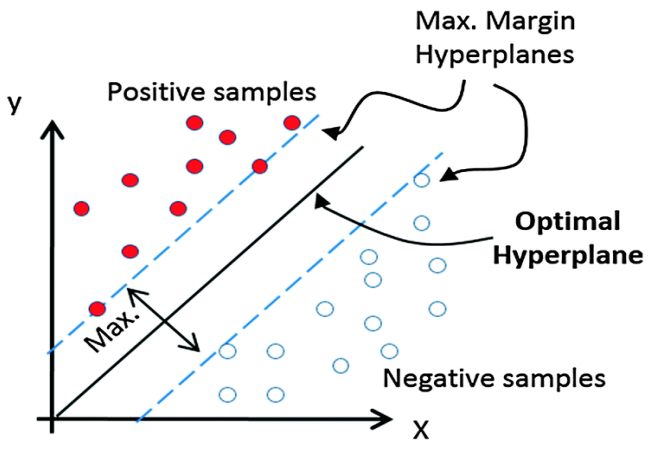


Figure 1.8 – Principe des SVM

1.6 Méthodes de suivi d'objets

De nombreuses méthodes de suivi d'objets dans la littérature ont été proposées. La différence entre ces méthodes réside en partie dans le choix de la représentation et de la forme des objets, des caractéristiques de l'image utilisées, de la nature du mouvement estimé, etc. Ce choix dépend de l'application ainsi que de la vidéo traitée. Il existe plus d'une catégorisation possible des algorithmes de suivi dans la littérature. Dans [80], les auteurs classent les algorithmes de suivi d'objets dans 4 catégories : algorithmes basés sur des régions, basés sur des contours actifs, basés sur des caractéristiques et basés sur un modèle. Dans [21], les auteurs ont publié une revue de littérature couvrant les travaux majeurs de suivi d'objets, en catégorisant les algorithmes de suivi selon la représentation de l'objet cible. Leur classification comprend trois catégories de méthodes : le suivi par points, le suivi par noyau et le suivi par silhouette. Une catégorisation similaire des algorithmes de suivi est apparue dans [22]. Récemment, une autre classification pour les algorithmes de suivi basée sur le modèle d'apparence utilisée. Dans [81] et [24], les auteurs distinguent deux catégories : les méthodes génératives et les méthodes discriminatives. Les méthodes génératives se concentrent sur la modélisation de l'apparence de l'objet, qui peut varier dans une trame différente. Les méthodes discriminatives distinguent l'objet par rapport à l'arrière-plan, en transformant le problème de suivi en un problème de classification binaire. Ces classifications ne sont pas strictes et certaines approches peuvent être représentées dans plusieurs catégories.

Dans cette section, nous reprenons la classification des algorithmes de suivi proposée dans [21]. Parce que le travail de Yilmaz décrit les méthodologies des méthodes de suivi d'objets existantes dans l'état de l'art, de manière claire et très compréhensible. Ce travail considère comme un bon cadre de référence. La figure 1.9 présente la taxonomie des méthodes de suivi d'objets proposées dans cet article.

1.6.1 Suivi de points

Le suivi d'objet peut être formulé comme un problème de mise en correspondance d'objets représentés par des points d'une trame sur l'autre. La correspondance de point est un problème complexe, particulièrement en présence de bruit dans les images, les occultations partielles ou totales, les erreurs de détection de points, etc. Globalement, les méthodes de correspondance de points peuvent être divisées en deux grandes catégories : les méthodes déterministes et les méthodes probabilistes

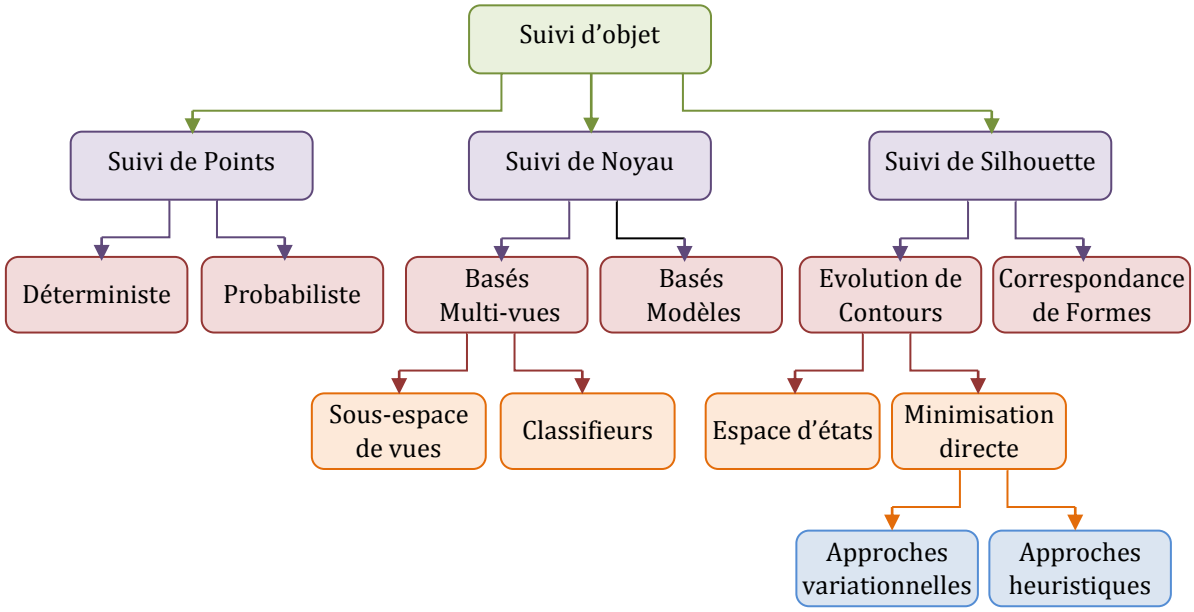


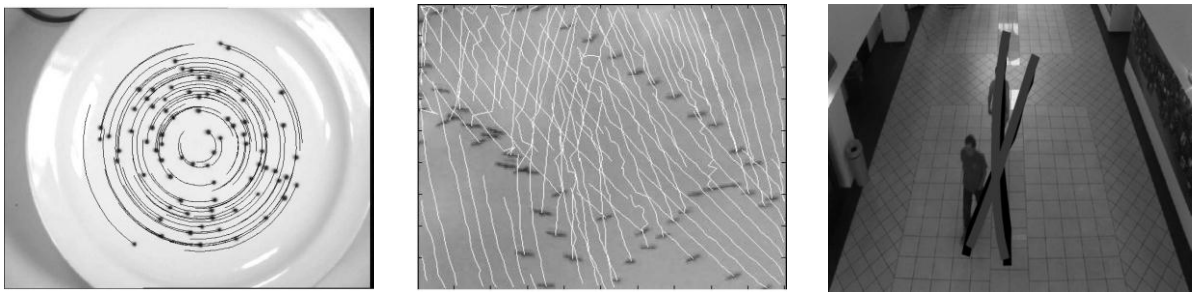
Figure 1.9 – Taxonomie des méthodes de suivi d'objets.

1.6.1.1 Méthodes déterministes

Dans les méthodes déterministes, le suivi s'effectue en minimisant un coût d'association entre les objets à l'image précédente et chaque objet unique à l'image courante. Le coût est en général défini comme une combinaison de contraintes de type proximité, rigidité, mouvement commun, vitesse maximale, etc.

Sethi et Jain. [82] résolvent le problème de mise en correspondance en utilisant une approche gloutonne (greedy) basée sur les contraintes de proximité et de rigidité. Leur algorithme considère deux trames consécutives, et est initialisé par une recherche des plus proches voisins. Les correspondances sont échangées itérativement pour minimiser le coût. Cette méthode ne permet pas de prendre en compte les occultations. Salari et Sethi. [83] traitent ce problème, en créant d'abord la correspondance pour les points détectés, puis étendent le suivi d'objets manquants en ajoutant de nouveaux points hypothétiques.

Veenman et al. [84] introduisent une contrainte de mouvement dans leur méthode de mise en correspondance. Cette contrainte fournit une contrainte forte pour suivre de manière cohérente un ensemble de points appartenant au même objet. L'algorithme est initialisé par la génération de la trajectoire initiale en utilisant un algorithme à deux passes et la fonction de coût est minimisée par l'algorithme d'assignation hongrois sur deux trames consécutives. Cette approche permet de gérer les occultations et les erreurs de détection de points. Cependant, elle n'est pas adaptée dans le cas où les points à suivre appartiennent à des objets ayant des mouvements différents. La figure 1.10 (a) illustre les résultats de suivi.



(a) (b) (c)

Figure 1.10 – Exemples de suivi de points : (a) En utilisant l'algorithme proposé par Veenman et al. [84], (b) En utilisant l'algorithme proposé par Shafique et Shah. [85], (c) En utilisant l'algorithme proposé par Scovanner et Tappen. [86].

Shafique et Shah. [85] ont proposé une approche multi-trames afin de préserver la cohérence temporelle de la vitesse et de la position des points. Les auteurs utilisent un algorithme glouton afin de résoudre le problème de correspondance étant défini comme un problème de théorie des graphes dans lequel la problématique devient la détection du meilleur chemin unique pour chaque point. La figure 1.10 (b) illustre les résultats de cet algorithme pour le suivi des oiseaux.

Dans [86], les auteurs présentent une méthode pour apprendre hors ligne certains paramètres de suivi, qui régissent les mouvements des piétons en observant les données vidéo. Leur cadre d'apprentissage repose sur l'apprentissage en mode variationnel, qui permet d'optimiser efficacement un modèle continu de coûts pour les piétons. La figure 1.10 (c) illustre un exemple de suivi par cette méthode.

1.6.1.2 Méthodes probabilistes

Les données perçues par les capteurs vidéo contiennent toujours du bruit. De plus, les mouvements d'objet peuvent subir des perturbations aléatoires, par exemple, des manœuvres de véhicules. Les méthodes de correspondance statistique résolvent ces problèmes de suivi en ajoutant une incertitude au modèle de l'objet et aux modèles des observations. Les méthodes de correspondance statistique utilisent l'approche de l'espace d'état pour modéliser les propriétés de l'objet telles que la position, la vitesse et l'accélération. Lors du suivi, l'état de l'objet est estimé par un modèle dynamique de transition, mis à jour et corrigé au cours du suivi en prenant des mesures de l'image. Parmi les méthodes d'estimation de l'état dans le contexte du suivi des points, nous citons le filtre de Kalman et le filtre à particules. Mais il convient de noter que ces méthodes peuvent être utilisées en général pour estimer l'état de n'importe quel système variant dans le temps.

Le filtre de Kalman, a été présenté en 1960 par Rudolph E. Kalman [87]. Il s'agit d'un estimateur optimal de processus aléatoires. Partant de l'hypothèse que le bruit de mesure suit une distribution gaussienne, le filtre de Kalman est un algorithme récursif en deux étapes, la prédiction et la correction de l'état. L'étape de prédiction est assurée par le modèle de mouvement linéaire calculé à l'instant précédent M_{t-1}. L'étape de correction corrige la prédiction d'état en utilisant l'écart entre l'observation prédite (modèle) et l'observation courante. À l'origine conçu pour le suivi de points (radar). Le filtre de Kalman a été largement utilisé dans les algorithmes de suivi d'objets. Broida et Chellappa. [88] ont utilisé le filtre de Kalman pour suivre les points dans des images bruyantes. Dans le suivi d'objet par caméra stéréo, Beymer et Konolige. [89] utilisent le filtre de Kalman pour prédire la position et la vitesse de l'objet dans les dimensions x-z. Ponsa et al. [90] ont utilisé le filtre de Kalman pour suivre les véhicules sur les images prises à partir d'une plate-forme mobile. Dans [91], Robert utilise le filtre de Kalman pour reconstruire la trajectoire de voitures dans une ambiance nocturne. Malgré la popularité du filtre de Kalman, cet algorithme reste insuffisant dans des conditions réelles d'application, comme la présence de la multi-modalité et le comportement non-linéaire des objets (changement brusque de direction, mouvement de caméra).

Une limitation du filtre de Kalman est qu'il n'est applicable que dans le cas des variables d'état qui suivent la distribution gaussienne. Cette limitation peut être surmontée en utilisant le filtrage de particules [92]. Le filtrage particulaire est une généralisation du filtrage de Kalman dans lequel la variable d'état n'est plus décrite par une gaussienne. Ce filtre se base principalement sur l'algorithme de simulation Monte Carlo séquentielle [93], dans laquelle des échantillons pondérés appelés particules explorent l'espace d'état et interagissent sous l'effet d'un mécanisme de sélection qui concentre automatiquement les particules dans les régions d'intérêt de l'espace d'état. Les particules font office de description de la distribution et sont mises à jour régulièrement dans un schéma similaire au filtrage de Kalman à l'aide d'une étape de prédiction, d'une étape de mesure et d'une étape de correction de l'état. Le filtrage particulaire est bien adapté aux trajectoires complexes des objets et aux occultations, par exemple deux piétons se croisant. Son coût de calcul dépend du nombre de particules utilisées. Le filtrage particulaire a été largement utilisé en suivi mono-objet et multi-objets. Dans le contexte de suivi par points, Arnaud et al. [94] ont proposé d'appliquer le filtrage particulaire qui permet une bonne qualité de suivi, même pour des variables d'état de distribution variable.

Le suivi multi-objets peut se faire utilisant le filtre de Bayes qui nécessite une méthode d'association. Les méthodes d'association ont pour but de relier les observations fournies par le détecteur aux pistes (trajectoires des objets). Parmi les méthodes statistiques d'association les plus utilisées sont le JPDAF (Joint Probability Data Association Filtering) et le MHT (Multiple Hypothesis Tracking). Le JPDAF a été utilisé pour la reconstruction 3D [95] ainsi que le suivi de régions [96], sa limitation majeure est son incapacité de gérer la variabilité du nombre de pistes. Reid et al. [97] ont été développés la technique MHT, pour remédier cette limitation. Cependant, MHT est très coûteuse à la fois en mémoire et en temps de calcul. Pour réduire le temps de calcul, Cox et Hingorani. [98] ont utilisé l'algorithme de Murty [99] pour déterminer les k meilleures hypothèses en un temps polynomial.

1.6.2 Suivi de Noyau

Le suivi de noyau s'effectue en calculant le mouvement de l'objet, qui est représenté par une forme géométrique (rectangle ou ellipse), d'une image sur l'autre. Le mouvement de l'objet est généralement sous la forme de mouvement paramétrique (translation, rotation, affine, etc.) ou le champ de déplacement dense calculé dans les images suivantes [21]. Il existe un grand nombre de méthodes de suivi de noyau. Celles-ci diffèrent en termes de la représentation de l'apparence utilisée, le nombre d'objets suivis, et la méthode utilisée pour estimer le mouvement de l'objet. Cependant, on peut regrouper ces méthodes en deux sous-catégories en fonction de la représentation de l'apparence utilisée : les méthodes basées sur des modèles (*Templates)* ou des modèles basés sur une densité de probabilité, et les méthodes basées sur une représentation multi-vues de l'objet.

1.6.2.1 Les méthodes basées sur des *Templates ou* densité de probabilité

Les méthodes basées sur des *Templat*es ou basées sur une densité de probabilité ont été largement utilisées dans les premiers jours en raison de leur performance, de leur simplicité et de leur faible coût calculatoire.

L'approche la plus simple pour suivre un objet dans ce paradigme est mise en correspondance de Templates (Template Matching) [100][101], où le contenu de l'image au sein de la boîte englobante dans la première trame sert de Template initial. Le suivi est ensuite effectué pour trouver la partie de l'image la plus similaire au Template initial, par des fonctions de similarités telle que la corrélation ou la somme des carrés des différences (SSD). Pour ce faire, la recherche est réalisée de manière exhaustive sur tout ou partie de l'image. Généralement, les informations utilisées sont l'intensité ou les composantes couleurs de l'image, ce qui rend ces méthodes sensibles aux changements d'illumination. Le gradient de

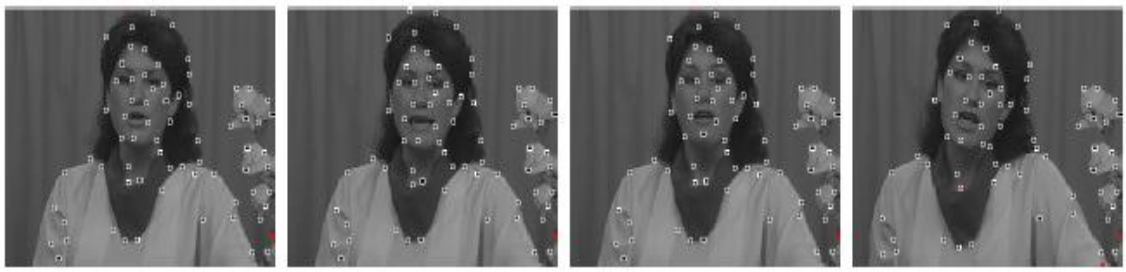


Figure 1. 11 – Suivi des caractéristiques en utilisant le suivi KLT [21].

l'image [100] est également utilisé pour former le Template. Récemment, on a proposé la méthode de suivi NCC [102] qui utilise la fonction de similarité NCC (normalized cross-correlation) pour faire la similarité entre le patch initial et les patchs extraits des régions candidates. Elle agit aussi comme l'un des composants importants des trackers plus avancés [103][8]. L'inconvénient majeur de mise en correspondance de Templates est la lenteur de la recherche exhaustive, surtout lorsque la taille du Template et/ou la zone de recherche est/sont grandes.

Le tracker KLT (Kanade-Lucas-Tomasi). [29][104]-[106] est la méthode la plus efficace basée sur la mise en correspondance de Templates. Le tracker KLT trouve des correspondances affines transformées entre deux trames successives au moyen de dérivés spatio-temporels. La nouvelle localisation de la cible est déterminée en mettant en correspondance sa position dans la trame précédente à la localisation dans la trame actuelle en utilisant la transformation affine estimée (Figure 1.11). Afin d'améliorer la robustesse de suivi KLT, Shi et Tomasi [56] ont proposé un critère de sélection de caractéristiques pour appliquer la mise en correspondance de Templates. Celui-ci calcule itérativement la translation d'une région (25 x 25 pixels) centrée en un point d'intérêt.

Vojir et al. [107] ont proposé le tracker FoT (Flock of Trackers) qu'un algorithme capable de suivre des objets non-rigides par analyse du flot optique obtenu par l'algorithme de suivi KLT (Kanade-Lucas-Tommasi) en utilisant l'opérateur médian pour estimer la direction de déplacement de l'objet. Cette méthode apporte une certaine robustesse par rapport aux occultations partielles. Elle est adaptée dans le cas d'objets rigides montrant des points saillants.

Comaniciu et al. [1][66][108] présentent la méthode Mean shift, capable de détecter, localiser et suivre une cible mobile représentée par un modèle statistique, même si les conditions d'acquisition des images sont défavorables (changements d'échelle, occultations partielles ou bruit). L'algorithme Mean shift est sans doute la méthode la plus populaire qui utilise une

Figure 1.12 – L'itération du Processus de suivi par Mean Shift. (a) Localisation estimée de l'objet à l'instant t-1, (b) Trame à l'instant t avec estimation de localisation initiale en utilisant la position d'objet précédente, (c), (d), Mise à jour de position en utilisant des itérations de Mean Shift, (f) Position finale de l'objet à l'instant t [21].

représentation de l'apparence d'un objet sous forme d'histogrammes. Il s'agit d'une méthode non-paramétrique qui maximise de façon itérative la similarité d'apparence (mesurée par la distance de Bhattacharyya) entre l'histogramme de couleurs pondéré représentant l'objet et l'histogramme correspondant représentant la position hypothétique, comme illustrée dans la figure 1.12. L'avantage de cette méthode est la réduction considérable du temps d'estimation de la localisation d'objet. Cependant, cet algorithme peut être confondu par les régions avec distribution de couleur similaire, en raison du manque d'information spatiale. Pour cette raison, plusieurs chercheurs ont proposés des méthodes de combinaison de plusieurs indices visuels [14][109]-[112] d'une part, et de filtre de Kalman et de filtre à particules, d'autre part [113]-[116]. Ces méthodes peuvent améliorer la convergence de cet algorithme, mais le choix des indices et la manière de les combiner restent des problèmes au cœur de la recherche. Le tracker Mean shift est détaillé dans le chapitre 2.

L'algorithme Camshift (Continuously Adaptive Mean Shift) [117]-[119] est une version étendue de Mean Shift qui permet de résoudre le problème de changement d'échelle de l'objet. Camshift consiste à une étape de mise à jour des histogrammes permettant s'adapter aux changements d'apparence des objets.

1.6.2.2 Les méthodes basées sur une représentation multi-vues de l'objet

Lorsque la vue de l'objet change considérablement pendant le suivi, le modèle d'apparence peut ne plus être valide, et l'objet à suivre pourrait être perdu. Pour pallier ce problème, les méthodes basées sur une représentation multi-vues de l'objet construisent leur modèle par apprentissage d'un ensemble de vues différentes de ce dernier [21]. Ces méthodes se divisent en deux catégories : suivi utilisant le modèle génératif (Sous espace) et suivi utilisant le modèle discriminatif (Classifieur).

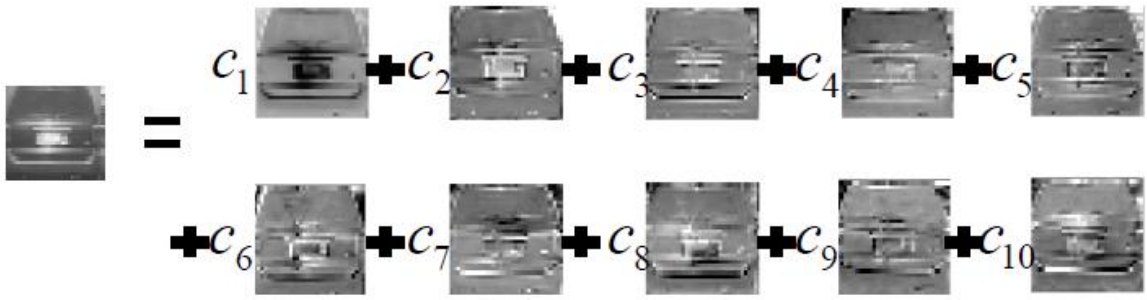


Figure 1.13 – Illustration des modèles linéaires de sous-espace ACP. La partie gauche montre un échantillon candidat, et la partie droite présente une combinaison linéaire d'échantillons propres [24].

a. Méthodes basées sur les modèles génératifs

Le suivi utilisant le modèle génératif se base sur l'apprentissage d'un modèle pour représenter l'objet cible, puis l'utilise pour trouver la région la plus similaire dans les trames suivantes.

Black et al. [33] ont proposé une approche basée sur un sous-espace propre, pour calculer la transformation affine de l'image courante de l'objet à l'image reconstruite en utilisant des vecteurs propres. La représentation de sous-espace de l'apparence d'un objet est construite en utilisant l'analyse en composantes principales (ACP), comme illustrée dans la figure 1.13, et la transformation par minimisation de la différence entre l'image d'entrée et l'image reconstruite à l'aide les vecteurs propres. Le suivi est effectué en estimant itérativement les paramètres de la transformation qui rendent la différence d'images minimale.

Ross et al. [2] ont proposé un algorithme de suivi IVT qui utilise un modèle sous-espace incrémental pour décrire l'objet cible afin d'adapter les changements d'apparence. L'idée de cet algorithme est d'apprendre incrémentielle une représentation de sous-espace de faible dimension, en s'adaptant en ligne aux changements de l'apparence de la cible. La mise à jour du modèle, basée sur des algorithmes incrémentiels pour l'analyse en composantes principales, comprend deux caractéristiques : une méthode pour mettre à jour correctement la particule moyenne, et un facteur d'oubli (forgetting factor) pour assurer moins de puissance de la modélisation est expiré pour les observations plus anciennes. Wang et al. [120] appliquent l'analyse des moindres carrés partiels pour apprendre un sous-espace caractéristique de faible dimension pour le suivi d'objets robustes.

Li et al. [121] tirent parti de la décomposition en tenseur en ligne pour construire un modèle d'apparence à base de tenseur pour le suivi d'objet visuel robuste. Dans [122], un algorithme d'apprentissage incrémental est développé pour le tenseur sous-espace pondéré (Weighted Tensor Subspace WTS) afin de l'adapter aux changements d'apparence pendant le suivi.

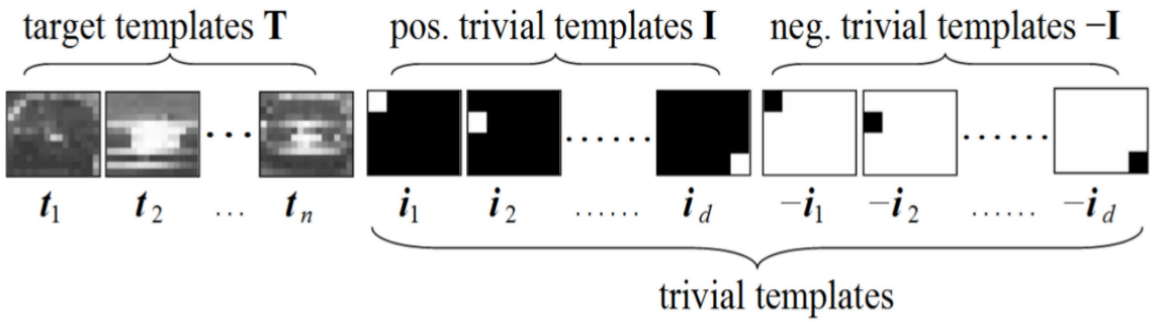


Figure 1.14 – Illustration des modèles utilisés pour la reconstruction L1 [3].

Cependant, les modèles d'apparence adoptés dans les approches de suivi sont généralement sensibles aux variations d'illuminations, point de vue et pose. En effet, il leur manque un critère de description d'objet compétent qui capture les propriétés statistiques et spatiales de l'apparence de l'objet [43]. Yang et al. [123] ont proposé pour l'algorithme incrémental d'analyse en composantes principale (IPCA) [2], un descripteur supplémentaire PCA-HOG pour le suivi de la main visuelle.

Shirazi et al. [124][125], ont proposé une approche de suivi basée sur des sous-espaces affines (construits à partir de plusieurs images) qui sont capable de gérer l'occultation, pose, et les variations d'illumination. Ils ont utilisé des sous-espaces affines non seulement pour représenter l'objet, mais aussi les régions candidates que l'objet peut occuper. En plus, ils ont proposé une nouvelle approche pour mesurer la distance affine sous-espace à sous-espace par l'utilisation de la géométrie non-euclidienne des variétés de Grassmann. Le suivi est effectué en utilisant le filtrage des particules.

La représentation parcimonieuse [126] (*Sparse representation*) a également été appliquée dans le suivi visuel d'objet pour déterminer la cible avec une erreur de reconstruction minimum à partir de l'espace du modèle. Mei et Ling. [3] ont proposé une méthode de suivi robuste, appelée le tracker L1, en traitant le suivi d'objets comme un problème d'approximation parcimonieuse et d'introduire le modèle trivial pour approcher le bruit et l'occultation. Pendant le suivi, les candidats cibles sont représentés sous la forme d'une combinaison linéaire parcimonieuse d'ensemble des modèles incluant des modèles de cible qui sont obtenus à partir de trames précédentes et des modèles triviaux, comme le montre la figure 1.14. Bien que, le tracker L1 fonctionne bien sur plusieurs scénarios difficiles, il nécessite des ressources de calcul élevé en raison de nombreux calculs de minimisation L1. Pour résoudre ce problème, Mei et al. [127] ont étendu le tracker L1. Les échantillons les plus insignifiants sont filtrés par la limite de l'erreur minimale avant de résoudre le calcul coûteux de fonction minimisation L1. Cette nouvelle stratégie peut améliorer la vitesse de suivi avec la

précision. Ainsi, pour réduire la sensibilité au bruit de fond dans la zone d'objet sélectionnée, Wang et al. [128] et Jia et al. [129] appliquent l'histogramme de la représentation parcimonieuse qui se base sur les patches locaux pour décrire l'objet. Zhang et al. [130][131] ont appliqué l'apprentissage multitâche parcimonieux et l'apprentissage faible rang parcimonieux pour étendre le tracker L1, et d'utiliser les relations sous-jacentes entre les échantillons de reconstruction.

b. Méthodes basées sur les modèles discriminatifs

Malgré les succès, les modèles génératifs généralement rencontrent des difficultés pour décrire l'objet cible sans considérer l'information de l'arrière-plan, en particulier lorsque l'apparence de l'objet cible change dramatiquement et/ou que l'arrière-plan est encombré. Au contraire, les modèles discriminatifs décrivent l'objet par rapport à l'arrière-plan, en transformant le problème de suivi en un problème de classification binaire pour distinguer l'objet cible de l'arrière-plan [132]. Ils visent à maximiser la séparation entre les régions objet et non-objet de manière discriminante. De plus, ils se concentrent sur la découverte de caractéristiques très informatives pour le suivi des objets visuels. Par conséquent, ils sont plus robustes aux scénarios complexes en modélisant explicitement l'arrière-plan comme échantillons d'entraînement négatifs [24]. Les méthodes basées sur le modèle discriminatif ont évolué rapidement ces dernières années.

Avidan. [133] a proposé une approche discriminative, où le classifieur SVM (Support Vector Machines) et un tracker basé sur le flot optique sont combinés pour suivre des véhicules sur de longues séquences vidéo. SVM est un schéma général de classification qui étant donné un ensemble d'exemples d'apprentissage positifs (objet) et négatifs (régions pouvant être confondues avec l'objet), et trouve le meilleur hyperplan de séparation entre les deux classes. Le suivi est effectué en maximisant le score de classification du SVM sur des régions de l'image afin d'estimer la position de l'objet. Ce type d'approche intègre explicitement la connaissance du fond de la scène au sein du suivi. L'approche de suivi par ensemble tracking [134] adopte l'algorithme Adaboost et divise la phase d'entraînement complexe en un ensemble de classifieurs faibles, qui peuvent être calculés en ligne. Le principe de base de l'ensemble tracking est la construction d'un classifieur fort mis à jour à chaque image dans le but de séparer les pixels du fond de ceux de l'objet [135].

Les Algorithmes de suivi basés sur Boosting en ligne souffrent du problème de dérive de modèle, car les classificateurs faibles mises à jour dans chaque trame ne sont pas robustes aux erreurs accumulées du suivi [24][132]. Afin de résoudre ce problème, les chercheurs adoptent

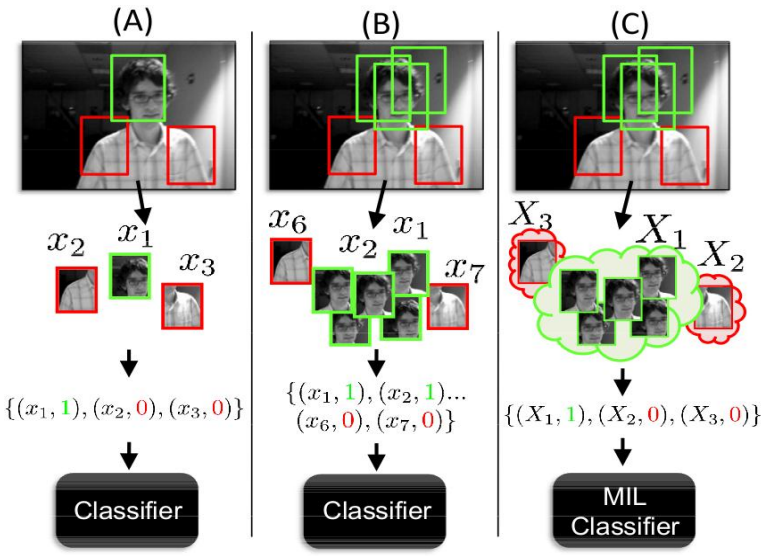


Figure 1.15 – Mise à jour d'un modèle d'apparence discriminative : (a) Utilisation d'un seul patch d'image positive. (b) Utilisation de plusieurs patchs d'image positive pour mettre à jour d'un classifieur traditionnel discriminatif. (c) Utilisation d'un sac positif constitué de plusieurs patchs d'image [34].

les techniques d'apprentissage semi-supervisé [136] pour le suivi des objets visuels. Grabner et al. [137] ont proposé le suivi SemiBoost pour explorer le continuum entre un classifieur précédent appris à partir de la trame initiale et tous les échantillons comme non étiquetés pour la mise à jour dans les trames suivantes. En autre terme, le tracker SemiBoost est de formuler le processus de mise à jour de manière semi-supervisée en tant que décision combinée d'un classifieur précédent et d'un classifieur en ligne.

Babenko et al. [34] ont proposé une méthode de suivi basée sur l'apprentissage en ligne à instance multiple au lieu des méthodes traditionnelles d'apprentissage supervisé, qui résout les incertitudes de l'endroit où prendre les mises à jour positives pendant le suivi. Comparé à d'autres méthodes traditionnelles (voir figure 1.15), en sélectionnant un seul échantillon positif imparfait ou en utilisant plusieurs échantillons bruyants positifs. Le tracker MIL traite les échantillons d'entraînement comme des «sacs». Un sac est considéré comme positif s'il contient au moins une instance positive, sinon le sac est mis à négatif. Le tracker MIL conserve suffisamment d'échantillons d'entraînement et tolère le bruit d'étiquetage lors de la mise à jour de son modèle. Au lieu de traiter les échantillons dans chaque sac [34], Zhang et al. [5] proposent le tracker WMIL (online Weighted MIL) qui intègre l'importance de l'échantillon dans le processus d'apprentissage en ligne « Boosting » en supposant que l'échantillon le plus important est connu lors entraîner le classificateur. Ce tracker donne des résultats de suivi robuste.

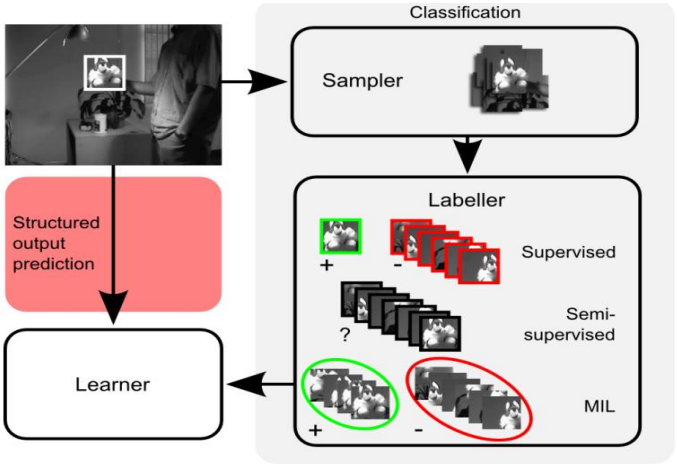


Figure 1.16 – Différents paradigmes adaptatifs de suivi par détection en utilisant le tracker Struck [6].

Hare et al. [6] ont proposé l'algorithme Struck pour le suivi adaptatif d'objets qui se base sur la prédiction de sortie structurée. Il aborde la limitation des suivis précédents tels que [137] et [34], qui séparent la localisation cible et la mise à jour du modèle en deux étapes distinctes, comme le montre la figure 1.16. L'algorithme est basé sur l'utilisation d'un noyau sur la sortie structuré de machine à vecteurs de support (SVM), qui est apprise en ligne pour fournir un suivi adaptatif. Cet algorithme utilise l'apprentissage multi-noyau (MKL). Les performances de suivi sont significativement améliorées en combinant un noyau Gaussien sur les caractéristiques de Haar 192-D avec un noyau d'intersection sur les caractéristiques de l'histogramme 480-D, mais à un coût en vitesse [102]. L'algorithme RobStruck [102] est une extension de Struck qui utilise des caractéristiques plus riches, il adapte l'échelle et applique un filtre de Kalman pour l'estimation de mouvement.

Kalal et al. [8] ont proposé un algorithme de suivi visuel robuste, appelé TLD (Tracking Learning and Detection). Cet algorithme décompose explicitement la tâche de suivi à long terme en trois sous-tâches : le suivi, l'apprentissage et la détection. Chaque sous-tâche est traitée par un composant unique et les composants fonctionnent simultanément. Le suivi suit l'objet d'une trame à l'autre. Le composant de détection localise l'objet dans toutes ses apparences qui ont été observés jusqu'à présent, et corrige le suivi si nécessaire. L'apprentissage estime les erreurs du détecteur et les mises à jour pour éviter ces erreurs dans le futur.

Récemment, les méthodes de suivi discriminatives basées sur le filtre de corrélation ont prouvé qu'elles étaient capables d'atteindre une vitesse assez élevée et des performances de suivi robustes [138]-[140]. Classiquement, les filtres de corrélation sont conçus pour produire des pics de corrélation pour chaque cible intéressée dans la scène tout en donnant de faibles

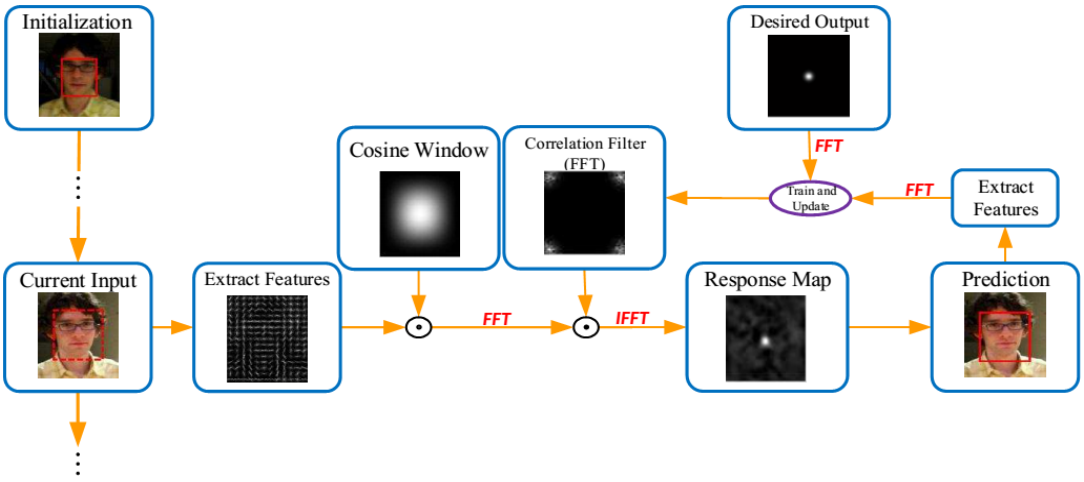


Figure 1.17 – Un flux de travail général pour des méthodes typiques de suivi basées sur le filtre de corrélation [140].

réponses à l'arrière-plan, qui sont généralement utilisés pour détecter les modèles attendus. Le cadre d'une méthode typique de suivi basée sur le filtre de corrélation peut être résumé comme suit [140]. Dans la première trame, un filtre de corrélation initial est entraîné avec un patch d'image recadrée à partir d'une position donnée de la cible. Ensuite, pour chaque trame suivante, différentes caractéristiques peuvent être extraites des données d'entrée brutes, et une fenêtre de cosinus est habituellement appliquée pour lisser les effets de frontière, comme le montre la figure 1.17. Par la suite, une carte de réponse (une carte de confiance) est générée efficacement par une transformée de Fourier rapide (FFT). La position avec la valeur maximale dans cette carte est prédite comme la nouvelle localisation de l'objet cible. Enfin, L'apparence à la position estimée est extraite pour la formation et la mise à jour du filtre de corrélation. Bolme et al. [141] ont proposé le filtre de la somme de sortie minimale de l'erreur au carré pour le suivi visuel sur des images en niveaux de gris. En autre terme, l'approche proposée est basée sur la recherche d'un filtre de corrélation adaptatif en minimisant la somme de sortie de l'erreur au carré (MOSSE). Le suivi basé sur le filtre MOSSE est efficace du point de vue du calcul avec une vitesse atteignant plusieurs centaines de trames par seconde, et robuste aux variations d'illumination, d'échelle, de pose et de déformations non rigides. Sur la base du cadre de base du filtre MOSSE, de nombreuses améliorations ont été apportés plus tard. Henriques et al. [142] ont amélioré le filtre MOSSE en introduisant la méthode du noyau. Cette approche appelée CSK, montre une excellente performance, tout en maintenant l'efficacité du calcul. L'approche CSK se fonde sur les caractéristiques d'intensité d'illumination et est encore améliorée par l'utilisation des caractéristiques du HOG dans l'algorithme de tracker KCF [143]. Les approches de suivi basées sur le filtre de corrélation DSST [144] et SAMF [145] ont appliqué une stratégie de recherche à multi-échelles pour

estimer l'échelle réelle de l'objet cible. Dans [139], Ma et al. Introduisent le classifieur de fougère aléatoire en ligne pour le suivi à long terme, la méthode est appelée LCT. Dans cette méthode, un composant re-détection est ajouté dans le système de suivi.

1.6.3 Suivi de Silhouettes

Les objets peuvent avoir des formes complexes, par exemple, les mains, la tête et les épaules qui ne peuvent pas être bien décrites par des formes géométriques simples (rectangle, ellipse...). Les méthodes basées sur la silhouette fournissent une description de forme précise pour ces objets, en utilisant l'information encodée à l'intérieur de la région objet. Le but de ces méthodes de suivi fondées sur l'utilisation de silhouettes est d'estimer la région d'objet dans chaque trame au moyen d'un modèle généré en utilisant les trames précédentes [21]. On distingue deux catégories de ces méthodes : la correspondance de formes et l'évolution du contour.

1.6.3.1 Méthodes de correspondance de formes

Les méthodes de correspondance de formes recherchent la silhouette de l'objet dans la trame courante. La recherche est effectuée en calculant la similarité de l'objet avec le modèle généré à partir de la silhouette d'objet hypothétique qui se base sur la trame précédente, la mise à jour du modèle (Le modèle d'objet est réinitialisé dans chaque trame après détection) permet de prendre en compte les changements d'illumination et de point de vue. En 1993, Huttenlocher et al. [146] ont effectué une correspondance de forme en utilisant une représentation basée sur les arêtes. Une surface de corrélation a été construite en utilisant la distance de Hausdorff et le minimum a été sélectionné comme la nouvelle position de l'objet. Cependant dans [146], la transformation non rigide de la forme ne peut pas être manipulée explicitement [147]. Li et al. [148] ont proposé d'utiliser la distance de Hausdorff pour la procédure de correspondance utilisée pour la vérification des trajectoires et le problème d'estimation de la pose. Le suivi est réalisé en évaluant le vecteur de flot optique calculé à l'intérieur de la silhouette hypothétique de sorte que le flot moyen fournit la nouvelle position d'objet.

La correspondance de formes peut aussi s'effectuer en calculant la distance qui sépare les modèles objets associés aux silhouettes détectées dans deux images consécutives. Kang et al. [149] ont utilisé des histogrammes de couleur et des arêtes comme modèles d'objets. Contrairement aux histogrammes traditionnels, ils proposent de générer des histogrammes à partir de cercles concentriques de différents rayons centrés sur un ensemble de points de contrôle sur un cercle de référence. Les histogrammes résultants sont invariants à de

nombreuses transformations et le score de correspondance est calculé par distance de Bhattacharya et divergence de Kullback-Leibler. Sato et Aggarwal [150] ont proposé la transformation TSV (Temporal Spatio-Velocity) pour le suivi où la trajectoire des silhouettes est générée en appliquant la transformée de Houg dans l'espace des vitesses aux silhouettes objets dans les trames consécutives. Les matrices de flot (vertical et horizontal) obtenues en appliquant la transformée de Houg fournissent l'image TSV (4D). L'image TSV encodant le mouvement principal d'une région en mouvement ainsi que la vraisemblance de ce mouvement. Il s'agit ici d'une mise en correspondance de mouvements.

Cai et al [151] ont proposé la méthode DGT (Dynamic Graph based Tracker). Le suivi de la cible par DGT est formulé comme un problème de correspondance entre le graphe cible et le graphe candidat. L'algorithme SLIC (Simple Linear Iterative Clustering) est utilisé pour oversegment la zone de recherche en plusieurs parties (superpixels), et d'exploiter l'approche Graph Cut pour séparer les superpixels d'avant-plan à partir de superpixels d'arrière-plan. Une matrice d'affinité basée sur le mouvement, l'apparence et les contraintes géométriques est construite pour décrire la fiabilité des correspondances. La correspondance optimale entre les superpixels candidats est obtenue à partir de la matrice d'affinité appliquant la technique spectrale. La localisation de la cible est déterminée par une série de parties avec succès correspondant selon leur fiabilité de correspondance.

1.6.3.2 Méthodes d'évolution du contour

Contrairement aux méthodes de correspondance de formes, les méthodes de suivi des contours font évoluer un contour initial dans la trame précédente à sa nouvelle position dans la trame courante. Cette évolution de contour nécessite qu'une partie de l'objet dans la trame courante se chevauchent avec la région d'objet dans la trame précédente. Le suivi par l'évolution d'un contour peut être effectué en utilisant soit les modèles d'espace d'état, soit la minimisation directe d'une certaine énergie [21][152].

a. Suivi par Modèle d'espace d'état

Dans les méthodes de suivi de contours qui utilisent les modèles d'espace d'état, l'état d'un objet est défini en termes de forme et de paramètres de mouvement du contour. L'état est mis à jour à chaque instant de manière à maximiser la probabilité a posteriori du contour. La probabilité posteriori dépend de l'état a priori et de la vraisemblance actuelle qui est définie en termes de la distance du contour à partir des bords.

Terzopoulos et Szeliski. [153] ont fusionné le snake (modèle de contour actif) et le filtre de Kalman dans le but de suivre les objets. Le Kalman snake utilise la dynamique de snake

Figure 1.18 – Suivi de voiture en utilisant la méthode des courbes de niveaux [21].

comme un modèle de système. Le nouvel état du contour est prédit en utilisant le filtre de Kalman. Les gradients d'image sont utilisés pour l'étape de correction. Isard et Blake. [44] ont utilisé de paramètres de forme de spline et de paramètres de mouvement affine pour définir l'état de l'objet et le filtre à particule pour la mise à jour de l'état. Les échantillons initiaux pour le filtre à particules sont obtenus en calculant les variables d'état à partir des contours extraits dans des trames consécutives pendant une phase d'entraînement. MacCormick et Blake [154] ont utilisé le principe d'exclusion afin de gérer les occultations. Chen et al [155] ont proposé un nouveau cadre de HMM (Modèles de Markov cachés) pour le suivi d'objets basé sur le contour. Le filtre JPDAF (Joint Probability Data Association Filtering) est utilisé pour le calcul des probabilités de transition d'état de HMM. En tenant compte de l'interrelation entre la mesure voisine.

b. Suivi par Minimisation directe de la fonctionnelle d'énergie du contour

L'énergie de contour est définie en termes d'informations temporelles de type gradient d'image temporelle (flux optique) ou les statistiques d'apparence générées par l'objet et les régions d'arrière-plan. Les méthodes de segmentation et de suivi basées sur le contour minimisent l'énergie soit par des méthodes greedy, soit par une descente en gradient.

Bertalmio et al [156] ont utilisé la contrainte de flot optique pour évoluer le contour dans les images successives en utilisant une représentation par courbes de niveau. Les auteurs utilisent deux fonctionnelles énergétiques: une fonctionnelle de suivi du contour et une fonctionnelle de modelage de l'intensité qui minimise les changements d'intensité d'une trame à la suivante, et les deux fonctionnelles sont minimisées simultanément. Mansouri [157] a appliqué la contrainte de flot optique pour calculer le vecteur de flot pour chaque pixel dans la région d'objet complète. L'énergie de contour est ensuite évaluée en fonction de la contrainte de constance de luminosité. L'énergie est minimisée en effectuant itérativement ce processus. La figure 1.18 montre les résultats de la méthode de suivi proposée par Mansouri dans une séquence de voiture.

(a)

(b)

Figure 1.19 – Résultats du suivi du contour en utilisant la méthode proposée par Yilmaz et al (a) le suivi d'un joueur de tennis, (b) le suivi en présence d'une occultation [21].

En 2003, Cremers et Schnorr, ont également utilisé le flot optique pour l'évolution du contour et des vecteurs de flot homogènes dans la région de l'objet. Les formes a priori sont générés à partir d'un ensemble de contours d'objet, de sorte que chaque point de contrôle du contour est associé à une gaussienne avec une moyenne et écart type des positions spatiales des points de contrôle correspondants sur tous les contours. Cette approche peut gérer l'occultation partielle. Yilmaz et al. [74][158][159] ont développé un contour d'objet en utilisant les modèles de couleurs et de texture générés dans une bande autour de la frontière de l'objet. (Voir la figure 1.19 (a)). La largeur de la bande sert à combiner les méthodes de suivi des contours basées sur la région et les frontières dans un cadre unique. Les auteurs ont modélisé la forme de l'objet ainsi que ses changements au moyen d'un modèle de forme basé sur des courbes de niveau. Dans ce modèle, les points des courbes de niveau contiennent les moyennes et écarts-types des distances des points aux contours de l'objet. Le modèle de forme basé sur des courbes de niveaux résout les occultations d'objet au cours du suivi (voir la figure 1.19 (b)).

Une méthode récente, pour le suivi des objets non-rigides est présentée dans [160], fournissant un cadre pour l'interaction de suivi et de segmentation. La méthode étend l'idée de Forêts de Hough et couple la transformation de Hough généralisée pour la détection d'objet avec la méthode de segmentation interactive de GrabCut. Ainsi, la méthode permet le suivi simultané des objets et la segmentation grossière dans chaque trame vidéo (figure 1.20). En réduisant la quantité des échantillons d'entraînement bruyants utilisée pour l'apprentissage en ligne. Plus tard, Duffner et Garcia, [161] ont étendu ce cadre pour faire la classification au niveau de pixel basée sur Hough. Ce qui améliore encore les performances de suivi, en particulier sur de petites régions.

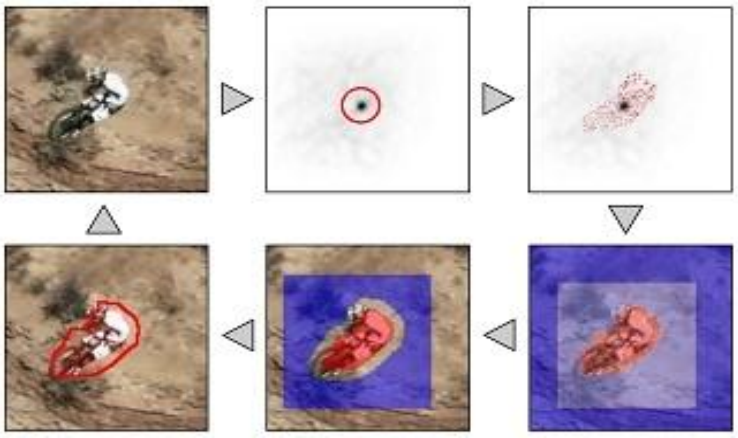


Figure 1.20 – Boucle de suivi de Hough, à partir de l'image supérieure gauche : Trame d'entrée, résultat de détection d'objet basé sur Hough, la projection arrière et supportant des positions d'image, Segmentation guidée, robuste mise à jour et le résultat de suivi, (Rouge: support d'avant-plan, segmentation et mises à jour; Bleu: segmentation d'arrière-plan et mises à jour) [160].

1.7. Conclusion

Dans ce chapitre, nous avons présenté dans un premier temps les différentes représentations de l'objet et les caractéristiques visuelles, celles-ci jouent un rôle fondamental dans le suivi visuel. Tandis que dans un deuxième temps, nous avons présenté un état de l'art sur la détection d'objet, qui représente le fondement de tout algorithme de suivi. Chaque méthode de suivi nécessite un mécanisme de détection d'objet, soit sur toutes les trames, soit lorsque l'objet apparaît pour la première fois dans la vidéo. Cependant, la détection visuelle est difficile, car l'apparence de l'objet peut varier en raison de nombreux facteurs (l'occultation, l'illumination, la texture et l'articulation, etc). Enfin, nous avons présenté un état de l'art sur le suivi d'objet visant à montrer la diversité des approches proposées dans ce domaine. La différence entre ces approches réside en partie dans le choix de la représentation des objets, des caractéristiques de l'image utilisées et de la nature du mouvement estimé. Ce choix dépend de l'application ainsi que de la vidéo traitée. Bien que de nombreuses approches de suivi d'objets ont été proposées dans la littérature, certains des défis fondamentaux n'ont pas été complètement résolus. Pour résoudre les problèmes du suivi, les trackers exploitent l'apparence de l'objet de manière dynamique considérant que la modélisation de l'apparence d'objet est une étape importante pour réussir à suivre l'objet correctement.

Chapitre 2

Suivi d'objet par l'algorithme Mean shift

Sommaire

2.1 Introduction

Le suivi d'objet dans des séquences d'images est, depuis ces dernières décennies, un thème de recherche très actif en vision par ordinateur. Bien que de nombreux algorithmes de suivi d'objets ont été développés ces dernières années, le suivi d'objet demeure un problème non résolu à cause du nombre élevé de facteurs environnementaux. Parmi les différents algorithmes de suivi, le tracker Mean shift est l'un des algorithmes de suivi les plus efficaces pour les applications en temps réel, en raison de sa simplicité et de sa robustesse. L'algorithme Mean shift est une procédure itérative d'estimation de mode (maximum local) d'une densité de probabilité non-paramétrique, où la position estimée de l'objet est déplacée vers un centre de gravité local jusqu'à convergence. Le tracker Mean Shift maximise la similarité d'apparence itérativement en comparant les histogrammes de l'objet modèle et une fenêtre autour de la position estimée d'objet candidat.

Dans ce chapitre, nous présentons d'abord, un état de l'art sur l'algorithme Mean shift. Ensuite, nous présentons le principe et les différentes étapes de l'algorithme de suivi

Mean shift qui utilise une fonction de densité des histogrammes de couleur pour représenter le modèle et le candidat cible. Enfin, nous présentons certains algorithmes qui peuvent gérer quelques problèmes de tracker Mean shift.

2.2 Etat de l'art sur l'algorithme Mean shift

L'algorithme Mean shift est un estimateur non paramétrique du gradient de densité, basé sur l'utilisation de noyau. Il a été initialement présenté par Fukunaga en 1975 [67], remis au goût du jour vingt ans plus tard par Cheng en 1995 [68]. Il s'agit d'un algorithme de clustering des données par l'estimation de leur densité. Dans une fenêtre de recherche de taille constante on trouve le centroïde (position moyenne des données) et la fenêtre qui est centrée sur le centroïde. Cette procédure est répétée jusqu'à la convergence, et le point de convergence est considéré comme le centre de cluster pour les données visitées (voir la figure 2.1). Cet algorithme a été utilisé dans le cadre de la segmentation, à partir de 1997 avec les travaux de Comaniciu et Meer [66][162]. En 1998, Bradski. [117] a modifié Mean shift et a développé l'algorithme Camshift pour suivre un visage en mouvement. L'algorithme Mean shift a été présenté à la vision par ordinateur par Comaniciu et al. [1] qui ont proposé son utilisation pour le suivi des objets, où le suivi est formulé, en maximisant le coefficient de Bhattacharyya entre l'histogramme de couleurs pondéré représentant l'objet et l'histogramme correspondant représentant la position hypothétique. Le tracker Mean shift est largement utilisé pour le suivi des cibles, robuste à l'occultation partielle, la rotation, le mouvement de fond et la déformation non-rigide de la cible, car il adopte l'histogramme du noyau pour représenter son modèle d'objet. Cependant, il existe également de nombreuses limitations telles que le manque de l'information spatiale, et le déclin de la performance pour le changement d'échelle

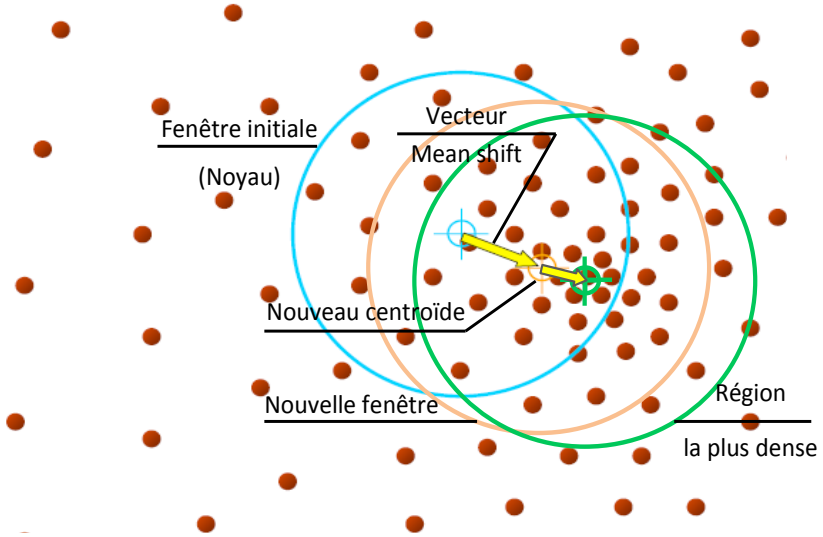


Figure 2.1 – Description intuitive de la convergence de la procédure Mean shift.

de l'objet, l'occultation complexe et lorsque la couleur de l'objet est similaire à la couleur de fond. De nombreux chercheurs ont proposé divers algorithmes améliorés qui peuvent donner une meilleure précision et performance dans une scène complexe.

Pour surmonter le problème de changement d'échelle, les chercheurs ont proposé une méthode de suivi qui permet d'adapter la taille de l'objet pendant le suivi basée sur Mean shift. Collins [163]. a essayé de combiner le Mean shift et la méthode de l'espace d'échelle, pour résoudre le problème de suivi des cibles lorsque la bande passante du noyau a changé en temps réel, mais la vitesse de l'algorithme était mauvaise. Allan et al. [118] ont amélioré la méthode Camshift en utilisant l'histogramme pondéré et l'histogramme de ratio pour représenter le modèle cible. Camshift peut s'adapter à la taille de la zone cible de sorte qu'il ait une bonne adaptabilité à la variation de la cible. Zivkovic et al. [164] ont proposé une méthode de suivi basée sur l'histogramme de couleur de 5 degrés de liberté (5-degree of freedom). Cette méthode estime simultanément la position de l'objet suivi et l'ellipse qui se rapproche de la forme de l'objet. Ning et al. [165] ont proposé un algorithme de suivi Mean shift adapté à l'échelle et à l'orientation (SOAMST) pour résoudre le problème de la façon d'estimer les changements d'échelle et d'orientation de l'objet dans le cadre de suivi par Mean shift. Récemment, Vojir et al. [166][107] ont proposé l'algorithme ASMS pour traiter le problème de l'adaptation de l'échelle en présentant un nouveau mécanisme théoriquement justifié d'estimation d'échelle. Cet algorithme repose uniquement sur la procédure de Mean shift pour la distance de Hellinger qui sert principalement à l'expansion de l'échelle.

L'utilisation d'histogrammes de couleurs pour modéliser l'objet cible rend l'algorithme Mean shift incapable de détecter l'information spatiale qui est perdue. Aussi, il ne peut pas distinguer entre la cible et le fond, lorsque la cible a une apparence similaire. Birchfield et al. [167] ont proposé le spatiogramme pour le tracker basé sur le noyau. Le spatiogramme est une version améliorée de l'histogramme de couleur incluant des moments d'ordre potentiellement plus élevés. Un histogramme est un spatiogramme d'ordre zéro, tandis que les spatiogrammes de second ordre contiennent des moyennes spatiales et des covariances pour chaque groupe d'histogrammes. Cette information spatiale permet toujours des transformations assez générales, comme dans un histogramme, mais capture une description plus riche de la cible pour augmenter la robustesse du suivi. Ning et al. [14] ont utilisé l'histogramme conjoint de couleur-texture pour représenter le modèle de la cible puis l'appliquer au cadre de Mean shift. Les caractéristiques de texture sont extraites à l'aide d'un motif binaire local LBP. Xiaorong et Zhihu. [19] ont utilisé l'histogramme conjoint de couleur-CLTP, qui exploite efficacement les informations structurelles de la cible. Ning et al. [165] ont proposé l'algorithme CBWH

(Corrected Background-Weighted Histogram) qui utilise le mécanisme de mise à jour de fond, dans le cadre de l'algorithme Mean shift. Cet algorithme peut efficacement réduire les interférences de fond dans la localisation de l'objet cible.

De nombreuses méthodes combinent l'algorithme Mean Shift avec le filtre Kalman [113][168] [169] ou le filtre à particules [170][171] ont été proposés, pour gérer le problème d'occultation. Alors que, Phadke et al. [172] ont résolu ce problème en incorporant les caractéristiques de couleur, de texture et de bord pour représenter l'apparence de la cible, et en appliquant l'algorithme Mean shift. Récemment, Dou et Li. [16] ont amélioré la robustesse du suivi en utilisant l'histogramme conjoint multi-caractéristiques (SIFT, Couleur, LBP, Bord), et en intégrant le filtre à particules et le Mean shift.

2.3 Principe de tracker Mean shift

Suivre un objet dans une séquence d'images est une opération qui consiste à localiser la région d'objet au fil du temps dans une vidéo. Le problème du suivi d'objet peut s'exprimer en termes de détection de l'objet au sein de chaque trame. Le tracker Mean shift est un algorithme qui concerne la représentation de la cible et la localisation d'objets. La représentation de la cible basée sur l'histogramme de couleurs est régularisée par un masque spatial avec un noyau isotrope tandis que la localisation de la cible est formulée en utilisant le bassin d'attraction des maxima locaux.

Comaniciu et al, [1] utilisent un histogramme pondéré calculé sur une région rectangulaire pour représenter l'objet. Au lieu de réaliser une recherche exhaustive pour localiser l'objet, ils utilisent le procédé Mean shift [66]. Le principe de tracker Mean shift est de maximiser la similarité d'apparence itérativement en comparant les histogrammes du modèle cible et une fenêtre autour de la position hypothèse de cible. La similarité entre deux histogrammes est définie en termes de coefficient de Bhattacharyya. À chaque itération, le vecteur Mean shift est calculé tel que la similarité entre les histogrammes est augmentée. Ce processus est répété jusqu'à ce que la convergence soit réalisée, qui s'effectue habituellement dix à quinze itérations. Pour la génération de l'histogramme, les auteurs utilisent une pondération définie par un noyau spatial qui donne un poids plus important aux pixels plus proches du centre de l'objet.

Le suivi par l'algorithme Mean shift utilise la densité de couleur de l'objet, effectue à partir de sa position initiale dans la première trame. L'initialisation s'effectue manuellement. L'objet d'intérêt est modélisé par une forme géométrique, sur laquelle on calcule sa distribution de

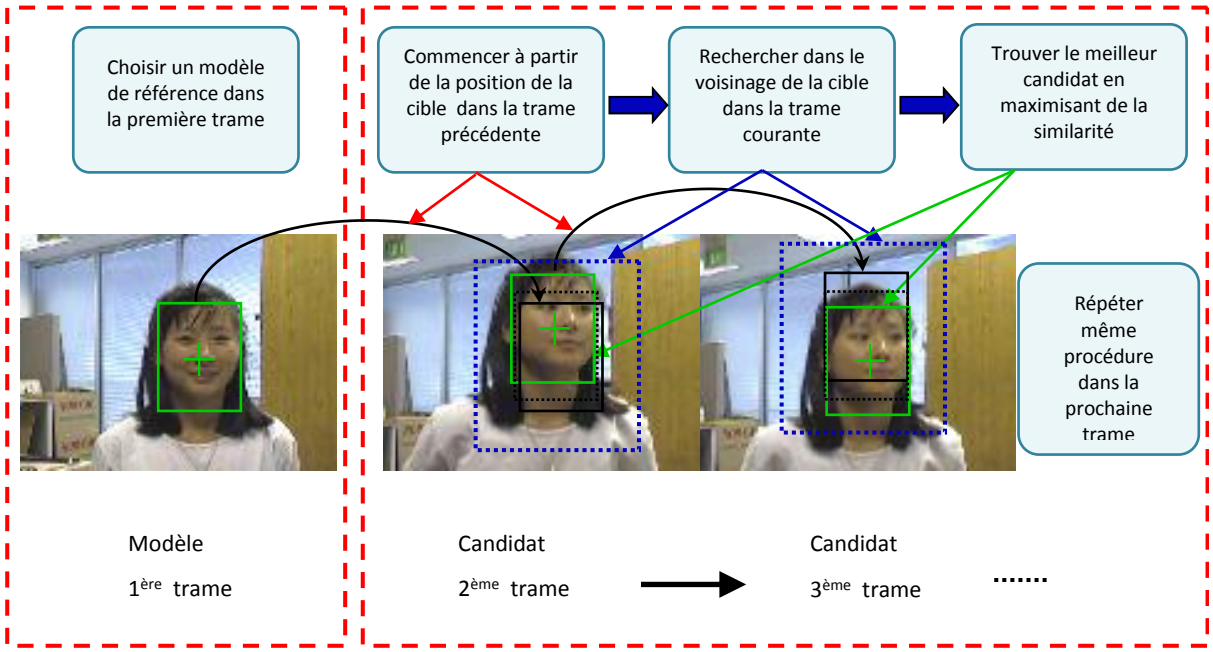


Figure 2.2 – Processus de suivi d'objet par le tracker Mean shift : (a) étape de représentation, (b) étape de localisation.

couleur. La distribution de couleur initiale est référencée en tant que modèle, et est ensuite comparée à celle des sites candidats pour déterminer la position la plus probable dans la trame suivante. La figure ci-dessus illustre le principe de tracker Mean shift.

2.4 Algorithme du Mean shift pour le suivi d'objet.

L'algorithme du Mean shift [108] est une procédure itérative d'estimation des modes d'une densité de probabilité non paramétrique à partir d'un ensemble d'échantillons $\{x_i\}_{i=1\ldots n}$. Il est similaire à une descente de gradient pour la recherche des modes d'une densité de probabilité (voir figure 2.3). La densité de probabilité sous-jacente est alors approchée sous la forme :

$$p(x) = C \sum_{i=1}^{n} K(x - x_i) \qquad (2.1)$$

Où la fonction *K* est une fonction noyau, qui pondère la contribution de chaque échantillon à la densité de probabilité *p*, et *C* une constante de normalisation.

Dans [1], la recherche de la cible à l'instant courant se base sur des distributions des couleurs dans une boîte englobante. La distribution est représentée par un histogramme de couleurs. L'algorithme consiste alors à déplacer une fenêtre d'analyse (noyau spatial) de manière à

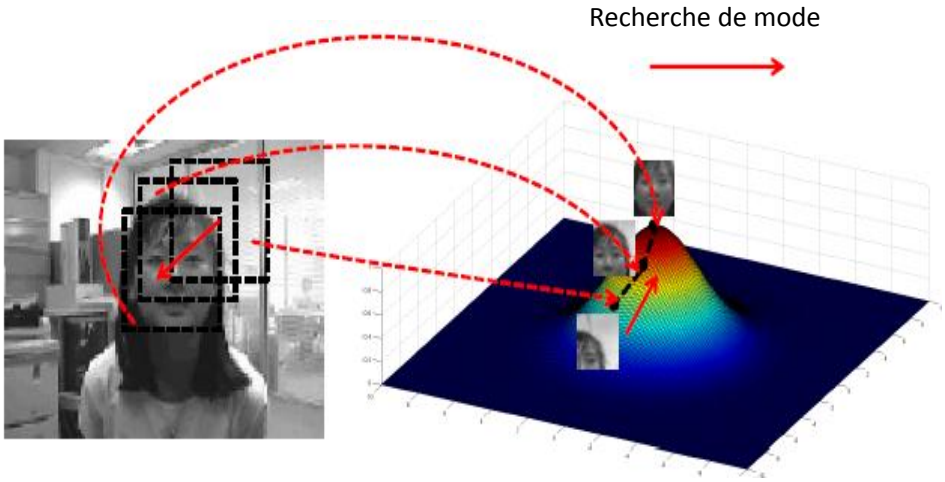


Figure 2.3 – Illustration le processus de la recherche de mode (maximum local) par Mean shift [24].

déterminer la fenêtre dont l'histogramme coïncide le mieux avec l'histogramme du modèle de référence. La similarité entre l'histogramme du modèle de référence et l'histogramme candidat est mesurée par le coefficient de Bhattacharyya. Un poids est associé à cette mesure de similarité et on peut alors calculer le vecteur Mean Shift, qui a pour but de fournir la nouvelle position estimée de la cible dans l'image courante. Le déplacement du noyau est contrôlé par une montée de gradient itérative. Dans cette section, nous présentons le fonctionnement de suivi d'objet par l'algorithme Mean Shift proposé dans [1].

2.4.1 Représentation de la cible

2.4.1.1 Modèle cible

La cible est généralement représentée par une région rectangulaire de taille (h_x, h_y) dans une image. Un espace caractéristique est choisi (l'espace RGB est généralement utilisé) pour déterminer un histogramme de la distribution des pixels dans la région cible. L'histogramme est représenté par le modèle cible $\hat{q} = \{ \hat{q}_u \}_{u=1\ldots m}$ de m classes de couleurs, qui utilise pour décrire l'apparence de l'objet situé dans la région cible. On note $\{x_i^*\}_{i=1\ldots n}$ l'ensemble des coordonnées des n pixels du modèle cible, centré à 0, et normalisé par les demi-rayons du rectangle h_x et h_y. La loi de probabilité des couleurs est calculée en utilisant une fonction de profil convexe (noyau isotrope) $k(x)$, qui donne plus d'importance aux pixels au voisinage du centre plutôt que ceux aux bords. La fonction $b : R^2 \rightarrow \{1 \ldots m\}$ associe à chaque pixel x_i^* l'indice $b(x_i^*)$ de sa couleur dans le m-histogramme. La probabilité de la couleur $u = 1 \ldots m$ dans le modèle cible est calculée comme suit :

$$\hat{q}_u = C \sum_{i=1}^{n} k(\|x_i^*\|^2)\, \delta[b(x_i^*) - u] \qquad (2.2)$$

Où δ est la fonction de Kronecker. La constante de normalisation *C* est dérivée en imposant la condition $\sum_{u=1}^{m} \hat{q}_u = 1$. Le constante *C* est donné par :

$$C = \frac{1}{\sum_{i=1}^{n} k\left(\left\|x_i^*\right\|^2\right)} \qquad (2.3)$$

2.4.1.2 Candidat cible

Typiquement, le modèle cible est formé à partir de la région cible dans la première trame d'une séquence vidéo. Le modèle cible est comparé aux régions candidates dans la trame courante pour déterminer la localisation de la cible dans la trame courante. Un candidat cible $\hat{p}(y) = \{\hat{p}(y)\}_{u=1\ldots m}$ est défini par un histogramme de la distribution de pixels d'une région dans la trame courante. On note $\{x_i\}_{i=1\ldots n_h}$ l'ensemble des coordonnées des n_h pixels du candidat région centré sur une position y dans la trame courante. En utilisant le même profil du noyau $k(x)$, mais avec un rayon h. La probabilité de la couleur $u = 1 \ldots m$ dans le candidat cible est donnée par :

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \qquad (2.4)$$

Où C_h une constante de normalisation, tel que $\sum_{u=1}^{m} \hat{p}_u(y) = 1$. Cette constante est donnée par :

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)} \qquad (2.5)$$

La figure 2.4 présente un exemple d'histogrammes de couleurs du modèle cible et du candidat en utilisant la composante S de l'image HSV.

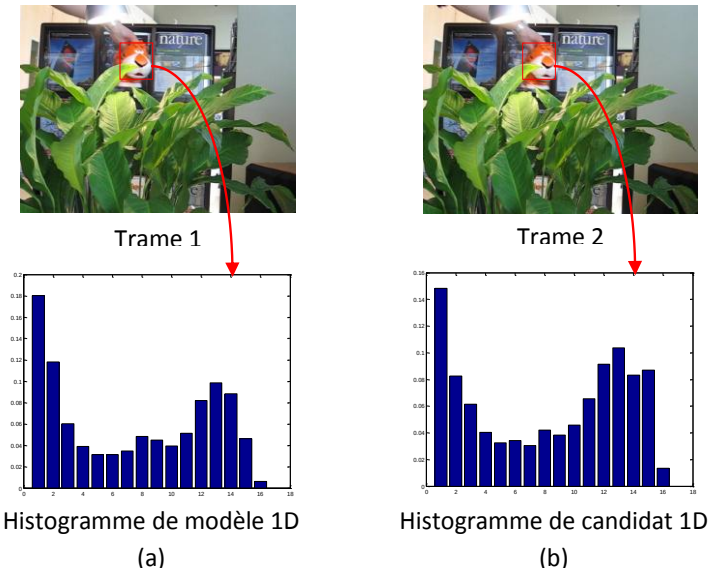


Figure 2.4 – Exemple d'histogramme de composante S : (a) modèle cible, (b) candidat cible.

- **Calcul de l'histogramme pondéré**

L'approche Mean-Shift consiste à calculer l'histogramme pondéré de couleurs pour une région qui englobe l'objet cible, certains cas englobe l'objet cible et des éléments du fond. L'utilisation de l'histogramme pondéré permet de limiter l'influence du fond et privilégier l'information pertinente. Pour la génération de l'histogramme, on utilise une pondération définie par un noyau spatial qui donne un poids plus important aux pixels plus proches du centre de l'objet, tandis que un poids plus faible aux pixels éloignés du centre de l'objet (pixels du fond). Pratiquement, lors du calcul de l'histogramme, nous pondérons chaque pixel de la région d'intérêt par la valeur du noyau spatial associé au même pixel, comme illustré dans la figure 2.5.

L'histogramme pour la couleur *u* est défini par :

$$q_u(y) = C \sum_{i=1}^{n} k(\|x_i\|^2)\, \delta[b(x_i) - u] \qquad (2.6)$$

Où k est un noyau spatial associant un poids à chaque position spatiale x_i.

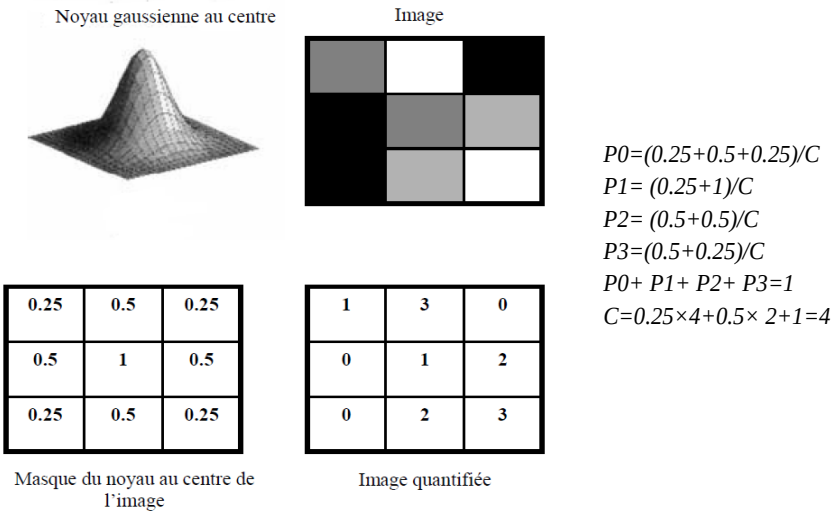


Figure 2.5 – Exemple de construction d'un histogramme pondéré par un Noyau gaussien d'une image en niveau de gris et l'intensité est quantifiée en 4 niveaux [173].

- **Fonction Noyau**

Un noyau défini sur un espace ε est une fonction à valeurs réelles positives qui vérifie les propriétés suivantes :

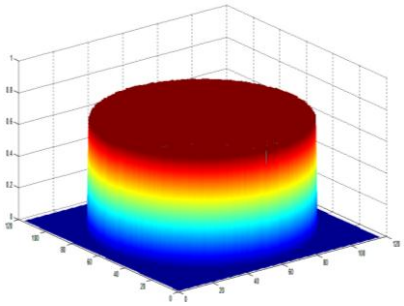
$$\int_{\varepsilon} K(x)dx = 1 \qquad (2.7)$$

$$\forall x \in \varepsilon, \qquad K(-x) = K(x) \qquad (2.8)$$

Le noyau donne une plus grande importance aux pixels du centre de l'objet, qui sont moins susceptibles d'être occultés. Mis à part le noyau uniforme qui donne un poids uniforme à tous les pixels. Ceci suggère que, les pixels aux bords du patch ont plus tendance à être affectés par des problèmes d'occultations où d'interférence avec le fond. De ce fait, le noyau considéré doit obligatoirement être "convexe" et "monotone" et d'autant plus "différentiable", afin de pouvoir calculer son gradient. Il existe une multitude de noyaux spatials utilisés. La Figure 2.6 présente des exemples de fonctions noyaux les plus couramment utilisées. L'algorithme Mean shift utilise le noyau d'Epanechnikov, puisqu'il utilise sa forme dérivée (noyau uniforme).

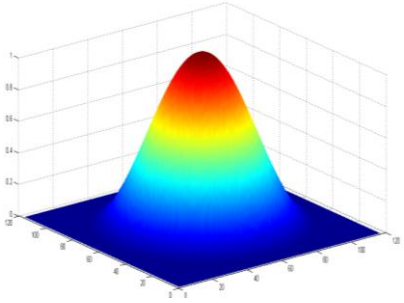
Noyau uniforme :

$$K(x) = \begin{cases} \lambda & si \ \ |x| \leq h \\ 0 & si \ \ |x| > h \end{cases}$$



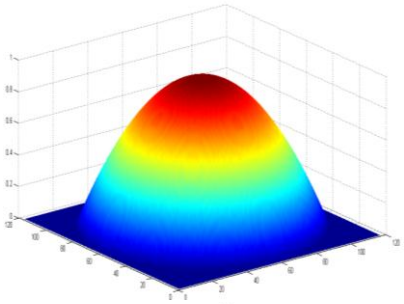
Noyau Gaussien :

$$K(x) = \begin{cases} \lambda \exp\left(-\frac{1}{2}|x|^2\right) & si \ \ |x| \leq h \\ 0 & si \ \ |x| > h \end{cases}$$



Noyau d'Epanechnikov :

$$K(x) = \begin{cases} \lambda(1 - |x|^2) & si \ \ |x| \leq h \\ 0 & si \ \ |x| > h \end{cases}$$



Noyau Triangulaire :

$$K(x) = \begin{cases} \lambda(1 - |x|) & si \ \ |x| \leq h \\ 0 & si \ \ |x| > h \end{cases}$$

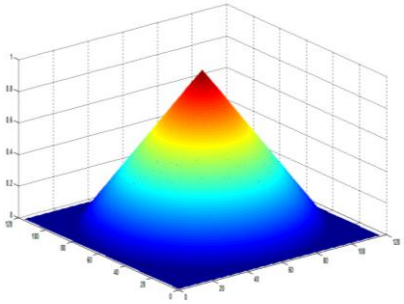


Figure 2.6 – Différents types de fonctions noyaux utilisables.

2.4.2 Mesure de similarité

La similarité entre les deux histogrammes de modèle cible et de candidat cible est mesurée par le coefficient de Bhattacharyya [174]. Ce coefficient représente le recouvrement de deux ensembles d'échantillons. La distance de Bhattacharyya est connue dans le monde des statistiques comme étant une mesure de similarité entre deux distributions statistiques. Le coefficient de Bhattacharyya est défini par l'équation :

$$\rho(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\, \hat{q}_u} \quad , \quad \forall\ \hat{p}(y), \hat{q},\ \ 0 \leq\ \rho(y) \leq 1 \qquad (2.9)$$

Le coefficient de Bhattacharyya a une interprétation géométrique directe par rapport à deux distributions. Ce coefficient est égal au cosinus de l'angle θ entre deux vecteurs m-dimensionnels représentant les histogrammes $\vec{\boldsymbol{p}} = \left(\sqrt{\hat{p}_1\,(y)}, \ldots, \sqrt{\hat{p}_m(y)}\right)^T$ et $\vec{\boldsymbol{q}} = (\sqrt{\hat{q}_1}, \ldots, \sqrt{\hat{q}_m})^T$, $(\|\boldsymbol{p}\|_2 = \|\boldsymbol{q}\|_2 = 1)$ (voire la figure 2.7).

Le coefficient de Bhattacharyya est donné par :

$$\rho(y) = \cos\theta = \frac{\boldsymbol{p}\, \boldsymbol{q}^T}{\|\boldsymbol{p}\|_2\, \|\boldsymbol{q}\|_2} = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \qquad (2.10)$$

La distance de Bhattacharyya peut s'évaluer en utilisant l'équation suivante :

$$d_{Bha}(y) = \sqrt{1 - \rho(y)} \qquad (2.11)$$

Cette distance a été utilisée par plusieurs travaux pour mesurer la similarité entre deux histogrammes dans le contexte du suivi d'objet. Localiser la région souhaitée revient à dire minimiser la distance de Bhattacharyya d_{Bha}, ou approximer le coefficient Bhattacharyya ρ à 1.

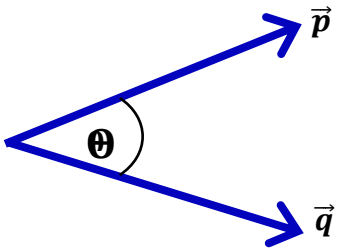


Figure 2.7 – Représentation de coefficient de Bhattacharyya.

Les itérations de Mean shift peuvent être utilisées pour maximiser le coefficient de Bhattacharyya comme fonction de *y* dans le voisinage d'une position donnée (figure 2.8).

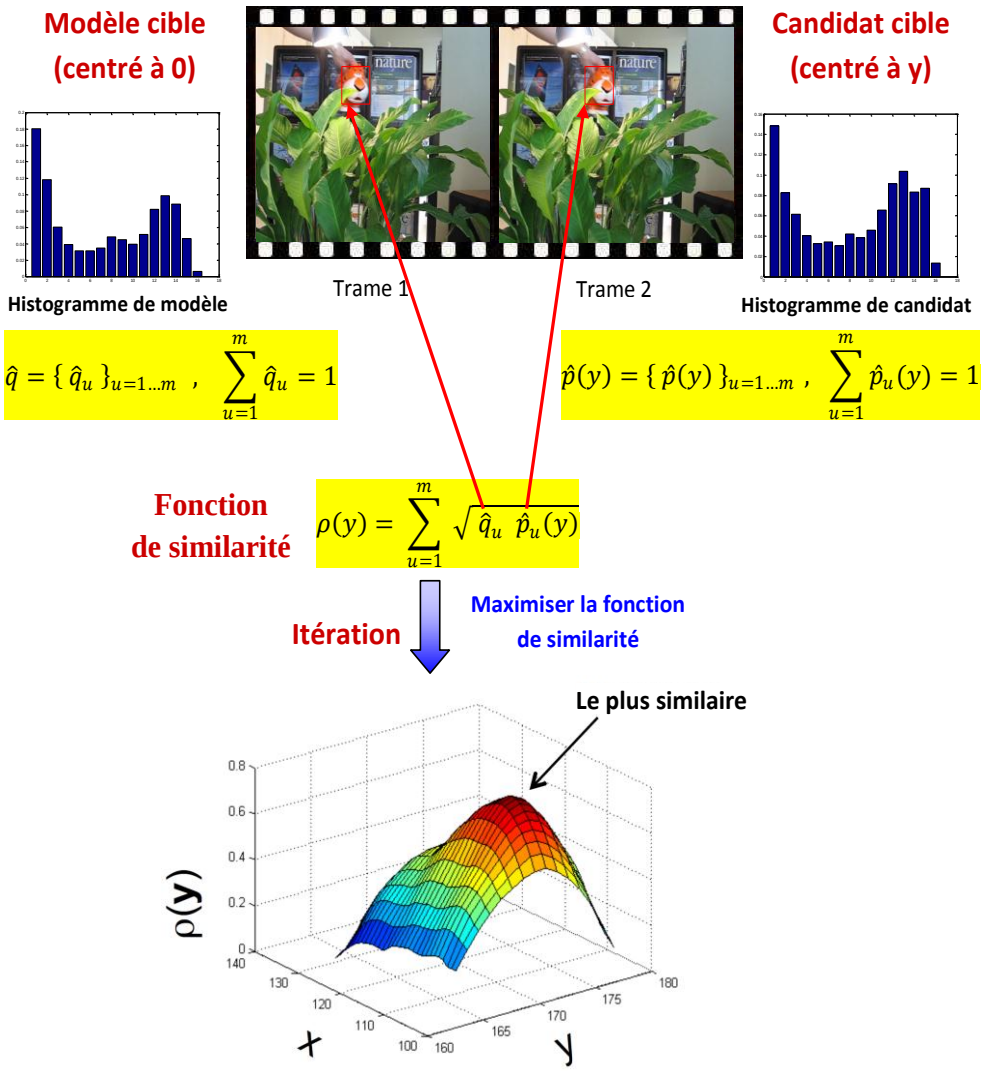


Figure 2.8 – Maximiser le coefficient de Bhattacharyya (la fonction de similarité).

2.4.3 Localisation de la cible

La recherche de l'objet dans la trame courante consiste à trouver la position y_1 qui maximise le coefficient de Bhattacharyya $\rho(y)$ (Eq. (2.9)), ce qui est équivalent à minimiser la distance dans l'équation (2.11). Le processus itératif d'optimisation est initialisé avec la localisation de la cible y_0 dans la trame précédente. En utilisant l'expansion de Taylor autour de $\hat{p}_u(y_0)$, l'approximation linéaire du coefficient de Bhattacharyya est obtenue comme suit :

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m} \sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m} \hat{p}_u(y) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \qquad (2.12)$$

En substituant l'équation (2.4) dans l'équation ci-dessus, le coefficient de Bhattacharyya s'exprime comme suit:

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \tag{2.13}$$

Où :

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}}\,\delta[b(x_i) - u] \tag{2.14}$$

Dans l'équation (2.13), le premier terme est indépendant de y. Afin de maximiser le coefficient de Bhattacharyya, le deuxième terme de cette équation doit obtenir le maximum. Le deuxième terme représente l'estimation de la densité du noyau dans *y*, et le poids de chaque pixel est calculé par w_i (voir figure 2.9). La densité du noyau se déplace vers un maximum local par le vecteur Mean shift et converge vers la position réelle de la cible par plusieurs itérations [175]. La position cible optimale y_1 dans la trame courante est représentée par l'équation suivante :

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \tag{2.15}$$

Où $g(x) = -k'(x) = 1$.

Lorsque nous choisissons le noyau g avec le profil Epanechnikov, Eq. (2.15) est réduite à :

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \tag{2.16}$$

En utilisant Eq. (2.16), l'algorithme de suivi par Mean shift trouve dans la nouvelle trame la région la plus similaire à celle de l'objet.

Image modèle Image candidat Image poids

Figure 2.9 – Exemple représenté l'image de poids. L'image de poids représente des valeurs élevées au niveau des régions ayant une couleur proche de la couleur de l'objet suivi.

La maximisation peut être effectuée efficacement en utilisant les itérations Mean Shift, à l'aide de l'algorithme 2.1, qui décrit les étapes de base :

Algorithme 2.1. Suivi d'objet par le tracker Mean shift

Soit $\{\hat{q}_u\}_{u=1\ldots m}$ l'histogramme du modèle cible, et y_0 le centre de l'objet cible dans la trame précédente.

1. Initialiser la position y_0 dans la trame courante, calculer la distribution de probabilité de $\{\hat{p}_u(y_0)\}_{u=1\ldots m}$ en utilisant (Eq. (2.4)). Et évaluer $\rho[\hat{p}(y_0), \hat{q}]$ en utilisant (Eq. (2.9)).
2. Calculer les poids $\{w_i\}_{i=1\ldots n_h}$ pour chaque pixel selon l'équation (Eq. (2.14)).
3. Calculer la nouvelle position du candidat cible selon l'équation (Eq. (2.15)).
4. Mettre à jour $\{\hat{p}_u(y_1)\}_{u=1\ldots m}$ en utilisant (Eq. (2.4)), puis évaluer $\rho[\hat{p}(y_0), \hat{q}]$ en utilisant (Eq. (2.9)).
5. Tant que $\rho[\hat{p}(y_1), \hat{q}] < \rho[\hat{p}(y_0), \hat{q}]$
   Faire $y_1 = \frac{1}{2}(y_0 + y_1)$.
   Evaluer $\rho[\hat{p}(y_1), \hat{q}]$ en utilisant (Eq. (2.9)).
6. Si $\|y_1 - y_0\| < \varepsilon$ Stop.
   Sinon $y_0 \leftarrow y_1$, et retourner à l'étape 2.

2.5 Suivi d'objet par Camshift

Camshift (Continuously Adaptive Mean shift) a été introduit pour la première fois comme une technique de suivi du visage et de la tête dans une interface utilisateur perceptive en tant qu'extension Mean shift [176]. Il est basé sur une adaptation de Mean Shift. Camshift peut ajuster de manière adaptative la taille de la zone d'objet cible de sorte qu'il ait une bonne adaptabilité à la variation de l'objet cible [177].

La différence principale entre les algorithmes Camshift et Mean Shift est que Camshift utilise des distributions de probabilité adaptative en continu, ce qui signifie que la distribution de probabilité de l'objet cible peut être recalculée dans chaque image. Cela permet de modifier la taille, la forme et l'apparence de l'objet cible dans chaque image, tandis que Mean Shift est basé sur des distributions statiques qui ne sont pas mises à jour à moins que l'objet cible ne subisse des changements significatifs en forme, en taille ou en couleur [118][176].

2.5.1 Procédure de suivi Camshift

Pour suivre les objets cibles, Camshift traditionnel [117] fonctionne essentiellement comme suit : D'abord, la fenêtre de recherche initiale de l'objet cible est sélectionnée et son histogramme couleur est calculé. Chaque image de la séquence est ensuite convertie en une

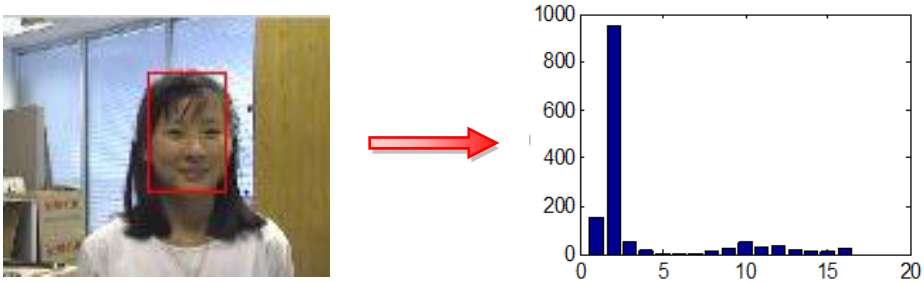


Figure 2.10 – Exemple sur l'histogramme de l'objet cible en utilisant la composante H de l'espace de couleur HSV.

image de distribution de probabilité par rapport à l'histogramme de l'objet cible. Ensuite, la localisation et la nouvelle taille de l'objet cible sont calculées via Mean shift à partir de cette image convertie et ils sont utilisées comme taille initiale et la localisation de la cible pour les itérations suivantes de l'algorithme. Dans la partie suivante, nous expliquons les étapes de la méthode de suivi d'objets Camshift et nous discutons trois variantes de cette méthode.

2.5.1.1 Initialisation de la fenêtre de recherche

Après avoir localisé l'objet cible par un rectangle environnant, son histogramme de couleur est calculé pour un traitement ultérieur dans les prochaines étapes.

2.5.1.2 Génération d'histogramme de couleur

Nous présentons dans cette section trois variantes de l'algorithme de suivi Camshift. L'algorithme Camshift traditionnel présenté dans [117] et les deux variantes qui ont proposé par Allan et al. dans [118]. Ces variantes ont améliore l'algorithme Camshift traditionnel en utilisant les meilleurs modèles cibles.

a. Camshift Original

L'algorithme original de Camshift [117] utilise l'histogramme unidimensionnel 1D comme modèle de l'objet cible. L'histogramme est quantifié en 16 bins qui regroupent des valeurs similaires et améliore ainsi les performances. Bradski prend le canal H de l'espace couleur HSV (teinte, saturation, luminosité) pour décrire l'objet cible par une gamme de teinte de couleur (voir figure 2.10). Selon l'occurrence d'une teinte dans l'histogramme, La valeur de probabilitée se situe dans [0,1].

b. Histogramme pondéré

Lorsque la région sélectionnée initiale contient des pixels de l'extérieur de l'objet (pixels de fond), notre image de distribution de probabilité 2D sera influencée par leur fréquence dans la rétroprojection de l'histogramme. Pour limiter l'influence du fond et privilégier l'information

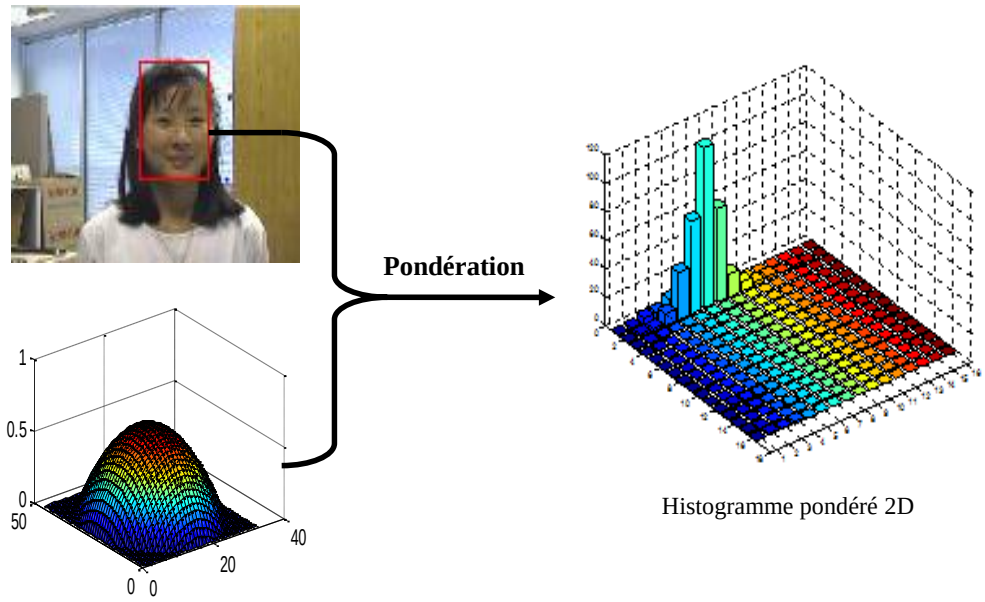


Figure 2.11 – Exemple sur l'histogramme pondéré de l'objet cible en utilisant les composantes H et S de l'espace de couleur HSV.

pertinente, Allen et al. ont utilisé l'histogramme pondéré pour calculer l'histogramme de l'objet cible, afin d'attribuer une pondération plus élevée aux pixels plus proches du centre de la région (plus les pixels sont loin du centre de l'objet, plus le poids pris en compte dans l'histogramme final est faible.). L'histogramme de modèle cible pour un bin *u* est donné par l'équation (2.2). Le profil du noyau utilisé pour générer l'histogramme pondéré est le noyau Epanechnikov. Ce noyau est une fonction de pondération utilisée dans les techniques d'estimation non-paramétrique. La figure 2.11 représente l'histogramme pondéré 2D en utilisant les deux composants H et S de l'espace couleur HSV.

c. Histogramme de Ratio

Allen et al. [118] ne sont pas satisfaits pour l'amélioration de Camshift qui utilise l'histogramme pondéré. Ils ont introduit également un histogramme de ratio (modèle du fond) pour répresenter l'objet cible. L'histogramme pondéré n'est pas suffisant pour localiser l'objet cible lorsque la rétroprojection de l'histogramme est utilisée pour générer l'image de distribution de probabilité 2D, parce que si l'histogramme de l'objet cible contient un nombre significatif de caractéristiques appartenant à l'image du fond ou aux objets voisins, la localisation et l'échelle de la cible ne peuvent pas être déterminées avec précision.

Un histogramme de ratio peut aider à résoudre le problème du fond en attribuant des caractéristiques de couleur qui appartiennent au fond avec des poids inférieurs. Dans notre

travail, nous calculons un histogramme pour une région en dehors de la localisation cible normalisé en utilisant un noyau avec le profil suivant :

$$k(r) = \begin{cases} ar & 1 < r \leq h \\ 0 & ailleurs \end{cases} \tag{2.17}$$

Où a est un facteur de mise à l'échelle et est le rayon de la nouvelle fenêtre de recherche. Une région de fond deux fois plus grande que la région cible a été utilisée dans notre travail. Un histogramme $\{\hat{o}\}_{u=1\ldots m}$ est calculé en utilisant l'équation (2.2) avec un rayon h puis pondéré en utilisant l'équation (2.10) où $\hat{o}^*$ est la plus petite entrée non nulle

$$\left\{\hat{w}_u = min\left(\frac{\hat{O}^*}{\hat{O}_u}, 1\right)\right\}_{u=1\ldots m} \tag{2.18}$$

L'histogramme de fond pondéré est donné comme suit :

$$\hat{q}_u = \hat{w}_u \sum_{i=1}^{n} k(\|x_i^*\|^2)\, \delta[c(x_i^*) - u] \tag{2.19}$$

La figure 2.12 représente l'histogramme de ratio en utilisant les composantes H et S de l'espace de couleur HSV.

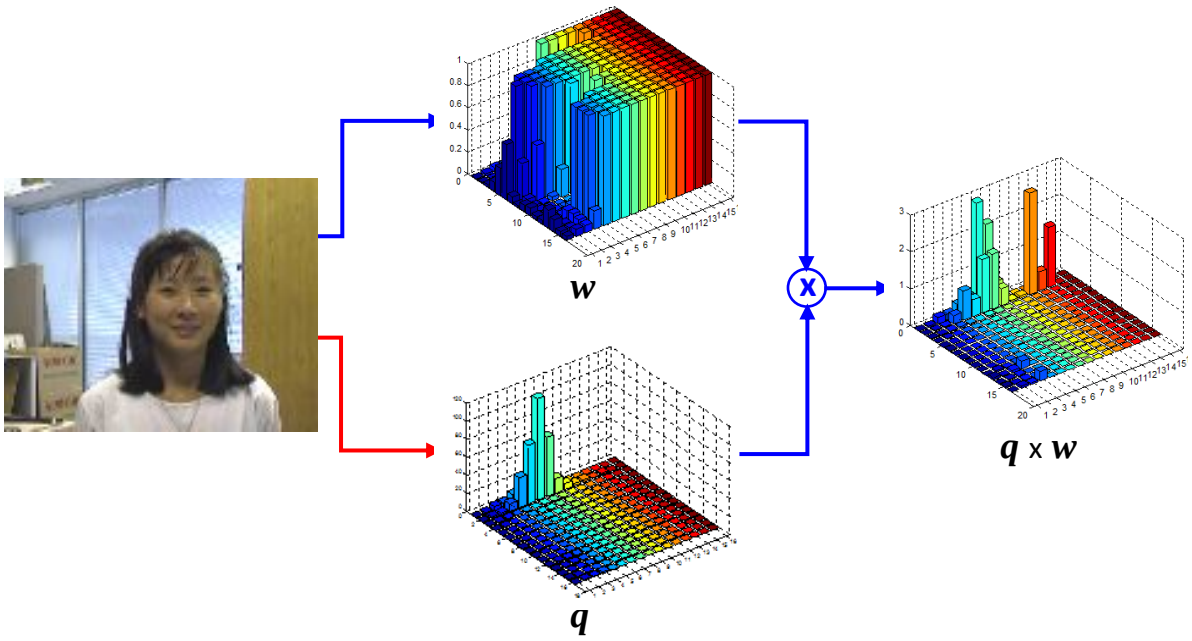


Figure 2.12 – Exemple sur l'histogramme de ratio des composants H et S : (w) l'histogramme pondéré du fond, (q) l'histogramme pondéré de l'objet cible, (q x w) l'histogramme de ratio.

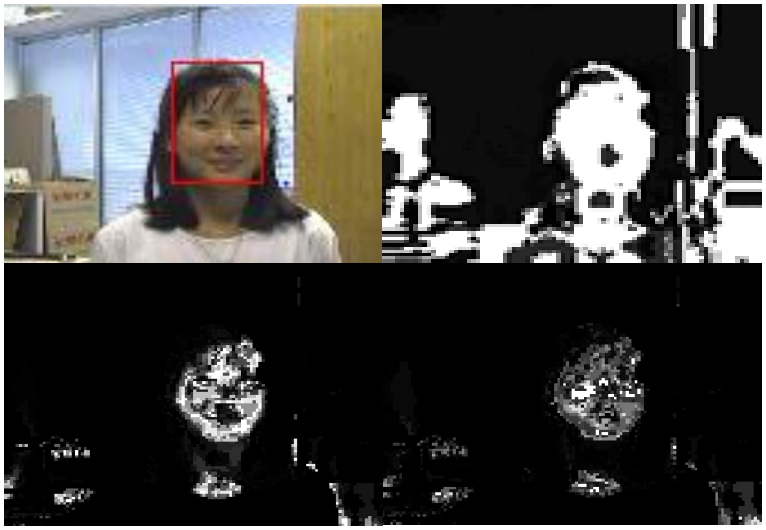


Figure 2.13 – Exemples sur l'image projection (l'image de la distribution de probabilité). L'image originale dans en haut à gauche. L'image projection 1-D (H) dans en haut à droite. L'image projection 3-D (HSV) dans en bas à gauche. L'image projection 3-D (HSV avec le fond) dans en bas à droite.

2.5.1.3 Rétroprojection de l'histogramme

La Rétroprojection de l'histogramme est une opération primitive qui associe les valeurs de pixel dans l'image à la valeur du bin d'histogramme correspondant. La Rétroprojection de l'histogramme de modèle d'objet cible avec n'importe quelle trame consécutive génère une image de probabilité où la valeur de chaque pixel caractérise la probabilité que le pixel d'entrée appartient à l'histogramme utilisé [118][176][178].

Dans tous les cas, les valeurs de bin d'histogramme sont mises à l'échelle pour se situer dans la plage de pixels discrets de l'image de distribution de probabilité 2D en utilisant l'équation (2.20).

$$I(x,y) = \left\lfloor \frac{\hat{q}_u}{max\{\hat{q}_u \mid j = 1,2,\dots,n\}} \times 255 \right\rfloor \qquad (2.20)$$

Où le symbole ⌊ ⌋ représente l'opération d'arrondi. Plus la valeur de gris du pixel est grande dans l'image de distribution de probabilité, plus grande la probabilité que le pixel appartienne à la zone cible. La figure 2.13 illustre les images de distribution de probabilité pour tous les histogrammes qui ont été étudiés.

2.5.1.4 Calcul de la taille de la fenêtre de recherche

a. Calcul du centre de masse

La localisation de centre dans la fenêtre de recherche de l'image de distribution de probabilité discrète est trouvée en utilisant de moments. Étant donné que $I(x,y)$ est l'intensité de l'image de probabilité discrète à (x,y) dans la fenêtre de recherche [118].

a) Calculer le moment zéro

$$M_{00} = \sum_x \sum_y \mathrm{I(x,y)} \tag{2.21}$$

b) Trouver le premier moment pour x et

$$\left.\begin{aligned} M_{10} &= \sum_x \sum_y \mathrm{x\ I(x,y)} \\ M_{01} &= \sum_x \sum_y \mathrm{y\ I(x,y)} \end{aligned}\right\} \tag{2.22}$$

c) Calculer la localisation de centre de la fenêtre de recherche

$$x_c = \frac{M_{10}}{M_{00}} \ ; \qquad y_c = \frac{M_{01}}{M_{00}} \tag{2.23}$$

b. Calcul de l'orientation et de l'échelle

L'utilisation de moments pour déterminer l'échelle et l'orientation d'une distribution en vision robotique et informatique est décrite dans [179] et a été utilisée pour la vision dans des jeux informatiques [180] et pour l'orientation et le suivi des têtes et des visages [117].

On détermine l'orientation (θ)du grand axe et l'échelle de la distribution en trouvant un rectangle équivalent ayant les mêmes moments que ceux mesurés à partir de l'image de distribution de probabilité 2D. Les premier et deuxième moments pour $x\ et\ y$ sont donnés par :

$$\left.\begin{aligned} M_{20} &= \sum_x \sum_y x^2 I(x,y) \\ M_{02} &= \sum_x \sum_y y^2\ I(x,y) \end{aligned}\right\} \tag{2.24}$$

$$M_{11} = \sum_x \sum_y xy\ \ I(x,y) \tag{2.25}$$

Les deux premières valeurs propres (la longueur et la largeur de la distribution de probabilité) sont calculées sous forme fermée comme suit. A partir des variables intermédiaires $a,\ b, et\ c$.

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \tag{2.26}$$

$$b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c\right) \tag{2.27}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \tag{2.28}$$

On trouve l'orientation du rectangle équivalent par :

$$\theta = \frac{1}{2}\, tan^{-1}\left(\frac{b}{a-c}\right) \tag{2.29}$$

Les distances L_1 et L_2 du centroïde de distribution (les dimensions du rectangle équivalent) sont données par :

$$L_1 = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \tag{2.30}$$

$$L_2 = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{2.31}$$

Où les paramètres extraits sont indépendants de l'intensité globale de l'image.

2.5.2 Algorithme de Camshift

L'algorithme Camshift peut être résumé dans les étapes suivantes [117][118].

Algorithme 2.2. Suivi par Camshift

1. Initialiser la localisation initiale et la taille de la fenêtre de recherche Mean shift, la localisation sélectionnée est la distribution cible à suivre.
2. Calculer la distribution de probabilité de couleur de la région centrée à la fenêtre de recherche Mean shift, dans la trame actuelle (l'image de distribution de probabilité).
3. Calculer la nouvelle localisation de la fenêtre de recherche cible en utilisant Mean shift (effectuer la recherche de la probabilité de densité maximale en utilisant le paramètre Mean shift pour la convergence). Conservez le moment zéro (zone de distribution) et la localisation de centre.
4. Calculer la nouvelle taille de la fenêtre de recherche cible en utilisant les moments 1er et 2ème ordre.
5. Utiliser la nouvelle localisation et la taille obtenue à l'étape 3 et 4 pour réinitialiser la fenêtre de recherche dans la nouvelle trame et retourner à l'étape 2.

2.6 Suivi par Mean shift avec filtre de Kalman (KaMS)

Afin d'augmenter la précision et la robustesse du traker Mean shift, et pour traiter le problème d'occultation partielle ou totale, le filtre de Kalman a été introduit pour l'algorithme Mean shift en plusieurs travaux [113][114][168][181]. L'idée est de prédire la position de l'objet suivi dans la nouvelle trame basée sur le mouvement précédent de l'objet. L'idée principale est de trouver la position de l'objet cible par le tracker Mean shift et il est considérée comme la mesure (observation) de filtre de Kalman. Puis cette position est transmise au filtre de Kalman pour estimer la position actuelle de l'objet cible.

2.6.1 Filtre de Kalman

Le filtre de Kalman est un algorithme d'estimation statistique développé par Rudolf Emil Kalman [182] à la fin des années 50. Il est utilisé pour prédire et corriger l'état courant d'un système dynamique linéaire dont l'état est caractérisé par un vecteur aléatoire, dont on connaît à l'instant t. La prédiction utilise des mesures linéaires perturbées aussi par du bruit additionnel. Dans le temps, les perturbations localisées sur les mesures sont représentées par du bruit gaussien avec une matrice de covariance connue à chaque temps de mesure [183].

Ce filtre est utilisé dans une large gamme de domaines technologiques (radar, vision électronique, communication, suivi d'objets ...). C'est un thème majeur de l'automatique et du traitement du signal. Le filtre de Kalman [182] a été utilisé pour le suivi mono-objet où il permet d'estimer le vecteur d'état x(t) décrivant généralement l'état cinématique de l'objet à suivre tel que sa position et sa vitesse ou d'autres attributs comme ses dimensions, son type ou sa classe, connaissant le vecteur de mesures bruitées (vecteur d'observation) z(t).

2.6.1.1 Principe du filtre de Kalman

Le filtre de Kalman est un filtre prédicteur-correcteur pour la résolution des problèmes numériques (figure 2.14). Ce filtre se compose de deux types d'équations pour estimer la valeur de la variable d'état :

- Les équations de prédiction.
- Les équations de correction (mise à jour).

Les équations de prédiction sont responsables de propager les estimées de l'état présent et les covariances d'erreur pour obtenir les estimées à priori de la prochaine étape. Alors que les équations de correction sont responsables d'introduire une nouvelle mesure avec l'estimée à priori afin d'obtenir une estimée à posteriori améliorée (figure 2.14).

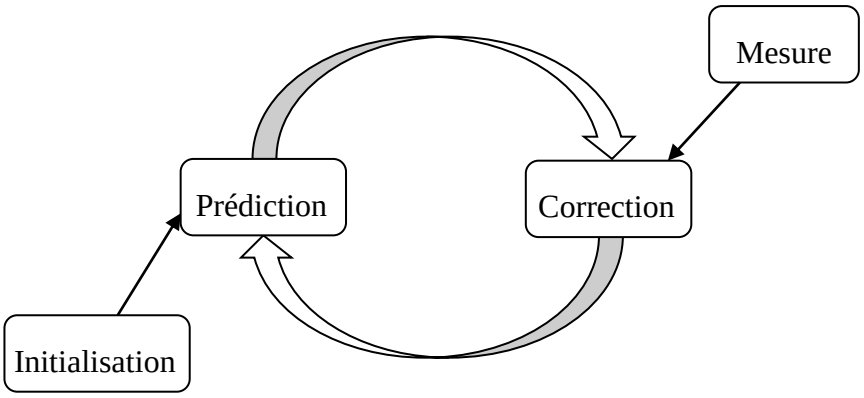


Figure 2.14 – Fonctionnement du filtre de Kalman

2.6.1.2 Calcul de l'estimateur de Kalman

On considère un système dynamique linéaire à temps discret défini par un vecteur d'équations aux différences, entaché d'un bruit blanc gaussien. Plus précisément, il existe un système de temps discret et son état à l'instant n est donné par le vecteur x_n. L'état de la prochaine étape $n+1$ est donné par [113] :

$$x_{n+1} = F_{n+1,n}\, x_n + w_{n+1} \tag{2.32}$$

Où $F_{n+1,n}$ est la matrice de transition d'état caractérisant l'évolution du système. w_{n+1} est un bruit blanc gaussien à moyenne nulle et de matrice de covariance Q_{n+1}.

L'équation de mesure est donnée par :

$$z_{n+1} = H_{n+1}\, x_{n+1} + v_{n+1} \tag{2.33}$$

Où z_{n+1} est le vecteur de mesure. H_{n+1} est la matrice de transformation de l'état vers l'espace des mesures. Le bruit de mesure est modélisé par un vecteur v_{n+1}, sa distribution est aussi une gaussienne de moyenne nulle et de covariance R_{n+1}.

Dans l'équation (2.33), le vecteur de mesure z_{n+1} ne dépend que du vecteur d'état actuel x_{n+1} et le vecteur de bruit v_{n+1} est indépendant du bruit w_{n+1}. Le filtre de Kalman calcule l'erreur d'estimation quadratique moyenne minimale de l'état x_k étant donné les mesures $z_1, \ldots\ldots, z_k$.

2.6.1.3 Prédiction du vecteur d'état et des mesures

Une prédiction de l'état futur des variables que l'on cherche à estimer est tout d'abord réalisée, à partir de l'état précédent. La prédiction de l'état et sa matrice de covariance au temps t_n sont données par le modèle dynamique :

$$\hat{\mathrm{x}}_n^- = \mathrm{F}_{n,n-1}\ \hat{\mathrm{x}}_{n-1} \tag{2.34}$$

$$\mathrm{P}_n^- = \mathrm{F}_{n,n-1}\ \mathrm{P}_{n-1}\,\mathrm{F}_{n,n-1}^{\mathrm{T}} + \mathrm{Q_n} \tag{2.35}$$

Où la matrice de covariance P est de dimensions $R^{D_x \times D_x}$ et le deuxième terme dans l'équation correspond à la covariance du vecteur de perturbation de l'état.

2.6.1.4 Correction de l'état.

Pour la mise à jour du vecteur d'état estimé $\hat{\mathrm{x}}$ et sa covariance P, il est nécessaire d'obtenir le gain du filtre G_n qui dépend directement des matrices de covariance obtenues pour l'état prédit.

$$\mathrm{G}_n = \mathrm{P}_n^- \mathrm{H}_n^{\mathrm{T}}[\mathrm{H}_n\,\mathrm{P}_n^- \mathrm{H}_n^{\mathrm{T}} + \mathrm{R}]^{-1} \tag{2.36}$$

$$\hat{\mathrm{x}}_n = \hat{\mathrm{x}}_n^- + \mathrm{G}_n\,(\mathrm{z}_n - \mathrm{H}_n\,\hat{\mathrm{x}}_n^-) \tag{2.37}$$

$$\mathrm{P}_n = (\mathrm{I} - \mathrm{G}_n\,\mathrm{H}_n)\mathrm{P}_n^- \tag{2.38}$$

Dans cette étape de correction, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise.

Le filtre de Kalman est appliqué en temps réel, parce que ce filtre ne nécessite pas toutes les données passées pour produire une estimation à l'instant courant, c-à-d il ne nécessite donc pas de mise en mémoire et de retraitement des données.

On résume les étapes de suivi par filtre de Kalman dans l'algorithme 2.3 suivant :

Algorithme 2.3. Filtre de Kalman

1. Initialisation : $\hat{\mathrm{x}}_0$ la position initiale de l'objet.

$$\hat{x}_0 = E[x_0]$$
$$P_0 = [(x_0 - E[x_0])(x_0 - E[x_0])^T]$$

2. Prédiction :

$$\hat{\mathrm{x}}_n^- = \mathrm{F}_{n,n-1}\ \hat{\mathrm{x}}_{n-1}$$
$$\mathrm{P}_n^- = \mathrm{F}_{n,n-1}\ \mathrm{P}_{n-1}\,\mathrm{F}_{n,n-1}^{\mathrm{T}} + \mathrm{Q_n}$$

3. Correction :

$$\mathrm{G}_n = \mathrm{P}_n^- \mathrm{H}_n^{\mathrm{T}}[\mathrm{H}_n\,\mathrm{P}_n^- \mathrm{H}_n^{\mathrm{T}} + \mathrm{R}]^{-1}$$
$$\hat{\mathrm{x}}_n = \hat{\mathrm{x}}_n^- + \mathrm{G}_n\,(\mathrm{z}_n - \mathrm{H}_n\,\hat{\mathrm{x}}_n^-)$$
$$\mathrm{P}_n = (\mathrm{I} - \mathrm{G}_n\,\mathrm{H}_n)\mathrm{P}_n^-$$

Retourner à l'étape de prédiction pour l'itération suivante.

2.6.2 Algorithme de la combinaison entre Mean shift et filtre de Kalman

Il existe de nombreuses approches pour définir les composants dans le filtre de Kalman. Par exemple, on peut définir l'état comme la position, et sa vitesse (x, y, v_x, v_y), ou on peut ajouter l'accélération dans l'état également ($x,\ y,\ v_x,\ v_y,,\ a_x,\ a_y$). Dans ce travail, Nous définissons le vecteur d'état x par la position uniquement (x, y), comme a défini par V.Karavasilis et al dans [113].

On suppose que l'objet est décrit par ses coordonnées du centre (x, y), les axes de rectangle sont (h_x, h_y), et que la taille du rectangle ne change pas dans le temps. Le vecteur d'état $\mathrm{x_n} = [x_n,\ y_n\ 1]^T$ représente la position réelle du centre de l'image en coordonnées homogènes ($x_n\ et\ y_n$ sont les coordonnées horizontale et verticale, respectivement), et sa position varie dans le temps par l'équation (2.32). La matrice de transition $\mathrm{F}_{n+1,n}$ est définie comme suit :

$$\mathrm{F}_{n+1,n} = \begin{bmatrix} 1 & 0 & \mathrm{d}x_{n+1,n} \\ 0 & 1 & dy_{n+1,n} \\ 0 & 0 & 1 \end{bmatrix}$$

Où $dx_{n+1,n} dy_{n+1,n}$ sont les translations horizontale et verticale du centre de l'objet. Les paramètres $dx_{n+1,n}\ dy_{n+1,n}$ ne sont pas constants dans le temps mais ils sont calculés dynamiquement comme il sera expliqué dans ce qui suit. Le vecteur de bruit $\mathrm{w}_{n+1} = [w_{n+1_x}, w_{n+1_y}, 1]^T$, et sa matrice de covariance Q **:**

$$\mathrm{Q} = \begin{bmatrix} \sigma_{Q_x} & 0 & 0 \\ 0 & \sigma_{Q_y} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Où $\sigma_{Q_x} = h_x$ et $\sigma_{Q_y} = h_y$.

On utilise le tracker Mean shift pour obtenir le vecteur de mesure $z_{n+1} = [x'_{n+1}, y'_{n+1}]^T$. Où x'_{n+1} et y'_{n+1} sont les coordonnées horizontales et verticales du centre du rectangle. En général, ces mesures diffèrent des variables d'état $x_{n+1}\ et\ \mathrm{y}_{n+1}$ du vecteur x_{n+1} en raison de la présence de bruit v_{n+1}. La relation entre la mesure z_{n+1} et l'état x_{n+1} est donné par Eq. (2.33), où :

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

Et le bruit de mesure $\mathrm{v}_{n+1} = [v_{n+1_x}, v_{n+1_y}, 1]^T$, et sa matrice de covariance R.

$$\mathrm{R} = \begin{bmatrix} \sigma_{R_x} & 0 \\ 0 & \sigma_{R_y} \end{bmatrix}$$

Où $\sigma_{R_x} = h_x$ et $\sigma_{R_y} = h_y$.

Le seul problème qui reste à résoudre est l'évaluation automatique des $dx_{n+1,n}dy_{n+1,n}$. En utilisant l'algorithme de Mean shift, on obtient:

- la mesure z_{n+1},
- La distance entre les histogrammes du modèle cible et du candidat cible.

L'idée principale est d'utiliser la distance calculée pour déterminer si l'objet a été détecté ou non. Ceci permet d'obtenir une mesure de la qualité de l'estimation actuelle de l'objet. Si la distance est petite alors le centre de l'objet est proche du centre prédit. Si cette distance est grande, alors la cible est perdue. Cette distance est donnée par:

$$a(y) = f\big(d(y)\big) \tag{2.39}$$

Où $d(y)$ la distance de Bhattacharyya est donnée par Eq. (2.9), f est une fonction décroissante. En se fondant sur la valeur de $a(y)$, le paramètre $d_{n+1,n} = [dx_{n+1,n}\,, dy_{n+1,\mathrm{n}}]^T$ est automatiquement mis à jour par:

$$d_{n+2,n+1} = (1-a)d_{n+1,n} + a(\hat{\mathrm{x}}_{n+1} - \hat{\mathrm{x}}_n) \tag{2.40}$$

Où $\hat{\mathrm{x}}_{n+1}$ est le vecteur contenant les valeurs estimées des coordonnées horizontales et verticales du centre de rectangle à l'instant $n+1$. Dans l'équation (2.40), l'estimation $\hat{\mathrm{x}}_{n+1}$ contribue aux mises à jour du déplacement $d_{n+2,n+1}$ quand l'estimation actuelle ressemble au modèle d'objet source, qui est alors $a(y) \to 1$, d'une part. D'autre part $(a(y) \to 0)$, les déplacements inclus dans la matrice d'état $\mathrm{F}_{n+2,n+1}$ restent inchangés, comme ils l'étaient à l'étape $n+1$, en considérant que l'objet est occulté.

On résume les étapes de suivi par Mean shift avec filtre de Kalman dans l'algorithme suivant :

Algorithme 2.4. Suivi par Mean shift avec filtre Kalman

1. Initialisation : $\hat{x}_0$ la position initiale de l'objet

$$P=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \quad R=\begin{bmatrix}hx & 0 & 0\\ 0 & hy & 0\\ 0 & 0 & 0\end{bmatrix} \quad Q=\begin{bmatrix}hx & 0 & 0\\ 0 & hy & 0\\ 0 & 0 & 0\end{bmatrix} \quad F_{1,0} = I_{3\times 3}$$

2. Calcul l'histogramme initial q dans la première trame en utilisant Eq. (2.2).
3. Etape de prédiction :

$$\hat{x}_n^- = F_{n,n-1}\ \hat{x}_{n-1}$$
$$P_n^- = F_{n,n-1}\ P_{n-1}\ F_{n,n-1}^T + Q_n$$

4. Etape de mesure : Calculer le nouvel centre (z_n), *p(y)* et la distance entre q et p en utilisant l'algorithme Mean shift
5. Etape d'estimation :

$$G_n = P_n^- H_n^T [H_n P_n^- H_n^T + R]^{-1}$$
$$\hat{x}_n = \hat{x}_n^- + G_n (z_n - H_n \hat{x}_n^-)$$
$$P_n = (I - G_n H_n) P_n^-$$

   $\hat{x}_n$ est la nouvelle position de l'objet

6. Mettre à jour les éléments de F_n en utilisant Eq. (2.40).
7. Retourner à l'étape de prédiction pour l'itération suivante.

2.7 Conclusion

Nous avons présenté dans ce chapitre, un état de l'art sur l'algorithme de suivi d'objets Mean shift, dans la première partie. Cet algorithme est plus efficace, particulièrement en temps réel, à cause de sa simplicité et de sa robustesse. Alors que, dans la seconde partie nous avons discuté le principe et les différentes étapes de tracker Mean shift, qui consiste à calculer l'histogramme pondéré de couleur pour une région qui englobe l'objet cible. Le principe de ce tracker est basé sur la maximisation de la similarité d'apparence itérativement. L'avantage d'un système de suivi par Mean shift est d'optimiser la phase de recherche. Cependant, cet algorithme est peu robuste en particulier à l'occultation totale, au changement d'échelle et lorsque la couleur de l'objet est similaire à la couleur de fond. Enfin, nous avons présenté l'algorithme de suivi Camshift (Continuously Adaptive Mean Shift) qui se base sur le même principe que l'algorithme Mean shift avec adaptation de la fenêtre de recherche pour surmonter le problème de changement d'apparence de l'objet (échelle). Ainsi que, nous avons présenté l'algorithme Mean shift combiné avec le filtre de Kalman a pour l'objectif de résoudre le problème d'occultation.

Chapitre 3

Etude de l'influence de l'espace couleur sur la performance du tracker Mean shift

Sommaire

3.1 Introduction

La couleur est l'une des caractéristiques d'image les plus largement utilisées dans de nombreux domaines de la vision par ordinateur, tels que le suivi d'objets, à cause de son efficacité et de son efficience. Bien que, l'information de couleur fournit des indices discriminatifs riches pour l'inférence visuelle, dans le suivi d'objets la plupart des trackers modernes reposent uniquement sur la version en niveaux de gris d'une séquence d'entrée, en négligeant les riches informations chromatiques. Malgré les efforts récents pour intégrer la couleur dans le suivi, il y a un manque de compréhension globale du rôle que peut jouer l'information couleur.

Parmi les trackers les plus utilisés de l'information couleur, les plus robustes ainsi que peuvent suivre en temps réel est le tracker Mean shift qui se base sur l'histogramme de couleurs pour représenter l'apparence de l'objet. L'algorithme Mean shift qui était proposé à

l'origine [1] utilise l'histogramme de couleurs 3D dans l'espace de couleurs RGB pour décrire le modèle cible et la région cible, ce qui rend le tracker est moins robuste aux occultations, changements d'illumination et en particulier lorsque la couleur de l'objet est similaire à la couleur de fond.

Par conséquent, dans ce chapitre, nous nous intéressons à l'étude et à l'analyse des différents effets de l'utilisation de différentes configurations d'espace de couleurs, sur l'efficacité et la qualité du suivi en utilisant le tracker Mean shift. Ainsi, nous essayons de comprendre l'influence de l'information couleur de chaque espace sur la performance de ce tracker.

3.2 Construction du modèle d'apparence

En général, le fonctionnement d'un tracker se décompose en quatre principales étapes : Initialisation de l'objet, modélisation de l'apparence de l'objet, prédiction de la localisation d'objet et mise à jour du modèle (figure 3.1). L'étape de modélisation de l'apparence de l'objet est une étape cruciale pour le système de suivi d'objets. Elle affecte d'une manière directe la performance d'un système de suivi d'objets. La modélisation est le processus d'extraction des caractéristiques discriminantes qui permette de décrire et de distinguer un objet d'intérêt dans la séquence vidéo.

L'apparence de l'objet est la principale information exploitée par les trackers, sa modélisation est donc une étape importante pour réussir à suivre l'objet correctement. La construction du modèle d'apparence consiste à associer à chaque objet cible des descripteurs qui permettent de caractériser son modèle d'apparence afin de le comparer avec d'autres objets candidats dans les trames suivantes. Les modèles d'apparence d'objets permettent de décrire le contenu des objets en termes de primitives basées sur l'intensité, couleur, texture, mouvement etc. La plupart des modèles se basent sur la distribution spatiale de la caractéristique visuelle

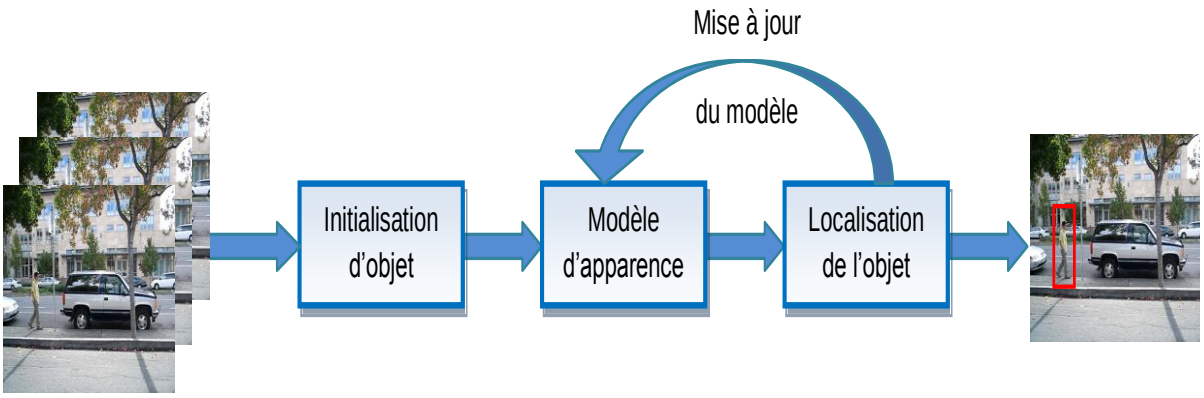


Figure 3.1 – Schéma générique de fonctionnement d'un tracker

considérée pour modéliser l'apparence. Il existe une diversité des modèles d'apparence d'objets tels que : template, Histogramme (probabiliste), Matrice de co-occurence (texture), Sous-espace de représentation... etc.

L'algorithme Mean shift est basé sur la distribution globale des caractéristiques des couleurs de l'objet cible. Il représente l'apparence de l'objet par un histogramme de couleurs 3D dans l'espace de couleurs RGB, et le nombre de classes (bins) utilisé pour chaque histogramme est 16 classes.

L'histogramme de couleur est un histogramme usuel souvent utilisé pour les applications de vision par ordinateur. Le succès des approches par histogramme provient de leur faible complexité calculatoire associée à une bonne robustesse vis-à-vis du bruit, de leur invariance aux rotations, aux occultations partielles et aux changements d'échelle. L'histogramme de couleur d'un objet est l'une des représentations décrivant son apparence. Il permet de représenter statistiquement la distribution des couleurs des pixels, c'est-à-dire la proportion de pixels répartis sur un ensemble de classes de couleurs. Il est basé sur l'encodage de l'information visuelle de couleur afin de modéliser l'objet cible sans prendre en compte l'information spatiale de chaque pixel. L'histogramme de couleurs ne peut pas être un modèle discriminant. Afin d'obtenir un modèle plus discriminant, Comaniciu et al. [1] ont utilisé l'histogramme pondéré de couleurs pour créer le modèle d'apparence. Cette modélisation est plus facile d'exploitation et permet d'intégrer plusieurs vues ou apparences de l'objet dans un espace de dimension faible mais ne permet pas aussi la conservation de l'information spatiale. La pondération de l'histogramme a pour but de favoriser les pixels les plus importants qui appartiennent à l'objet cible et de défavoriser les pixels de bruit qui appartiennent au fond. La figure 3.2 présente la procédure de calcul de l'histogramme pondéré de couleur 1D de 16 classes pour la composente R de l'espace de couleurs RGB en utilisant le noyau Epanechnikov. Ce noyau est une fonction de pondération utilisée dans les techniques d'estimation non-paramétrique. Le noyau Epanechnikov est donné par l'équation suivante :

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1}(d+2)(1-\|x_i\|^2) & si\ \|x\| \leq 1 \\ 0 & ailleurs \end{cases} \quad (3.1)$$

Où c_d est le volume de la dimension d ($c_d = \pi$, $d=2$).

Dans le tracker Mean shift, un modèle pour l'objet à suivre est sélectionné et l'histogramme pondéré de couleurs pour la région qui englobe l'objet cible est ainsi construit, dans la première trame. Le modèle est statique puisque seule l'apparence initiale de l'objet est utilisée

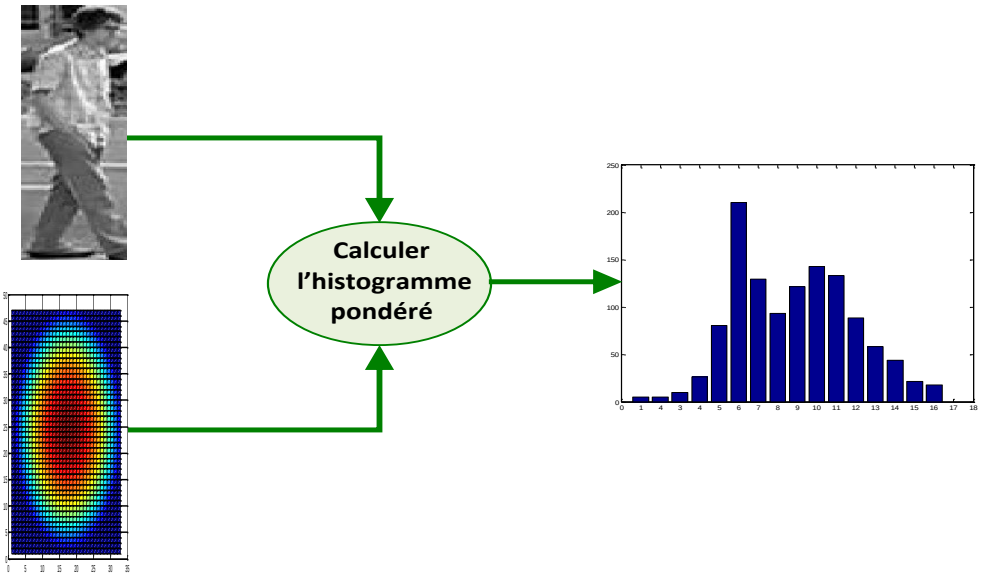


Figure 3.2 – Procédure de calcul de l'histogramme pondéré de couleur 1D de 16 classes pour la composente R.

pour le suivi d'objet tout au long de la séquence. Par la suite, l'algorithme Mean shift recherche l'objet cible dans l'image suivante en utilisant la distance de Bhattacharyya afin de comparer les histogrammes associés au modèle cible et des régions candidates. La recherche de la cible dans une trame repose sur la procédure itérative de recherche du maximum d'une densité de probabilité. À chaque itération, le vecteur Mean shift est calculé tels que la similarité entre les histogrammes est augmentée. Cette procédure est répétée jusqu'à ce que la convergence soit réalisée et que la valeur de similarité ne dépasse pas un seuil ou que le nombre limite d'itérations atteint (de quatre à six itérations en général). La figure 3.3 illustre la procédure de tracker Mean shift.

L'histogramme de couleurs rend le tracker Mean shift est moins robuste en particulier lorsque les couleurs de l'objet cible ne sont pas suffisamment contrastées par rapport au fond. De plus, le choix du nombre de classes (bins) d'un histogramme est délicat. Un trop faible nombre de classes détruit de l'information et mène à supprimer les contrastes pouvant exister entre certains objets de la séquence étudiée. Au contraire, un trop grand nombre de classes mène à des graphiques incohérents où toutes les classes sont faiblement représentées. Cette problématique a été étudiée dans [184]. En général, le nombre de classes de couleurs utilisé dans l'algorithme Mean shift est 16 ou 32 classes pour calculer l'histogramme.

Le problème de sélection d'un espace de couleur pour modéliser l'objet cible avait un effet important sur la robustesse de tracker Mean shift. Les informations de couleurs qui sont extraites de n'importe quel espace de couleurs sont différentes d'un tracker à l'autre, en raison

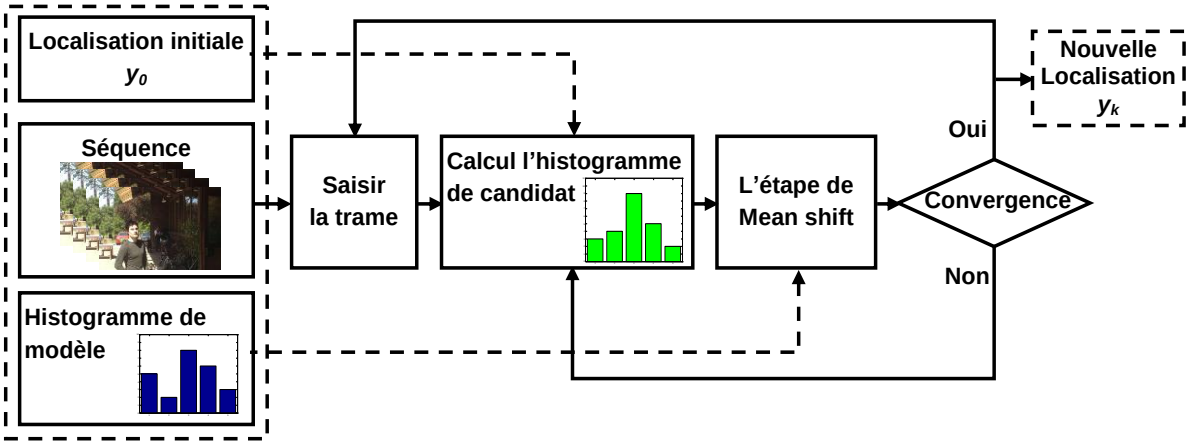


Figure 3.3 – La procédure de tracker Mean shift.

des caractéristiques spécifiques de chaque espace de couleurs. Bien que plusieurs algorithmes de suivi visuel ont été utilisés l'information de couleur, mais il n'y a pas une étude systématique sur les effets de l'utilisation de l'espace de couleurs pour le suivi visuel. On va essayer de comprendre de l'influence de l'information couleur de chaque espace sur la performance de tracker Mean shift, dans les sections suivantes.

3.3 Les espaces colorimétriques

Un espace colorimétrique, plus communément appelé espace de couleur est une méthode de description et de représentation des couleurs d'une manière standard. Une couleur est généralement représentée par trois composantes. Ces composantes définissent un espace des couleurs. Plusieurs études ont été réalisées sur l'identification d'espaces colorimétriques les plus discriminants, mais sans succès, puisqu'il n'existe pas un espace de couleurs idéal. Cela rend le choix d'un espace de couleur est une décision très importante car il peut influencer considérablement sur les résultats du traitement. Dans la littérature, il existe de nombreuses méthodes de représentation des couleurs, certaines se trouvent dans [185]-[189]. Beaucoup d'espaces de couleurs ont été appliqués avec succès dans des applications de vision par ordinateur. Chaque espace de couleur a ses propres caractéristiques qui le rendent plus approprié que d'autres pour des applications spécifiques. Certains espaces colorimétriques sont perceptuellement linéaires, tandis que d'autres non linéaires. Certains espaces de couleur sont intuitifs à utiliser alors que d'autres ne le sont pas. Enfin, certains espaces des couleurs sont liés à un équipement spécifique (télévision, imprimantes, caméras), alors que d'autres sont également valables sur n'importe quel dispositif utilisé [186]. Dans la suivante, nous présenterons les espaces de couleurs les plus utilisés dans la vision par ordinateur : RGB, YCbCr, YUV, YIQ, HSV, XYZ, I1I2I3, CIE Lab, CIE Luv et OPP.

3.3.1 L'espace RGB

L'espace colorimétrique RGB est l'un des espaces colorimétriques les plus utilisés dans la technologie informatique (traitement des images et des vidéos, ...etc). Il est basé sur le mélange additif de trois couleurs primaires R (rouge), G (vert), B (bleu). L'importance de l'espace colorimétrique RGB réside dans le fait qu'il est étroitement lié à la façon dont l'œil humain perçoit la couleur. La représentation des couleurs dans cet espace donne un cube de Maxwel qui se base sur le système de coordonnées cartésiennes comme illustré dans la figure 3.4. Le système de représentation RGB introduit par la CIE (Commission Internationale de l'Eclairage) en 1931 reste un système de référence même s'il présente quelques inconvénients. Le problème principal dans cet espace est que les canaux sont très corrélés car l'espace RGB ne sépare pas les informations de luminance et de chrominance (c-à-d chaque canal contient des données de luminance) [186]. Ce problème affecte les performances de l'espace colorimétrique RGB dans l'analyse des couleurs et les algorithmes de reconnaissance basés sur la couleur. Le deuxième problème est l'existence d'une partie négative dans les spectres et par conséquent, l'impossibilité de reproduire un certain nombre de couleurs par superposition des trois spectres (voir la figure 3.5).

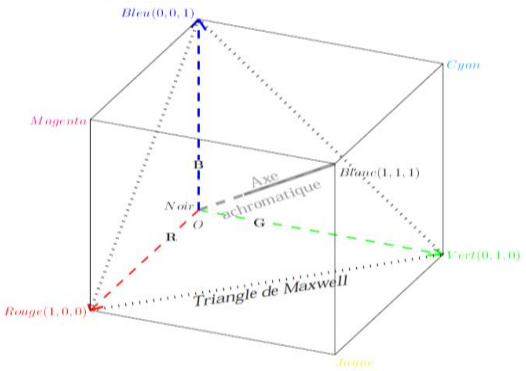


Figure 3.4 – Cube des Couleurs RGB.

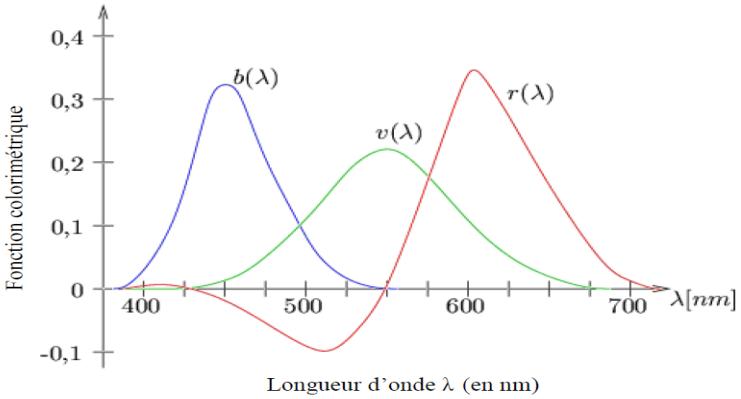


Figure 3.5 – Les courbes d'appariement R(λ), G(λ) et B(λ) correspondant aux Expériences d'égalisation avec standardisées par la CIE en 1931.

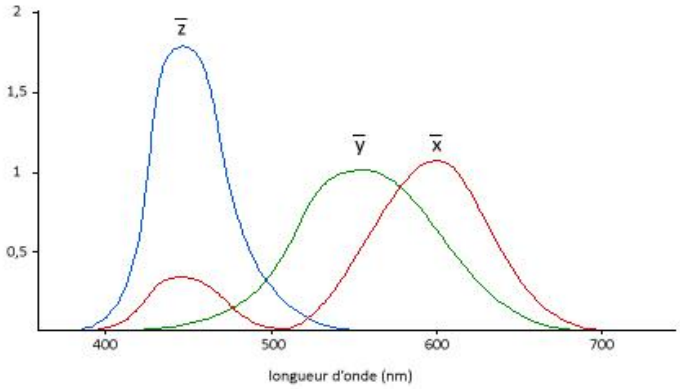


Figure 3.6 – Les fonctions colorimétriques X(λ), Y(λ) et Z(λ).

3.3.2 L'espace XYZ

L'espace de couleur XYZ est un standard international développé par la CIE en 1931 et constitue le fondement de toute colorimétrie [186]. Réalisé à partir d'une série d'expériences sur la perception des couleurs par l'œil humain, ce modèle sert de référence pour définir d'autres modèles. XYZ sont connus sous le nom de valeurs tristimulus. Cet espace définit toutes les couleurs visibles (gamme humaine) en utilisant uniquement des valeurs positives; par conséquent, les primaires X, Y et Z ne sont pas elles-mêmes visibles [190]. Le primaire Y est la luminance, tandis que les primaires X et Z donnent des informations de couleur (chrominances). Le système de référence colorimétrique XYZ permet de pallier le problème des valeurs négatives de la composante R du système RGB (voir figure 3.6). Chaque composante de ce système est une combinaison linéaire des composantes RGB. Ce système est rarement utilisé directement mais il sert plutôt de système de transition entre un système RGB et un autre système. Le passage d'un système RGB au système XYZ est donné comme suit:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 2.7690 & 17518 & 1.1300 \\ 1.0000 & 4.5907 & 0.0601 \\ 0.0000 & 0.0565 & 5.5943 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.2}$$

3.3.3 L'espace Lab et Luv

La CIE a proposé en 1976 deux espaces de couleur non linéaires CIELuv et CIELab qui sont perceptuellement uniformes. Ils sont dérivés de l'espace standard CIE XYZ et du point de référence blanc [191]. La distance euclidienne entre deux points de couleur dans les espaces colorimétriques CIELuv / CIELab correspond à la différence de perception entre les deux couleurs par le système de vision humaine. L'espace Lab sépare les informations de couleur

en Luminosité (luminance) (L) et les informations de couleur (chrominance) (a, b), (a) est en corrélation avec les composantes rouge-vert et (b) corrèle avec les composantes jaune-bleu. Ces deux systèmes se déduisent de XYZ par les transformations suivantes :

$$L = \begin{cases} 116(\frac{Y}{Y_b})^{1/3} - 16, & si\ \frac{Y}{Y_b} > 0.008856 \\ 903.3\ \frac{Y}{Y_b}, & sinon \end{cases} \quad (3.3)$$

$$u^* = 13\ L\ (\ u' - u'_b)\ \ avec\ \ u' = \frac{4X}{X + 15Y + 3Z} \quad (3.4)$$

$$v^* = 13\ L\ (\ v' - v'_b)\ \ avec\ \ v' = \frac{4X}{X+15Y+3Z} \quad (3.5)$$

$$a^* = 500\ (f\left(\frac{X}{X_b}\right) - f\left(\frac{Y}{Y_b}\right)) \quad (3.6)$$

$$b^* = 200\ (f\left(\frac{Y}{Y_b}\right) - f\left(\frac{Z}{Z_b}\right)) \quad (3.7)$$

avec $$f(x) = \begin{cases} (x)^{1/3}, & si\ x > 0.008856 \\ 7.787\ x + \frac{16}{116}, & sinon \end{cases} \quad (3.8)$$

Les termes $X_b, Y_b, Z_b, u'_b\ et\ v'_b$ sont à associer au blanc de référence.

3.3.4 L'espace HSV

L'espace HSV (teinte, saturation, value), est un espace colorimétrique basé sur l'idée du système visuel humain, a été développé pour être plus intuitif dans la manipulation des couleurs et a été conçu pour approcher la perception et l'interprétation des couleurs. Il est largement utilisé dans l'ordinateur graphique. L'espace HSV est une transformation non linéaire d'une représentation de coordonnées cartésiennes (RGB) à une représentation de coordonnées cylindriques (figure 3.7). HSV est conçu pour représenter la couleur de manière plus intuitive, simplifiant ainsi la tâche de quantifier une couleur perçue. L'espace HSV représente la couleur sous la forme d'un triplet : teinte H (Hue), saturation S (Saturation) et luminosité V (Value). Les transformations sont effectuées comme suit :

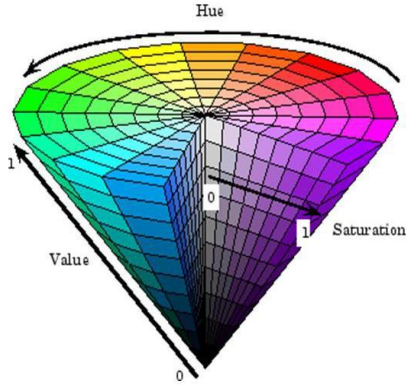


Figure 3.7 – Représentation du modèle HSV.

$$V = \frac{R + G + B}{3} \tag{3.9}$$

$$S = 1 - \frac{3\min(R, G, B)}{R + G + B} \tag{3.10}$$

$$H = cos^{-1}\left(\frac{0.5((R-G)+(R-B))}{\sqrt{((R-G)^2+(R-B)(G-B)}}\right) si\ \ B < G, \qquad sinon\ \ H = 2\pi - H \tag{3.11}$$

Avec ces équations, les intervalles de variation sont $H \in [0;\ 2\pi]$, $S \in [0;\ 1]$ et $V \in [0;\ 255]$.

3.3.5 Les espaces de type YCrCb

Ce système a été à l'origine développé pour assurer une compatibilité entre les téléviseurs couleurs et les téléviseurs noir et blanc, d'où la séparation des composantes de luminance et de chrominance. L'espace de couleur YCrCb défini par l'IRCC (International Radio Consultative Committee) est conçu pour améliorer l'efficacité du stockage et de la transmission en exploitant des informations perceptuellement significatives [190]. Il a dédié au codage digital des images de la télévision numérique, correspond à la norme ITU.BT-601 et fait partie du standard de compression JPEG2000. Cet espace sépare RGB en des composantes de luminance (Y) et de chrominance (Cb, Cr) en utilisant une transformation linéaire (figure 3.8). Cette transformation diffère suivant les standards de télévision NTSC (National Television System Committee), PAL (phase alternating line), ou SECAM (Séquentiel couleur à mémoire).

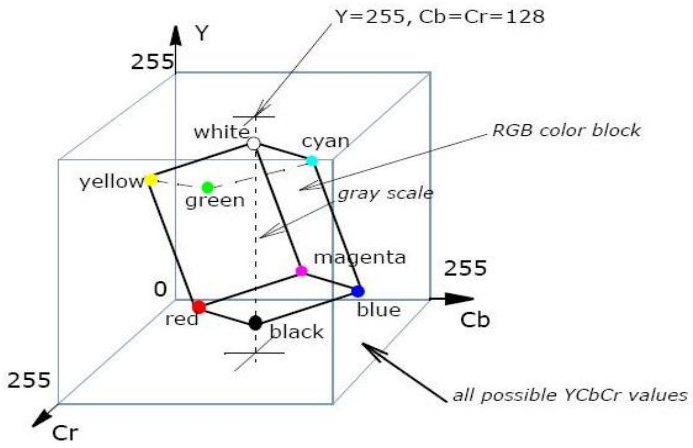


Figure 3.8 – Cube des Couleurs de l'espace YCrCb.

Comme déjà souligné, il existe plusieurs systèmes de type YCrCb. Les systèmes YIQ et YUV sont des espaces de couleur standards utilisés la transmission de télévision analogique. Le système YIQ est celui qui correspond à la norme NTSC, il est conçu pour exploiter les caractéristiques de réponse de la couleur de l'œil humain afin de maximiser l'utilisation d'une bande passante de transmission fixe [190]. Le système YUV est celui qui correspond à la norme PAL. Ces espaces de couleur similaires à YCrCb qui dérivent de l'espace RGB, où Y est la composante de la luminance, et U, V et I, Q sont des composantes de la chrominance.

Les principales transformations de l'espace RGB vers l'espace YCrCb, YIQ et YUV sont données par les équations suivantes :

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.12}$$

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.13}$$

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{3.14}$$

3.3.6 L'espace I1I2I3

Cet espace a été proposé par Ohta et al. [192] dans les années 80. Il s'appuie sur l'utilisation de l'analyse en composantes principales qui permet d'obtenir des composantes décorrélées. En effet, il est inspiré de la transformation de Karhunen-Loeve afin de déterminer les trois axes de plus grande variance de l'ensemble des couleurs. L'espace I1I2I3 appartient également à la famille des systèmes de type luminance-chrominance, puisque I1 correspond à la luminance, et I2 et I3 aux composantes de chrominance qui représentent respectivement une opposition bleu-rouge et une opposition magenta-vert.

Cet espace est une transformation linéaire à partir de l'espace RGB définie par les formules suivantes:

$$I_1 = \frac{R+G+B}{3} \tag{3.15}$$

$$I_2 = \frac{R-B}{2} \tag{3.16}$$

$$I_3 = \frac{2G-R-B}{4} \tag{3.17}$$

$$avec \qquad I_1 \in [0;255], \qquad I_2 \in \left[-\frac{255}{2};\frac{255}{2}\right], \qquad I_3 \in \left[-\frac{255}{2};\frac{255}{2}\right].$$

3.3.7 L'espace OPP

L'espace de couleur Opponent OPP est transformé à partir de l'espace de couleur RGB, les informations d'intensité sont contenues dans O3 et les informations chromatiques dans O1 et O2. En raison de la soustraction dans O1 et O2, les décalages s'annulent s'ils sont égaux pour tous les canaux (par exemple, une source de lumière blanche). Par conséquent, ces modèles de couleur sont invariants par rapport à l'intensité lumineuse. Le canal d'intensité O3 n'a pas de propriétés d'invariance [193][194]. Les formules exprimant la transformation de l'espace RGB à l'espace OPP sont données par :

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{3.18}$$

3.4 L'influence des espaces couleurs sur la performance de suivi

La couleur est une caractéristique très populaire qui est largement utilisée dans les algorithmes de suivi visuel. Elle acquiert sa popularité car elle est invariante à la translation, à la rotation, à l'occultation partielle, au changement de point de vue, aux variations de pose, au changement d'échelle et à la résolution, etc. De plus, elle est également facile à extraire. L'exploitation de l'information couleur pour le suivi visuel est un défi difficile. Les mesures de couleur peuvent varier considérablement sur une séquence d'images en raison des variations de l'éclairage, des ombres, des ombrages, de caméra et de géométrie d'objet [195]. Pour capturer l'information chromatique, plusieurs algorithmes de suivi utilisent des espaces de couleurs différents [194]. Dans [196], le filtre à particules de couleur introduit pour le suivi d'objets. Il calcule la probabilité de chaque particule en comparant son histogramme de couleur de l'espace de couleur HSV avec le modèle de couleur de référence. Dans [197], la distribution des couleurs RGB a été utilisée pour décrire le modèle cible et les candidats. Et l'objet cible a été localisé en minimisant la distance Kullback Leibler entre les distributions de couleurs du modèle cible et les candidats à l'aide d'une méthode de région de confiance. Mean shift [1], utilise l'histogramme de couleurs quantifié par l'espace de couleur RGB pour représenter le modèle cible et les candidats cibles. VTD [4] combine de différents modèles d'observation et de mouvement, et quatre modèles d'observation de base, qui utilisent respectivement des modèles de teinte, de saturation, d'intensité et de contour, sont adoptés. LOT [198] utilise l'espace de couleur HSV pour décrire l'apparence de chaque pixel de cible et le candidat. MEEM [199] utilise les caractéristiques extraites de l'espace de couleurs Lab. Dans le travail le plus récent [195], le tracker CSK [142] est étendu avec des noms de couleurs [200], et pour accélérer, la dimension des noms de couleurs d'origine est réduite par une technique de réduction de dimensionnalité adaptative.

Malgré des plusieurs algorithmes de suivi visuel ont utilisé l'information de couleur, mais il n'y a pas un seul espace de couleurs utilisé pour tous les trackers. Ce qui pose quelques questions : Quelle est l'influence des espaces de représentation de la couleur sur les performances des trackers ? Quelles sont les représentations chromatiques les plus adaptées au suivi visuel d'objets ?

Le choix des caractéristiques des couleurs est crucial pour la réussite d'un suivi visuel. Dans cette section, nous nous intéressons à examiner attentivement l'influence des espaces de représentation de la couleur sur les performances du tracker Mean shift (voir figure 3.9), puisqu'il se base sur l'histogramme de couleur et bien qu'il ait été introduit depuis plus d'une

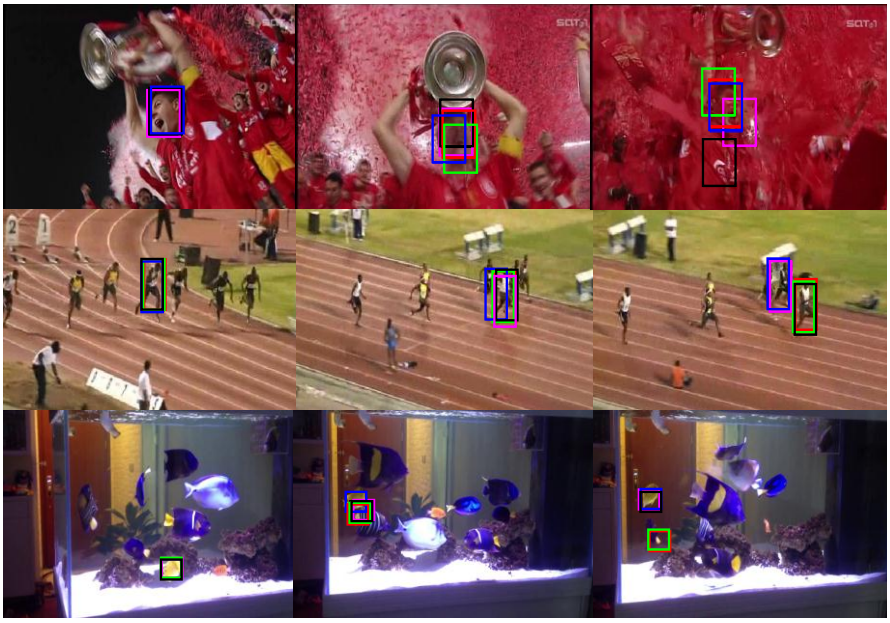


Figure 3.9 – Comparaison les résultats de suivi par le tracker Mean shift en utilisant différents espaces de couleurs en situations difficiles. Les résultats de Meanshift en utilisant HSV, RGB, Lab, YCbCr et OPP sont représentés par les boîtes : rouge, magenta, verte, bleue et noire, respectivement.

décennie mais son utilisation dans la littérature augmente encore et de nombreuses modifications ont été développées pour améliorer sa performance. Cependant, nous avons remarqué qu'il y a peu d'efforts dépensés dans la littérature pour étudier les effets de différents espaces de couleurs sur le processus de suivi lors de l'utilisation de l'algorithme de Mean shift.

Il existe de nombreux espaces de couleurs utilisés dans la vision par ordinateur, tels que RGB, HSV, YCbCr, YUV, YIQ, OPP, XYZ, Lab, rgb, I1I2I3, etc. Plusieurs d'entre eux ont été appliqués avec succès dans le domaine de suivi visuel d'objets. Les caractéristiques de chaque espace de couleur varient selon l'invariance photométrique (changements de la luminance ou de la couleur des images) et le pouvoir discriminatif aux représentations de couleurs biologiquement inspirées. Dans notre étude, nous utilisons les espaces de couleurs les plus utilisés dans le suivi d'objets pour étudier leurs effets sur les performances de tracker Mean shift. Ces espaces sont HSV, RGB, Lab, YCbCr et OPP. Cependant, les résultats d'utilisation de différents espaces de couleurs dans le tracker Mean shift seront présentés dans le chapitre 5.

3.4.1 Indicateurs de bon comportement

Comme nous avons pu le voir dans la section 3.2 la modélisation de l'apparence de l'objet joue un rôle capital dans la gestion des différents types de perturbations (variations d'apparence, occultation, mouvement, etc.). Puisque l'histogramme du modèle cible du tracker Mean shift est statique (car seule l'apparence initiale de l'objet - calculé sur la première trame - est utilisée pour le suivi de l'objet tout au long de la séquence). Cet histogramme est comparé avec les candidats cibles en utilisant le coefficient de Bhattacharyya pour trouver la localisation d'objet cible. Nous essayons de comprendre l'effet des espaces de couleurs à partir des informations intrinsèques de ce tracker. Ces informations sont le coefficient de Bhattacharyya défini dans la section (2.4.3), et la carte de rétroprojection qui est une carte de probabilité de l'image. Cette carte est un exemple d'information intrinsèque du modèle pouvant servir à caractériser le bon comportement du tracker Mean shift. Considérons la rétroprojection de l'histogramme modèle sur une fenêtre locale (image candidate) ou l'image dans laquelle l'objet est recherché : on lit pour chaque pixel de l'image, la probabilité associée à sa classe dans l'histogramme modèle. La figure 3.10 montre la carte de rétroprojection et l'image de poids de l'image candidat et leur coefficient de Bhattacharyya pendant l'itération du tracker Mean shift en appliquant l'histogramme de couleur 1D de la composante H dans l'espace HSV. L'image de poids représente le poids de chaque pixel dans l'image de candidat cible, définie dans la section (2.4.4)

La carte de rétroprojection révèle la localisation des classes des pixels qui ont des valeurs de probabilités plus grandes et plus faibles trouvées dans l'histogramme du modèle. Les valeurs de probabilités plus grandes indiquent aux pixels de l'objet cible et les valeurs de probabilités plus faibles indiquent aux pixels du fond (indiquées par la couleur bleue dans cette carte), comme illustré dans la figure 3.10. La distribution spatiale des pixels qui ont des valeurs de probabilités plus grandes peut renseigner sur la précision de localisation de l'objet, par exemple la présence de ces pixels en bas de la carte de rétroprojection, comme montré dans la carte de rétroprojection 1, qui indique que la position de la cible n'est pas dans le centre de la boîte englobante de l'objet, ce qui signifie que le tracker Mean shift ne peut pas détecter l'objet cible correctement dans cette trame. L'itération du tracker Mean shift permet de trouver l'objet cible en utilisant le coefficient de Bhattacharyya jusqu'à la convergence, comme montré dans l'image candidat 5. Dans ce cas, la valeur du coefficient de Bhattacharyya est maximale et les pixels qui ont des valeurs de probabilités plus grandes concentrés dans le centre de la carte de rétroprojection 5, ils sont indiqués par la couleur rouge dans cette carte.

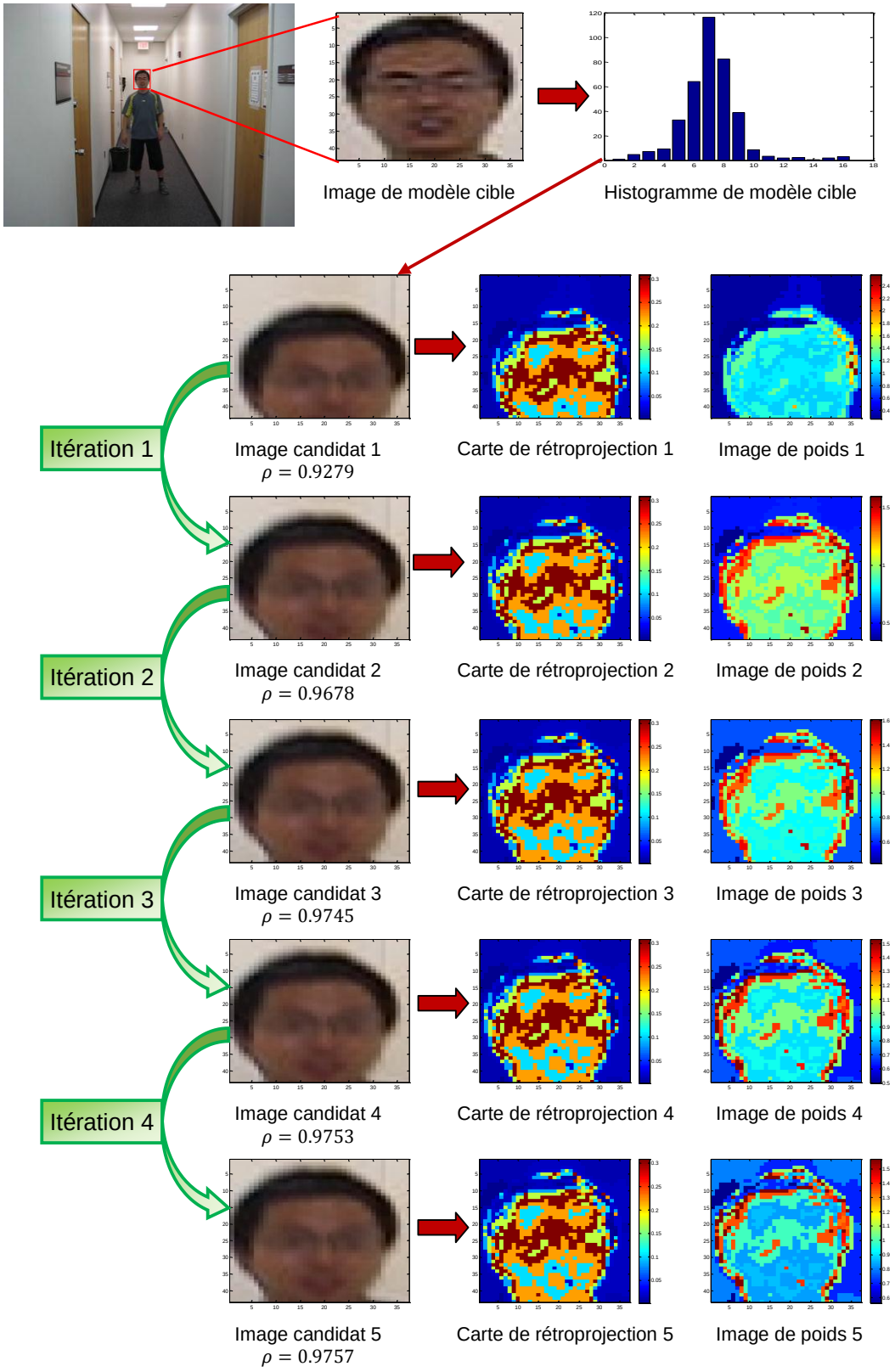


Figure 3.10 – Carte de rétroprojection et l'image de poids du tracker Mean shift pour des candidats pendant cinq itérations, en utilisant l'histogramme de couleur 1D de la composante H dans l'espace HSV.

3.4.2 Etude de l'effet des espaces de couleurs

Dans cette étude, nous avons choisi des séquences d'images qui contiennent plusieurs difficultés (changements d'illumination, variation d'échelle, occultation, déformation, flou de bougé, similarité entre objet et fond, objets similaires, etc.). Ces séquences couvrent 5 catégories d'objet à suivre (visage, personne et vélo, cerf, poisson). Nous essayons de comprendre l'influence du choix de l'espace de couleurs sur la performance du tracker Mean shift, et ceci à travers le comportement des cartes rétroprojections et les valeurs de coefficient de Bhattacharyya de l'objet cible pour chaque espace. Dans les figures 3.11, 3.12, 3.13 et 3.14, nous montrons le comportement des cartes à différents espaces de couleurs aux séquences Boy, Deer, MountainBike et Fish_ce1, respectivement, et à différents instants d'une même séquence.

Les cartes qui semblent traduire un bon comportement du tracker Mean shift sont signifiées par ✓, celles qui semblent traduire une dérive par ✗. Les cartes marquées par **?** présentent un comportement ambigu où elles semblent indiquer un mauvais fonctionnement du tracker Mean shift (mêmes allures que lors d'une dérive du tracker) alors que ce n'est pas le cas.

Les figures 3.11(a), 3.12(a), 3.13(a) et 3.14(a) présentent les histogrammes des classes qui ont des valeurs de probabilités différentes de 0, ont pour but de donner les nombres des classes détectées dans les trois composantes de chaque espace. Ce nombre représente la quantité des informations extraites pour chaque espace. L'espace HSV donne toujours le nombre des classes supérieures aux autres espaces Lab, RGB, YCbCr et OPP. Cependant, les nombres des classes détectées par Lab et YCbCr ne dépassent pas la valeur 70, dans cette étude. Cette différence retourne aux caractéristiques spécifiques pour chaque espace de couleurs.

- **L'analyse de la carte de rétroprojection**

La rétroprojection d'un histogramme de modèle sur l'image de l'objet suivi (image modèle cible, images candidates) sépare entre les pixels de l'objet cible et les pixels du fond, mais cela dépend des caractéristiques de couleurs extraites par les espaces couleurs utilisés, comme illustré dans les figures 3.11(b), 3.12(b), 3.13(b) et 3.14(b).

Dans la figure 3.11 (b) l'espace de couleur HSV permet de bien séparer le visage de l'objet du fond (carte encadrée en rouge), aussi pour Lab (carte encadrée en magenta). Tandis que, les espaces RGB (carte encadrée en vert), YCbCr (carte encadrée en bleu) et OPP (carte encadrée en noir) séparent la tête de l'objet du fond, c-à-d, ils détectent la couleur des cheveux avec le visage. Par conséquent, l'espace HSV détecte précisément la couleur de peau par rapport aux

autres espaces. Le tracker Mean shift est robuste pour suivre le visage en utilisant ces espaces (figure 3.11(c)). Un bon comportement de ce tracker est observé, bien qu'à la déformation de l'objet et au flou de bougé, comme illustré dans la figure 3.11(d). Cependant, un mauvais comportement correspond à un changement d'échelle de l'objet cible pour les espaces Lab (carte encadrée en magenta), RGB (carte encadrée en vert), YCbCr (carte encadrée en bleu) et OPP (carte encadrée en noir), en raison de diminution de l'information de couleurs du fond, rendant compte de la mauvaise détermination de la localisation (voir figure 3.11(e), sauf la carte encadrée en rouge). Au contraire, un bon fonctionnement de ce tracker est observé lorsque l'espace HSV est utilisé (carte encadrée en rouge).

Dans la figure 3.12 (b), les cartes du modèle cible ne montrent pas l'objet à suivre (tête du cerf) avec précision. La distribution spatiale des pixels qui ont des valeurs de probabilités plus grandes existe dans le coté droit de cette carte pour les espaces de couleurs utilisés, parce que la position de l'objet à suivre n'est pas au centre de la boîte englobante, et aussi la similarité entre l'objet et le fond. Dans ce cas, ne peut pas renseigner sur la précision de la localisation de l'objet. Le tracker Mean shift donne une bonne performance en utilisant l'espace couleur OPP comme montré dans les cartes encadrées en noir (figure 3.12). Pour les espaces HSV, Lab, RGB et YCbCr un bon comportement de ce tracker est observé pour les cartes qui présentent la même distribution spatiale des pixels de la carte du modèle cible (les cartes marquées par ✓). En revanche, un mauvais comportement correspond à une déformation importante de cette région par rapport à la forme de la carte du modèle cible (les cartes marquées par ✗). Dans cette séquence, le tracker Mean shift dérive lorsqu'il utilise les espaces HSV, Lab, RGB et YCbCr, puisqu'il y a des objets similaires (figure3.12(d)). La grande similarité entre l'objet cible et le fond a influé sur le modèle d'apparence qui est devenu inadapté au suivi.

La figure 3.13 illustre que le tracker Mean shift est robuste pour le suivi d'objet cible (humain et vélo) en utilisant l'espace YCbCr, et moins robuste pour l'espace HSV. Tandis que, ce tracker dérive lorsqu'il utilise les espaces Lab, RGB et OPP, à cause du clutter de fond et la rotation de l'objet cible. La figure 3.13 (b) montre que l'espace de couleur YCbCr permet de bien séparer l'humain et le vélo du fond (carte encadrée en bleu). Alors que, l'utilisation des espaces HSV et Lab démontre que les pixels de l'objet à suivre sont clairement différents du fond (carte encadrée en rouge et en magenta). Tandis que, les espaces RGB (carte encadrée en vert) et OPP (carte encadrée en noir) ne séparent pas bien l'objet à suivre du fond. A travers les cartes du modèle cible on peut voir que les pixels qui ont des valeurs de

probabilités plus grandes trouvées dans l'histogramme de modèle sont des pixels du fond et les pixels qui ont des valeurs de probabilités plus faibles sont des pixels de l'objet cible, car le tracker MS donne une importance à des pixels qui existent dans le centre de l'objet cible. Un bon fonctionnement du tracker est observé lorsque les cartes de candidats ont la même distribution spatiale des pixels de la carte du modèle cible, rendant compte de la précision de la localisation (les cartes marquées par ✓). Un mauvais comportement est observé lorsque les cartes présentent une déformation importante de cette région, rendant compte d'une mauvaise précision de localisation (les cartes marquées par ✗). Cependant, des cas ambigus existent où la carte présente est différente par rapport à la carte du modèle cible mais le tracker localise correctement la cible (les cartes marquées par **?**). Dans ce cas, l'objet cible fait une rotation importante comme montré dans la figure 3.13 (d).

Les cartes de rétroprojections de l'image du modèle dans la figure 3.14 (b) montre que l'objet à suivre a plusieurs couleurs pour la plupart des espaces utilisés. Bien que, l'objet cible est un poisson de couleur jaune citron, ce qui signifie que l'intervalle de variance des valeurs de chaque espace de couleur est différent. Le tracker MS est peu robuste dans cette séquence à cause des déformations de l'objet dans le plan, changements d'illumination et aux occultations partiales et totales, tout cela se produit en même temps.

Un bon comportement du tracker pour tous les espaces est observé lorsque les cartes de candidats ont la même distribution spatiale des pixels de la carte du modèle cible (les cartes marquées par ✓), car ces cartes existent en début de la séquence et l'objet cible n'est pas affecté par aucun facteur. Cependant, Les cartes ayant une déformation importante sont souvent accompagnées d'une dérive du tracker (les cartes marquées par ✗) car la localisation de l'objet est imprécise. Tandis que, les cartes indiquent la dérive du tracker alors que celui-ci fonctionne correctement (les cartes marquées par **?**), comme illustré lorsque ce tracker utilise l'espace YCbCr. Alors, l'interprétation des cartes n'est pas toujours évidente.

- **L'analyse des valeurs de coefficient Bhattacharyya**

Dans le tracker Mean shift, pour localiser la cible dans une image, on cherche à maximiser le coefficient de Bhattacharyya entre l'histogramme du modèle cible et l'histogramme du candidat cible, défini dans la section (2.4.3). Lorsque le tracker Mean shift utilise des espaces de couleurs différents, les valeurs du coefficient de Bhattacharyya varient d'un espace à l'autre, bien qu'il utilise une même séquence d'images (figures 3.11, 3.12, 3.13 et 3.14). Cette différence de valeur de coefficient est liée au nombre des classes détectées dans les trois

composantes de chaque espace, comme nous avons pu le voir dans l'histogramme de chaque espace (figures 3.11(a), 3.12(a), 3.13(a) et 3.14(a).

Le coefficient de Bhattacharyya n'est donc pas permis de connaître le comportement du tracker Mean shift. La détermination de l'espace de couleurs qui donne la meilleur performance à travers de ce coefficient est impossible, comme illustré dans les figures 3.11(c), 3.12(c), 3.13(c) et 3.14(c). Les cartes indiquent un bon comportement du tracker pour tous les espaces de couleurs alors que leurs valeurs de coefficient de Bhattacharyya sont différentes. Les valeurs de coefficient Bhattacharyya pour l'espace HSV sont toujours inférieures aux autres valeurs, car son nombre des classes est supérieur aux autres espaces. Si le nombre des classes est important, cela réduit la valeur de coefficient Bhattacharyya, et augmenter la précision de la comparaison entre les deux histogrammes modèle cible et candidat cible. De plus, on peut observer que le tracker dérive, mais la valeur de coefficient Bhattacharyya est supérieure à 0.5. Ce qui signifie que l'histogramme de modèle et l'histogramme de candidat sont convergents, comme montré dans la figure 3.13(d) et (e) (carte encadrée en noir et carte encadrée en magenta, respectivement), c-à-d les couleurs des régions du modèle cible et du candidat cible sont proches.

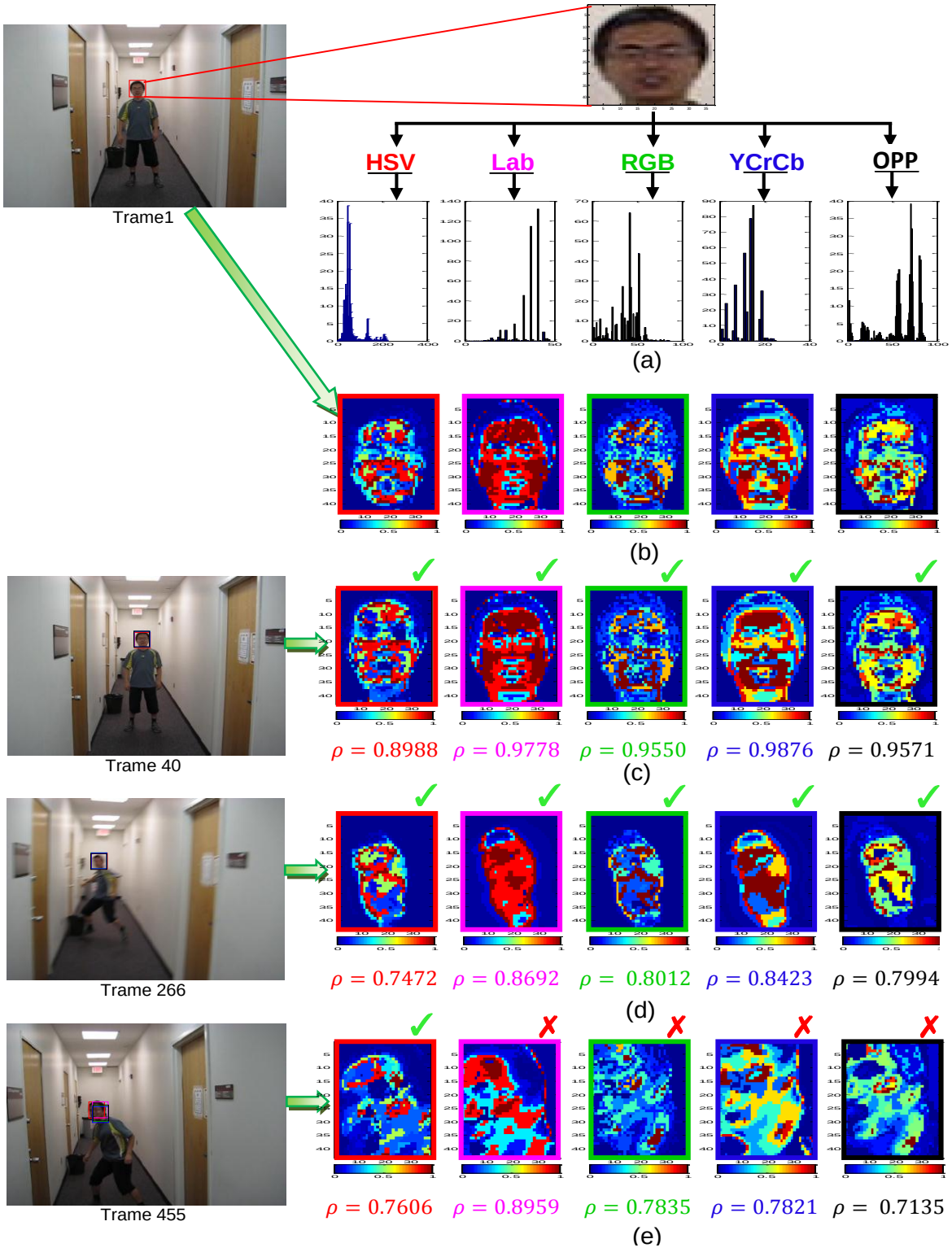


Figure 3.11 – Carte de rétroprojection de séquence Boy en utilisant différents espaces de couleurs. Les images encadrées en rouge utilisent HSV, en violet utilisent Lab, en vert utilisent RGB, en bleu utilisent YCbCr, en noir utilisent OPP. (a) L'histogramme de modèle cible pour chaque espace de couleur. (b) La carte de rétroprojection de l'image de l'objet modèle (trame1). (c), (d) et (e) Sont les cartes de rétroprojections des images d'objets cibles et leurs valeurs de coefficient de Bhattacharyya aux instants (t=40, 266 et 455), respectivement.

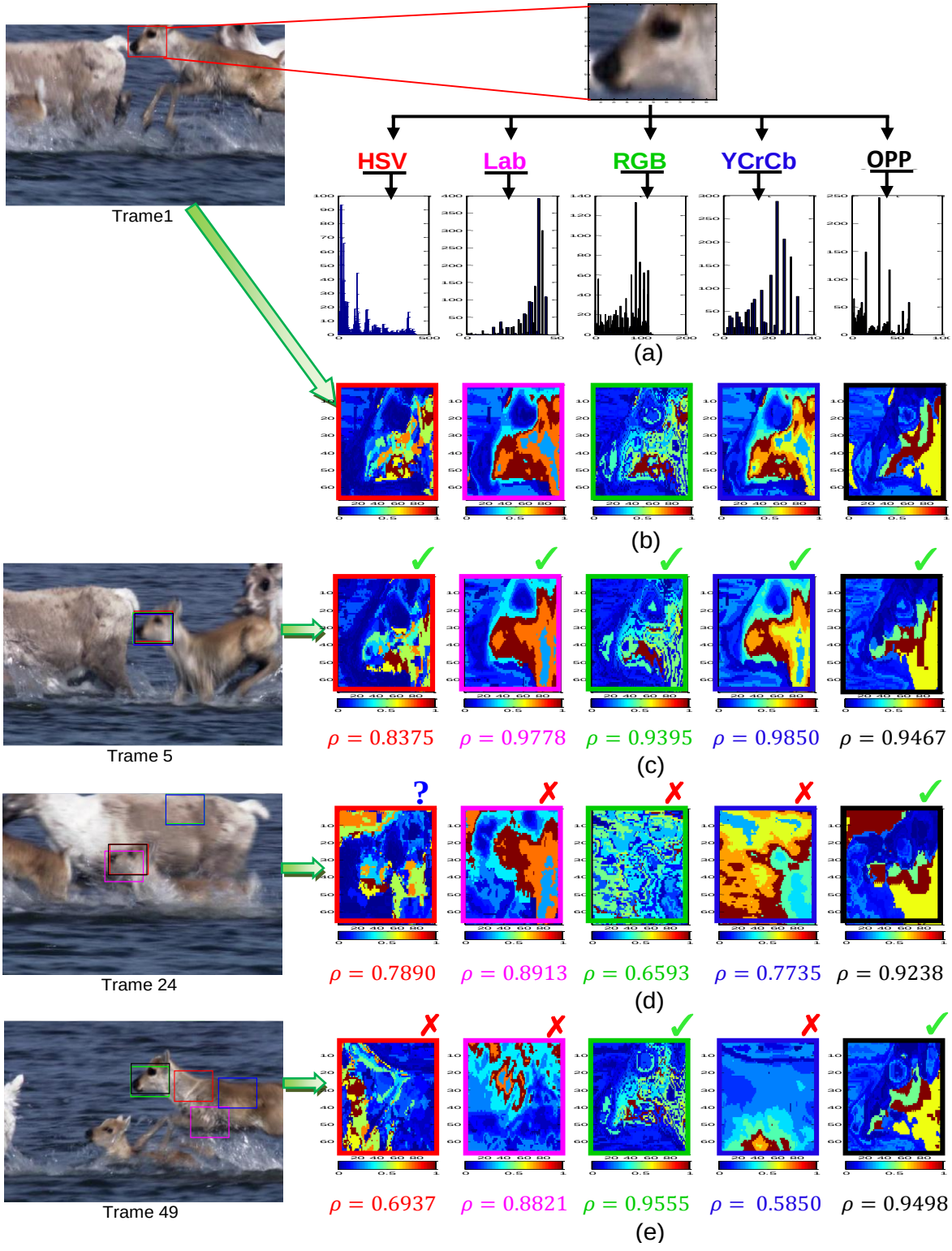


Figure 3.12 – Carte de rétroprojection de séquence Deer en utilisant différents espaces de couleurs. Les images encadrées en rouge utilisent HSV, en violet utilisent Lab, en vert utilisent RGB, en bleu utilisent YCbCr, en noir utilisent OPP. (a) L'histogramme de modèle cible pour chaque espace de couleur. (b) La carte de rétroprojection de l'image de l'objet modèle (trame1). (c), (d) et (e) Sont les cartes de rétroprojections des images d'objets cibles et leurs valeurs de coefficient de Bhattacharyya aux instants (t=5, 24 et 49), respectivement.

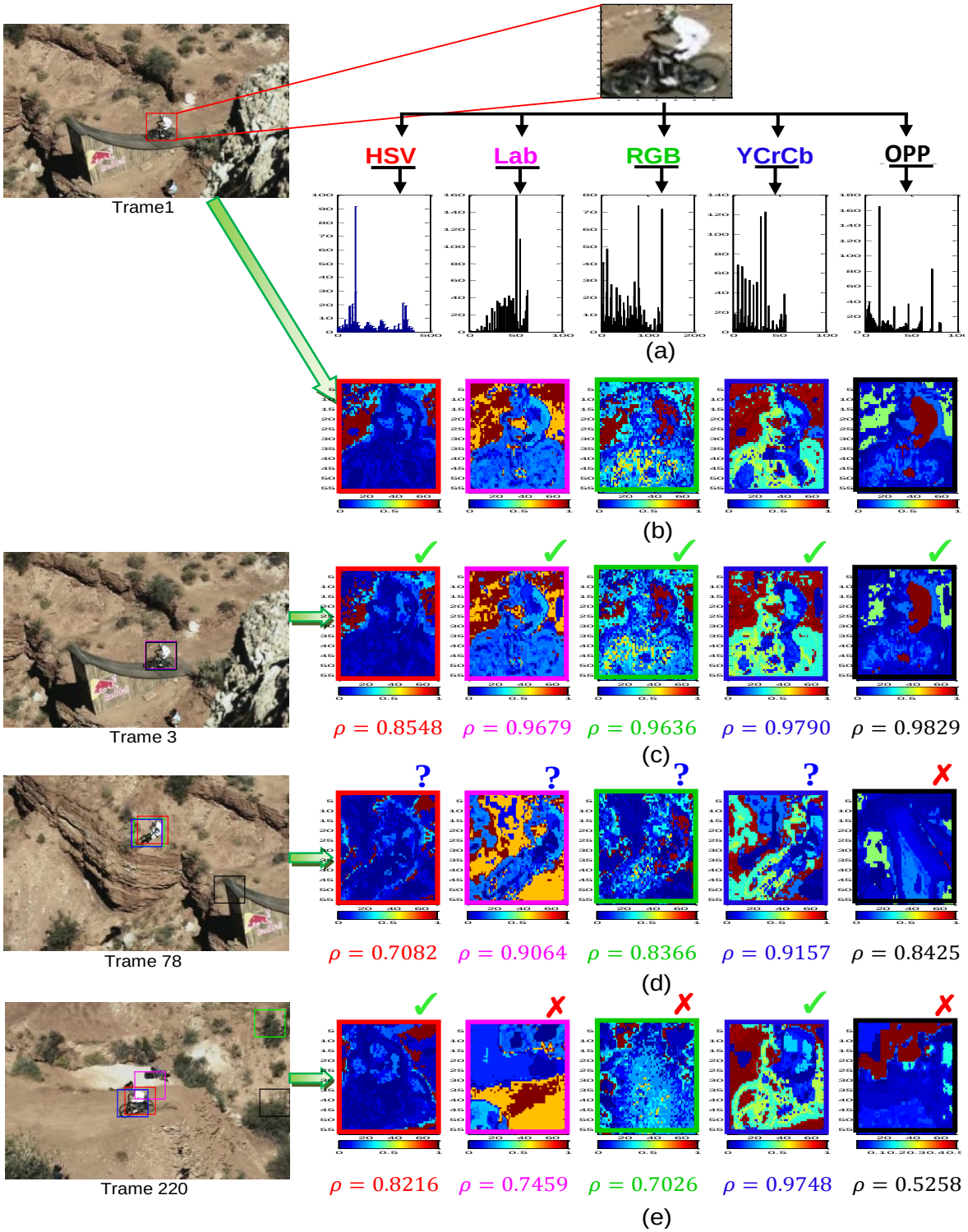


Figure 3.13 – Carte de rétroprojection de séquence MountainBike en utilisant différents espaces de couleurs. Les images encadrées en rouge utilisent HSV, en violet utilisent Lab, en vert utilisent RGB, en bleu utilisent YCbCr, en noir utilisent OPP. (a) L'histogramme de modèle cible pour chaque espace de couleur. (b) La carte de rétroprojection de l'image de l'objet modèle (trame1). (c), (d) et (e) Sont les cartes de rétroprojections des images d'objets cibles et leurs valeurs de coefficient de Bhattacharyya aux instants (t=3, 78 et 220), respectivement.

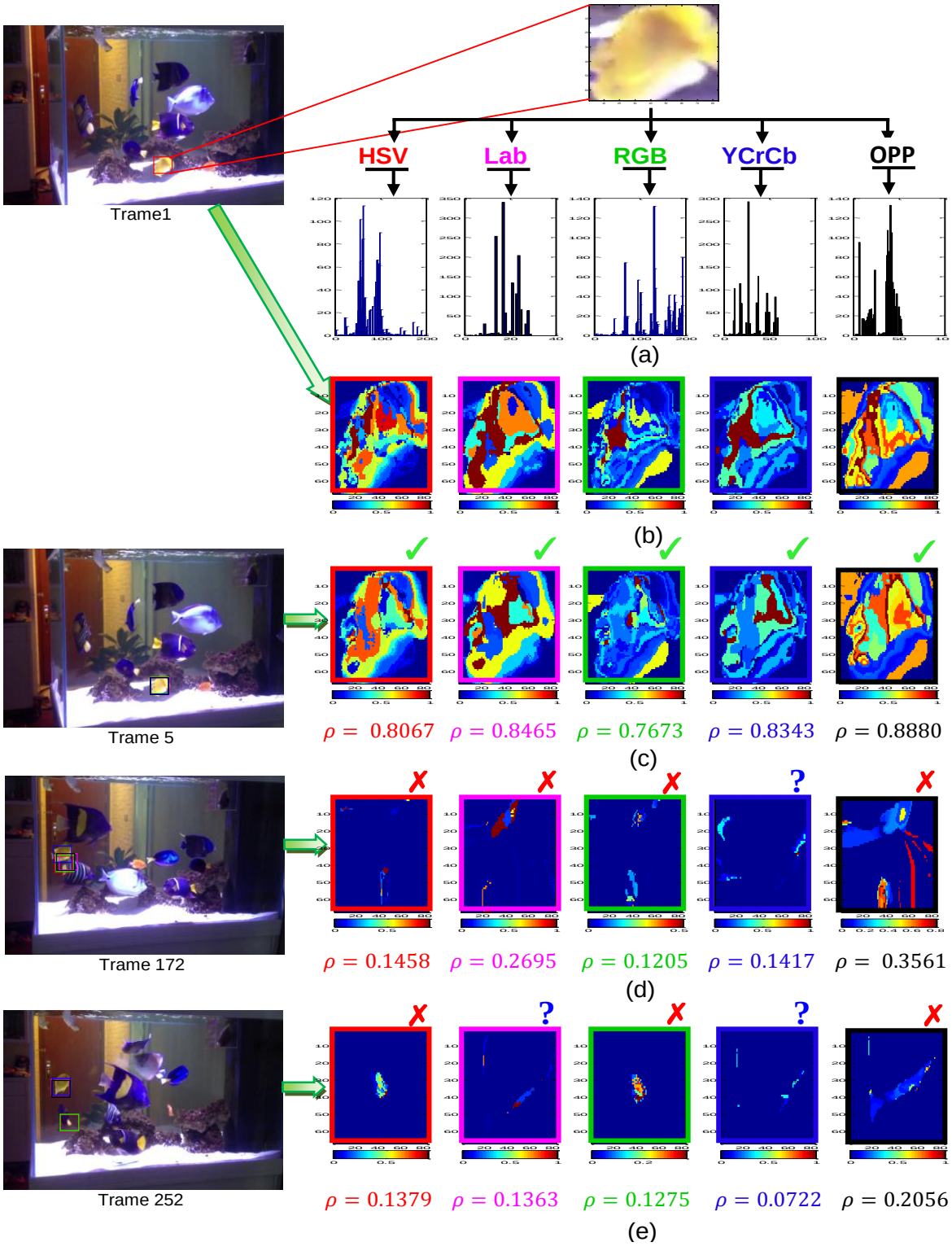


Figure 3.14 – Carte de rétroprojection de séquence Fish_ce1 en utilisant différents espaces de couleurs. Les images encadrées en rouge utilisent HSV, en violet utilisent Lab, en vert utilisent RGB, en bleu utilisent YCbCr, en noir utilisent OPP. (a) L'histogramme de modèle cible pour chaque espace de couleur. (b) La carte de rétroprojection de l'image de l'objet modèle (trame1). (c), (d) et (e) Sont les cartes de rétroprojections des images d'objets cibles et leurs valeurs de coefficient de Bhattacharyya aux instants (t=5, 172 et 252), respectivement.

3.5 Conclusion

Les travaux décrits dans ce chapitre sont centrés sur l'étude et l'analyse de l'effet de l'espace de couleur sur le suivi d'objets à l'aide du tracker Mean shift qui se base sur l'histogramme de couleur. Puisque l'information couleur joue un rôle capital dans la construction du modèle d'apparence du tracker Mean shift, nous avons appliqué plusieurs espaces de couleurs pour construire ce modèle et pour voir l'effet de ces espaces sur la performance de ce tracker. Pour évaluer la qualité du modèle d'apparence du tracker Mean shift nous avons utilisé des indicateurs de comportement de ce tracker. Ces indicateurs de comportement exploitent des caractéristiques intrinsèques du modèle (carte de rétroprojection et le coefficient de Bhattacharyya) qui traduisent une certaine qualité de la prédiction. Cette étude montre qu'il n'existe pas un seul espace de couleurs approprié pour tous les défis liés à la déformation, à la rotation, aux changements d'illumination, à la variation d'échelle, aux occultations, au flou de bougé, à la similarité entre l'objet et le fond et des objets similaires, etc. Aussi, le choix de l'espace colorimétrique peut être très important et très difficile, car l'espace couleur approprié dépend de la situation actuelle qui peut varier entre les trames et de la capacité de distinguer entre l'objet et son fond. Dans le chapitre 5, nous allons présenter les résultats de suivi du tracker Mean shift en utilisant plusieurs espaces de couleurs, afin de déterminer l'espace de couleur approprié pour construire une représentation robuste du modèle d'apparence d'objet cible.

Chapitre 4

Suivi d'objets robuste en utilisant un histogramme conjoint couleur-texture

Sommaire

4.1 Introduction

Comme nous avons pu le voir précédent, le modèle d'apparence affecte d'une manière directe la performance d'un système de suivi d'objets. La validité et la robustesse des opérations de suivi dépendent de la qualité représentative de caractéristiques visuelles extraites à partir des objets cibles. Le modèle d'apparence est extrait en utilisant le descripteur de couleur, le descripteur épars, le descripteur de mouvement et le descripteur de l'information spatiale. Chaque descripteur permet de décrire un objet selon un caractère particulier, par exemple l'histogramme de couleurs ne peut pas être un modèle discriminant dans le cas de la similarité des caractéristiques de l'objet et du fond, tandis qu'il est moins sensible aux transformations géométriques. Afin d'obtenir la modélisation la plus robuste possible des objets cibles, les efforts majeurs dans le suivi d'objet ont porté sur la fusion de modèles d'apparence différents. Diversifier les modèles d'apparence en combinant des caractéristiques de couleur, de forme ou de texture, augmente la représentativité des apparences des objets et du contexte, permettant de mieux gérer les variations d'apparence rencontrées au cours du suivi.

Motivés par la rapidité et la robustesse du tracker Mean shift, nous proposons une nouvelle représentation de modèle d'apparence de cible et qui se base sur une mixture des caractéristiques de couleur et de texture LPQ, LBP ou BSIF pour améliorer la robustesse et la précision de ce tracker. Ce chapitre présente donc le suivi d'objet en utilisant l'histogramme pondéré conjoint de couleurs et de textures dans le cadre du tracker Mean shift.

4.2 Aperçu du Modèle d'apparence proposé

Actuellement, l'histogramme de couleur est largement utilisé pour représenter un objet cible, à cause de sa simplicité et sa robustesse. Il consiste à estimer la distribution des valeurs d'intensité pour l'objet cible. Le tracker Mean shift repose sur l'histogramme pondéré de couleur afin de modéliser l'objet cible. Il est robuste à l'occultation partielle, à l'échelle, la rotation et la déformation non-rigide de la cible. Cependant, l'utilisation uniquement l'histogramme de couleur pour modéliser l'objet cible pose certains problèmes. Premièrement, les informations spatiales de la cible sont perdues. Deuxièmement, lorsque la cible a une apparence similaire à celle du fond, l'histogramme de couleur devient incapable pour les distinguer [14] [19][201].

Pour cette raison, plusieurs chercheurs [14][19][202] affirment que la combinaison de plusieurs caractéristiques visuelles peuvent améliorer la convergence de cet algorithme dans des conditions complexes, mais le choix des caractéristiques et la manière de les combiner restent des problèmes au cœur de la recherche. La combinaison de caractéristiques offre donc de nombreux avantages, mais toutes les caractéristiques ne sont pas discriminantes. Afin d'améliorer le tracker Mean shift plusieurs caractéristiques ont été utilisées pour les combiner avec un histogramme de couleur; telles que les caractéristiques spatiotemporelles, les caractéristiques en gradient, les caractéristiques de texture [43][203].

Récemment, de nombreux chercheurs ont proposé diverses méthodes améliorées [14] [16] [19][204][205] qui utilisent l'histogramme conjoint couleur-texture (section 2.2). Il est plus fiable que d'utiliser uniquement l'histogramme de couleur dans le suivi des scènes complexes. Les motifs de texture [206][207], qui reflètent la structure spatiale de l'objet sont des caractéristiques efficaces pour représenter et reconnaître les cibles. Contrairement à la couleur, la texture n'est pas la propriété d'un pixel, mais d'une région. Étant donné que les caractéristiques de texture présentent de nouvelles informations que l'histogramme des couleurs ne donne pas, généralement, ils ne soumettent pas à l'impact de la lumière et la couleur du fond. Bien que de nombreuses méthodes d'analyse de texture ont été proposés

telles que les matrices de cooccurrence des niveaux de gris [208], le filtre de Gabor [206], LBP [209][210] et CLTP [211]. Cependant, la façon d'utiliser efficacement les caractéristiques de l'intensité de la couleur et de la texture, et le choix de bon descripteur de texture, reste des problèmes difficiles.

En raison du pouvoir discriminant et la simplicité de calcul des descripteurs de textures LPQ (Local Phase Quantization), LBP (Local Binary Patterns) et BSIF (Binarized Statistical Image Features) dans diverses applications de vision par ordinateur, nous proposons des nouveaux modèles d'apparences en combinant l'histogramme pondéré de couleur HSV et la texture LPQ, LBP ou BSIF à l'objectif de rendre le tracker Mean shift plus robuste et plus précis. Dans ce travail, nous utilisons le descripteur LPQ, le descripteur LBP et le descripteur BSIF pour représenter les caractéristiques de textures de l'objet cible, puis nous proposons une nouvelle méthode de combiner ces caractéristiques avec l'histogramme pondéré de couleur pour créer un histogramme conjoint couleur-texture de manière plus distinctive et efficace.

Comparé à la représentation traditionnelle basée sur l'histogramme de couleur RGB, les modèles d'apparences proposés exploitent efficacement les informations structurelles de l'objet cible et permettent d'obtenir de meilleures performances de suivi avec moins d'itérations de tracker Mean shift et plus grande robustesse dans les scènes complexes, en particulier à diverses interférences du fond et au flou de mouvement. Dans la classification de la texture, l'opérateur LPQ est très robuste au flou que l'opérateur Local Binary Pattern (LBP) qu'il utilise dans [14]. La figure 4.1 illustre un exemple sur un modèle d'apparnce proposé dans le cadre du tracker Mean shift.

4.3 Descripteurs de texture

De nombreux algorithmes d'extraction des caractéristiques de texture ont été proposés dans la littérature du traitement d'image. Les descripteurs de texture ont été considérés comme l'une des méthodes les plus efficaces adoptées au système de vérification, à la détection des régions d'intérêt et le système de suivi, etc. Les descripteurs de texture représentent les images de manière discriminative avec leurs micros caractéristiques d'apparence locale. Ces caractéristiques sont invariantes à la rotation, l'échelle, le flou et l'éclairage.

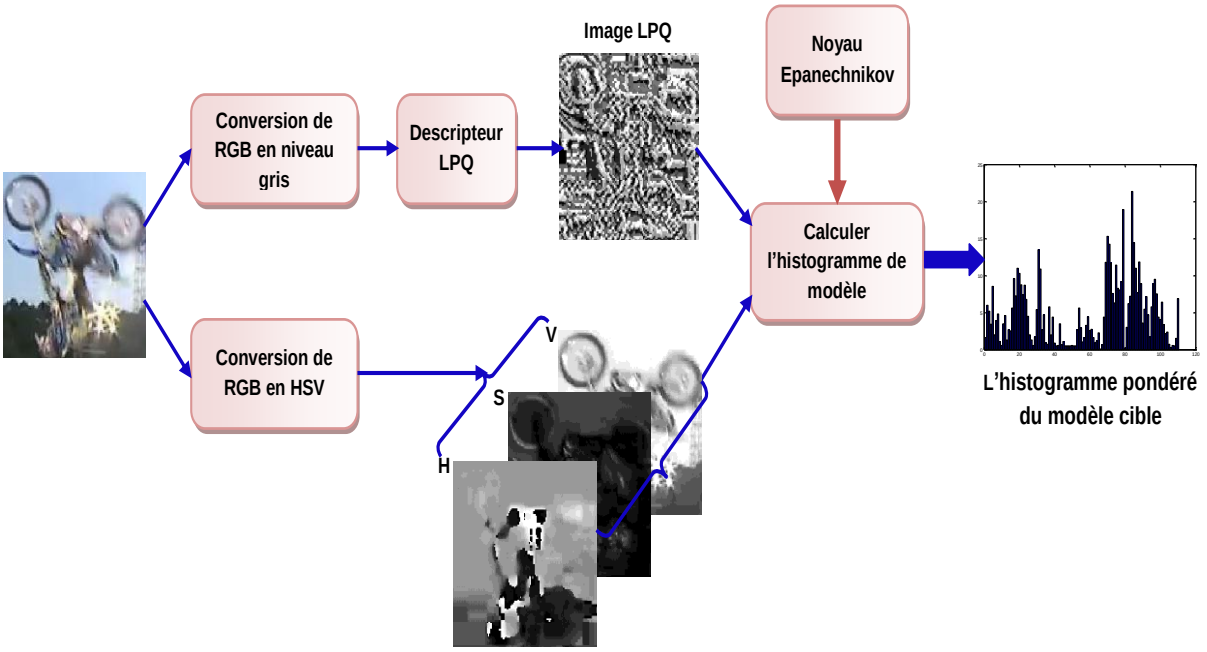


Figure 4.1 – Schéma général de la méthode proposé pour combiner l'histogramme de couleur HSV avec la texture LPQ pour représenter le modèle cible.

Dans cette section, on va présenter certains descripteurs les plus utilisés dans la littérature, en particulier dans les systèmes de reconnaissance de visage. Ces descripteurs sont le descripteur LPQ qui a utilise dans notre modèle proposé décrit dans la section précédente, et les descripteurs LBP (Local Binary Patterns) et BISF (Binarized Statistical Image Features) qui utilisent pour la comparaison.

4.3.1 Le descripteur LPQ

La texture d'un objet est une caractéristique relativement stable qui pourrait refléter l'information des caractéristiques de la région d'objet [203]. L'opérateur de la quantification de phase locale (Local Phase Quantization LPQ) a été proposé par Ojansivu et al. [212] pour la description de la texture. Il a été démontré que l'opérateur est robuste au flou et le plus performant que l'opérateur LBP (section 4.3.2) [210] dans la classification de la texture. Le descripteur LPQ permet d'améliorer la classification de textures pour être robuste aux artéfacts générés par différentes formes de flou présents dans une image. Pour cela, le descripteur est construit de façon à ne retenir dans une image que l'information locale invariante à un certain type de flou. Il est insensible au flou central symétrique, tel que celui causé par le mouvement linéaire et hors du foyer du capteur [212]. Dans notre travail, nous proposons le descripteur LPQ comme une méthode efficace pour résoudre le problème de l'information spatiale.

Les auteurs [212] ne considèrent en effet que les flous pouvant être représentés par une fonction d'étalement du point (PSF, "Point Spread Function") présentant une symétrie centrale. Cette hypothèse sur la PSF ne limite pas pour autant l'utilisation de cette méthode étant donné que la réponse à une source ponctuelle de la majorité des capteurs et des systèmes d'imagerie peut être modélisée par ce type de fonctions mathématiques qui peuvent également présenter des symétries d'ordre supérieur (axiale ou radiale par exemple). Une fois les conditions sur le flou définies, une transformée de Fourier à fenêtre glissante est calculée pour plusieurs fréquences u choisies pour respecter les critères de la fonction d'étalement. Les coefficients ainsi obtenus sont quantifiés afin d'obtenir un mot de 8 bits.

LPQ utilise la propriété d'invariance de flou du spectre de la phase de Fourier [212]. Il est basé sur la phase quantifiée de la transformée de Fourier discrète calculée localement pour les petits patchs d'image [213].

Le flou spatial est représenté par une convolution entre l'intensité de l'image et une fonction d'étalement du point (PSF). Dans le traitement d'image numérique, le modèle discret pour le flou invariant spatialement d'une image originale $f(x)$ résultant en une image observée $g(x)$ peut être exprimé par une convolution, donnée par:

$$g(x) = (f * h)(x) \tag{4.1}$$

Où $h(x)$ est la fonction d'étalement du point PSF du flou, * représente la convolution 2-D et x est un vecteur de coordonnées $[x, y]^T$. Dans le domaine de Fourier, cela correspond à

$$G(u) = F(u)\ .\ H(u) \tag{4.2}$$

où $G(u), F(u)$ et $H(u)$ sont les transformées de Fourier discrètes (DFT) de l'image flou $g(x)$, l'image originale $f(x)$ et la PSF $h(x)$, respectivement, et u est un vecteur de coordonnées $[u, v]^T$ dans le Domaine fréquentiel. Les parties magnitude et phase peuvent être séparées de l'équation (4.2), résultant en :

$$\begin{aligned} |G(u)| &= |F(u)|\ .\ |H(u)| \quad \text{et} \\ \angle G(u) &= \angle F(u)\ .\ \angle H(u) \end{aligned} \tag{4.3}$$

Le flou PSF $h(x)$ est centralement symétrique, c'est-à-dire $h(x) = h(-x)$, sa transformée de Fourier est toujours de valeur réelle et, par conséquent, sa phase n'est qu'une fonction à deux valeurs, donnée par:

$$\angle H(u) = \begin{cases} 0 & if \quad H(u) \geq 0 \\ \pi & if \quad H(u) < 0 \end{cases} \tag{4.4}$$

Dans la méthode LPQ, on suppose que dans la bande de fréquence très basse, la valeur de $H(u)$ est positive avec $\angle H(u) = 0$, de sorte que l'information de phase de $G(u)$ et $F(u)$ est la même et, par conséquent, une représentation invariante de flou peut être obtenue à partir de la phase.

$$\angle G(u) = \angle F(u) \qquad pour\ tous \quad \angle H(u) \geq 0 \tag{4.5}$$

Dans LPQ, L'extraction de l'information de la phase est calculée dans une région $N_x$de M x M voisins pour chaque position de pixel x dans l'image $f(x)$. Ces spectres locaux sont calculés à l'aide d'une transformée de Fourier à court terme (STFT) définie par :

$$F(u,x) = \sum_{y \in N_x} f(x-y) e^{-j2\pi u^T y} \tag{4.6}$$

Où $x \in \{x_1, x_2, \dots, x_N\}$ consistent simplement par une convolution en 1-D pour les lignes et les colonnes successivement. Les coefficients locaux de Fourier sont calculés à quatre points de fréquence $u = [u_1, u_2, u_3, u_4]$, où $u_1 = [a, 0]^T$, $u_2 = [0, a]^T, u_3 = [a, a]^T$ et $u_4 = [a, -a]^T$. La valeur a est la plus haute fréquence scalaire pour laquelle $H(u_i) > 0$. Pour chaque position de pixel, il en résulte un vecteur :

$$F_x = [F(u_1, x), F(u_2, x), F(u_3, x), F(u_4, x)] \tag{4.7}$$

L'information de phase dans chaque coefficient de Fourier est enregistrée, en observant les signes des parties réelles (Re) et imaginaires (Im) de chaque composant dans F_x. Cela se fait en utilisant un quantificateur scalaire simple :

$$q_j(x) = \begin{cases} 1 & if \quad g_i(x) \geq 0 \\ 0 & if \quad otherwise \end{cases} \tag{4.8}$$

Où $g_i(x)$ représente la $j^{ième}$ composante du vecteur $G_x = [Re\{F_x\}, Im\{F_x\}]$. Les huit coefficients binaires obtenus $q_j(x)$ sont représentés par des valeurs entières entre 0 et 255 en utilisant un codage binaire simple pour obtenir les étiquettes de LPQ, f_{LPQ} est définie par :

$$f_{LPQ}(x) = \sum_{j=1}^{8} q_j(x)\ 2^{j-1} \tag{4.9}$$

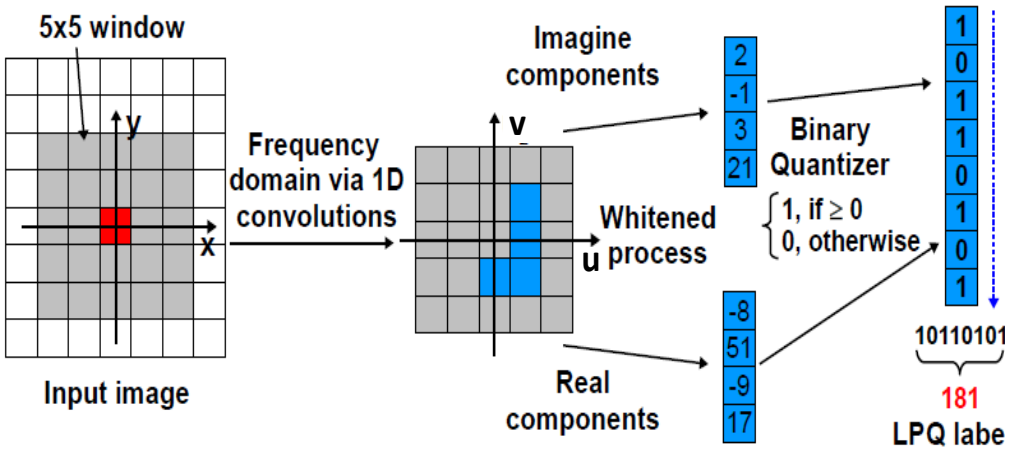


Figure 4.2 – Organigramme de l'ensemble des étapes nécessaires à la construction du descripteur LPQ

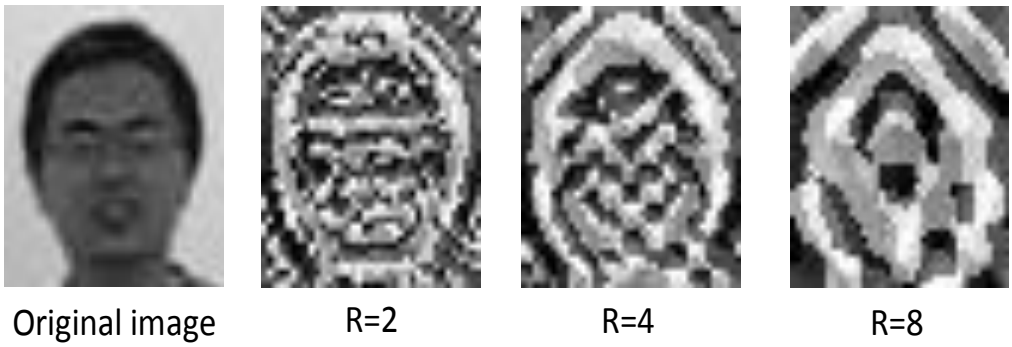


Figure 4.3 – Représentation d'une image par le descripteur LPQ sous différents voisinage de pixel.

En conséquence, nous obtenons l'étiquette d'image f_{LPQ}, dont les valeurs sont le flou invariant des étiquettes de LPQ. Le processus d'application d'un opérateur LPQ, LPQ(M, ρ) sur un pixel d'image est démontré à la figure 4.2, où ρ est le coefficient de corrélation et $M = 2 * R + 1$.

La figure 4.3 illustre une représentation de l'image LPQ par le descripteur LPQ en utilisant trois rayons ($R = 2, 4, 8$).

4.3.2 Le descripteur LBP

L'opérateur LBP (Local Binary Pattern) a été introduit par Ojala et al. [210] en 2002, dans le but d'exprimer la texture des patchs de l'image. En effet, il a d'abord été présenté en 1996 comme une mesure complémentaire du contraste de l'image locale [214]. Le Motif binaire local (LBP) a été largement appliqué avec succès en tant que méthode d'extraction de

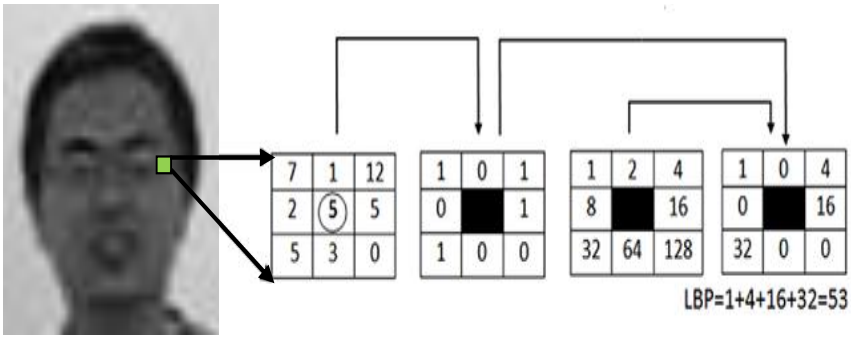


Figure 4.4 – Une illustration de LBP basique

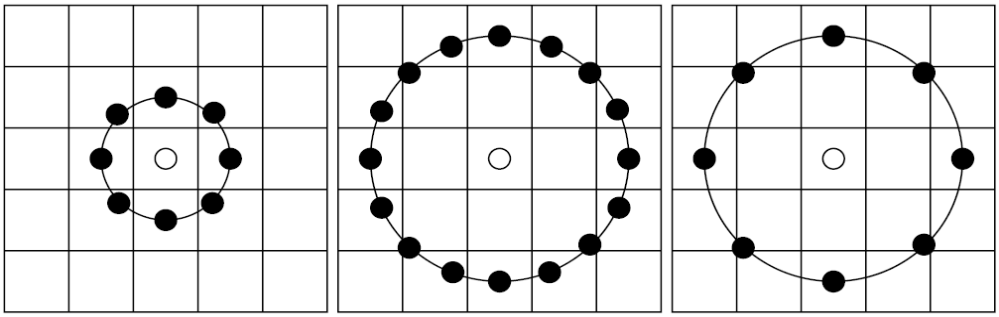
caractéristiques locales dans la reconnaissance faciale, la détection et le suivi des objets dans une séquence d'image. L'opérateur LBP est une méthode puissante de description de texture basée sur l'analyse statistique et montre son utilisation pratique dans la description de texture. Le concept du LBP basique consiste à attribuer un motif binaire pour chaque pixel de l'image à analyser. Le LBP basique fonctionne dans un bloc de 3 x 3 pixels d'une image. Les pixels de ce bloc sont seuillés par la valeur de pixel central, multipliés par des puissances de deux, puis additionnés pour obtenir une étiquette pour le pixel central. Comme le voisinage se compose de huit pixels, un total de 2^8 = 256 étiquettes différentes peuvent être obtenues en fonction des valeurs à niveaux de gris relatives du centre et des pixels dans le voisinage. La figure 4.4 illustre la procédure de calcule de LBP sur une fenêtre de taille 3x3. Le LBP a été étendu ultérieurement en utilisant de différents rayons de voisinages R et différents points d'échantillonnage P [215] ce qui permet d'extraire les caractéristiques dans différentes échelles. Le code $LBP_{P,R}$ du pixel courant est calculé comme suit :

$$LBP_{P,R} = \sum_{p=0}^{P-1} s\left(g_p - g_c\right)2^p \qquad (4.10)$$

Où g_p et g_c sont respectivement les niveaux de gris d'un pixel voisin et du pixel central, et la fonction s (x) est défini comme :

$$s(x) = \begin{cases} 1 & si \quad x \geq 0 \\ 0 & si \quad x < 0 \end{cases} \qquad (4.11)$$

Dans Le LBP étendu, un cercle de rayon R autour du pixel central est considéré. Les valeurs des P points échantillonnés sur le bord de ce cercle sont prises et comparées avec la valeur du pixel central. Pour obtenir les valeurs de tous les points échantillonnés P dans le voisinage pour tout rayon R, une interpolation est nécessaire. La notation (P, R) utilisée pour définir le voisinage de P points de rayon R d'un pixel. La figure 4.5 illustre trois voisinages pour des valeurs de R et P différentes.



(a) (b) (c)

Figure 4.5 – Exemples de d'operateur LBP P.R; (a) LBP 8.1, (b) LBP 16.2, (c) LBP 8.2.

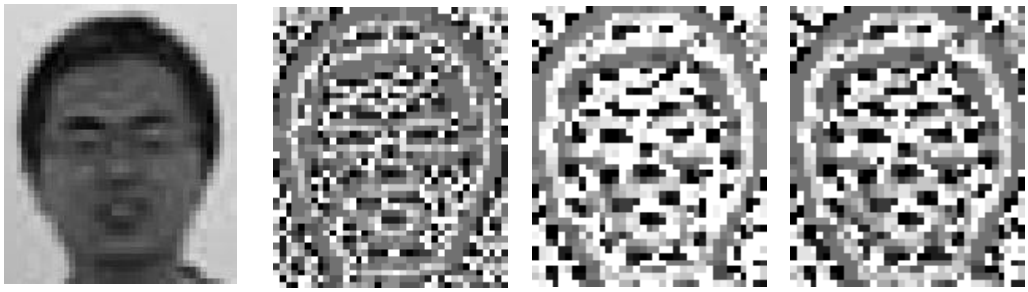


Image d'intensité $LBP_{P.R}(8,1)$ $LBP_{P.R}(16,2)$ $LBP_{P.R}(8,2)$

Figure 4.6 – Représentation d'une image avec le descripteur LBP

Le nombre total des valeurs des différentes sorties 2^P peut être généré par l'opérateur $LBP_{P.R}$ avec certaines valeurs correspondantes aux mêmes motifs suivant la rotation [216]. Dans le but d'améliorer encore la puissance discriminante du LBP et d'éliminer l'effet de la rotation, d'autres méthodes étendues sont proposées, telle que le Rotation-invariant LBP ($LBP_{P.R}^{ri}$) et l'Uniform-rotation-invariant LBP ($LBP_{P.R}^{uri}$) [216][217][214]. La figure 4.6 illustre une représentation d'image avec le descripteur LBP et différentes valeurs de *P* et *R*

4.3.3 Le descripteur BSIF

Le descripteur BSIF (Binarized Statistical Image Features) a été proposé par J. Kannala et E. Rahtu [218] en 2012 pour la reconnaissance faciale et la classification de texture. Il est inspiré par la méthodologie LBP et LPQ, BSIF calcule également un code binaire pour chaque pixel dans une image pour représenter la structure locale d'une image [219]. Contrairement à LBP et LPQ qui peuvent être utilisées pour calculer les statistiques d'étiquettes dans les voisinages des pixels locaux, BSIF utilise un ensemble prédéfini manuellement des filtres linéaires et binarisation des réponses du filtre. La valeur de chaque bit dans une chaîne de code binaire est calculée en binarisant la réponse d'un filtre linéaire avec un seuil à zéro. Chaque bit est associé à un filtre différent et la longueur désirée de la chaîne de bits détermine le nombre de

Figure 4.7 – Les 13 images naturelles utilisées pour l'apprentissage des filtres dans le descripteur BSIF.

filtres utilisés. L'ensemble de filtres sont automatiquement appris basés sur des propriétés statistiques d'un petit ensemble d'images naturelles. L'ICA (Independent Component Analysis) est utilisée pour l'apprentissage de l'ensemble des filtres linéaires en maximisant l'indépendance statistique des réponses de filtres [218]. Les filtres linéaires utilisés dans toutes les expériences de Kannala et Rahtu [218] sont tirés d'un ensemble de 13 images naturelles donné par A. Hyvärinen et al dans [220] (figure 4.7).

Compte tenu d'un patch image *X* de taille $l \times l$ pixels et un filtre linéaire *Wi* de la même taille (voir la figure 4.8), la réponse du filtre S_i est donnée par l'équation suivante :

$$S_i = \sum_{u,v} W_i(u,v)X(x,v) = w_i^T\, x \qquad (4.12)$$

Où les vecteurs w et x contiennent les pixels de Wi et X respectivement. La chaîne de code binaire b est obtenue par la binarisation de chaque élément S_i. La fonction binarisée b_i est calculée par :

$$b_i = \begin{cases} 0 & si \;\; S_i > 0 \\ 1 & ailleurs \end{cases} \qquad (4.13)$$

Etant donné *n* filtres linéaires W_i, nous pouvons les empiler sur une matrice W de taille $n \times l^2$. La longueur de la chaîne de bits n avec la taille du filtre l sont des paramètres variables pour évaluer le descripteur BSIF. Toutes les réponses sont calculées à la fois, c'est-à-dire $s = W.x$.

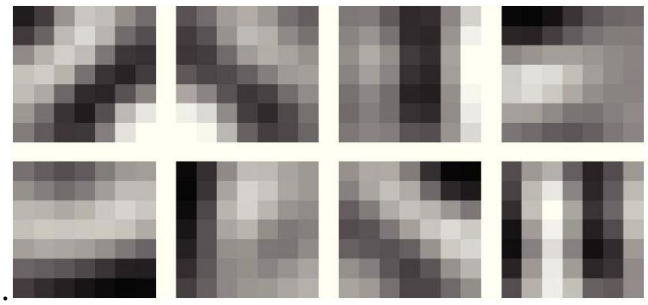


Figure 4.8 – Filtres tirés de taille *l*=7 et nombre de bits *n*=8

Afin d'obtenir un ensemble utile de filtres Wi Kannla et Rahtu [218] ont estimé les filtres en maximisant l'indépendance statistique des S_i. Pour estimer les composantes indépendantes, il faut décomposer la matrice de filtres W en deux parties par :

$$S = Wx = UVx = Uz \qquad (4.14)$$

Où Vx=z et U une matrice carrée ($n \times n$) qui sera estimée par l'algorithme ICA et la matrice V effectue le prétraitement canonique, c.à.d, la réduction de la dimension des échantillons d'apprentissage *x* [220]. La réduction de données basée sur l'algorithme PCA est utilisée pour réduire la taille de x, ne conservant que les *n* premières composantes principales qui sont divisées par leur écart-type pour obtenir les échantillons de données. Finalement, on obtient la matrice du filtre $W=UV$, qui peut être directement utilisée pour le calcul BSIF. La figure 4.9 illustre la représentation BSIF d'une image avec différentes taille du filtre (l) et différentes longueur de la chaîne de bits n.

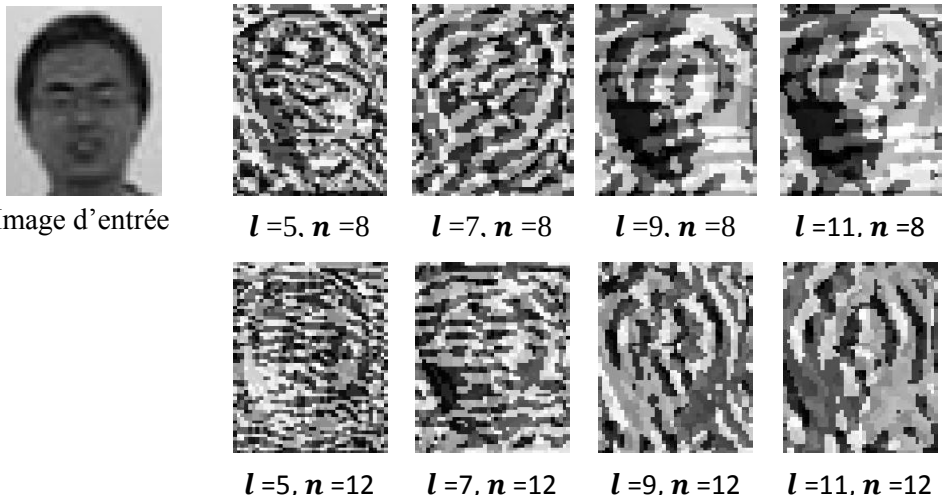


Figure 4.9 – La représentation BSIF d'une image avec différentes tailles ($\boldsymbol{l}$) du filtre et différentes longueur de la chaîne de bits $\boldsymbol{n}$.

4.4 Suivi d'objet par le tracker Mean shift avec l'histogramme conjoint couleur-texture proposé

Dans cette section, nous présentons une nouvelle méthode de la représentation de la cible en utilisant les caractéristiques de la couleur HSV et de la texture LPQ, LBP ou BSIF dans le cadre de l'algorithme Mean shift. Cet algorithme a été prouvé être robuste à une occultation partielle, au changement d'échelle, la rotation et la déformation non-rigide de la cible [1]. Cependant, l'utilisation de l'histogramme de couleur uniquement dans le tracker Mean shift présente quelques problèmes [14]. D'abord, l'information spatiale de la cible est perdue. Deuxièmement, lorsque la cible a une apparence similaire à celle de fond ou le mouvement rapide de la cible et de la caméra (flou de mouvement), l'histogramme de couleur devient invalide pour les distinguer. Pour surmonter la limitation de l'histogramme de couleur du tracker Mean shift, nous utilisons l'espace de couleur HSV au lieu de l'espace de couleur RGB, et les caractéristiques de texture LPQ, LBP ou BSIF en combinaison avec l'histogramme de couleur qui explore la propriété de flou invariant et l'information spatiale [201]. Les détails de la méthode proposée sont expliqués dans les sous-sections suivantes et nous prenons les caractéristiques de la texture LPQ comme un exemple pour illustrer l'histogramme conjoint HSVcouleur-texture.

4.4.1 Représentation de cible avec l'histogramme conjoint HSVcouleur-LPQ texture

Dans l'algorithme de suivi Mean shift traditionnel, l'histogramme de couleur RGB a été utilisé pour la représentation de la cible. Cependant, l'espace RGB n'est pas un espace de couleur perceptivement uniforme (la différence entre les couleurs de l'espace RGB ne correspond pas aux différences de couleur perçues par l'œil). De plus, l'inconvénient majeur de RGB est sa forte corrélation entre les composantes couleurs [21]. Pour améliorer les performances de l'algorithme de suivi Mean shift, nous avons appliqué l'histogramme de couleur HSV, qui permet de suivi robuste en conditions d'éclairage, car HSV possède un certain degré d'invariance contre les changements d'illumination. L'espace HSV (teinte, saturation et value) est un espace de couleur approximativement uniforme. Cet espace est conçu pour représenter la couleur de manière intuitive, en simplifiant la tâche de quantifier une couleur perçue. Ceci est obtenu en séparant les composantes de luminance (S, V) et de chrominance (H); cependant, le HSV n'est pas perceptuellement linéaire [221].

Pour modéliser l'objet cible de manière efficace, nous calculons tout d'abord la caractéristique LPQ de chaque pixel dans la région cible pour obtenir la région cible LPQ,

dont la valeur est comprise entre 0 et 255. Ensuite, nous utilisons les canaux HSV et les modèles LPQ pour représenter conjointement la cible par les caractéristiques de couleur et de texture. Pour obtenir la distribution des couleurs et des textures de la région cible, nous utilisons l'équation suivante :

$$\begin{cases} \hat{q}_{HC} = \{\hat{q}_{u_{HC}}\}_{u_{HC}=1\ldots m} \\ \hat{q}_{u_{HC}} = C\sum_{i=1}^{n} k(\|x_i^*\|^2)\,\delta[b_{HC}(x_i^*) - u_{HC}] \end{cases} \tag{4.15}$$

Où $\hat{q}_{u_{HC}}$ représente la probabilité de la caractéristique conjointe u_{HC} (couleur et texture) dans le modèle cible, $\hat{q}_{HC}$ correspond à l'histogramme conjoint HSV couleur-LPQ texture. $m = N_H \times N_S \times N_V \times N_{LPQ}$ est le nombre d'espaces des caractéristiques conjoints. Les trois dimensions $N_H \times N_S \times N_V$ (i.e. 16 x 16 x 16) représentent les classes quantifiées des canaux de couleurs HSV et la quatrième dimension N_{LPQ} (i.e. 16) est constituée des classes de la caractéristique de texture LPQ. La distribution de couleur et de texture du modèle cible $\hat{q}_{HC}$ consiste à quatre dimensions 4D (i.e. 16 x 16 x 16 x 16).

De même, la probabilité du candidat cible $\hat{p}_{u_{HC}}(y)$ est calculée par :

$$\begin{cases} \hat{p}_{HC} = \{\hat{p}_{u_{HC}}\}_{u_{HC}=1\ldots m} \\ \hat{p}_{u_{HC}}(y) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\,\delta[b_{HC}(x_i) - u_{HC}] \end{cases} \tag{4.16}$$

La figure 4.10 illustre un exemple de la distribution conjointe de la couleur HSV et de la texture LPQ du modèle cible $\hat{q}$ dans les caractéristiques $u_H = 3,\ u_S = 2, u_V = 4,\ u_{LPQ} = 1$.

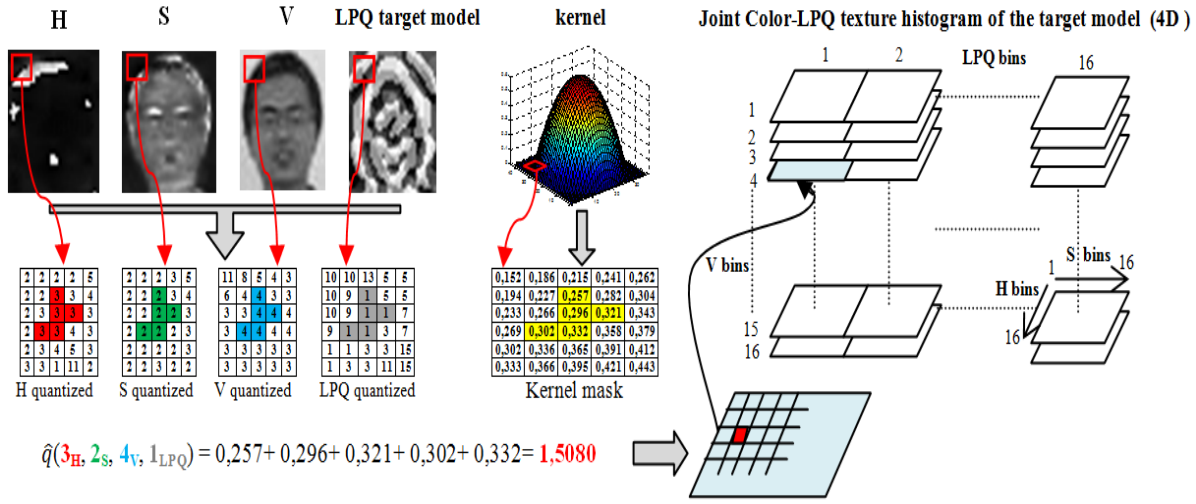


Figure 4.10 – Exemple d'un histogramme conjoint HSV couleur-LPQ texture pour représenter le modèle cible dans l'algorithme Mean shift [201].

Afin obtenir des histogrammes conjoints HSV couleur-LBP texture et HSV couleur-BSIF texture nous utilisons la méthode de la combinaison présentée dans la figure au-dessus, qui montre les étapes principales de la construction de l'histogramme conjoint HSV couleur-LPQ texture. Aussi, les équations (4.15) et (4.16) sont utilisées pour calculer la distribution de couleur HSV avec la texture LBP ou BSIF du modèle cible et du candidat cible, respectivement. Bien que, les caractéristiques de texteur LBP ont été utilisées dans la littérature pour construire l'histogramme conjoint mais notre histogramme est différent, puisque la méthode de la combinaison proposée entre les caractéristiques de couleur HSV et de texture LBP.

4.4.2 L'algorithme de suivi avec l'histogramme conjoint couleur-texture

Après la modélisation de l'objet cible à l'aide de l'histogramme conjoint HSV couleur-texture, on utilise la distance de Bhattacharyya afin de mesurer la similarité entre l'histogramme du modèle et les histogrammes des régions candidates (sélectionnées autour de la dernière position connue de l'objet à suivre). Cette opération est répétée jusqu'à ce que la valeur de similarité ne dépasse pas un seuil ou que le nombre limite d'itérations atteint (de quatre à six itérations en général). La figure 4.11 illustre la procédure de l'algorithme de suivi Mean shift avec l'histogramme conjoint HSV couleur-texture en utilisant les caractéristiques de texture LPQ. Ainsi, le processus de construction de l'histogramme conjoint HSV couleur-LPQ texture proposé est décrit dans la figure 4.12. Le même processus a été utilisé pour les histogrammes conjoints HSV couleur-LBP texture et HSV couleur-BSIF texture en remplaçant les caractéristiques de texture LPQ par LBP et BSIF.

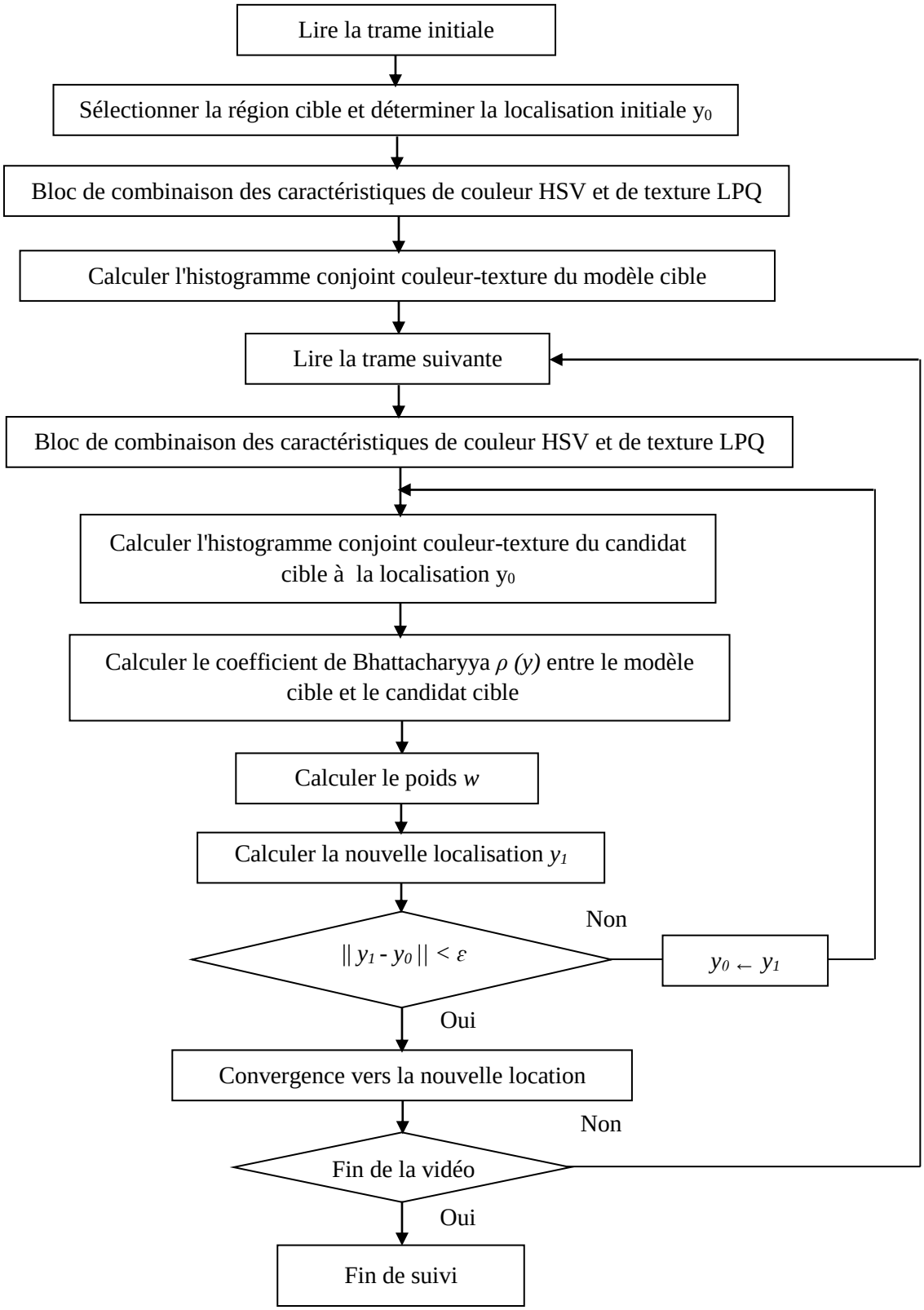


Figure 4.11 – Organigramme de l'algorithme de suivi Mean shift avec l'histogramme conjoint HSV couleur- LPQ texture.

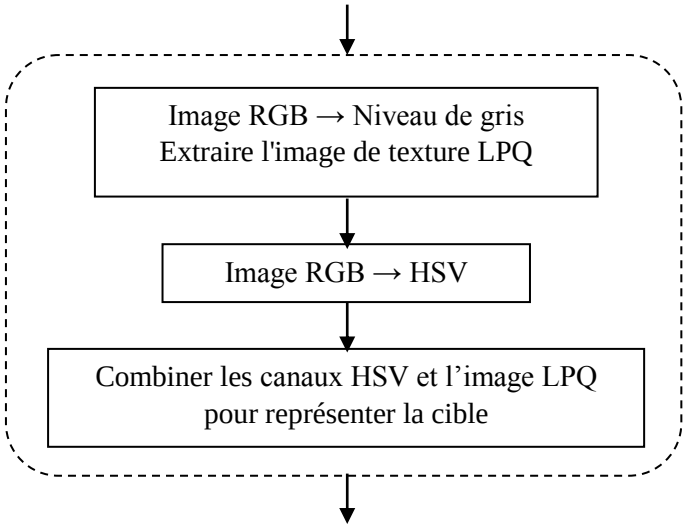


Figure 4.12 – Bloc de combinaison des caractéristiques de couleur HSV et de texture LPQ.

4.5 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode efficace et robuste de suivi d'objets qui utilise l'histogramme conjoint HSV couleur-texture, pour représenter l'objet cible, et en l'appliquant au cadre de l'algorithme de suivi Mean shift. Les descripteurs LPQ, LBP et BSIF ont été utilisés pour représenter les informations structurelles de l'objet, puisqu'ils plus discriminants et insensibles au flou. Comparé à l'algorithme traditionnel Mean shift qui ne considère que les informations statistiques de couleur de l'objet, les représentations proposées de l'objet cible (Histogramme conjoint HSV-LPQ, HSV-LBP et HSV-BSIF) sont utilisées pour distinguer efficacement la cible et son fond. Ces représentations permettent d'obtenir de meilleures performances de suivi avec moins d'itérations et plus de robustesse à différentes interférences du fond et au flou de mouvement. De plus, la méthode de combinaison des caractéristiques de couleur et de texture est très simple et très rapide.

Chapitre 5

Résultats Expérimentaux et Evaluation des Performances

Sommaire

5.1 Introduction

D'après les connaissances théoriques apportées dans les chapitres précédents, du principe de tracker Mean shift, l'influence des espaces couleurs sur les performances de ce tracker et l'histogramme conjoint couleur-texture proposé qui utilise pour représenter l'objet cible au lieu de l'histogramme de couleur. Dans ce chapitre, nous allons présenter et évaluer les différents résultats de tracker Mean shift avec différents espaces couleurs d'une part, et d'autre part, avec l'histogramme conjoint couleur-texture proposé. Ensuite, nous comparons l'efficacité de la méthode proposée avec des méthodes de l'état de l'art. L'étude expérimentale de suivi d'objets sur les bases de données OTB et VOT2013 sont réalisées afin de valider ce travail.

5.2 Bases de données pour le suivi d'objet

Les bases de données jouent un rôle critique dans presque toutes les tâches de vision par ordinateur. Dans le cas du problème de classification d'objets, il y a eu une évolution considérable de Caltech101 [222] à PASCAL COV [223] puis à ImageNet à grande échelle [224]. Bien qu'une telle évolution se soit également produite dans le cas du suivi d'objets, elle s'est faite à plus petite échelle et à un rythme plus lent. Jusque dans les années 2010, un tracker était expérimentalement évalué sur un nombre restreint de vidéos (quelques vidéos choisies par l'auteur) et selon des métriques d'évaluation propres à l'auteur. Une telle évaluation est insuffisante pour mesurer les forces et les faiblesses de chacun des trackers pour les nombreux phénomènes existants en suivi d'objet (illumination, occultation, variations d'apparence, etc.) [225]. La plupart des séquences vidéo dans les ensembles de données initiales ont été enregistrées dans un environnement expérimental non-naturel ou dans certains cas sélectionnées, pour mettre en évidence les avantages du tracker proposé. De plus, ils n'ont pas de protocole commun pour l'annotation de vérité de terrain, et sont généralement peu nombreux. Il existe plusieurs bases connues, créées dans le cadre de la vidéo-surveillance et de la détection d'événements telles que VIVID[1] [226], CAVIAR[2] et PETS [227], mais les catégories d'objets d'intérêt sont assez restreintes (piétons, véhicules) et l'arrière-plan est statique. Cependant, ces bases ne sont pas suffisamment génériques (catégories d'objet peu variées) et représentatives des difficultés qu'il est possible de rencontrer en suivi d'objet [225]. Ces problèmes sont traités par des bases de données récentes et benchmarks. Les bases récentes sont celles collectées par [23] (base OTB[3]), [228] (base ALOV++[4]), [229],[230][102] (base VOT[5]) visant à couvrir un grand nombre de situations possibles. Dans cette thèse, nous avons utilisé deux bases de vidéos OTB et VOT, qui sont les plus populaires, pour évaluer les performances de suivi des trackers. Ces bases présentent des objets et des scènes variées soumis à différentes perturbations (mouvement de caméra, zoom, changements d'illumination, occultations, objets déformables, changements d'apparence rapides, mouvements d'objet, etc.). Nous discutons ces bases dans ce qui suit.

[1] http://vision.cse.psu.edu/data/vividEval/datasets/datasets.html
[2] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/
[3] http://www.visual-tracking.net
[4] http://www.alov300.org.
[5] http://www.votchallenge.net/

5.2.1 La base OTB (Objet Tracking Benchmark)

La base OTB est la base de données la plus couramment utilisée dans la littérature, qui comprit OTB2013 [23] et OTB2015 [231]. OTB2013 contient 50 séquences entièrement annotées qui sont collectées à partir de séquences de suivi couramment utilisées. OTB2015 est l'extension d'OTB2013 et contient 100 séquences vidéo. Certaines nouvelles séquences sont plus difficiles à suivre. Les séquences d'OTB sont des séquences annotées suivant 11 difficultés, dont la variation d'illumination (IV), la variation d'échelle (SV), l'occultation (OCC), déformation (DEF), flou de mouvement (MB), mouvement rapide (FM), rotation dans le plan (IPR), rotation hors plan (OPR), hors-vue (OV), clutter de fond (BC) et basse résolution (LR). Ces séquences font partie de celles habituellement utilisées en suivi d'objet. La première trame de chaque séquence dans OTB2013 et OTB2015 est illustrée sur la figure 5.1 et 5.2, respectivement. Afin de présenter la progression des algorithmes de suivi et de définir un benchmark général, 29 trackers sont comparées dans [23][231]. Deux méthodologies d'évaluation bien adoptées sont utilisées: la précision et le succès [132]. La précision reflète l'erreur de localisation du centre. Elle est mesurée comme le pourcentage de trames dont la localisation prédite de l'objet (centre de la boîte prédite) se trouve à une distance variant entre 0 et 50 pixels du centre de la boîte de vérité terrain. Le score de précision est le pourcentage lorsque la distance de seuil est fixée à 20 pixels. La mesure du succès est basée sur le chevauchement de la boîte englobante. Elle montre le pourcentage de trames dont l'intersection sur le chevauchement d'union avec l'annotation de vérité de terrain est sur un seuil, variant entre 0 et 1.

Figure 5.1 – Séquences de la base OTB2013 [23]. Les images correspondent à la première trame de chaque séquence avec l'objet d'intérêt détouré par une boîte englobante rouge.

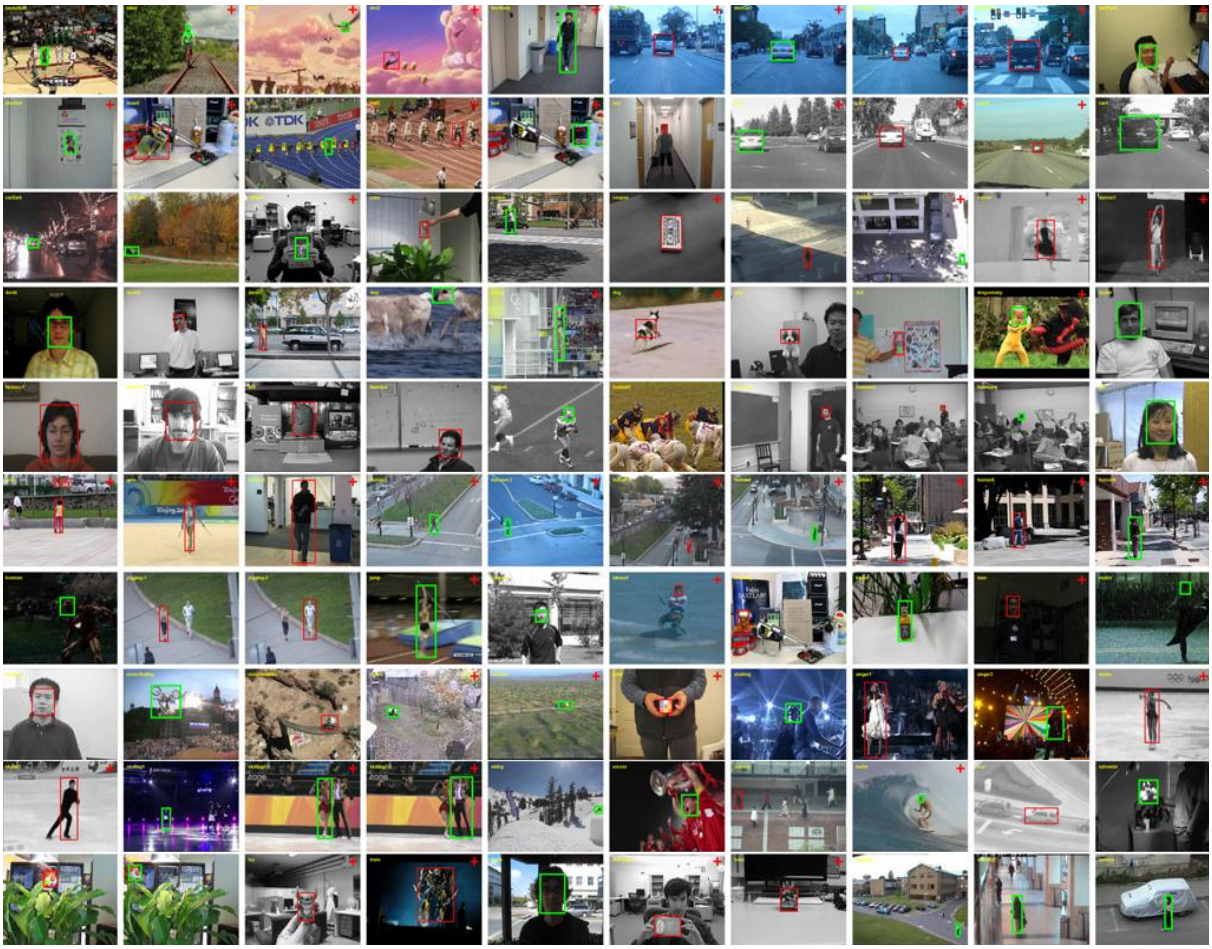


Figure 5.2 – Séquences de la base OTB2015 [231]. Les images correspondent à la première trame de chaque séquence avec l'objet d'intérêt détouré par une boîte englobante. Les 50 cibles marquées par des boîtes englobantes vertes sont sélectionnées pour des évaluations approfondies. Les séquences nouvellement ajoutées par rapport à OTB2013 [23] sont indiquées par une croix rouge dans le coin supérieur droit de chaque image.

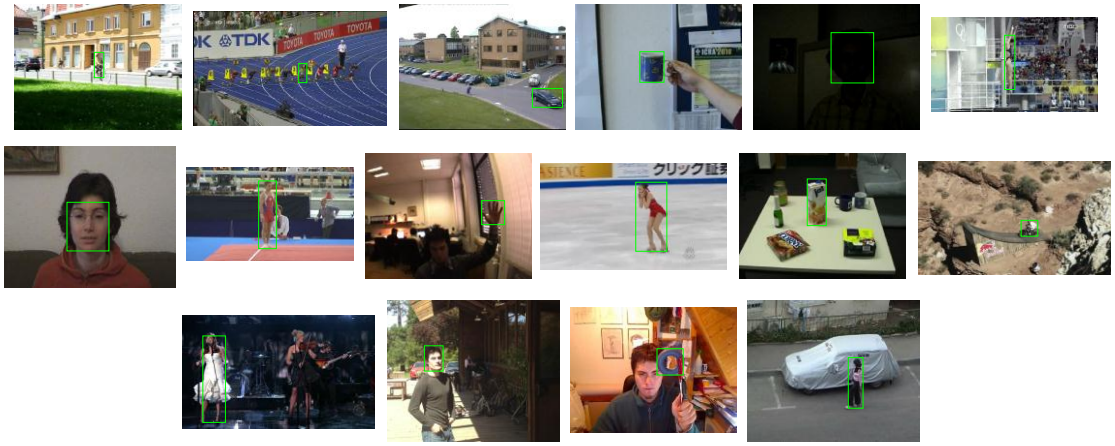


Figure 5.3 – Séquences de la base VOT2013 [229]. Les images correspondent à la première trame de chaque séquence avec l'objet d'intérêt détouré par une boîte englobante verte.

5.2.2 La base VOT (Visual object tracking (VOT) challenge)

Le benchmark VOT a été introduit en 2013 dans le but de fournir une plate-forme standardisée pour évaluer une seule caméra, une seule cible, sans modèle, algorithmes de suivi causal à court terme. Il est devenu la référence en suivi d'objet. C'est sur ce dernier que les trackers actuels s'évaluent et se comparent. Depuis, une nouvelle édition est organisée chaque année [230][102] et s'étend au suivi d'objet dans des images infra-rouge (VOT-TIR2015).

Le schéma d'évaluation du challenge VOT utilise des mesures de précision et de robustesse pour comparer les trackers, en raison de leur haut niveau d'interprétabilité [233]. La précision brute est calculée comme la moyenne de l'intersection sur le score d'union entre la boîte prédite et de vérité de terrain, sur toute la séquence ou sur une base de vidéos (tout en supprimant dix trames immédiatement après une dérive de tracker pour réduire davantage le biais dans la mesure de la précision), et la robustesse brute est le nombre de fois le suivi a dérivé. Une dérive de tracker est signalée dans une trame t si la boîte prédite ne chevauche pas avec la boîte de vérité de terrain.

Les bases d'évaluation de VOT2013 [229] (16 vidéos) et VOT2014 [230] (25 vidéos) se composent de séquences sélectionnées à partir d'OTB et ALOV++ de façon semi-automatique selon les phénomènes présents (occultation, changement d'illumination, changement de taille, mouvement objet, mouvement caméra). Tandis que, VOT2015 [102] se compose de 60 vidéos difficiles qui sont automatiquement sélectionnés à partir des bases OTB, ALOV++, PTR [166], et 30 autres séquences annotées selon 11 attributs globaux (figure 5.3). La base de données VOT2016 [234] est constituée des mêmes vidéos de VOT2015, mais la vérité de terrain a été ré-annotée (plus précise).

5.3 Métriques de performances

L'évaluation des performances d'un algorithme de suivi (figure 5.4) n'est pas une tâche facile. En fait, plusieurs facteurs peuvent limiter la validité des résultats tels que les métriques de performances choisies, le choix des seuils d'évaluation. Par ailleurs, afin de comparer un algorithme de suivi avec d'autres trackers de la littérature, il faut utiliser les mêmes métriques et les mêmes paramètres de mesure. Il existe de nombreuses métriques de performance en suivi d'objet proposées dans la littérature, et détaillées dans [23][228][233]. Dans le cadre de cette thèse, on a choisi d'utiliser les métriques les plus populaires et qui sont largement utilisées dans la littérature. Comme toutes ces métriques supposent que les annotations manuelles sont données pour une séquence. La définition générale d'une description d'état d'objet dans une séquence de longueur N, est donnée par [233] :

$$\Lambda = \{(A_t, x_t)\}_{t=1}^{N} \tag{5.1}$$

Où $x_t \in R^2$ dénote un centre de la boîte de l'objet et A_t dénote la boîte de l'objet à l'instant t. Les métriques de performance visent à résumer l'étendue à laquelle l'annotation prédite du tracker, Λ_P, est en accord avec l'annotation de vérité de terrain Λ_G. Ces métriques sont décrites ci-dessous.

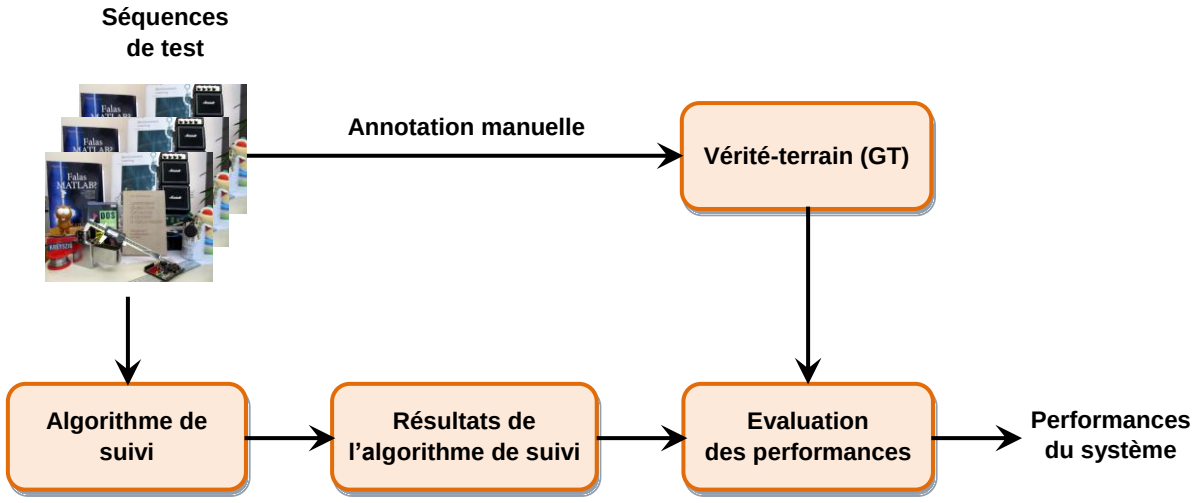


Figure 5.4 – Vue d'ensemble d'un processus d'évaluation des performances d'un système de suivi.

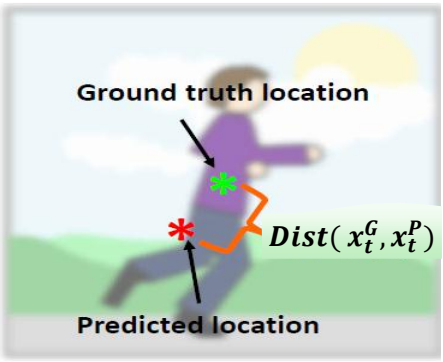


Figure 5.5 – Une illustration de l'erreur de localisation du centre entre les centres prédite et la vérité terrain.

5.3.1 Erreur de localisation du centre CLE

L'erreur de localisation du centre (center location error (CLE)) est le moyen le plus ancien de mesurer la performance, qui a ses racines dans l'aéronautique [233]. Il est une mesure courante [2][4][34][110] consistant à mesurer la distance moyenne entre les centres des boîtes prédites $\{x_t^P\}_{t=1}^N$ et de la vérité terrain $\{x_t^G\}_{t=1}^N$, donnée par l'équation (5.2) [233]. Cette mesure ne rend pas compte de la précision en taille des boîtes prédites.

$$\Delta_{\mu}(\Lambda^{\mathrm{G}}, \Lambda^{\mathrm{P}}) = \frac{1}{N}\sum_{t=1}^{N}\|x_t^G - x_t^P\| \tag{5.2}$$

La popularité de la mesure de centre prédite provient de son effort d'annotation minimale, c'est-à-dire, un seul point par trame. La figure 5.5 illustre la métrique de l'erreur de localisation du centre.

5.3.2 Précision selon un seuil sur l'erreur de localisation

Récemment, la métrique de la précision selon un seuil sur l'erreur de localisation (*Precision plots*) a été adoptée pour mesurer la performance globale du suivi [23]. Elle mesure la proportion d'images, entre [0, 1], pour lesquelles la distance entre les centres de la boîte prédite et de la vérité terrain est inférieure à un seuil en nombre de pixels. Une courbe de proportion d'images en fonction du seuil sur l'erreur de localisation peut être calculée (voir la figure 5.6(a)). Le seuil habituellement utilisé pour comparer la précision entre différents trackers est de 20 pixels [225].

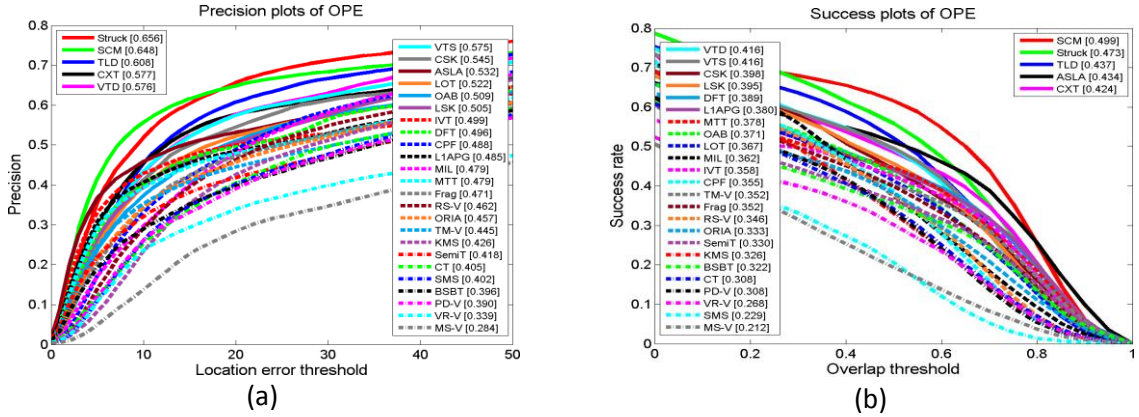


Figure 5.6 – Métriques d'évaluation. (a) Courbe de proportion d'images en fonction du seuil sur l'erreur de localisation pour différents trackers. (b) Courbe de proportion d'images en fonction du seuil sur le taux de recouvrement pour différents trackers [23].

5.3.3 Taux de recouvrement moyen VOR

Une autre métrique d'évaluation est le taux de recouvrement (overlap ratio (VOR)) entre la boîte prédite A_t^P et la vérité terrain A_t^G. Il est défini comme étant le rapport des aires d'intersection et d'union des boîtes, donné par [233] :

$$\Phi\left(\Lambda^{\mathrm{G}}, \Lambda^{\mathrm{P}}\right) = \{\phi_t\}_{t=1}^{N} \quad , \quad \phi_t = \frac{A_t^G \cap A_t^P}{A_t^G \cup A_t^P} \tag{5.3}$$

La figure 5.7 illustre le taux de recouvrement ϕ_t. Ce taux est une mesure d'erreur plus précise que l'erreur de localisation du centre puisqu'il tient compte de la taille des boîtes. Pour mesurer la performance sur une séquence de trames, nous comptons le nombre de trames réussies dont le taux de recouvrement ϕ_t est supérieur au seuil donné τ (Le seuil est souvent utilisé pour l'évaluation des performances de suivi est 0,5) [23]. Pour rendre le score final plus comparable entre les différentes séquences, le nombre de trames correctement suivies est divisé par le nombre total de trames :

$$\mathrm{P}_{\tau}\left(\Lambda^{\mathrm{G}}, \Lambda^{\mathrm{P}}\right) = \frac{\|\{t|\phi_t > \tau\}_{t=1}^{N}\|}{N} \tag{5.4}$$

Où τ dénote le seuil du recouvrement.

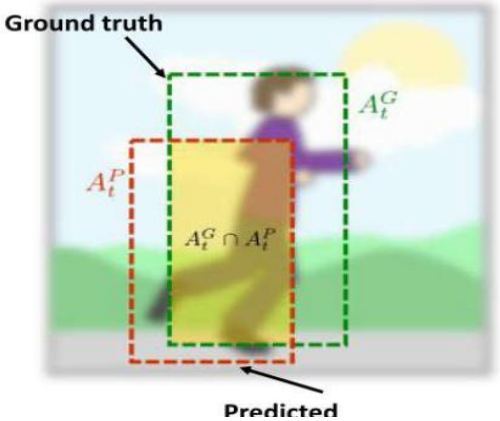


Figure 5.7 – Une illustration du recouvrement des boîtes prédite (rouge) avec de vérité terrain (vert).

5.3.4 Précision selon un seuil sur le taux de recouvrement

Une autre mesure largement utilisée par les trackers actuels trace une courbe de proportion d'images en fonction du seuil sur le taux de recouvrement (*Success plots*) compris entre [0, 1] [23], Chaque valeur du taux de recouvrement correspond à la proportion d'images de la séquence ayant un taux de recouvrement avec la vérité terrain, inférieur à cette valeur. De cette courbe, on tire une valeur représentative du comportement du tracker qui est l'aire sous la courbe (Area Under Curve) (voir la figure 5.6 (b)). Cette métrique est souvent utilisée conjointement avec la précision selon un seuil sur l'erreur de localisation [225].

5.4 Présentation des résultats

Dans cette section, nous présentons les détails de mise en œuvre du tracker Mean shift (MS) et les améliorations sur ce tracker (Camshift, Kalman et Mean shift (KaMS)). Ensuite, l'application de différents espaces de couleurs sur le tracker Mean shift pour calculer l'histogramme de couleur de l'objet cible. Enfin, les algorithmes proposés qui utilisent les histogrammes conjoints couleur-texture (HSV-LPQ, HSV-LBP et HSV-BSIF) pour modéliser l'objet cible. Pour évaluer l'efficacité du tracker Mean shift, nous avons utilisé l'histogramme de couleur HSV, le nombre d'itération égal à 20 et le seuil d'erreur de position ζ égal à 0,5. La figure 5.8 illustre l'organigramme de suivi d'objet par le tracker Mean shift. Notre système de suivi est appliqué sur les bases de données OTB et VOT2013. Les résultats expérimentaux obtenus sont exprimés sur deux plans : quantitatif (erreur de localisation du centre et taux de recouvrement) et qualitatif (visuel).

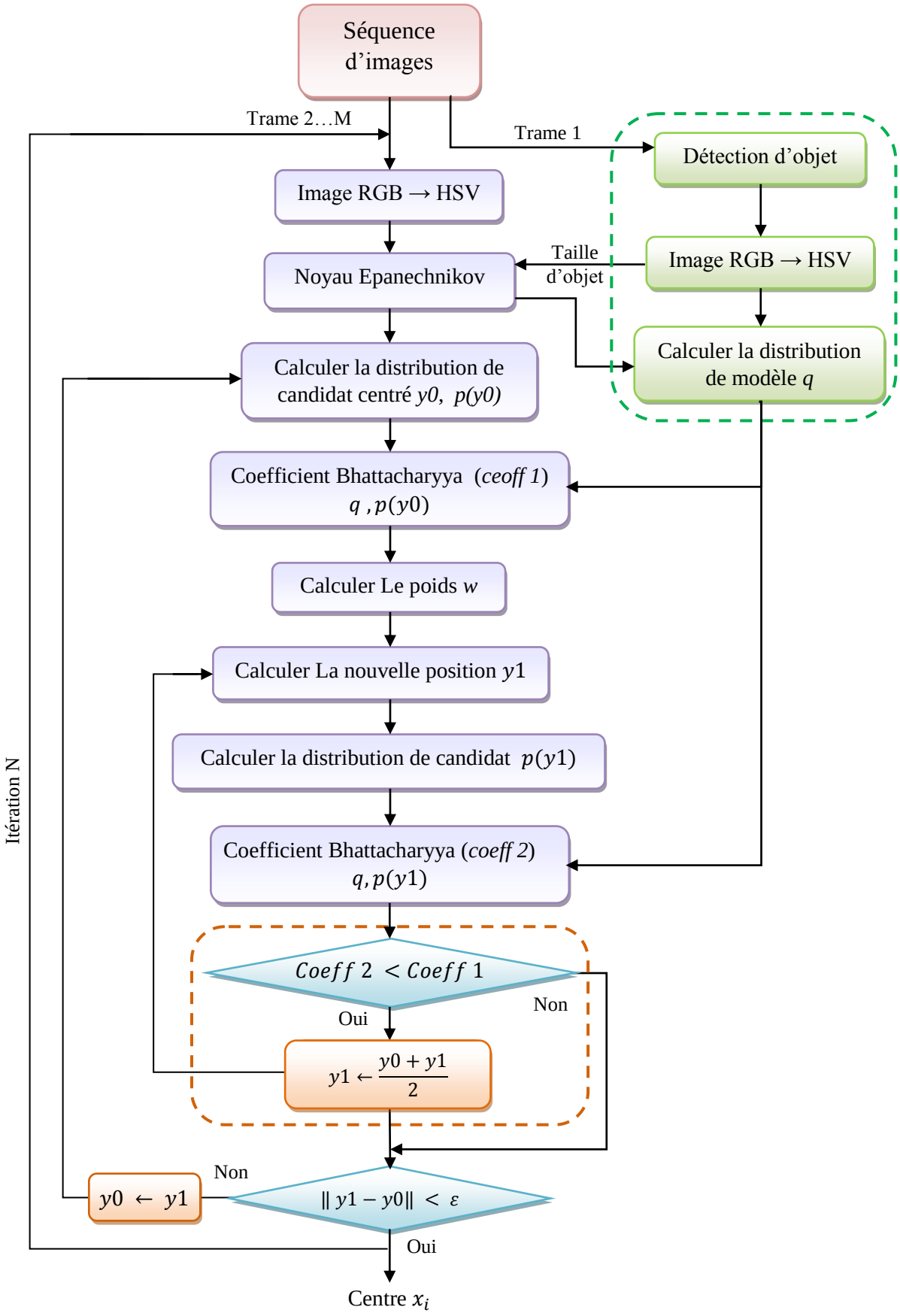


Figure 5.8 – Organigramme du suivi d'objet par le tracker Mean shift.

5.4.1 Comparaison du tracker Mean shift avec Camshift et KaMS

Le tracker Mean shift est largement utilisé dans le temps réel, plus robuste à l'occultation partielle, la rotation, le mouvement de fond et la déformation non-rigide de la cible. Cependant, ce tracker moins robuste au changement d'échelle de l'objet cible ainsi que, il dérive lorsqu'il y a une occultation complexe. Comme nous l'avons vu dans le chapitre 2, les trackers améliorés Camshift et KaMS (Kalman combiné avec Mean shift) ont résolu les problèmes du changement d'échelle et d'occultation complexe, respectivement. Dans cette section, nous présentons une étude comparative entre le tracker Mean shift et les trackers améliorés Camshift et KaMS, pour démontrer l'efficacité de ces trackers. Pour évaluer les performances, nous avons utilisé des séquences de bases de données OTB qui contiennent la plupart des attributs de changement d'échelle et d'occultation complexe.

- **Résultats quantitatifs**

Les tableaux 5.1 et 5.2 résument la comparaison entre le tracker Mean shift et les trackers Camshift et KaMS, en utilisant les métriques CLE et VOR pour chaque séquence testée. À partir du tableau 5.1, nous pouvons clairement voir que l'erreur de localisation du centre minimum et le taux de recouvrement maximal dans toutes les séquences testées sont obtenus par Camshift (19.721 / 0.5714), en particulier dans la séquence CarScale (8.6812 / 0.7490). Cela signifie que, Camshift a atteint de meilleures performances par rapport au Mean shift. Dans le tableau 5.2, l'CLE minimum et le VOR maximal dans toutes les séquences testées montrent que KaMS a les meilleures performances moyennes (16.339 / 0.6155) par rapport au Mean shift, en particulier dans la séquence Jogging (20.957 / 0.5097).

Tableau 5.1 – Les moyens du taux de recouvrement (VOR) et de l'erreur de localisation du centre (CLE) pour les deux trackers Mean shift et Camshift. Le chiffre en gras indique la meilleure performance.

Sequence	Mean shift	Camshift
Walking2	0.3006 / 37.881	**0.3314 / 37.382**
Lemming	**0.6397** /13.949	0.6340 / **13.161**
CarScale	0.3885 / 9.0093	**0.7490 / 8.6219**
Moyenne	0.4429 / 20.279	**0.5714 / 19.721**

Tableau 5.2 – Les moyens du taux de recouvrement (VOR) et de l'erreur de localisation du centre (CLE) pour les deux trackers Mean shift et KaMS. Le chiffre en gras indique la meilleure performance.

Sequence	Mean shift	KaMS
David3	0.6721 / 11.898	**0.6851 / 10.785**
Jogging	0.1589 / 72.485	**0.5097 / 20.957**
Girl2	0.5338 / 47.013	**0.6517 / 17.276**
Moyenne	0.4549 / 43.799	**0.6155 / 16.339**

L'erreur de localisation du centre (CLE) (section 5.3.1) et le taux de recouvrement (VOR) (section 5.3.3) au fil du temps (trame par trame) sont utilisés comme des critères quantitatifs, comme représentés dans les figures 5.9 et 5.10. La figure 5.9 représente les résultats obtenus par les trackers Mean shift et Camshift sur les séquences CarScale, Lemming et Walking2. La figure 5.9 (a) montre que les courbes (CLE) des deux trackers sont presque identiques, car la procédure de la localisation du nouvel centre pour Camshift et Mean shift est la même. On peut voir que, l'erreur de localisation centre est inférieure à 20 pixels sur la plupart des trames des séquences utilisées, excepté dans Walking2 lorsque l'occultation complète s'est produite (à partir de trame 190), le CLE arrive à 100 pixels. À partir de la figure 5.9 (b), nous pouvons clairement voir que le taux de recouvrement par Camshift est le plus fort (supérieur à 0.5) sur la plupart des trames des séquences. Spécifiquement, dans CarScale, le VOR du Camshift est supérieur à Mean shift, car la taille de l'objet cible change considérablement au fil du temps. Cependant, dans Walking2 après l'occultation complète de l'objet cible le VOR diminue jusqu'à 0.

La figure 5.10 représente les résultats obtenus par les trackers Mean shift et KaMS sur les séquences Davide3, Jogging et Girl2. On remarque que, dans la séquence David3 l'erreur de localisation du centre (CLE) et le taux de recouvrement (VOR) sont presque identiques, sauf dans quelques trames où il y a une occultation partielle. Dans la séquence Jogging, après la trame 71 le CLE du KaMS est inférieur à Mean shift et le VOR du KaMS supérieur à 0.5 sur la plupart des trames, mais le VOR du Mean shift égal à 0 pour toutes les trames. Cela indique que l'objet cible est occultée complètement. Dans Girl2, le CLE du KaMS est inférieur à 20 pixels et le VOR est supérieur à 0.5, sur la plupart des trames. Cependant, les résultats obtenus (CLE et VOR) par Mean shift sont presque identiques, sauf dans les trames 114 à 322 le CLE est supérieur à 100 pixels et le VOR égal à 0 car il existe une occultation totale. Cela signifie que l'objet à suivre par Mean shift est perdu.

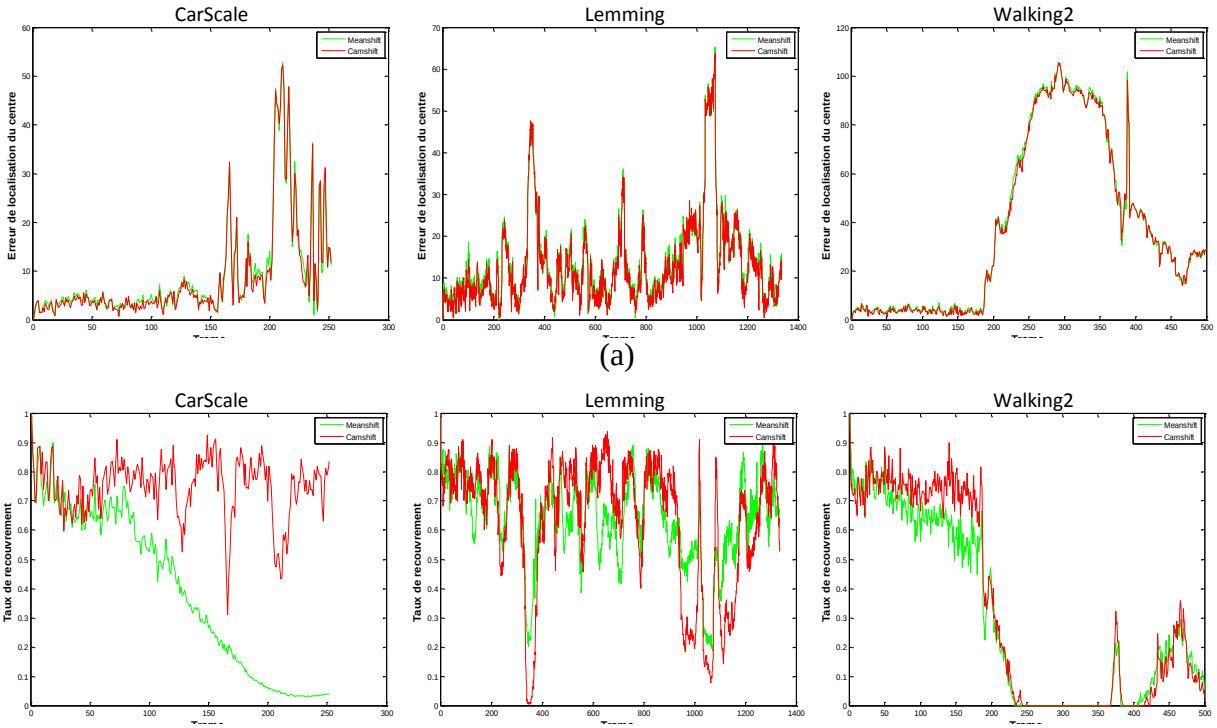


Figure 5.9 – Comparaison quantitative entre Meanshift et Camshift sur les séquences CarScale, Lemming et Walking2. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

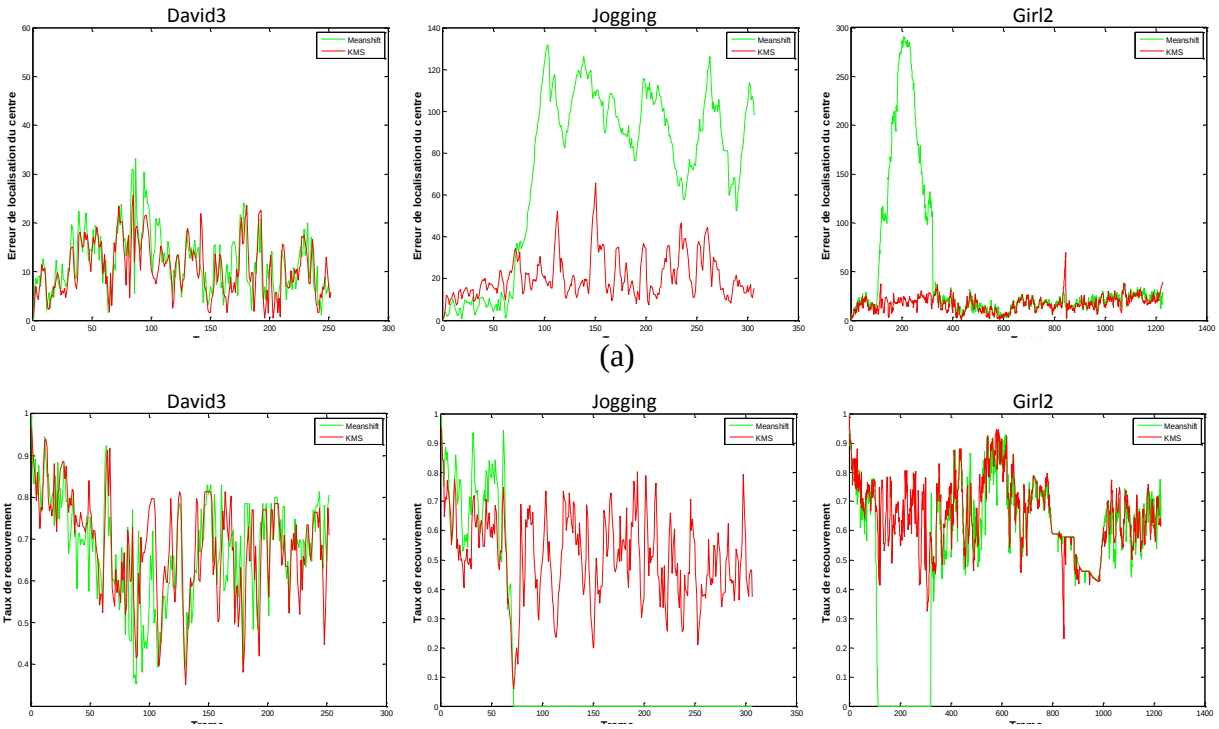


Figure 5.10 – Comparaison quantitative entre Meanshift et KaMS sur les séquences David3, Jogging et Girl2. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

- **Résultats qualitatifs**

La figure 5.11, représente les résultats visuels de suivi par les trackers Mean shift et Camshift sur des séquences vidéo CarScale, Lemming et Walking2. Nous avons utilisé l'histogramme de ratio (section 2.5.1.2.c) pour représenter le modèle de la cible dans Camshift, parcequ'il rend Camshift plus robuste. Les résultats obtenus pour les trois séquences utilisées montrent que Camshift donne de meilleures performances par rapport Mean shift, en particulier dans la séquence CarScale (voir figure 5.11 (a)), car son attribut de changement d'échelle est bien évident. Camshift permet d'adapter la taille de la zone d'objet pendant le suivi, c-à-d, il réalise une estimation d'échelle de l'objet à chaque instant, son utilisation permet d'améliorer la précision de suivi. Tandis que, Mean shift fonctionne à échelle fixe, comme illustré dans les trames de chaque séquence présentées dans cette figure. Dans la figure 5.11 (a), (b), on peut voir que les trackers Mean shift et Camshift arrivent à gérer l'occultation partielle dans CarScale (trame 169) et Lemming (trame 338). Cependant, Camshift estime de manière adaptative la taille de la fenêtre de l'objet cible avec précision, contrairement à Meanshift. Dans la figure 5.11 (c), les trackers Mean shift et Camshift dérivent en même temps à partir de la trame 201, car ils ne sont pas capables de suivre l'objet cible lorsqu'il y a un autre objet similaire. En conséquence, Camshift traite le problème du Mean shift qu'est le changement d'échelle de l'objet. Cependant, Mean shift et Camshift dérivent lorsqu'il existe une occultation totale et sont moins robustes à la similarité entre l'objet et le fond ou objets similaires.

La figure 5.12, représente les résultats visuels de suivi par les trackers Mean shift et KaMS sur des séquences vidéo David3, Jogging et Girl2. Dans la séquence David3 (figure 5.12(a)) les trackers Mean shift et KaMS peuvent suivre l'objet dans la plupart des trames. Cependant, ils sont moins efficaces pendant une occultation partielle de l'objet (trames 84, 188), malgré le tracker Mean shift est robuste dans ce cas. La cause de dérives de ces trackers à l'occultation partielle dans cette séquence est l'existence d'un autre attribut qui est la similarité entre l'objet et le fond. À partir des résultats présentés à la figure 5.12 (b) et (c), les trackers Mean shift et KaMS peuvent suivre l'objet cible de manière efficace jusqu'aux trames 73 et 111 dans les séquences Jogging et Girl2, respectivement. Dans ces trames, les objets à suivre sont occultés complètement (occultation totale). Lorsque les objets cibles réapparaissent, KaMS reprend le suivi parce que le filtre de Kalman sert à faire l'association temporelle entre deux images (les trames 80 et 119 dans les séquences Jogging et Girl2, respectivement). Contrairement, le tracker Mean shift dérive vers le fond en raison de mises à jour incorrectes. Dans la séquence Girl2 (figure 5.12(c)), le tracker Mean shift peut récupérer

la cible par hasard, en croisant la trajectoire de la cible. A la fin de la séquence, Mean shift et KaMS sont moins robustes, car il existe un changement d'échelle de l'objet (trame 985). En conséquence, le filtre de kalman peut améliorer la robustesse du Mean shift à l'occultation totale, puisque ce filtre prend les informations sur l'état de l'objet à ce moment-la. Puis, il utilise ces informations pour prédire où se trouve l'objet dans la prochaine trame. Cependant, KaMS est moins robuste au changement d'échelle d'objet et à la similarité entre l'objet et le fond ou des objets similaires.

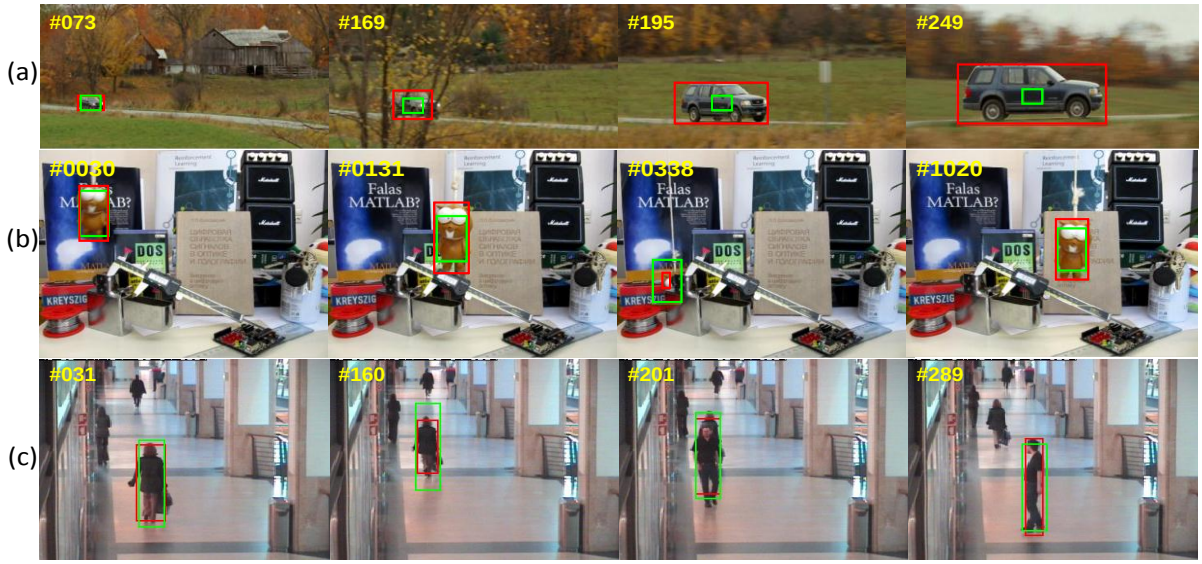


Figure 5.11 – Résultats de suivi des trackers Mean shift (rectangle vert) et Camshift (rectangle rouge) sur : (a) CarScale, (b) Lemming, (c) Walking2. Les index des trames sont indiqués en haut à gauche de chaque trame.

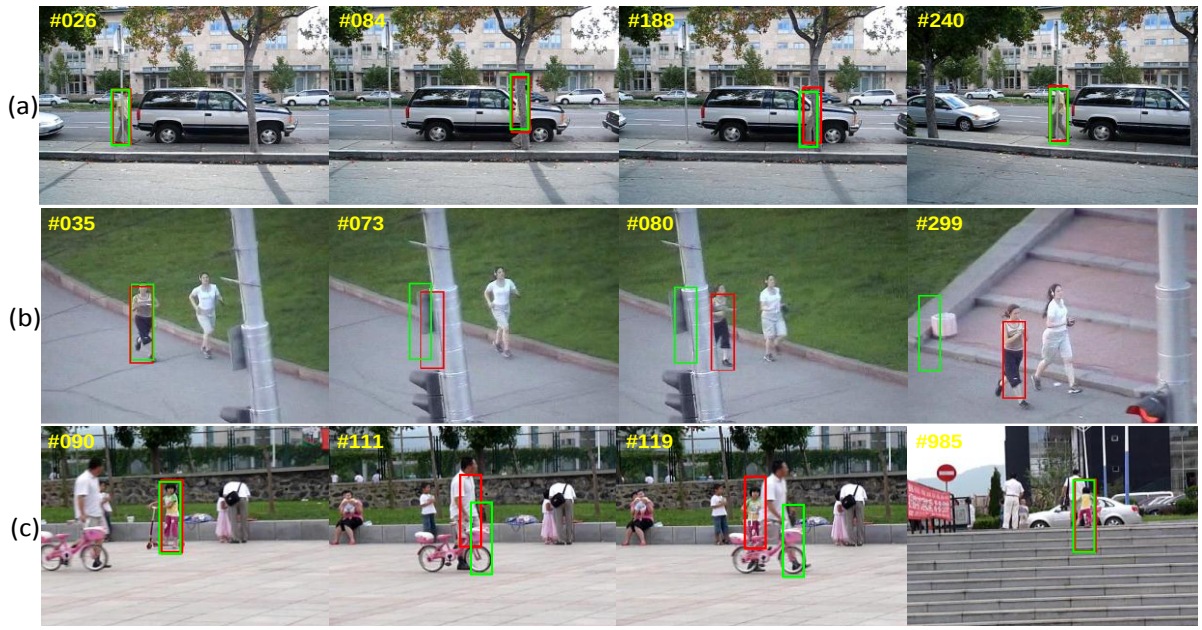


Figure 5.12 – Résultats de suivi des trackers Mean shift (rectangle vert) et MSK (rectangle rouge) sur : (a) David3, (b) Jogging, (c) Girl2. Les index des trames sont indiqués en haut à gauche de chaque trame.

5.4.2 Influence des espaces de couleurs sur les performances de tracker Mean shift

Dans cette section, nous présentons les résultats du suivi par le tracker Mean shift en utilisant les espaces de couleurs les plus utilisés dans la vision par ordinateur : HSV, RGB, YCrCb, YIQ, YUV, I1I2I3, XYZ, Lab, Luv et OPP. En effet, l'algorithme traditionnel Mean shift extrait les informations de couleurs de l'espace RGB pour construire l'histogramme de couleurs du modèle cible. Afin d'évaluer l'influence des espaces de couleurs sur les performances de tracker Mean shift nous avons utilisé des séquences d'images couleurs des bases de données OTB 2013 et 2015. Le tracker Mean shift a été exécuté en utilisant ces séquences avec chacun des espaces couleurs mentionnés précédents.

- **Résultats quantitatifs**

Les performances du tracker Mean shift obtenues pour chacun des espaces de couleurs choisies et des quelques séquences vidéo, sont indiquées dans le tableau 5.3. La meilleure performance correspond à l'erreur (CLE) la plus faible et au taux de recouvrement (VOR) le plus fort. Les résultats montrent que les performances de ce tracker différent d'un espace à l'autre comme le montre le tableau 5.3. Cette différence semble résulter de l'information de couleur extraite pour chaque espace et les défis des séquences vidéo. D'après les résultats de tableau, on peut voir que les espaces HSV et YIQ obtiennent globalement le taux moyen le plus fort (0.4080 pour HSV, 0.4040 pour YIQ) et l'erreur moyenne la plus faible sur l'ensemble des séquences comparées aux autres espaces, mais l'erreur moyenne des YIQ (40.127) est légèrement inférieure de HSV (42.451). Tandis que, l'espace RGB obtient 0.3849 pour VOR et 46.061 pour CLE.

La figure 5.13 représente les résultats obtenus de l'erreur de localisation du centre (CLE) et le taux de recouvrement (VOR) par le tracker Mean shift en utilisant les différents espaces de couleurs sur la séquence Deer. Nous pouvons clairement voir que l'espace de couleur OPP donne des performances supérieures aux autres espaces comme le montre la figure 5.13 (a), (b) : l'erreur de localisation du centre de OPP est inférieure à 20 pixel et sa valeur de taux de recouvrement est supérieur à 0.5 dans la plupart des trames. Tandis que, les espaces HSV, RGB, YCrCb, YIQ, YUV, I1I2I3, Lab et Luv donnent de mauvaises performances à partir des trames 12 et 18, bien qu'ils récupérent l'objet cible par hasard dans quelques instants, comme le montre la figure. L'espace RGB peut récupérer finalement l'objet cible dans la trame 37 et pour les espaces HSV, YIQ et XYZ dans la trame 57. Cette pondération inégale semble résulter de la similarité entre l'objet et un autre objet dans quelques instants.

Tableau 5.3 – Les moyens du taux de recouvrement (VOR) et de l'erreur de localisation du centre (CLE) pour le tracker Mean shift en utilisant les différents espaces de couleurs sur quelques séquences. La dernière ligne du tableau (Moyenne) correspond à une moyenne de VOR et de CLE. Le chiffre en rouge indique la meilleure performance, tandis que le bleu indique la deuxième meilleure performance.

Séquence	Défis	HSV	RGB	YCrCb	YIQ	YUV	I1I2I3	XYZ	Lab	Luv	OPP
CarDark	IV, BC	0,4001/ 32,177	0,3287/ 34,171	0,3580/ 33,445	0,3192/ 33,849	0,3235/ 35,419	0,2222/ 36,521	0,3135/ 34,279	0,3158/ 35,646	0,3355/ 34,470	0,2804/ 35,487
BlurCar2	SV, MB, FM	0,1623/ 150,48	0,1912/ 104,18	0,2201/ 82,798	0,2726/ 90,016	0,1851/ 101,31	0,2148/ 108,49	0,2085/ 98,206	0,2159/ 100,04	0,1780/ 104,45	0,2233/ 105,59
Bolt	OCC, DEF, IPR, OPR	0,1140/ 75,458	0,2078/ 66,842	0,5468/ 9,5395	0,5463/ 9,7162	0,1872/ 82,949	0,1726/ 104,48	0,5742/ 8,8887	0,2069/ 81,396	0,3659/ 30,332	0,1759/ 141,47
Bolt2	DEF, BC	0,5683/ 21,505	0,5546/ 36,187	0,4825/ 32,774	0,4909/ 42,722	0,4958/ 31,268	0,5805/ 19,699	0,5686/ 21,354	0,5654/ 17,979	0,5292/ 26,084	0,5108/ 30,843
Human2	IV, SV, MB, OPR	0,5772/ 31,968	0,6062/ 28,770	0,6154/ 24,888	0,6099/ 28,129	0,6541/ 18,541	0,6328/ 25,443	0,6146/ 28,877	0,6098/ 29,004	0,6018/ 29,030	0,6361/ 25,183
Woman	IV, SV, OCC, DEF, MB, FM, OPR	0,6608/ 8,5500	0,1056/ 118,68	0,5781/ 12,039	0,5942/ 13,203	0,0993/ 142,86	0,1259/ 123,91	0,1250/ 121,81	0,5472/ 17,027	0,5767/ 14,505	0,5342/ 18,981
Girl	SV, OCC, IPR, OPR	0,6100/ 7,5596	0,5392/ 13,541	0,4854/ 15,511	0,5055/ 13,827	0,4666/ 15,959	0,5166/ 15,692	0,4660/ 18,161	0,4924/ 15,206	0,4976/ 15,399	0,1160/ 71,673
DragonBaby	SV, OCC, MB, FM, IPR, OPR, OV	0,5683/ 21,505	0,5546/ 36,187	0,4825/ 32,774	0,4909/ 42,722	0,4958/ 31,268	0,5805/ 19,699	0,5686/ 21,354	0,5654/ 17,979	0,5292/ 26,084	0,5108/ 30,843
Panda	SV, OCC, DEF, IPR, OPR, OV, LR	0,0451/ 85,700	0,2670/ 68,044	0,1607/ 75,049	0,2389/ 51,787	0,3000/ 37,990	0,5380/ 6,5528	0,3189/ 58,845	0,3227/ 47,667	0,1044/ 78,625	0,1836/ 66,990
Deer	MB, FM, IPR, BC, LR	0,3092/ 62,937	0,4904/ 45,454	0,1631/ 173,66	0,2962/ 109,69	0,1412/ 123,22	0,1963/ 161,22	0,3062/ 87,961	0,1469/ 100,72	0,1743/ 111,38	0,6341/ 12,420
Football1	IPR, OPR, BC	0,5326/ 9,6961	0,4656/ 10,137	0,4209/ 12,951	0,4978/ 10,406	0,3935/ 14,685	0,3508/ 16,420	0,4165/ 13,238	0,4867/ 10,642	0,5371/ 8,7243	0,3667/ 14,598
KiteSurf	IV, OCC, IPR, OPR	0,2082/ 16,229	0,1621/ 19,309	0,0412/ 145,26	0,0981/ 36,668	0,1370/ 20,369	0,1357/ 32,314	0,0880/ 45,869	0,1872/ 18,591	0,0500/ 70,422	0,0763/ 124,11
Moyenne	---	0.4080/ 42.451	0.3849/ 46.061	0.3772/ 54.604	0.4040/ 40.127	0.3154/ 61.1067	0.3446/ 57.357	0.3714/ 47.857	0.3747/ 42.7671	0.3849/ 44.345	0.3694/ 54.571

.

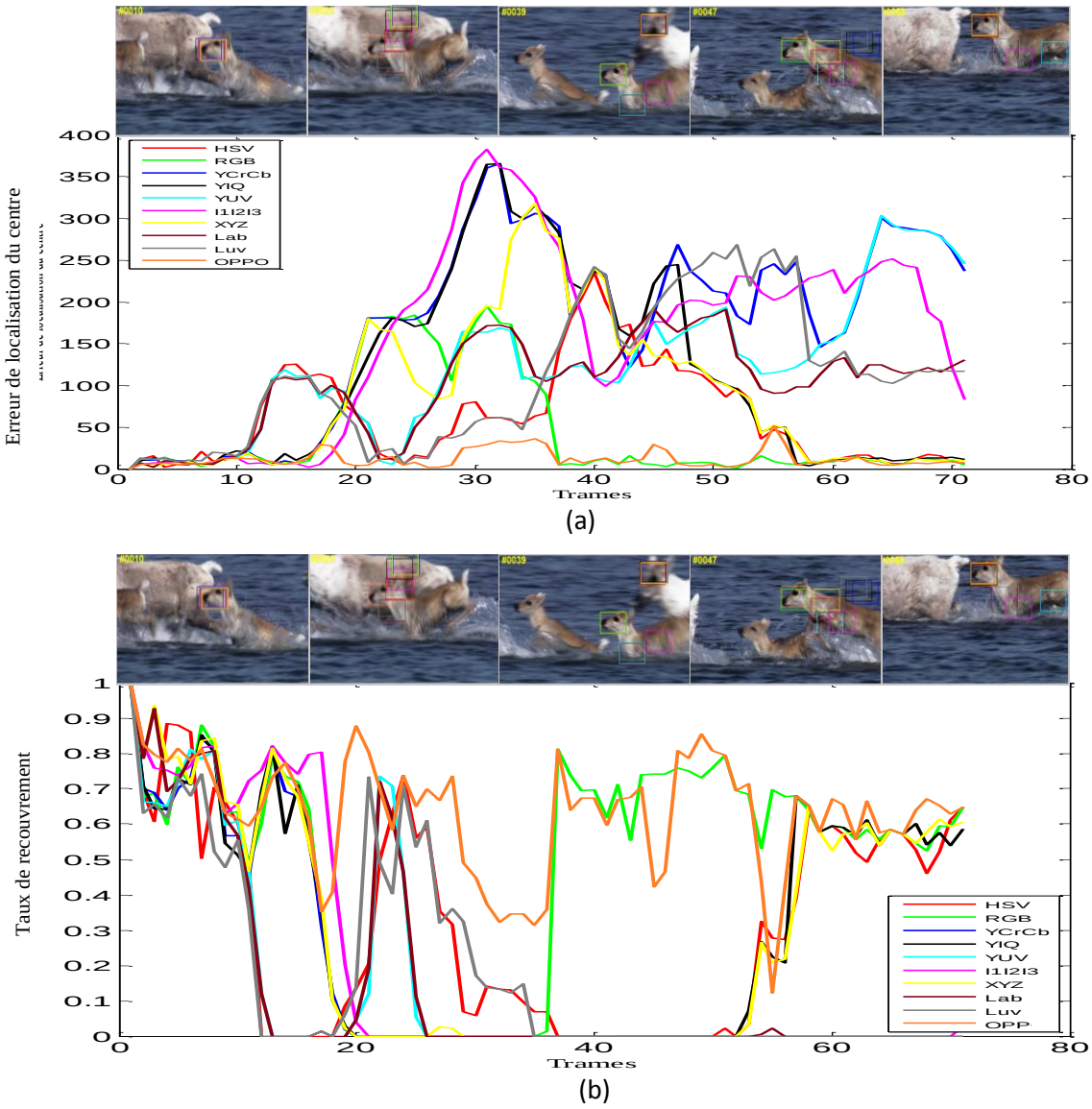


Figure 5.13 – Comparaison quantitative du tracker Meanshift en utilisant les différents espaces de couleurs sur la séquence Deer. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

Afin déterminer quel est l'espace de couleur qui rend le tracker MS plus robuste au lieu de l'espace RGB, nous appliquons ce tracker en utilisant les différents espaces de couleurs sur des séquences d'images couleurs des bases de données OTB 2013, 2015. Puis nous traçons les courbes de *Precision plots* et *Success plots* (sections 5.3.2 et 5.3.4) suivant le protocole d'évaluation défini dans [23]. La figure 5.14 montre les performances globales du tracker MS pour les espaces de couleurs sélectionnés précédents en utilisant de l'évaluation en une seule passe (OPE). L'évaluation de la robustesse par "OPE" consiste à les exécuter tout au long d'une séquence de test avec une initialisation à partir de la position de vérité-terrain (GT) dans la première trame et les critères d'évaluation (le score de précision et le score de réussite (ACU)), comme l'illustre la figure 5.14 (a), (b), respectivement. Selon les résultats obtenus,

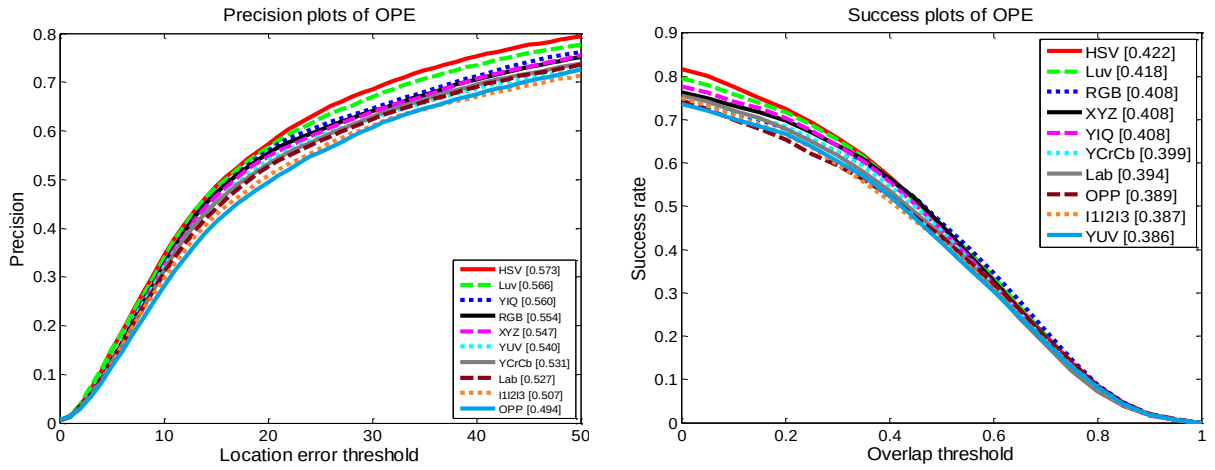
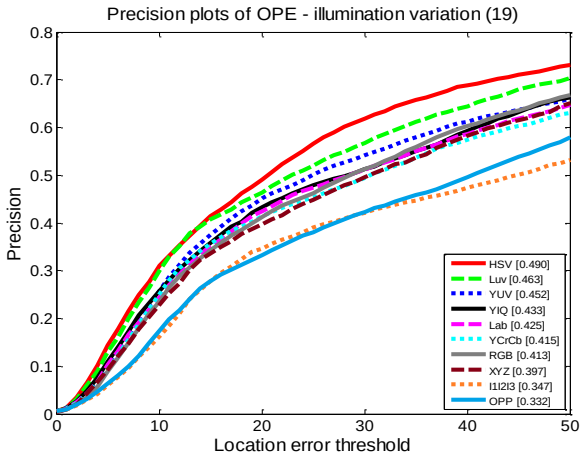


Figure 5.14 – *Precision plots* et *Success plots* de l'OPE sur des séquences d'images couleurs des bases OTB. Nous comparons le tracker MS en utilisant les différents espaces de couleurs. Les courbes de précision sont résumées avec le score de précision au seuil d'erreur de 20 pixels et les courbes de succès avec l'aire sous la courbe. Ces valeurs sont indiquées dans la légende.

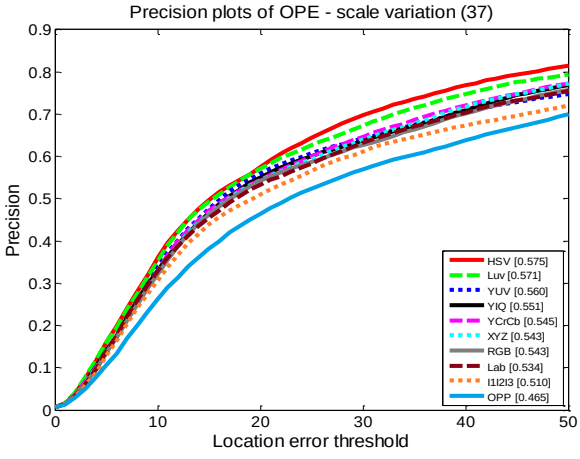
l'espace de couleur HSV donne meilleurs résultats dans les deux courbes *Precision plots* et *Success plots.* Plus précisément, l'utilisation de l'espace HSV pour calculer l'histogramme de couleur du tracker MS obtient un score de précision de 0.573 comparé à 0.554 de l'espace RGB, et un score de réussite de 0.422 comparé à 0.408 de l'espace RGB. Par conséquent, l'utilisation de l'espace HSV dans le tracker MS rend le plus précis et plus robuste que les autres espaces de couleurs, en particulier RGB. Le score de précision et de réussite de l'espace HSV se sont améliorés de 2% par rapport à celui de l'espace RGB.

- **Évaluation basée sur les défis (attributs)**

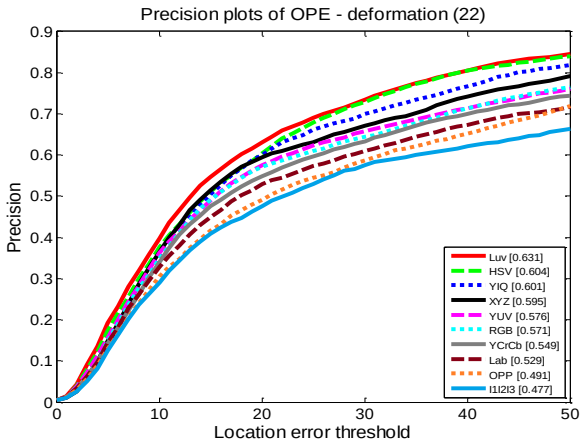
Plusieurs facteurs peuvent affecter les performances d'un tracker visuel. Dans la benchmark d'évaluation [23], les séquences sont annotées suivants 11 défis (attrubits) différents, comme nous avons pu le voir dans la section 5.2.1. Nous effectuons une comparaison pour les différents espaces de couleurs sur les séquences annotées par rapport aux 11 défis. Les figures 5.15 et 5.16 montrent les courbes de *Precision plots* et de *Success plots*, respectivement des différents défis. On peut remarque que l'espace de couleur HSV fonctionne favorablement sur 5 défis : IV, SV, FM, IPR et OV dans *Precision plots* et sur 4 difficultés : IV, SV, IPR et BC dans *Success plots*. Tandis que, l'espace RGB fonctionne favorablement sur 2 défis : MB et BC dans *Precision plots* et sur le défi FM dans *Success plots*. Selon les résultats obtenus, on peut dire que l'espace HSV donne de meilleures performances dans la plupart des difficultés, puisqu'il permet d'extraire les informations de couleurs de façon mielleure que les autres espaces.


Precision plots of OPE - illumination variation (19)
Precision
Location error threshold
HSV [0.490]
Luv [0.463]
YUV [0.452]
YIQ [0.433]
Lab [0.425]
YCrCb [0.415]
RGB [0.413]
XYZ [0.397]
I1I2I3 [0.347]
OPP [0.332]

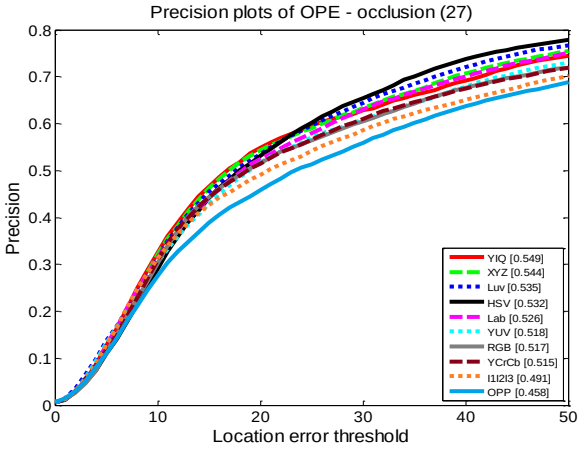
(a)


Precision plots of OPE - scale variation (37)
Precision
Location error threshold
HSV [0.575]
Luv [0.571]
YUV [0.560]
YIQ [0.551]
YCrCb [0.545]
XYZ [0.543]
RGB [0.543]
Lab [0.534]
I1I2I3 [0.510]
OPP [0.465]

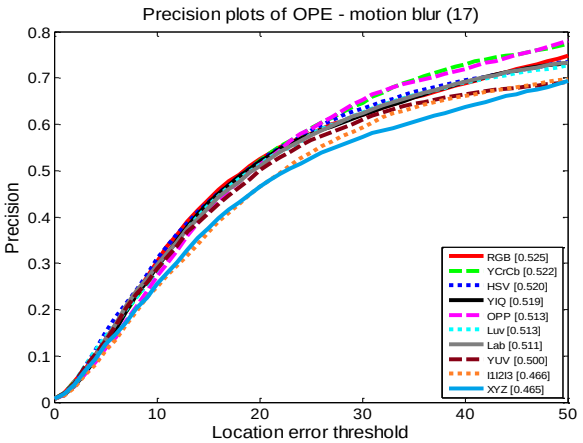
(b)


Precision plots of OPE - deformation (22)
Precision
Location error threshold
Luv [0.631]
HSV [0.604]
YIQ [0.601]
XYZ [0.595]
YUV [0.576]
RGB [0.571]
YCrCb [0.549]
Lab [0.529]
OPP [0.491]
I1I2I3 [0.477]

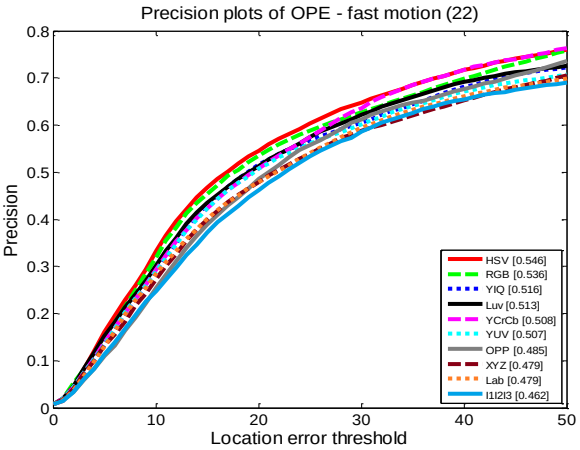
(c)


Precision plots of OPE - occlusion (27)
Precision
Location error threshold
YIQ [0.549]
XYZ [0.544]
Luv [0.535]
HSV [0.532]
Lab [0.526]
YUV [0.518]
RGB [0.517]
YCrCb [0.515]
I1I2I3 [0.491]
OPP [0.458]

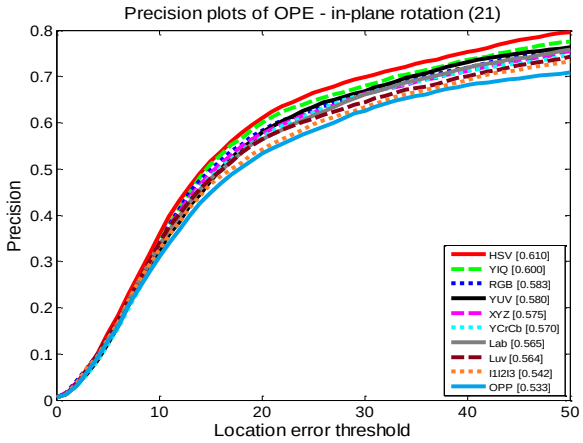
(d)


Precision plots of OPE - motion blur (17)
Precision
Location error threshold
RGB [0.525]
YCrCb [0.522]
HSV [0.520]
YIQ [0.519]
OPP [0.513]
Luv [0.513]
Lab [0.511]
YUV [0.500]
I1I2I3 [0.466]
XYZ [0.465]

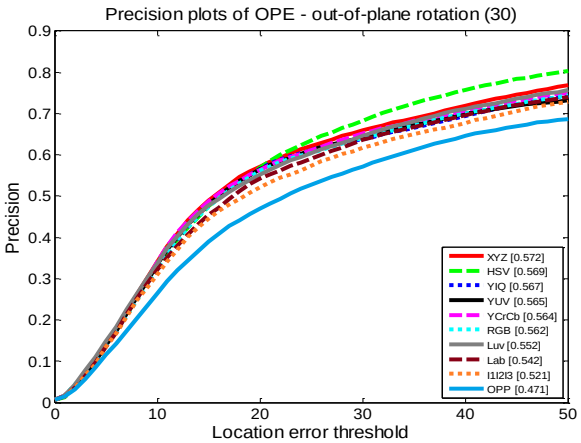
(e)


Precision plots of OPE - fast motion (22)
Precision
Location error threshold
HSV [0.546]
RGB [0.536]
YIQ [0.516]
Luv [0.513]
YCrCb [0.508]
YUV [0.507]
OPP [0.485]
XYZ [0.479]
Lab [0.479]
I1I2I3 [0.462]

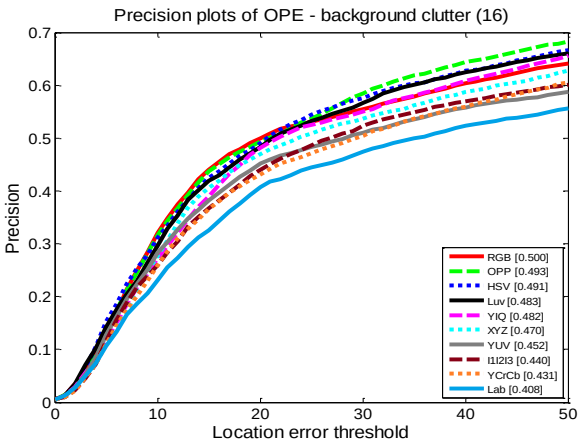
(f)



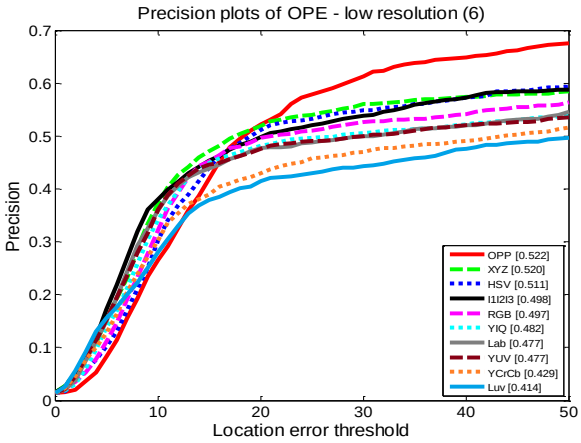
(g)



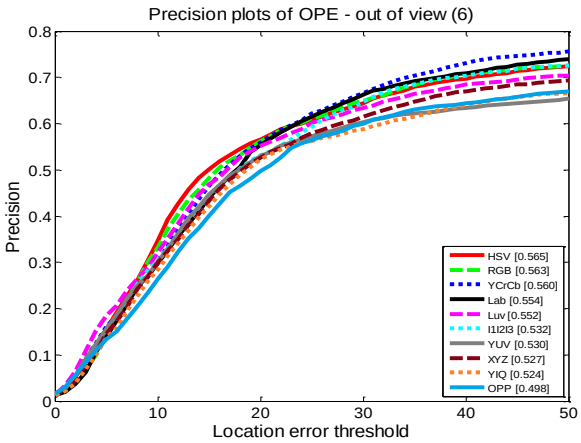
(h)



(i)

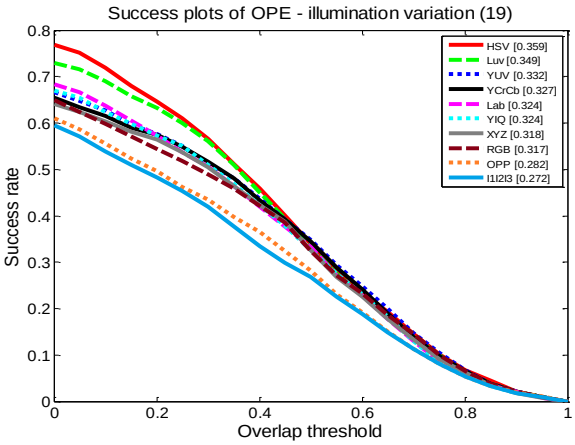


(j)

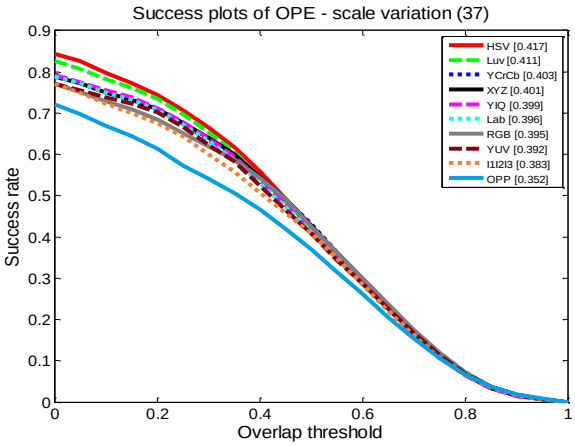


(k)

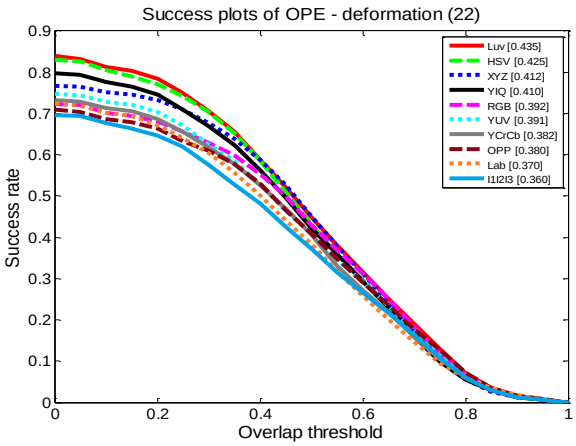
Figure 5.15 – *Precision plots* pour les différents difficultés du tracker Mean shift en utilisant les espaces de couleurs sélectionnés : (a) IV, (b) SV, (c) DEF, (d) OCC, (e) MB, (f) FM, (g) IPR, (h) OPR, (i) BC, (j) LR, (k) OV. La légende contient le score de précision pour chaque espace. La valeur apparaissant dans le titre indique le nombre de vidéos associées à la difficulté correspondant.


Success plots of OPE - illumination variation (19)
HSV [0.359]
Luv [0.349]
YUV [0.332]
YCrCb [0.327]
Lab [0.324]
YIQ [0.324]
XYZ [0.318]
RGB [0.317]
OPP [0.282]
I1I2I3 [0.272]
Success rate
Overlap threshold

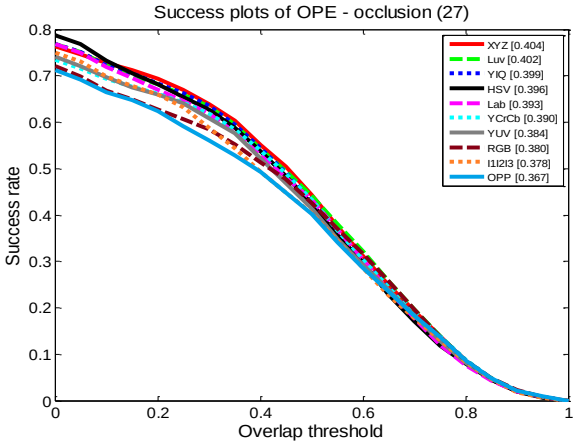
(a)


Success plots of OPE - scale variation (37)
HSV [0.417]
Luv [0.411]
YCrCb [0.403]
XYZ [0.401]
YIQ [0.399]
Lab [0.396]
RGB [0.395]
YUV [0.392]
I1I2I3 [0.383]
OPP [0.352]
Success rate
Overlap threshold

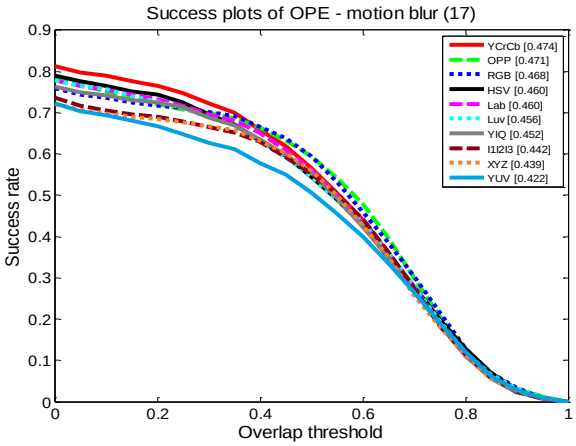
(b)


Success plots of OPE - deformation (22)
Luv [0.435]
HSV [0.425]
XYZ [0.412]
YIQ [0.410]
RGB [0.392]
YUV [0.391]
YCrCb [0.382]
OPP [0.380]
Lab [0.370]
I1I2I3 [0.360]
Success rate
Overlap threshold

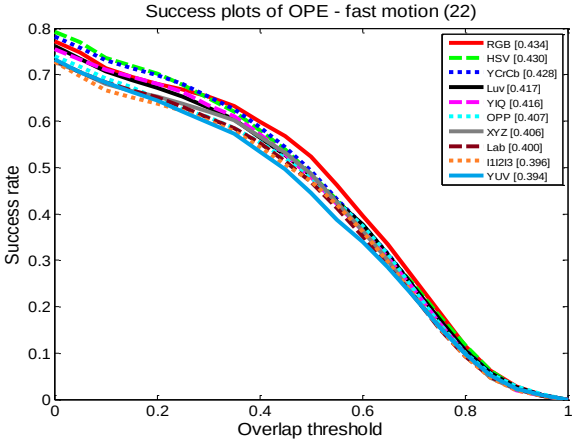
(c)


Success plots of OPE - occlusion (27)
XYZ [0.404]
Luv [0.402]
YIQ [0.399]
HSV [0.396]
Lab [0.393]
YCrCb [0.390]
YUV [0.384]
RGB [0.380]
I1I2I3 [0.378]
OPP [0.367]
Success rate
Overlap threshold

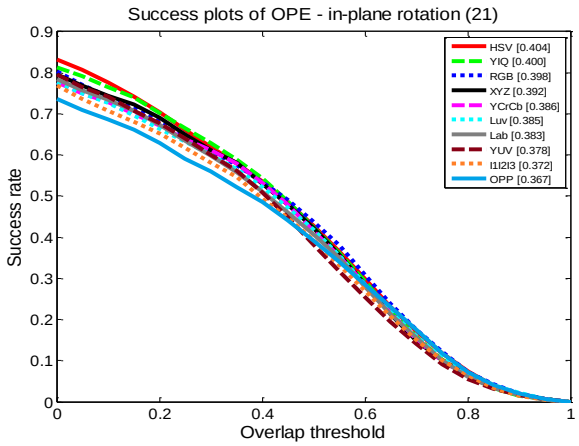
(d)


Success plots of OPE - motion blur (17)
YCrCb [0.474]
OPP [0.471]
RGB [0.468]
HSV [0.460]
Lab [0.460]
Luv [0.456]
YIQ [0.452]
I1I2I3 [0.442]
XYZ [0.439]
YUV [0.422]
Success rate
Overlap threshold

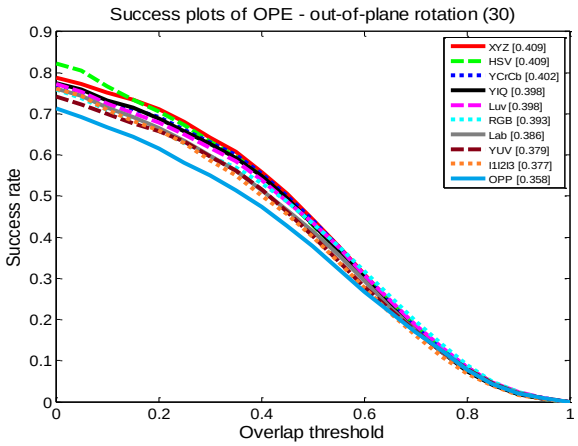
(e)


Success plots of OPE - fast motion (22)
RGB [0.434]
HSV [0.430]
YCrCb [0.428]
Luv [0.417]
YIQ [0.416]
OPP [0.407]
XYZ [0.406]
Lab [0.400]
I1I2I3 [0.396]
YUV [0.394]
Success rate
Overlap threshold

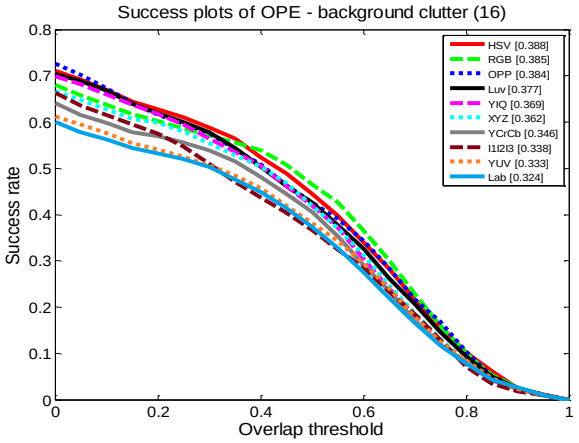
(f)



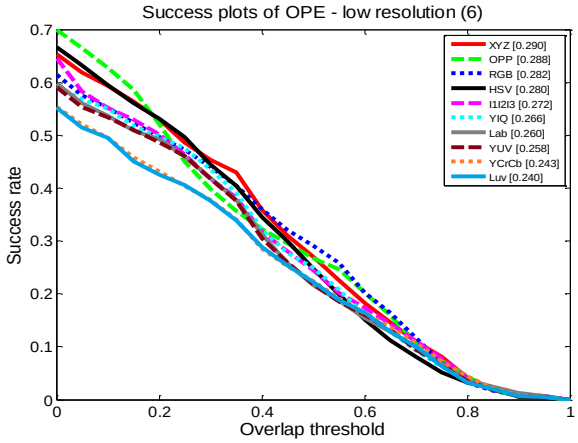
(g)



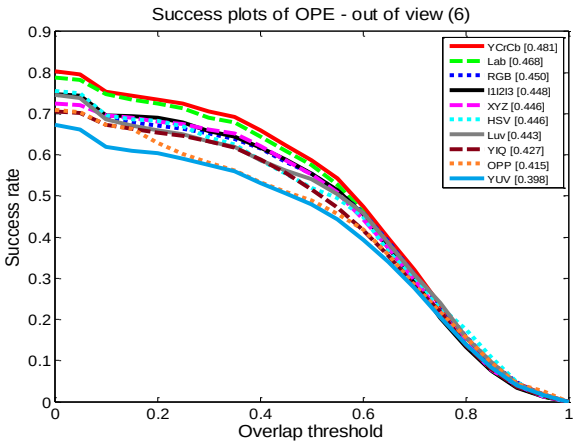
(h)



(i)



(j)



(k)

Figure 5.16 – *Success plots* pour les différents difficultés du tracker Mean shift en utilisant les espaces de couleurs sélectionnés : (a) IV, (b) SV, (c) DEF, (d) OCC, (e) MB, (f) FM, (g) IPR, (h) OPR, (i) BC, (j) LR, (k) OV. La légende contient le score AUC pour chaque espace. La valeur apparaissant dans le titre indique le nombre de vidéos associées à la difficulté correspondant.

- **Résultats qualitatifs**

La figure 5.17 montre quelques résultats qualitatifs sur des trames des séquences CarDrak, Panda, Girl, DragonBaby, Human2 et Bolt. Ces séquences contiennent plusieurs défis et divers types des objets suivis qui sont : voiture, animal, visage, humain. Dans la séquence CarDrak (figure 5.17 (a)) l'objet à suivre est une voiture sous les effets de la variation d'éclairage et du fond clutter. Bien que, l'objet cible est petit et ne peut pas le distinguer visuellement de fond, le tracker Mean shift pour tous les espaces de couleurs peut suivre la voiture de manière efficace jusqu'à la trame 40. Après cette trame, la plupart des espaces restent à suivre l'objet cible, mais avec moins de précision. Puis à partir de la trame 250, le tracker échoue pour tous les espaces en raison de la grande similarité entre l'objet et le fond. La figure 5.17 (b) illustre que la représentation de modèle cible par l'espace de couleur I1I2I3 donne meilleurs résultats, où le tracker peut suivre l'objet cible dans toutes les trames. Contrairement, le tracker Mean shift échoue par les autres espaces sauf Luv et YUV qui peuvent récupérer la cible par hasard (voir les trames 760 et 999). L'utilisation de l'espace HSV dans la séquence Girl (figure 5.17 (c)) rend le tracker plus robuste et plus précis par rapport aux autres espaces, car l'objet cible est un visage et HSV détecte précisément la couleur de peau. Les trames 99 et 115 montrent que l'efficacité de cet espace. La séquence vidéo dragonBaby contient plusieurs défis en même temps IV, OCC, MB, FM, IPR, OPR, OV. Cependant, le tracker peut suivre l'objet cible pour tous les espaces de couleurs, mais de façon inégale. Toutefois, il dérive complètement de trame 43 à 44 (figure 5.17 (d)) car il existe un grand déplacement (c-à-d mouvement rapide). La figure 5.17 (e) illustre la robustesse et la précision du tracker Mean shift au long de la séquence en utilisant tous les espaces de couleurs, malgré des changements d'illumination et d'échelle et l'occultation partiale. La cause de bonnes performances de ce tracker est de la distinction claire entre l'objet cible et le fond, ainsi que la grande taille de l'objet cible. Dans la dernière séquence, le tracker Mean shift a pu suivre l'objet cible au long de la séquence pour les espaces YCrCb, YIQ et XYZ, comme montré dans les trames 197 et 314 (figure 5.17 (f)). Mais, il a échoué pour les autres espaces en raison de la rapidité de l'objet cible et la similarité entre un autre objet et le fond. En conséquence, le choix de l'espace colorimétrique est important pour obtenir un histogramme de couleur plus robuste, car l'espace couleur approprié dépend de la situation actuelle qui peut varier entre les trames. L'espace colorimétrique choisi doit avoir une bonne capacité de distinguer l'objet de son fond.

Figure 5.17 – Résultats de suivi par le tracker Mean shift en utilisant les différents espaces de couleurs sur quelques séquences : (a) CarDrak, (b) Panda, (c) Girl, (d) dragonBaby, (e) Human2 et (f) Bolt. Les index des trames sont indiqués en haut à gauche de chaque trame.

5.4.3 L'efficacité de l'histogramme conjoint couleur-texture proposé

Les caractéristiques utilisées pour modéliser l'objet cible à l'aide de l'histogramme conjoint proposé sont les caractéristiques de couleur de l'espace HSV et les caractéristiques de texture extraies par le descripteur LPQ, le descripteur LBP ou le descripteur BSIF. L'efficacité des histogrammes conjoints proposés est évaluée dans cette section.

5.4.3.1 Performance de tracker MS à travers la variation de la valeur de rayon du descripteur LPQ

Dans cette section, nous concentrons sur la précision et la robustesse du tracker MS (Mean shift) en utilisant les caractéristiques de textures LPQ à travers d'étude la variation de la valeur de rayon du descripteur LPQ. Cette dernière, est variée pour fixer la valeur optimale. Le tableau 5.4 présente les performances du tracker Mean shift en utilisant l'histogramme conjoint couleur HSV- texteur LPQ en fonction du rayon R de LPQ dans quelques séquences d'images couleurs des bases de données OTB. Nous pouvons clairement constater que R=17 donne de meilleures performances que les autres rayons R=3, 5, 7, 9, 11, 13 et 15 (le moyen de taux VOR égale à 0.5892 et l'erreur CLE égale à 18,166).

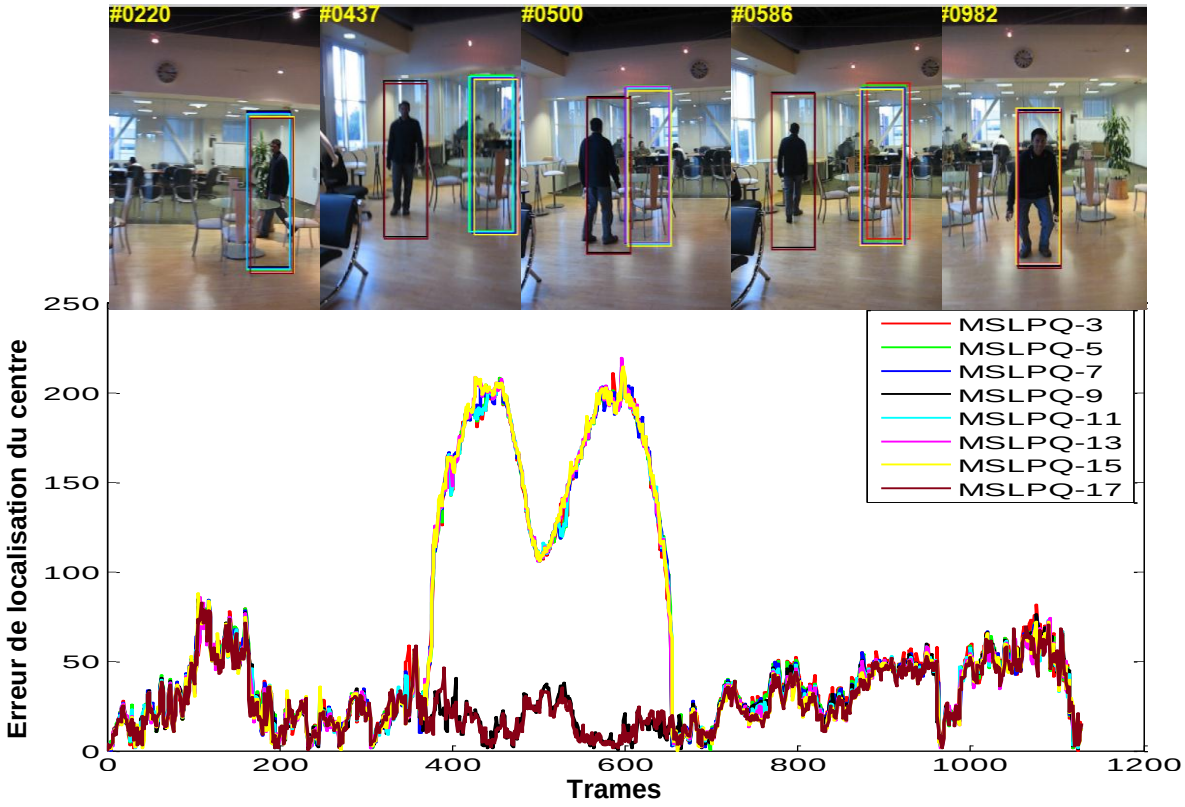
La figure 5.18 représente les résultats obtenus de l'erreur de localisation du centre (CLE) et le taux de recouvrement (VOR) par le tracker MS en utilisant les différentes valeurs du rayon du descripteur LPQ sur la séquence Human2. Nous pouvons clairement voir que les valeurs de rayons R=17 et R=9 donne des performances supérieures à celle des autres rayons comme les montre dans la figure 5.18 (a), (b) : l'erreur de localisation du centre de R=17 et R=9 est inférieure à 20 pixel et sa valeur de taux de recouvrement est supérieur à 0.5 dans la plupart des trames. Par contre, les rayons R=3, 5, 7, 11, 13 et 15 donnent de mauvaises performances à partir de trame 360 à 660 (le tracker MS dérive) en raison d'occultation partielle, de flou de mouvement et une présence des autres objets similaires, bien qu'ils aient récupéré l'objet cible par hasard dans la trame 661, comme montré dans les courbes ELC et VOR et les trames présentées. Ce qui signifie que les informations de texture obtenus par R=17 et R=9 peuvent ajouter à l'histogramme conjoint des informations utiles qui rendent le tracker Mean shift plus robuste aux défis mentionnés. Par conséquent, le choix du paramètre R est important pour le pouvoir de discrimination du descripteur LPQ et pour obtenir un histogramme conjoint HSV couleur- LPQ texteur plus robuste.

Les performances globales du tracker MS avec le rayon du descripteur LPQ sont représentées par la figure 5.19. Cette figure présente une comparaison des erreurs de localisation centre et des taux de recouvrement du tracker MS en fonction de la variation de rayon R du descripteur

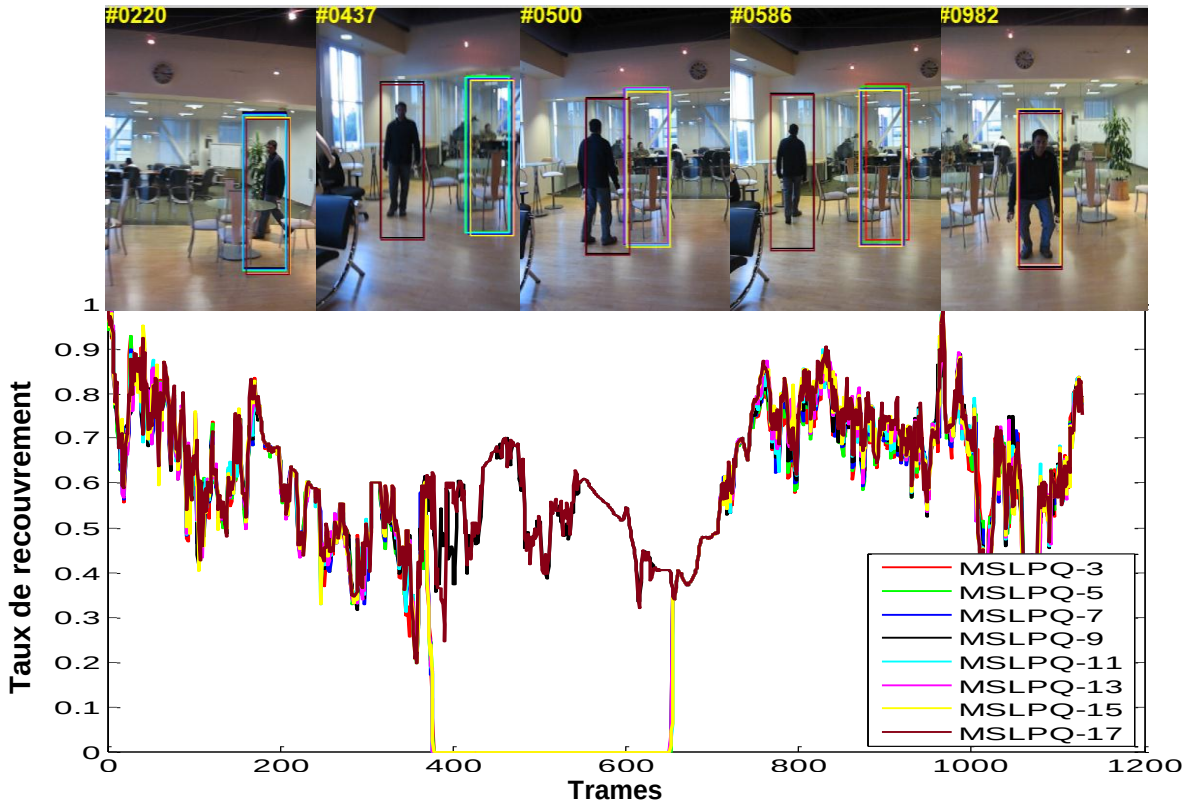
LPQ par les courbes de *Precision plots* et de *Success plots.* Afin obtenir un rayons optimal, nous appliquons ce tracker sur un ensemble des séquences d'images couleurs des bases de données OTB. Les deux courbes montrent que, les résultats obtenus pour R=17 sont globalement bons (le score de précision est 0.608 et le score de ACU est 0.541). D'après les résultats expérimentaux, il est évident que l'histogramme conjoint proposé HSV-LPQ qui utilise la valeur de rayon R=17 est plus robuste que les autres rayons. Le changement du rayon R contribue significativement à l'amélioration des performances du tracker MS. Cela montre l'importance de la considération d'un voisinage de pixel lors de l'application du descripteur LPQ pour extraire les caractéristiques de texture LPQ.

Tableau 5.4 – Les moyens du taux de recouvrement (VOR) et de l'erreur de localisation du centre (CLE) pour le tracker Mean shift en utilisant les différentes valeurs du rayons du descripteur LPQ sur quelques séquences. Le chiffre en rouge indique la meilleure performance, tandis que le bleu indique la deuxième meilleure performance.

Séquence	Défis	R=3	R=5	R=7	R=9	R=11	R=13	R=15	R=17
boy	SV, MB, FM, IPR, OPR	0,7511/ 3,8471	0,7264/ 4,6197	0,7382/ 4,3133	0,7383/ 4,3691	0,7459/ 4,1477	0,7514/ 3,9525	0,7564/ 3,7530	**0,7611/ 3,6606**
BlurCar3	MB, FM	0,4304/ 56,473	0,4947/ 38,108	0,3676/ 61,218	0,2531/ 78,349	0,3678/ 62,757	0,5370/ 31,823	0,4685/ 41,076	**0,5877/ 24,552**
CarDrak	IV, BC	0,4505/ 30,261	0,3881/ 23,563	0,3744/ 24,014	0,3986/ 24,231	0,4003/ 26,616	0,4334/ 23,583	0,4713/ 20,908	**0,4778/ 20,870**
skating2	SV, OCC, DEF, FM, OPR	**0,5083/ 29,696**	0,5040/ 30,622	0,5050/ 30,654	**0,5063/** 30,226	0,5055/ 30,301	0,5050/ **29,954**	0,5039/ 30,005	0,5046/ 29,995
Girl	SV, OCC, IPR, OPR	0,4963/ 17,770	0,4610/ 25,707	0,5479/ 13,513	0,5027/ 17,080	0,6252/ 7,3450	0,6347/ 6,9287	**0,6551/ 5,6822**	**0,6642/ 5,2458**
David3	OCC, DEF, OPR, BC	**0,7304/ 8,5173**	0,7212/ 8,9813	0,7245/ 9,1565	0,7284/ 8,9063	**0,7323/ 8,3196**	0,7261/ 8,8240	0,7266/ 8,7599	0,7236/ 9,2345
Human2	IV, SV, MB, OPR	0,4647/ 65,344	0,4653/ 64,991	0,4668/ 64,677	**0,5990/ 28,236**	0,4704/ 64,151	0,4741/ 63,810	0,4748/ 63,759	**0,6084/ 26,759**
faceocc1	OCC	0,6943/ 21,094	0,6948/ 21,177	0,6950/ 21,174	0,6959/ 21,159	0,6970/ 21,065	**0,6975/** 20,989	0,6973/ **20,982**	**0,7000/ 20,675**
Rubik	SV, OCC, IPR, OPR	0,3967/ 29,083	0,4061/ 28,056	0,3969/ 29,204	0,4004/ 29,069	0,4028/ 28,821	0,4056/ 28,490	**0,4155/ 27,563**	**0,4242/ 26,218**
kiteSurf	IV, OCC, IPR, OPR	0,2571/ 23,408	0,3481/ 18,387	**0,4370/ 13,269**	0,3344/ 19,799	0,3602/ 18,046	0,3769/ 16,617	0,3741/ 15,211	**0,4400/ 14,452**
Moyenne	---	0,5180/ 28,549	0,5210/ 26,421	0,5253/ 27,119	0,5157/ 26,142	0,5307/ 27,157	0,5542/ **23,497**	**0,5544/** 23,770	**0,5892/ 18,166**

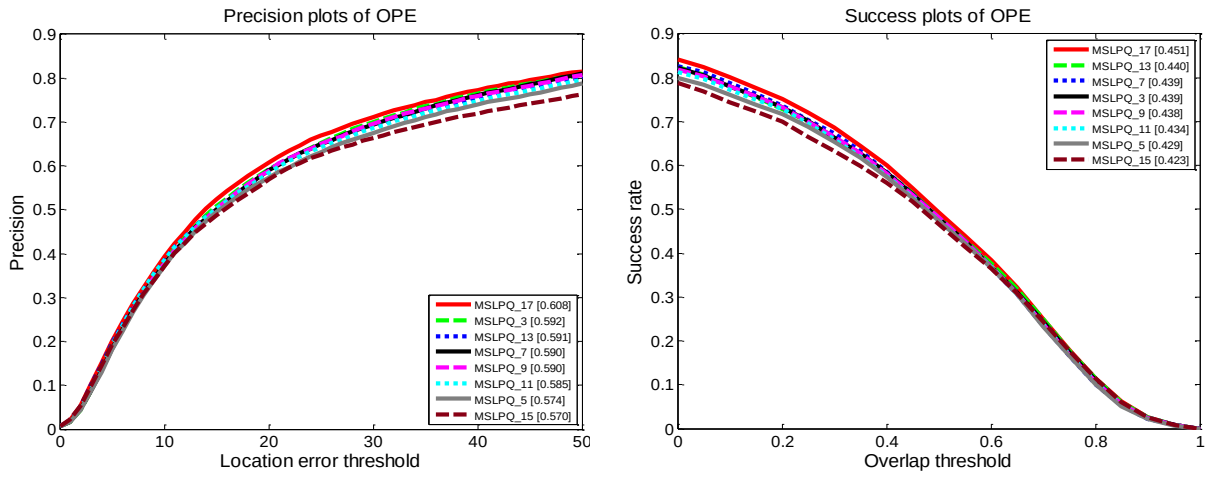


(a)



(b)

Figure 5.18 – Comparaison quantitative avec qualitative du tracker Meanshift en utilisant les différents rayons du descripteur LPQ sur la séquence Human2. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).



(a) (b)

Figure 5.19– *Precision plots* et *Success plots* de l'OPE pour la comparaison quantitative du tracker MS en utilisant les différents rayons du descripteur LPQ sur des séquences d'images couleurs des bases OTB.

5.4.3.2 Performances du tracker Mean shift par les histogrammes conjoints proposés HSV-LPQ, HSV-LBP et HSV-BSIF

Dans cette section, nous évaluons les algorithmes proposés MSLPQ, MSLBP, MSBSIF (le tracker MS en utilisant les histogrammes conjoints proposés HSV-LPQ, HSV-LBP et HSV-BSIF) en faisant une comparaison avec cinq algorithmes populaires dans l'état de l'art: (1) le traditionnel Mean shift tracking (KMS) [1] ; (2) Structured output tracking with kernels (Struck) [6]; (3) Tracking-Learning-Detection (TLD) [8]; (4) Visual Tracking Decomposition (VTD) [4] ; (5) multiple instance learning (MIL) [34] et avec notre tracker MS(HSV). Nous évaluons les performances de chaque tracker en fonction de l'erreur de localisation du centre et du taux de recouvrement suivant le protocole d'évaluation défini dans [23]. Nous présentons notre évaluation empirique sur deux ensembles des bases de données : OTB [23] et VOT2013 [229] et seules les séquences d'images couleurs sont utilisées. Nous utilisons l'espace de couleur HSV pour extraire les caractéristiques de couleurs. Comme pour les caractéristiques de textures nous utilisons pour le descripteur LPQ la valeur de rayon R=17 qui a donné de bons résultats dans la section précédente, pour le descripteur LBP ($LBP_{P,R}(8,1)$) LBP basique et pour le descripteur BSIF le filtre de taille l=9 et nombre de bits n=8. Les paramètres des descripteurs LBP et BSIF sont choisis selon les études de la reconnaissance de visage.

a) Résultats obtenus par OTB

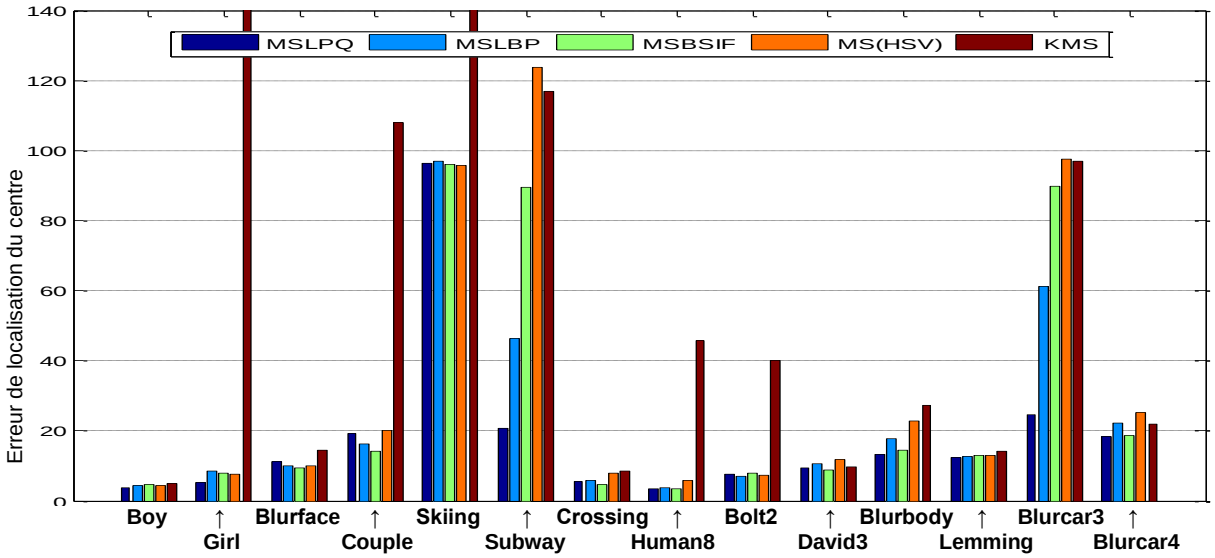
- **Résultats quantitatifs**

Les résultats du tableau 5.5 résument la comparaison entre les algorithmes proposés et les cinq trackers de comparaison mentionnés au-dessus, sur les 14 séquences d'images qui contiennent des nombreux défis. Le tableau 5.5 montre que nos algorithmes (MSLPQ, MSLBP et MSBSIF) ont des meilleures performances que les méthodes de comparaison MS(HSV), KMS, Struck, LTD, VTD et MIL, en taux de recouvrement (VOR) et en erreur de localisation du centre (CLE). Plus précisément, l'algorithme proposé MSLPQ a globalement obtenu le taux moyen le plus fort (VOR=0.6091) et l'erreur moyenne la plus faible (CLE=17.898) sur l'ensemble des séquences présentées, puis MSLBP avec VOR= 0.5833, CLE=27.272 et MSBSIF avec VOR=0.5782, CLE=23.066. Alors que, le tracker Struck a obtenu des performances inférieures que MSLPQ, MSLBP et MSBSIF avec VOR= 0.5246, CLE=45.653, bien qu'il a obtenu de meilleurs résultats dans plusieurs séquences (Girl, Subway, Blurcar3 et Blurcar4), mais pour les autres séquences il a obtenu de mauvais résultats comme la séquence david3 (Struck : VOR= 0.2916 et CLE=106,50, MSLPQ : VOR=0.7236 et CLE=9.2345, MSLBP : VOR=0.7022 CLE= 10.491, MSBSIF : VOR=0.7251 et CLE=8.8455). Les trackers VTD et MIL donnent de mauvaises performances, cependant le tracker TLD en échec sur la plupart des séquences. La comparaison entre les algorithmes proposés et l'algorithme traditionnel Mean shift (KMS) montre qu'il y a une importante amélioration des performances de ce dernier, en raison de l'utilisation des informations de texture pour construire l'histogramme de l'objet cible. Le taux VOR de KMS est 0.4150 inferieur au taux de MSLPQ par 0.19, et l'erreur CLE de tracker KMS est 165,944 supérieurs à l'erreur de MSLPQ par 149 pixels. De plus, MS (HSV) proposé semble être plus robuste que le KMS et a obtenu de meilleures performances dans cet ensemble de séquences, en particulier dans la séquence Girl où KMS a échoué.

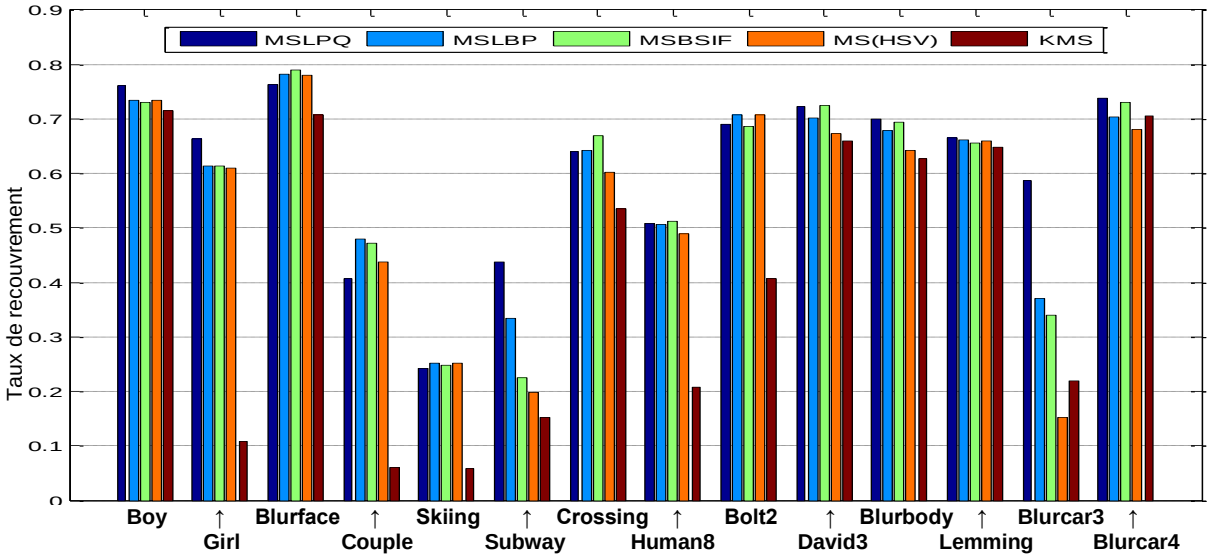
Tableau 5.5 – Les moyens du taux de recouvrement (VOR) et de l'erreur de localisation centre (CLE) pour les algorithmes proposés et cinq trackers de l'état de l'art sur quelques séquences des bases OTB. Le chiffre en rouge indique la meilleure performance, tandis que le bleu indique la deuxième meilleure performance. NaN : Le tracker a échoué.

Séquence	Défis	MIL	VTD	TLD	Struck	KMS	MS(HSV)	MSBSIF	MSLBP	MSLPQ
Boy	SV, MB, FM, IPR, OPR	0,4910/ 12,835	0,6256/ 7,5736	0,6616/ 4,4942	**0,7598/ 3,8448**	0,7146/ 4,8324	0,7349/ 4,2176	0,7314/ 4,4807	0,7341/ 4,3643	**0,7611/ 3,6606**
Girl	SV, OCC, IPR, OPR	0,3980/ 13,666	0,5510/ 8,5977	0,5722/ 9,7925	**0,7453/ 2,5731**	0,1076/ 1560.05	0,6100/ 7,5596	0,6139/ 7,8625	0,6133/ 8,4381	**0,6642/ 5,2458**
BlurFace	MB, FM, IPR	0,2754/ 71,975	0,4164/ 40,688	**0,8809/ 3,7374**	0,4691/ 42,353	0,7076/ 14,539	0,7802/ 9,9785	**0,7897/ 9,4850**	0,7816/ 9,8854	0,7624/ 11,233
Couple	SV, DEF, FM, OPR, BC	0,4978/ 34,525	0,0644/ 104,25	**0,7721/ 2,5433**	**0,5361/ 11,332**	0,0609/ 107,94	0,4374/ 20,094	0,4713/ 14,076	0,4797/ 16,123	0,4069/ 19,266
Skiing	IV, SV, DEF, IPR, OPR	0,0550/ 266,97	0,0660/ 263,27	NaN/ NaN	0,0340/ 251,92	0,0587/ 254,72	**0,2527/ 95,773**	0,2470/ **95,931**	**0,2509/** 96,926	0,2418/ 96,284
Subway	OCC, DEF, BC	**0,6481/ 7,5953**	0,1567/ 141,31	NaN/ NaN	**0,6537/ 4,4690**	0,1519/ 117,02	0,1973/ 123,86	0,2255/ 89,573	0,3342/ 46,458	0,4380/ 20,739
Crossing	SV, DEF, FM, OPR, BC	**0,7274/ 3,1768**	0,3168/ 26,125	0,4032/ 24,342	**0,6767/ 2,8082**	0,5357/ 8,4002	0,6013/ 8,0296	0,6693/ 4,4849	0,6424/ 5,9087	0,6396/ 5,4513
Human8	IV, SV, DEF	0,1234/ 74,947	0,2877/ 18,998	0,1279/ 65,971	0,1318/ 63,785	0,2071/ 45,699	0,4895/ 5,6854	**0,5115/ 3,5455**	0,5069/ 3,5845	**0,5088/ 3,5306**
Bolt2	DEF, BC	0,6827/ 7,3915	0,5019/ 17,115	NaN/ NaN	0,2223/ 86,407	0,4077/ 39,932	**0,7081/ 7,1510**	0,6874/ 7,8174	**0,7066/ 7,0947**	0,6897/ 7,6050
David3	OCC, DEF, OPR, BC	0,5367/ 29,680	0,4025/ 66,721	NaN/ NaN	0,2916/ 106,50	0,6590/ 9,6120	0,6721/ 11,898	**0,7251/ 8,8455**	0,7022/ 10,491	**0,7236/ 9,2345**
BlurBody	SV, DEF, MB, FM, IPR	0,0385/ 206,74	0,2361/ 146,90	NaN/ NaN	**0,7239/ 13,973**	0,6271/ 27,331	0,6428/ 22,895	0,6938/ 14,369	0,6792/ 17,596	**0,6989/ 13,305**
Lemming	IV, SV, OCC, FM, OPR, OV	0,6485/ 12,064	0,4337/ 79,224	NaN/ NaN	0,4808/ 37,752	0,6473/ 14,171	0,6601/ 13,096	0,6564/ 12,882	**0,6613/ 12,595**	**0,6661/ 12,246**
BlurCar3	MB, FM	0,2727/ 138,15	0,1865/ 107,79	NaN/ NaN	**0,7729/ 6,5226**	0,2192/ 97,104	0,1526/ 97,486	0,3408/ 89,696	0,3699/ 61,198	**0,5877/ 24,552**
blurCar4	MB, FM	0,0627/ 197,41	0,0721/ 85,276	NaN/ NaN	**0,8454/ 4,9019**	0,7050/ 21,876	0,6799/ 25,099	0,7304/ 18,755	0,7032/ 22,259	**0,7382/ 18,227**
Moyenne	---	0.3899/ 76.9391	0.3084/ 86.7042	NaN/ NaN	0.5246/ 45.653	0.4150/ 165,944	0.5443/ 32.345	**0.5833/** 27.272	0.5782/ **23.066**	**0.6091/ 17.898**

Pour illustrer mieux l'amélioration des performances du tracker Mean shift en utilisant les histogrammes conjoints couleur-texture, nous présentons la figure 5.20 qui montre la valeur moyenne de l'erreur CLE et le taux VOR sur les 14 séquences choisies (tableau 5. 5) par une barre pour chaque tracker. On peut clairement voir que le KMS donne de mauvais résultats (CLE supérieure et VOR inferieur) dans la plupart des séquences, en particulier dans les séquences difficiles comme Girl, Couple, Subway et Human8. Cela signifie que les informations de texture LPQ, LBP ou BSIF et les informations de couleur HSV améliorent le tracker Mean shift de manière efficace, en particulier dans les défis difficiles. De plus, les algorithmes proposés sont rapides comme l'algorithme traditionnel Mean shift, car les descripteurs de texture utilisés sont très rapides et la méthode de combinaison est très simple.



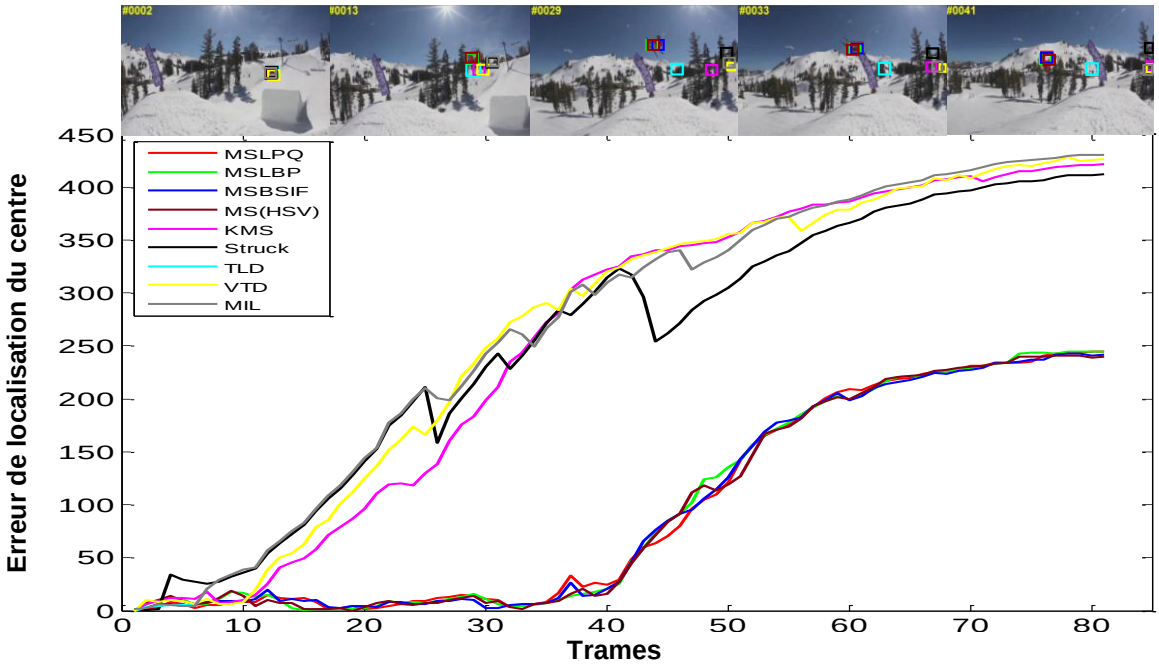
(a)



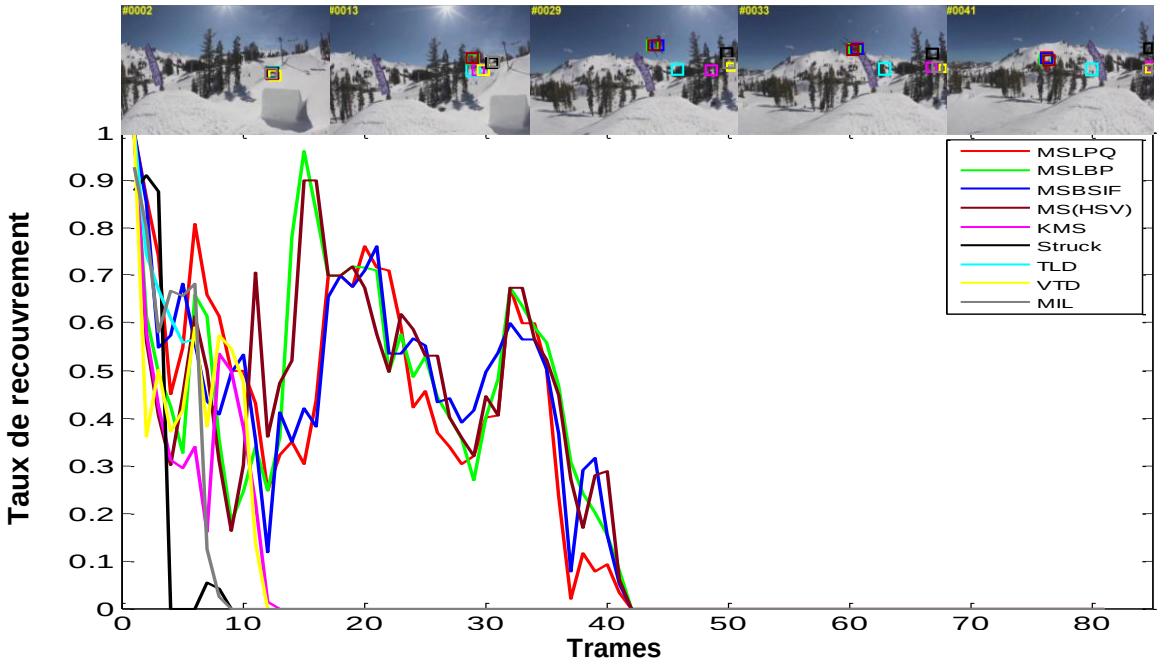
(b)

Figure 5.20 – Comparaison entre les algorithmes proposés et l'algorithme traditionnel Mean shift sur quelques séquences des bases de données OTB. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

Les figures 5.21 à 5.24 représentent les résultats obtenus de l'erreur de localisation du centre (CLE) et le taux de recouvrement (VOR) par les algorithmes proposés et les trackers de comparaison pour les séquences d'images Skiing, Bolt2, Blurcar4 et Blurface, respectivement. Dans la figure 5.21, l'objet cible est très petit mais malgré ça, les trackers MSLPQ, MSLBP, MSBSIF et MS(HSV) peuvent suivre la cible jusqu'à la trame 41 comme illustré dans les courbes (CLE et VOR) et les trame présentées. Par contre, les autres trackers échouent à partir de la trame 13, en raison du changement de couleur du fond et de la similarité entre l'objet et le fond. La figure 5.22 illustre les résultats de la séquence Bolt2. Dans cette séquence, la pose de la cible change rapidement et l'apparence se déforme fréquemment, D'après les résultats obtenus, MSLPQ, MSLBP, MSBSIF, MS(HSV) et MIL réussissent à suivre tout au long de la séquence, alors que KMS dérive après la trame 185 lorsque le fond a la même couleur que la cible, mais TLD dérive rapidement en début (trame 10). Cependant, VTD est seulement capable de suivre une partie de la cible mais ne la perd pas, mais Struck échoue tout au long de la séquence. La figure 5.23 montre le taux VOR et l'erreur CLE de la séquence BlurBody qu'inclut le mouvement de flou et le DEF. MIL perd la cible rapidement à cause du flou, alors que VTD est capable de suivre au début, mais échoue après la trame 99. TLD dérive souvent et récupère rapidement la cible par hasard, c'est pourquoi il garde une mauvaise performance. Alors que, Struck, MSLPQ, MSLBP et MSBSIF obtiennent les meilleures performances par rapport à MS(HSV) et KMS, comme le montre les trames présentées (trames 111, 160 et 275). Dans la figure 5.24, les résultats expérimentaux montrent que TLD obtient les meilleures performances du suivi de visage, comme le montre les courbes et les trames présentées. Cependant, MSLPQ, MSLBP, MSBSIF, MS(HSV) et KMS peuvent suivre la cible avec succès, mais KMS dérive dans quelques trames lorsqu'un mouvement de flou sévère se produit. Alors que, les trackers MIL, VTD et Struck, ne parviennent pas à suivre la cible à partir des premières trames.

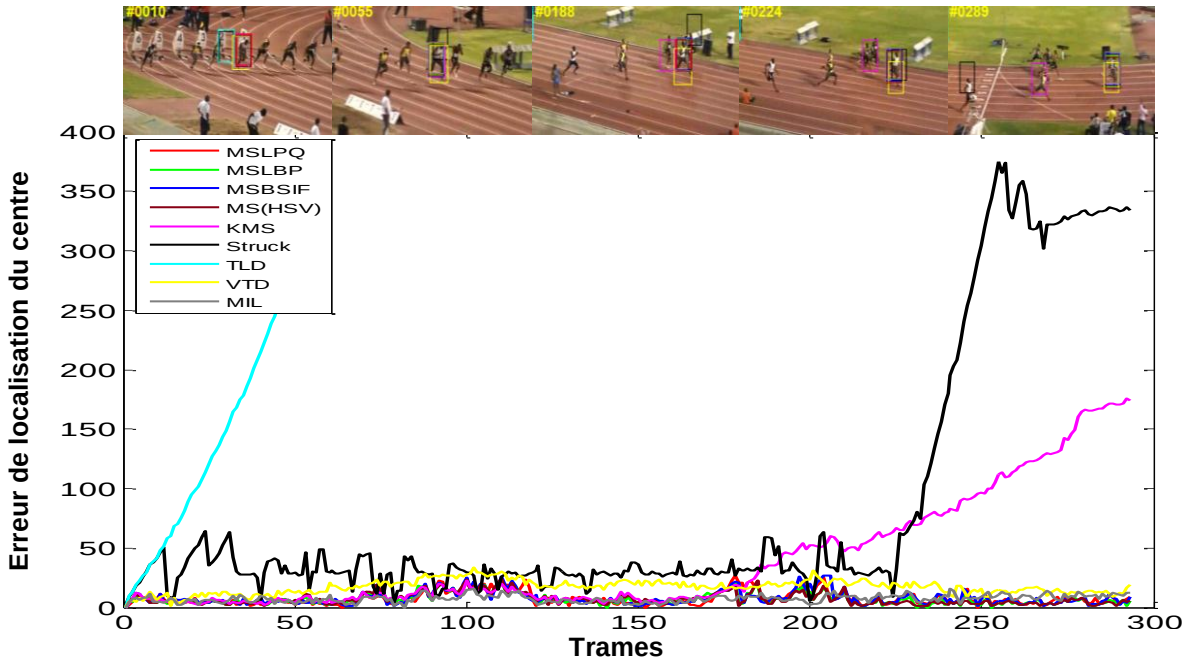


(a)

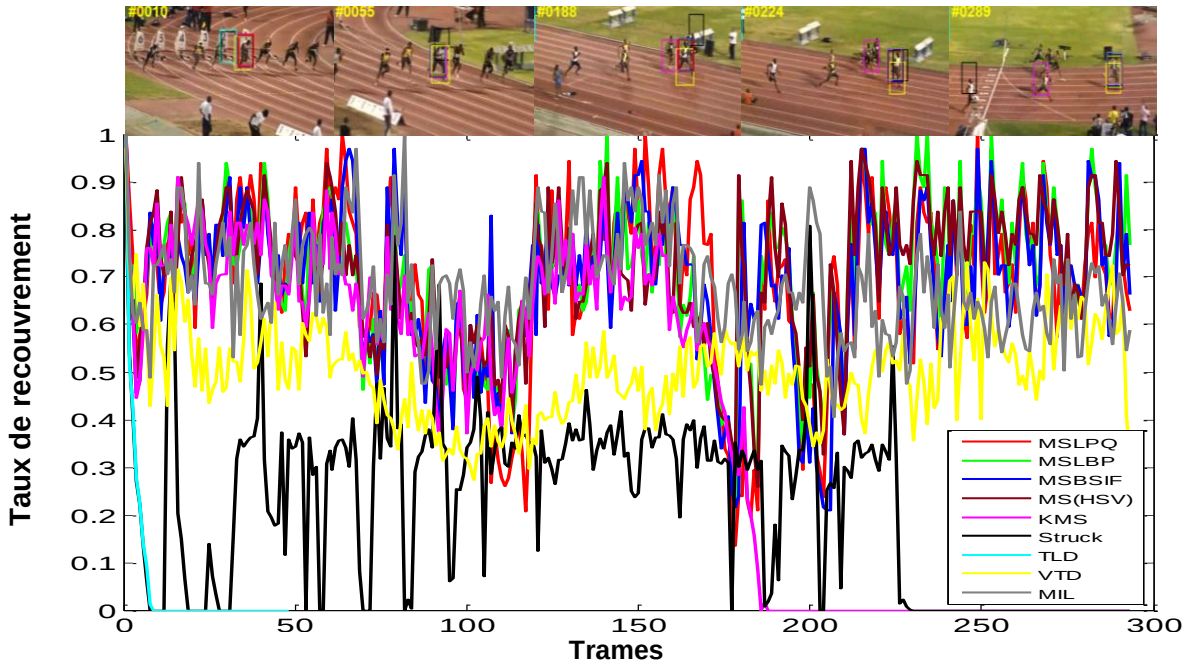


(b)

Figure 5.21 – Comparaison quantitative et qualitative pour les algorithmes proposés et cinq trackers de l'état de l'art sur la séquence Skiing. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

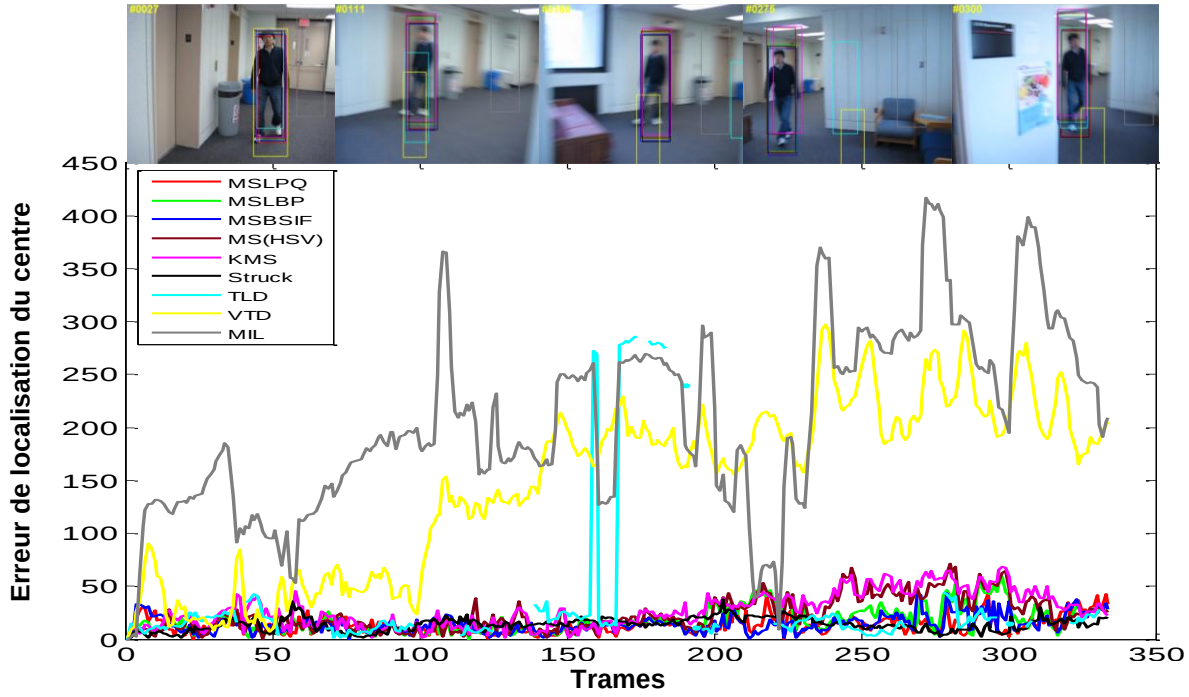


(a)

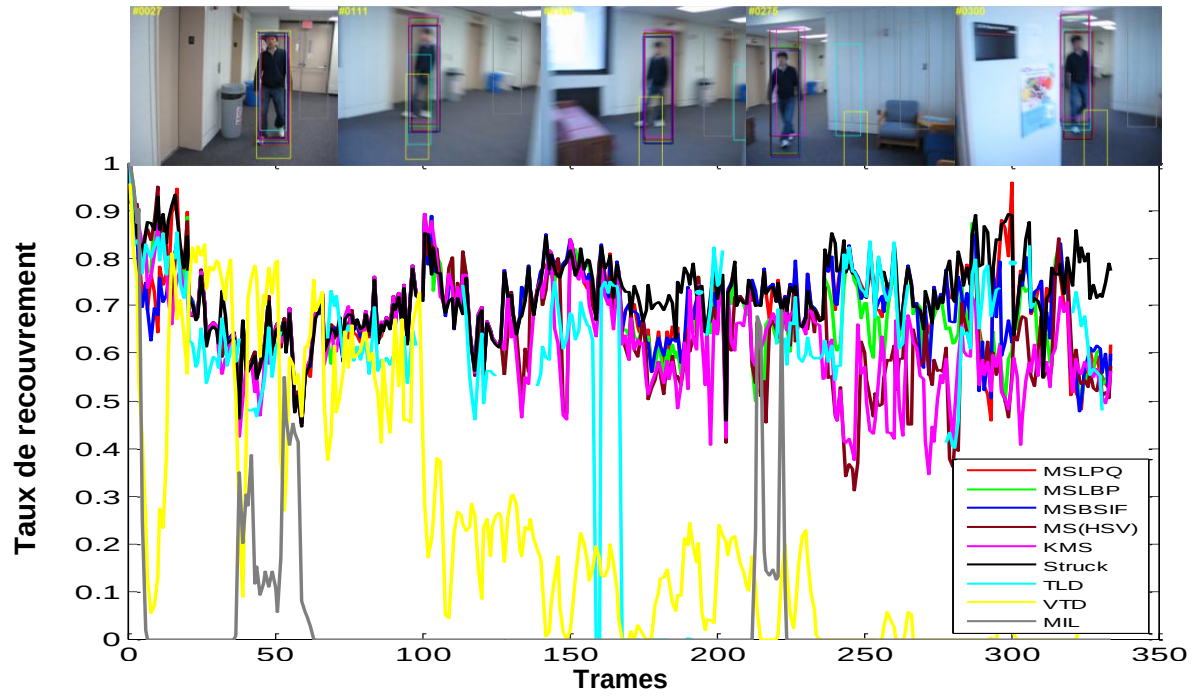


(b)

Figure 5.22 – Comparaison quantitative et qualitative pour les algorithmes proposés et cinq trackers de l'état de l'art sur la séquence Bolt2. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

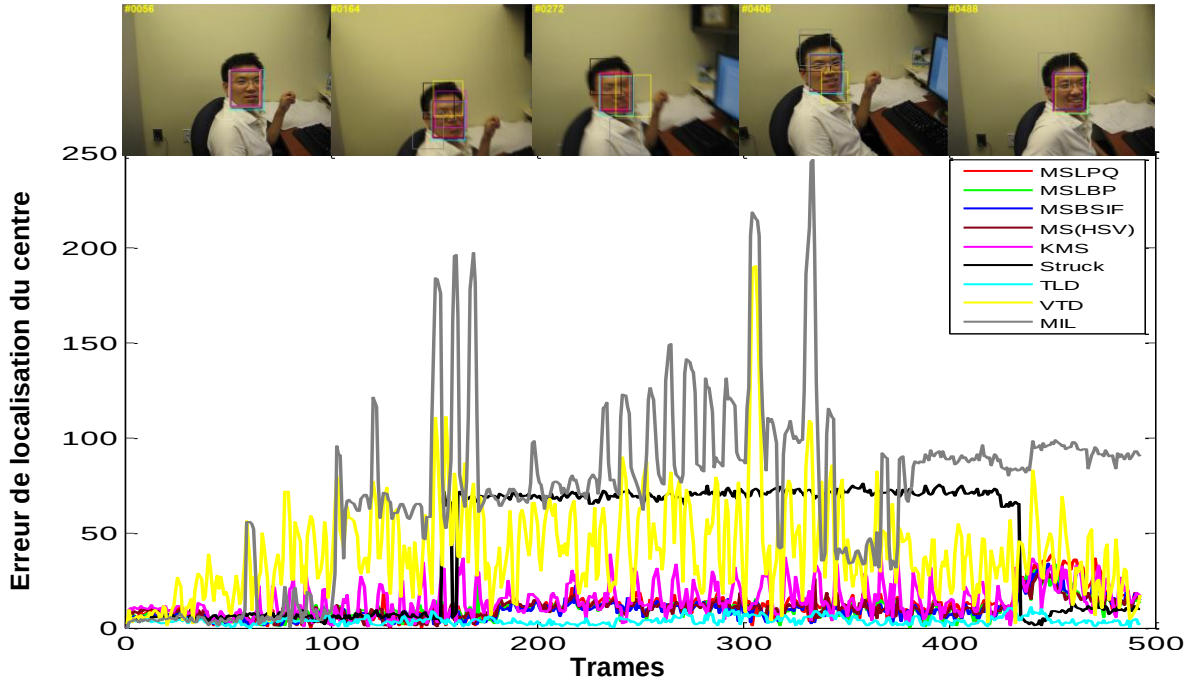


(a)

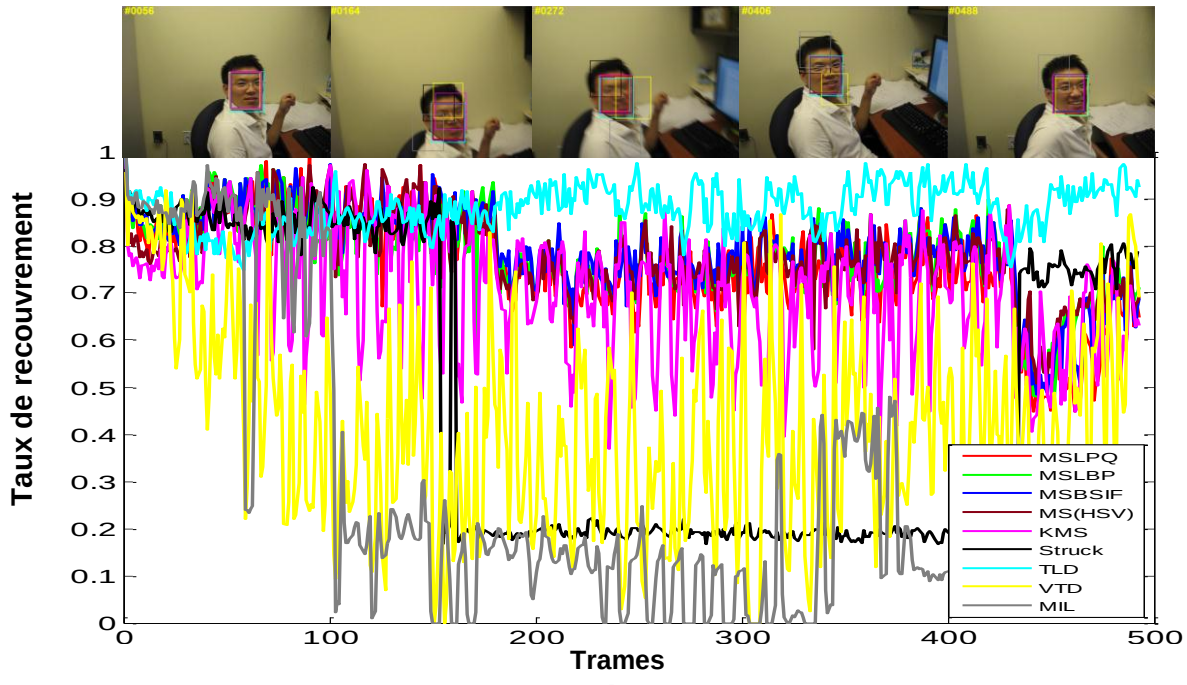


(b)

Figure 5.23 – Comparaison quantitative et qualitative pour les algorithmes proposés et cinq trackers de l'état de l'art sur la séquence BlurBody. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).



(a)



(b)

Figure 5.24 – Comparaison quantitative et qualitative pour les algorithmes proposés et cinq trackers de l'état de l'art sur la séquence BlurFace. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

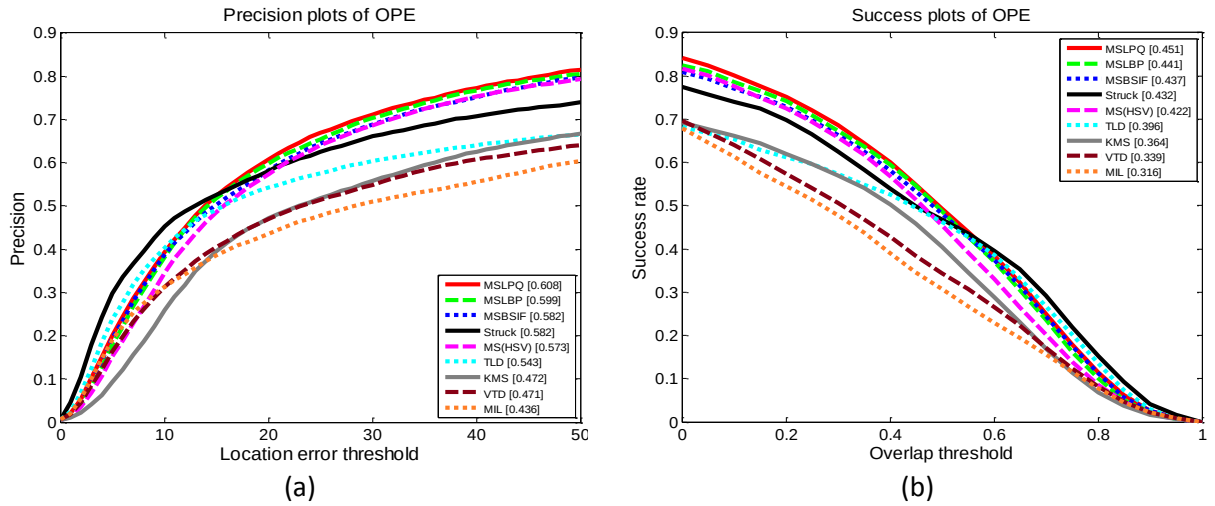
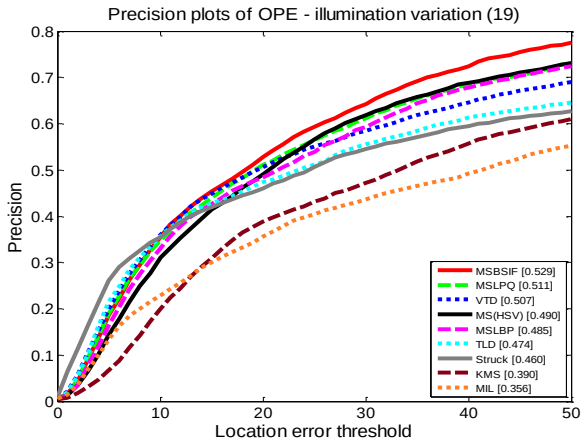


Figure 5.25 – *Precision plots* et *Success plots* de l'OPE pour la comparaison quantitative des algorithmes proposés et cinq trackers de l'état de l'art sur des séquences d'images couleurs des bases OTB.

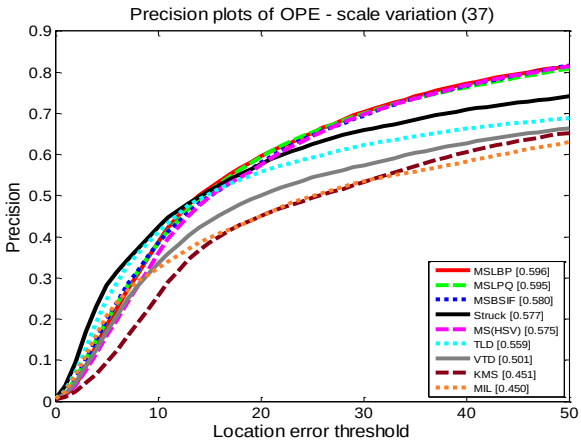
Les résultats globaux des algorithmes proposés etde cinq trackers de comparaison sont représentés par les courbes de *Precision plots* et *Success plots* suivant le protocole d'évaluation OTB dans la figure 5.25. Les critères d'évaluation sont le score de précision et le score de réussite (ACU) comme l'illustre la figure 5.25 (a), (b), respectivement. D'après les résultats obtenus, on peut clairement voir que la performance globale de nos trackers MSLPQ, MSLBP et MSBSIF est supérieure à celle cinq autres trackers, spécialement le tracker Struck qui a une performance supérieure dans benchmark OTB et le tracker traditionnel Mean shift (KMS) qui utilise l'histogramme de couleur RGB. Il est bien évident que MSLPQ semble être plus robuste que MSLBP et MSBSIF et obtient des meilleurs résultats avec un score de précision égale à 0.608 et un score de réussite 0.451. De plus, les résultats des MSLPQ, MSLBP et MSBSIF montrent qu'une grande amélioration des performances du KMS de l'ordre de 13.6%, 12.7% et 11%, respectivement, pour le score de précision et de l'ordre de 7.8%, 7.7% et 7.3%, respectivement, pour le score de réussite. Tandis que, les résultats de tracker MS(HSV) qu'utilise l'histogramme de couleur HSV montre qu'une amélioration de 10.1% pour le score de précision et 5.8% pour le score de réussite.

- **Évaluation basée sur les défis (attributs)**

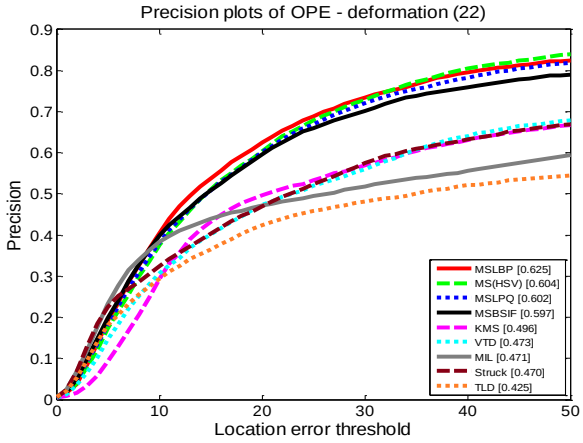
Les figures 5.26 et 5.27 montrent les courbes de *Precision plots* et de *Success plots*, respectivement pour les 11 défis dans les bases de données OTB. Selon les résultats obtenus, les trackers proposés MSLPQ, MSLBP, MSBSIF et MS(HSV) sont plus robustes pour tous les défis en comparaison avec KMS tracker parce que les algorithmes proposés peuvent effectivement tirer parti de caractéristiques complémentaires. Cependant comparées à Struck qui le meilleur tracker dans le benchmark OTB, ils sont moins robustes au défi LR en raison de la petite taille de la cible et le manque des informations spatiales qui mènent à réduire les caractéristiques de textures.


Precision plots of OPE - illumination variation (19)
Precision
Location error threshold
MSBSIF [0.529]
MSLPQ [0.511]
VTD [0.507]
MS(HSV) [0.490]
MSLBP [0.485]
TLD [0.474]
Struck [0.460]
KMS [0.390]
MIL [0.356]

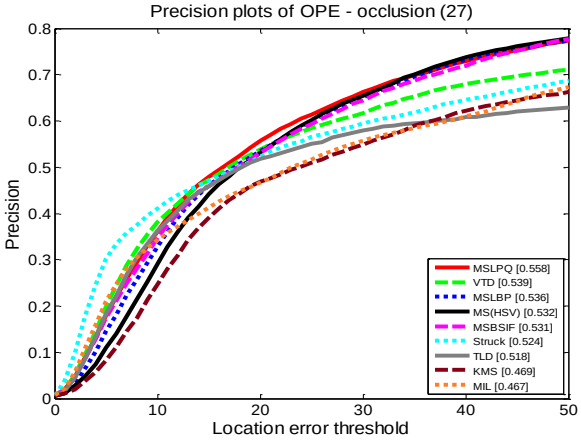
(a)


Precision plots of OPE - scale variation (37)
Precision
Location error threshold
MSLBP [0.596]
MSLPQ [0.595]
MSBSIF [0.580]
Struck [0.577]
MS(HSV) [0.575]
TLD [0.559]
VTD [0.501]
KMS [0.451]
MIL [0.450]

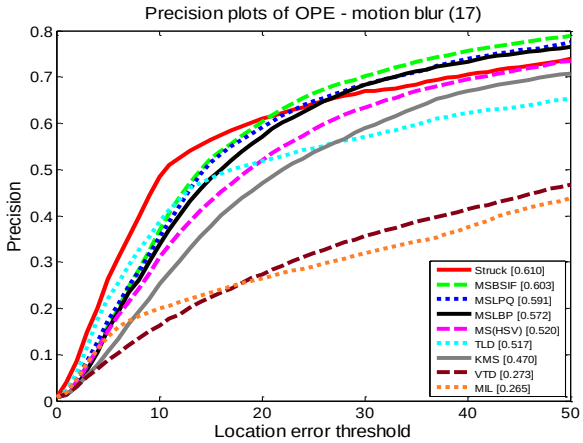
(b)


Precision plots of OPE - deformation (22)
Precision
Location error threshold
MSLBP [0.625]
MS(HSV) [0.604]
MSLPQ [0.602]
MSBSIF [0.597]
KMS [0.496]
VTD [0.473]
MIL [0.471]
Struck [0.470]
TLD [0.425]

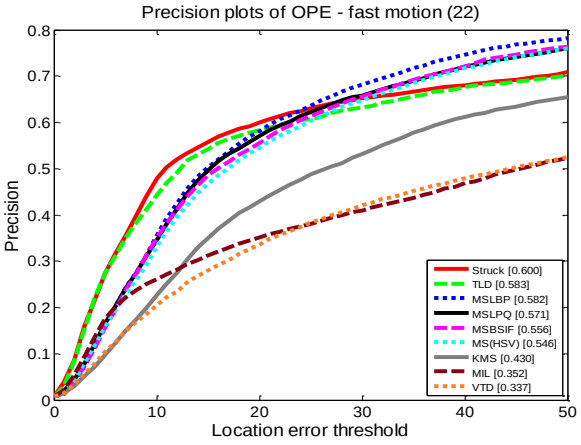
(c)


Precision plots of OPE - occlusion (27)
Precision
Location error threshold
MSLPQ [0.558]
VTD [0.539]
MSLBP [0.536]
MS(HSV) [0.532]
MSBSIF [0.531]
Struck [0.524]
TLD [0.518]
KMS [0.469]
MIL [0.467]

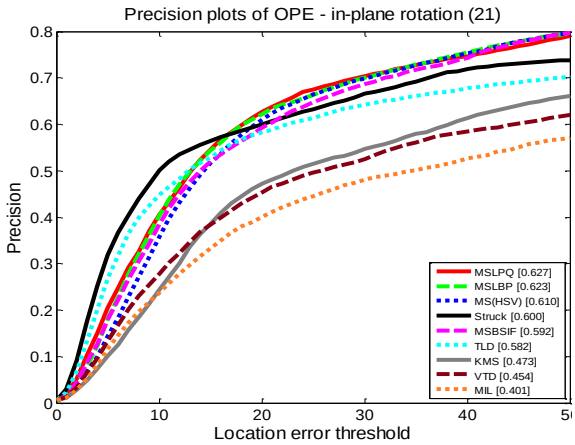
(d)


Precision plots of OPE - motion blur (17)
Precision
Location error threshold
Struck [0.610]
MSBSIF [0.603]
MSLPQ [0.591]
MSLBP [0.572]
MS(HSV) [0.520]
TLD [0.517]
KMS [0.470]
VTD [0.273]
MIL [0.265]

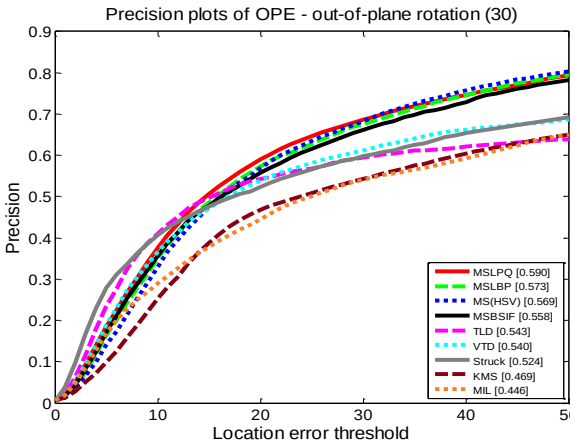
(e)


Precision plots of OPE - fast motion (22)
Precision
Location error threshold
Struck [0.600]
TLD [0.583]
MSLBP [0.582]
MSLPQ [0.571]
MSBSIF [0.556]
MS(HSV) [0.546]
KMS [0.430]
MIL [0.352]
VTD [0.337]

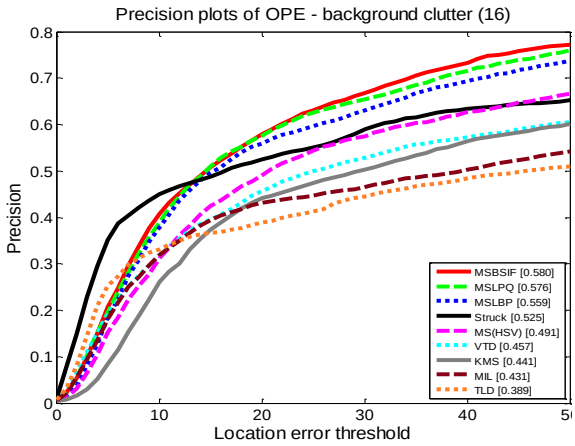
(f)



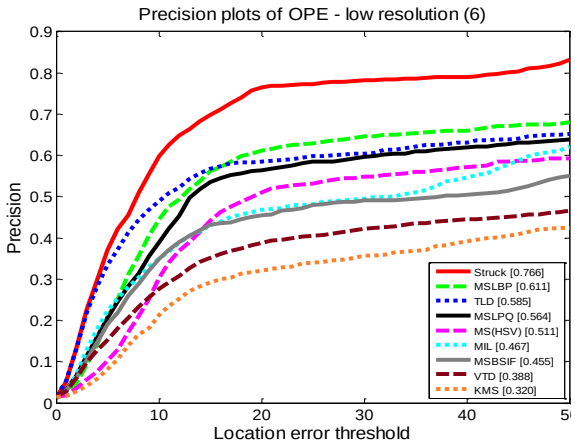
(g)



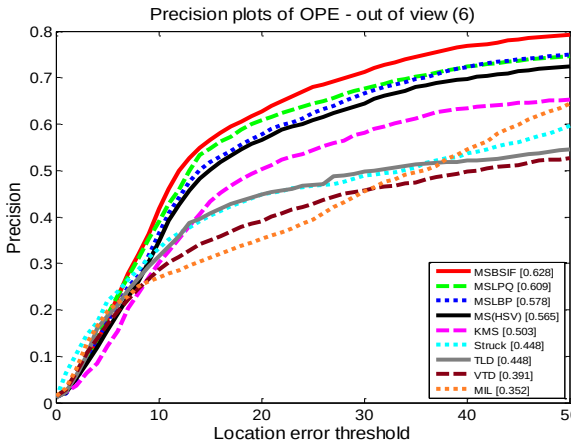
(h)



(i)

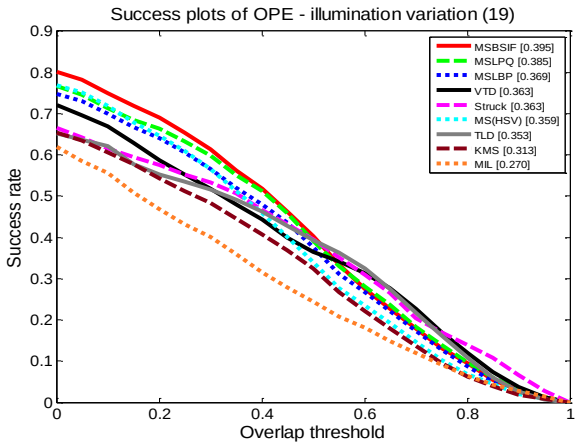


(j)

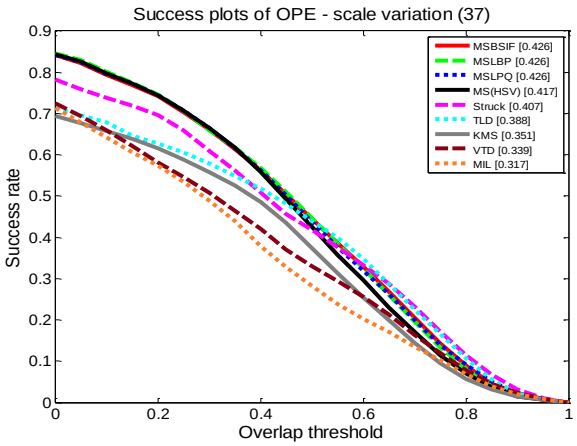


(k)

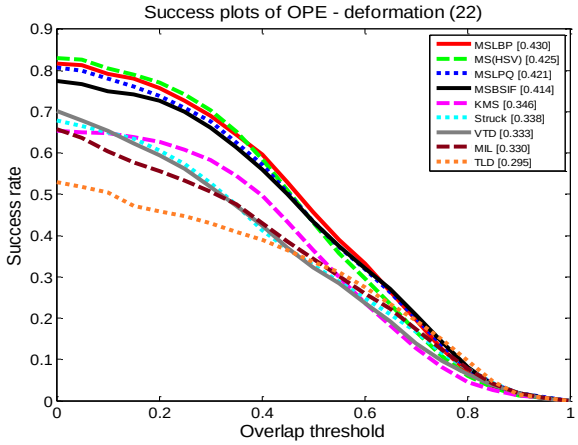
Figure 5.26 – *Precision plots* des différents défis pour les algorithmes proposés et cinq trackers de l'état de l'art : (a) IV, (b) SV, (c) DEF, (d) OCC, (e) MB, (f) FM, (g) IPR, (h) OPR, (i) BC, (j) LR, (k) OV. La valeur apparaissant dans le titre indique le nombre de vidéos


Success plots of OPE - illumination variation (19)
MSBSIF [0.395]
MSLPQ [0.385]
MSLBP [0.369]
VTD [0.363]
Struck [0.363]
MS(HSV) [0.359]
TLD [0.353]
KMS [0.313]
MIL [0.270]
Success rate
Overlap threshold

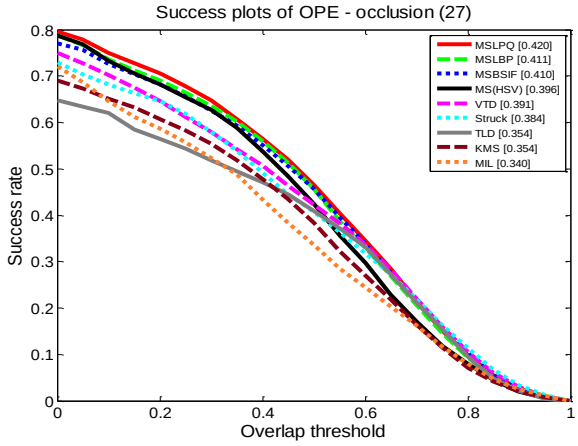
(a)


Success plots of OPE - scale variation (37)
MSBSIF [0.426]
MSLBP [0.426]
MSLPQ [0.426]
MS(HSV) [0.417]
Struck [0.407]
TLD [0.388]
KMS [0.351]
VTD [0.339]
MIL [0.317]
Success rate
Overlap threshold

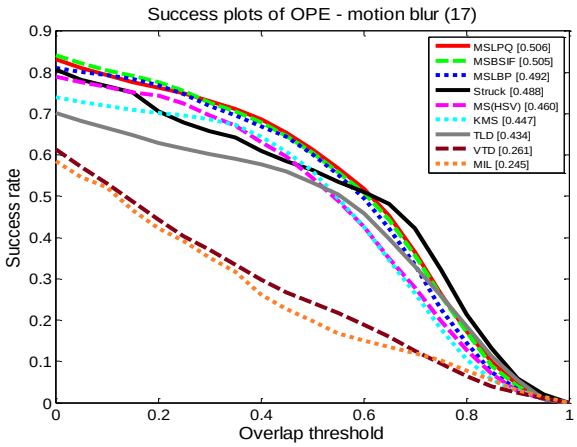
(b)


Success plots of OPE - deformation (22)
MSLBP [0.430]
MS(HSV) [0.425]
MSLPQ [0.421]
MSBSIF [0.414]
KMS [0.346]
Struck [0.338]
VTD [0.333]
MIL [0.330]
TLD [0.295]
Success rate
Overlap threshold

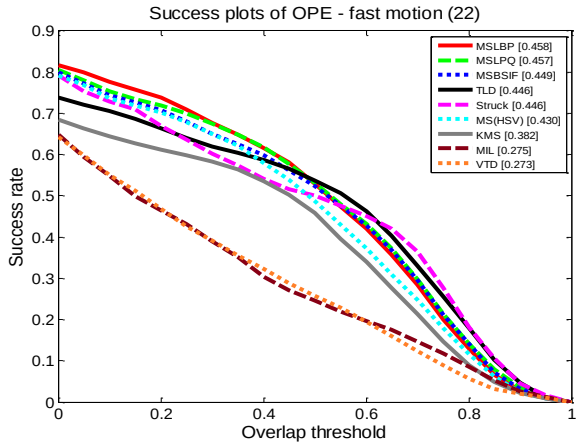
(c)


Success plots of OPE - occlusion (27)
MSLPQ [0.420]
MSLBP [0.411]
MSBSIF [0.410]
MS(HSV) [0.396]
VTD [0.391]
Struck [0.384]
TLD [0.354]
KMS [0.354]
MIL [0.340]
Success rate
Overlap threshold

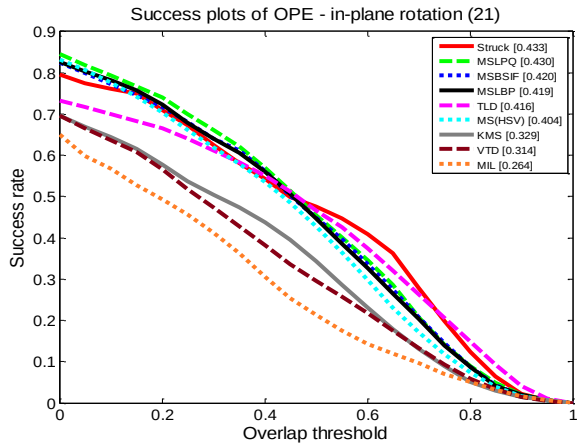
(d)


Success plots of OPE - motion blur (17)
MSLPQ [0.506]
MSBSIF [0.505]
MSLBP [0.492]
Struck [0.488]
MS(HSV) [0.460]
KMS [0.447]
TLD [0.434]
VTD [0.261]
MIL [0.245]
Success rate
Overlap threshold

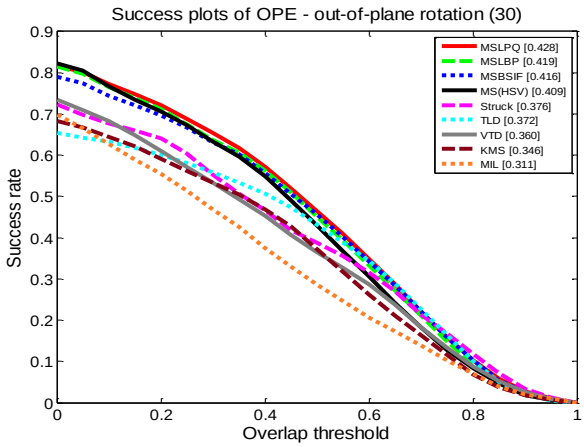
(e)


Success plots of OPE - fast motion (22)
MSLBP [0.458]
MSLPQ [0.457]
MSBSIF [0.449]
TLD [0.446]
Struck [0.446]
MS(HSV) [0.430]
KMS [0.382]
MIL [0.275]
VTD [0.273]
Success rate
Overlap threshold

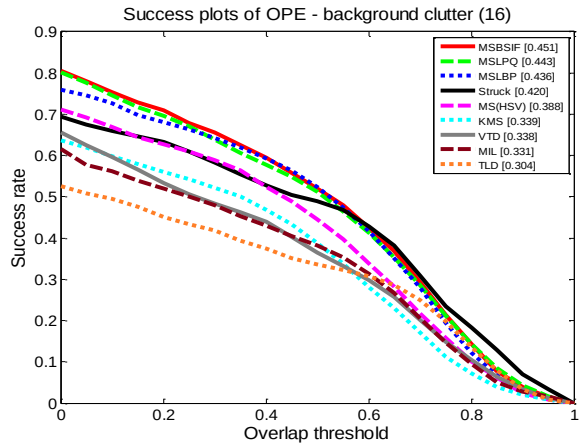
(f)



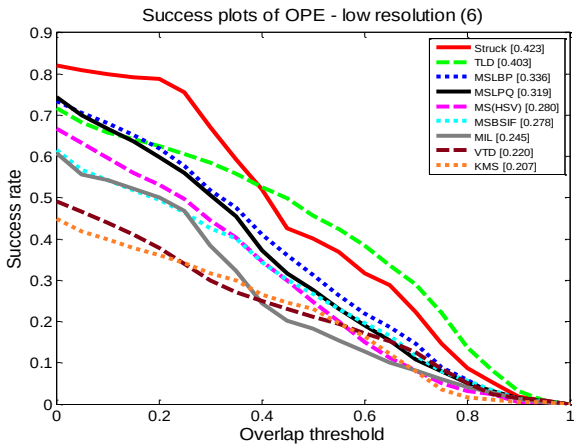
(g)



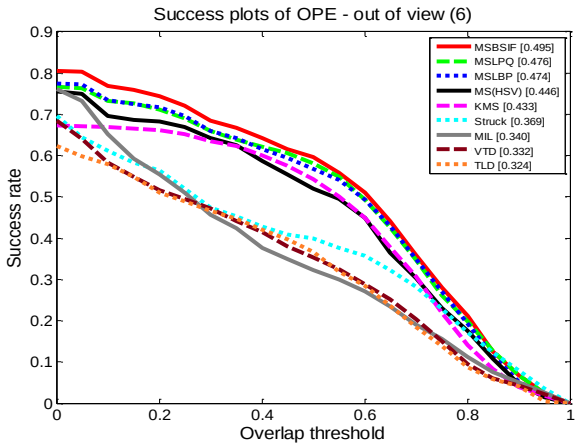
(h)



(i)



(j)



(k)

Figure 5.27 – *Success plots* des différents défis pour les algorithmes proposés et cinq trackers de l'état de l'art : (a) IV, (b) SV, (c) DEF, (d) OCC, (e) MB, (f) FM, (g) IPR, (h) OPR, (i) BC, (j) LR, (k) OV . La valeur apparaissant dans le titre indique le nombre de vidéos

- **Résultats qualitatifs**

Afin d'illustrer plus clairement la comparaison qualitative, la figure 5.28 illustre les résultats de suivi de différents trackers sur plusieurs séquences difficiles. Les séquences Boy et Girl suivent les visages humains sous SV, OCC, MB, FM, IPR, OPR. Les résultats expérimentaux montrent que MSLPQ et Struck obtiennent les meilleures performances comme le montre la figure 5.28 (a), (b). Dans la séquence Boy, le MIL a échoué à suivre la cible à partir de l'image 486, en raison des mouvements rapides et le flou de mouvement sévère. De plus, les MSLPQ, MSLPB et MSBSIF peuvent suivre la cible avec précision lorsqu'il existe un flou de mouvement, un mouvement rapide et un changement de pose (trames 207, 486 et 588), tandis que tous les autres trackers ont échoué. Pour la séquence Girl, le tracker KMS dérive au début (trame 100), parce que la cible a fait une rotation. Dans le même cas TLD et MSLBP ont échoué bien qu'ils récupèrent la cible dans les trames suivantes (voir les trames 116, 324 et 461).

Dans la figure 5.28 (c-e), nous présentrons quelques exemples des trames pour les séquences Subway, David3 et Human8. Les objets à suivre sont les corps humains dans différents environnements. Dans la séquence Subway (figure 5.28 (c)), les deux meilleures trackers sont Struck et MSLPQ, mais MSLPQ est moins robuste que Struck parce qu'il dérive dans la trame 122 en raison de la similarité entre le fond et la cible. Tandis que, les autres trackers dérivent quand l'objet cible occulté par un autre objet similaire ou non (voir les trames 58 et 94). La figure 5.28 (d) représente la séquence David3 qui contient de nombreux défis tels que l'occultation partielle et la similarité entre le fond et la cible. Struck, TLD, VTD, et MIL ne peuvent pas à suivre la cible lorsqu'une occultation partielle se produit, comme illustrée dans la trame130. Cependant, les méthodes proposées et KMS sont robustes, mais les trackers MSLBP, MS(HSV) et KMS sont moins précis (trames 130 et 197). Pour la séquence Human8, la plupart des trackers Struck, TLD, VTD, MIL et KMS perdent la cible à partir des premières trames (trames 47 et 88) car l'apparence de la cible est similaire au fond. Alors que, les méthodes proposées MSLPQ, MSLPB, MSBSIF et MS(HSV) peuvent suivre avec précision la cible comme dans les trames 88 et 122, malgré la présence de nombreux défis tels que le clutter de fond, les changements d'échelle et d'éclairage.

Dans le dernier groupe des séquences testées, les tâches varient de suivi de la poupée en mouvement dans la séquence Lemming au suivi de la voiture dans la route dans les séquences BlurCar4 et BlurCar3. La figure 5.28 (f-h) montre quelques trames de ces séquences. La séquence Lemming est beaucoup plus difficile en raison des changements d'apparence

significatifs, d'occultation totale, mouvement rapide et des rotations hors plan (OPR). Dans cette séquence, tous les trackers réussissent à suivre la cible au début jusqu'à la trame 373. Le VTD échoue comme dans la trame 738 en raison d'une occultation totale, alors que Struck dérive progressivement après la trame 948 et perd la cible totalement dans la trame 963 (trames 1023 et 1108) en raison des changements de pose. Cependant, MSLPQ, MSLBP, MSBSIF, MS(HSV), KMS et MIL ne peuvent suivre qu'une partie de la cible sans la perdre (trame 1108), en raison du changement de taille de la cible. De plus, nous pouvons voir que MSLPQ, MSLBP, MSBSIF et MS(HSV) sont plus précis que KMS et MIL, car ils utilisent les caractéristiques de couleur HSV avec les caractéristiques de texture pour représenter la cible. Dans la séquence BlurCar4 (figure 5.28 (g)), VTD et MIL perdent la cible dès le départ (trame 100). Les autres trackers sont capables de suivre la voiture en mouvement malgré le mouvement dramatiquemennt et le flou de mouvement. Cependant, ces trackers sont sujettes à la dérive pendant le suivi en raison de la similarité entre la voiture cible et le fond ou une autre voiture. En outre, Struck et MSLPQ atteignent les meilleures performances que MSLBP, MSBSIF, KMS et MS(HSV), comme le montre dans la trame 375. La figure 5.28 (h) représente la séquence BlurCar3 qui contient de nombreux défis tels que le mouvement rapide et le flou de mouvement et la cible à suivre est petite. Struck et MSLPQ atteignent les meilleures performances dans toutes les trames. Tandis que, les autres trackers ont échoué à suivre de la cible (voir les trames 83, 240 et 344), malgré TLD, MSLBP et MSBSIF récupèrent la cible par hasard (trame 344).

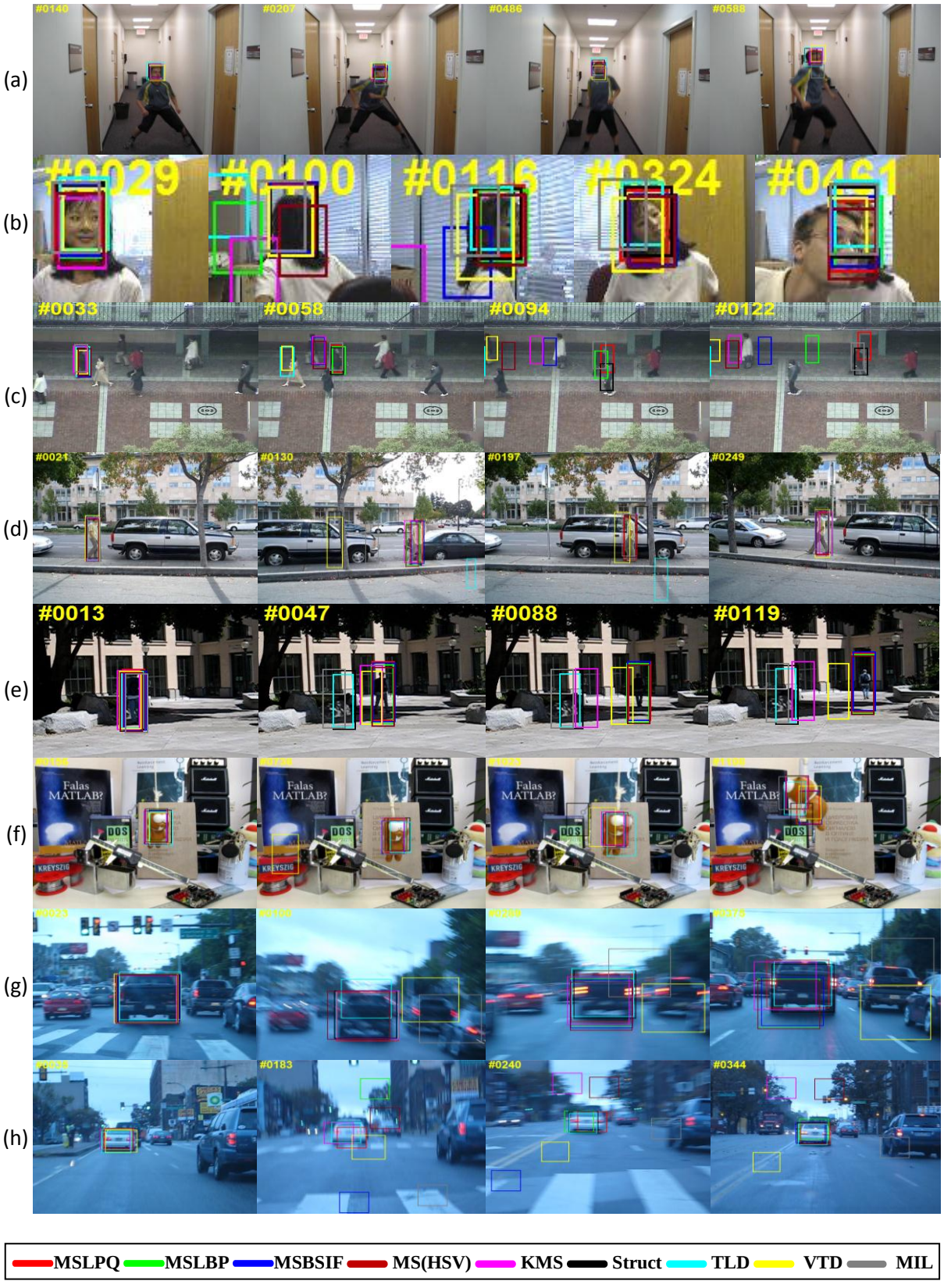


Figure 5.28 – Résultats de suivi de différents trackers sur quelques séquences de bases OTB : (a) Boy, (b) Girl, (c) Subway, (d) David3, (e) Human8, (f) Limming, (g) BlurCar4, et (h) BlurCar3. Les index des trames sont indiqués en haut à gauche de chaque trame.

b) Résultats obtenus par VOT13

- **Résultats quantitatifs**

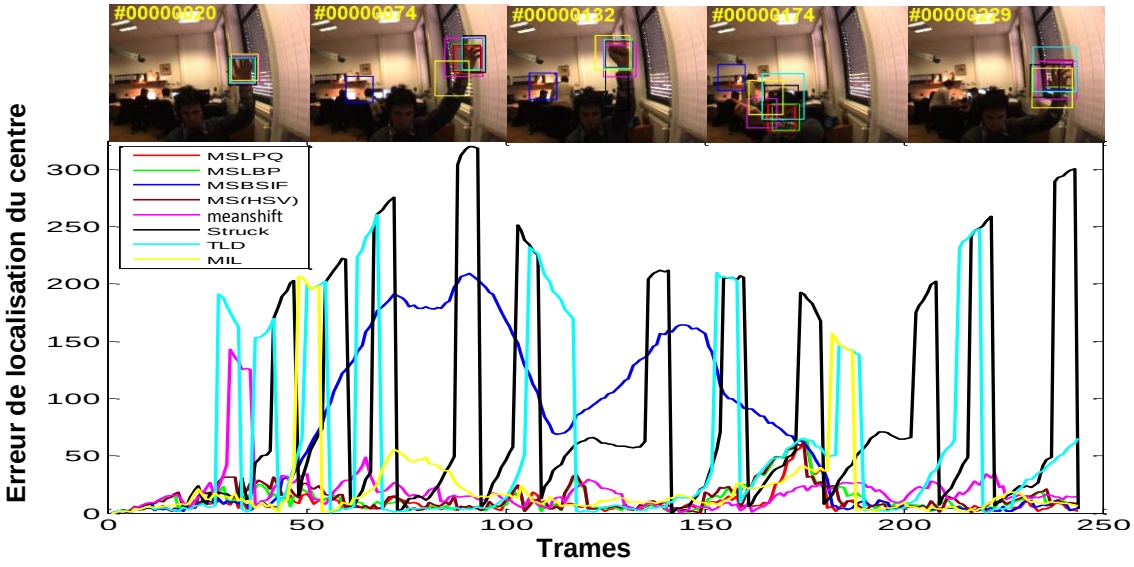
Les performances des algorithmes proposés (MSLPQ, MSLBP, MSBSIF et MS(HSV)) et quatre trackers de la benchmark VOT2013 (Struck, TLD, MIL et meanshift) sur 6 séquences d'images qui contiennent des nombreux défis, sont indiquées dans le tableau 5.6. Le protocole d'évaluation utilisé, qui a défini dans [23], c'est un protocole de la benchmark OTB, qui est l'erreur de localisation du centre CLE et le taux de recouvrement VOR. Le tracker meanshift de la benchmark VOT2013 utilise la technique d'estimation d'échelle. Les résultats montrent que les trackers MSLPQ et Struck obtiennent globalement des performances supérieures aux autres trackers, comme le montre le tableau 5.6. De plus, les trackers proposés sont plus robustes que le meanshift et ils obtiennent de meilleures performances dans la plupart des séquences sélectionnées, en particulier dans les deux séquences Hand et Sunshade.

Tableau 5.6 – Les moyens du taux de recouvrement (VOR) et de l'erreur de localisation du centre (CLE) pour les algorithmes proposés et quatre trackers de l'état de l'art sur quelques séquences de la base VOT2013. Le chiffre en rouge indique la meilleure performance, tandis que le bleu indique la deuxième meilleure performance.

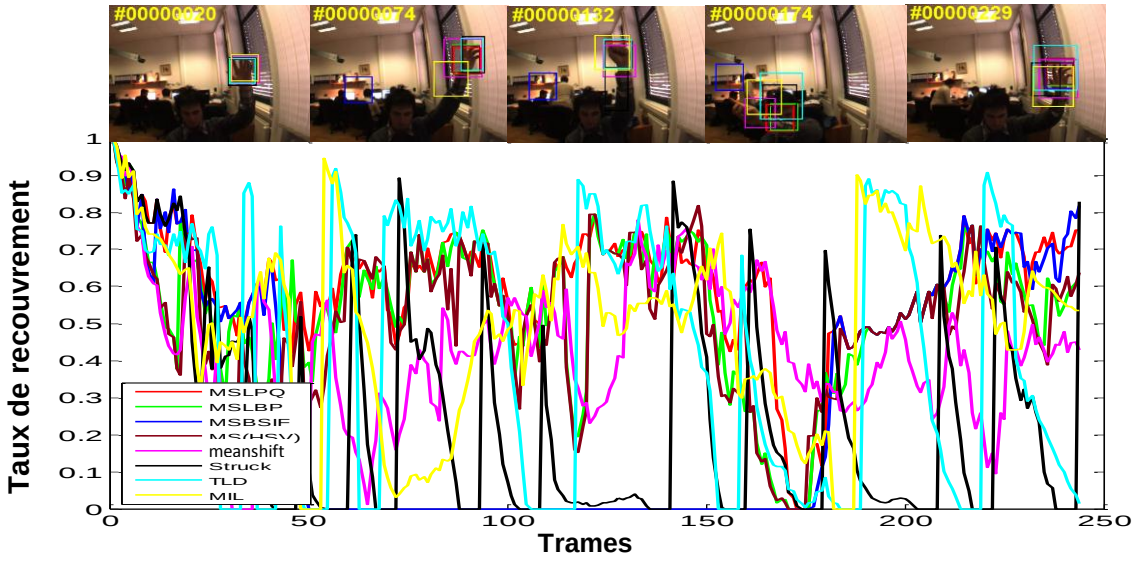
Séquence	Défis	Struck	TLD	MIL	meanshift	MS(HSV)	MSBSIF	MSLBP	MSLPQ
Car	mot, occl, size	0.4189/ 35.678	0.3283/ 10.528	**0.4365/** 34.500	0.3486/ 16.780	0.3595/ **4.9177**	**0.4230/** 11.336	0.3563/ **6.4738**	0.3488/ 35.993
Cup	camera, mot	0.7280/ 7.0119	0.6934/ 9.7220	0.7451/ 4.5212	0.6311/ 9.5196	0.3931/ 42.181	**0.7528/ 4.3930**	0.6128/ 15.561	**0.7601/ 3.7922**
Hand	mot, size	0.5005/ 22.531	0.4313/ 57.093	0.2451/ 85.329	0.4250/ 18.842	0.5060/ 13.387	0.2836/ 72.298	**0.5369/ 12.188**	**0.6055/ 8.4865**
Sunshade	camera, illum, mot	0.6044/ 21.002	0.5873/ 37.287	**0.7498/ 4.7037**	0.4238/ 32.328	0.5998/ 9.9500	**0.7196/ 5.9368**	0.6682/ 7.4885	0.6494/ 7.8312
Juice	camera, mot	0.6132/ **3.8629**	**0.7801/ 3.9560**	**0.6463/** 5.8400	0.5893/ 10.564	0.5872/ 12.394	0.5696/ 13.279	0.6024/ 11.388	0.4900/ 22.356
Torus	mot, size	0.5536/ 27.249	0.5133/ 42.412	0.2397/ 66.755	0.5694/ **13.238**	0.3126/ 46.018	**0.5867/** 14.759	0.4152/ **4.8219**	**0.5958 /** 13.604
Moyenne	---	**0.5698/ 14.688**	0.5556/ 26.968	0.5104/ 28.678	0.4979/ 16.879	0.4597/ 26.405	0.5559/ 20.199	0.5320/ 19.521	**0.5749/ 15.344**

occl : occultation, illum : illumination change, mot: object motion, size : object size change, camera : camera motion

La figure 5.29 représente les résultats obtenus de l'erreur de localisation du centre (CLE) et le taux de recouvrement (VOR) par les algorithmes proposés et les trackers de comparaison pour la séquence d'images Hand. D'après les résultats, on peut voir que les dérives sont beaucoup plus nombreuses pour les trackers MSBSIF, MIL, Struck et TLD, comme montré dans les courbes de l'erreur CLE et du taux VOR. Tandis que, MSLPQ, MSLBP réussissent à suivre tout au long de la séquence, sauf dans la trame 174 où la cible et le fond sont similaires (voir la figure). Le tracker meanshift présenté dans la base VOT2013 trait le problème de la variation d'échelle, comme montré dans les trames présentées, mais il a obtenu de performances inférieures que MSLPQ, MSLBP et MS(HSV), en raison de son utilisation de l'histogramme de couleur RGB qui ne conserve pas les informations spatiales.



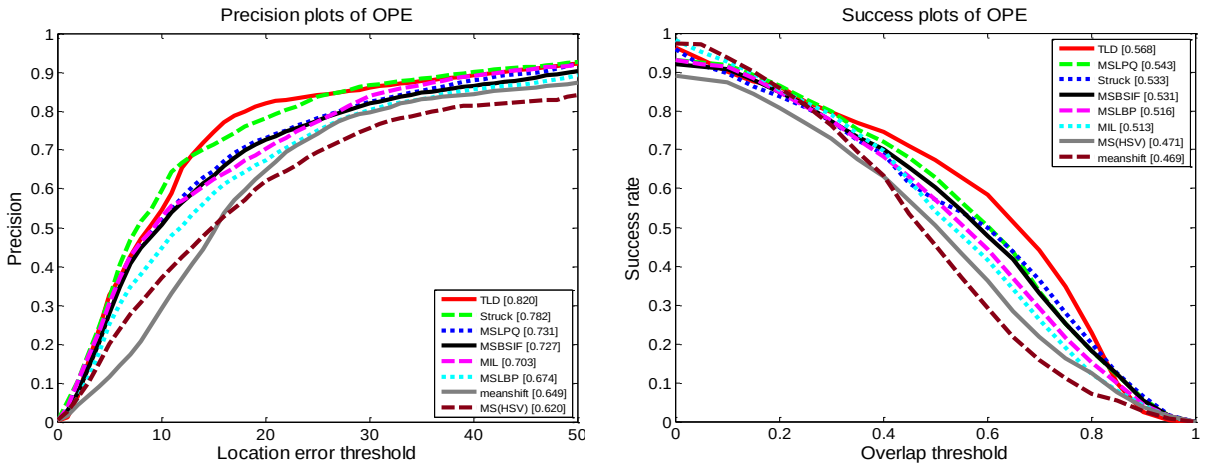
(a)



(b)

Figure 5.29 – Comparaison quantitative avec qualitative pour les algorithmes proposés et quatre trackers de l'état de l'art sur la séquence Hand. (a) Erreur de localisation du centre (CLE), (b) Taux de recouvrement (VOR).

Les performances globales des algorithmes proposés et quatre trackers de comparaison sur la base de données VOT2013 sont représentées par le *Precision plots* et le *Success plots* suivant le protocole d'évaluation OTB dans la figure 5.30. Il est bien évident que, le tracker TLD est plus robuste que les autres trackers et a obtenu de meilleurs résultats avec un score de précision égale à 0.820 et un score de réussite 0.568. Alors que, MSLPQ et Struck atteignent de performances inférieures avec de score de précision égale à 0.731 et 0.782 et de score de réussite 0.543 et 0.533, respectivement. De plus, le tracker MS(HSV) atteigne une performance légèrement meilleure que meanshift. Tandis que, les résultats des MSLPQ, MSLBP et MSBSIF montrent qu'une amélioration des performances du meanshift de l'ordre de 8.2%, 2.5% et 7.8%, respectivement, pour le score de précision et de l'ordre de 7.4%, 4.7% et 6.2%, respectivement, pour le score de réussite. Cela montre l'importance de la représentation de l'objet cible à l'aide un histogramme conjoint couleur-texture. L'intérêt de cette représentation est qu'elle conserve l'information spatiale.



(a) (b)

Figure 5.30 – *Precision plots* et *Success plots* de l'OPE pour la comparaison quantitative des algorithmes proposés et quatre trackers de l'état de l'art sur des séquences d'images de la base de données VOT2013.

- **Résultats qualitatifs**

La figure 5.31 montre quelques résultats qualitatifs de suivi de différents trackers sur des trames des séquences de la base de données VOT2013. Les séquences sont Car, Cup, Juice, Sunhand et Torus. Dans la figure 5.31 (a) la cible est une voiture sous les défis de changement d'échelle, d'occultation et de fond clutter. Struck et MIL sont robustes tout au long de la séquence. meanshift et TLD sont robustes mais moins précis malgré l'estimation d'échelle (trames 249 et 338). MSLPQ, MSLBP et MS(HSV) dérivent en même temps après la trame 273, parce qu'ils ne sont pas capables de suivre la cible lorsqu'il y a un autre objet similaire, bien que MSLPQ et MSLBP utilisent l'information de texture. Tandis que, MSBSIF peut suivre la cible mais il est moins robuste. Dans la séquence Cup, MSLPQ, MSBSIF et Struck peuvent suivre la cible avec une grande précision, bien qu'il y a changement de fond au long de la séquence. MS(HSV) dérive totalement comme le montre dans la trame 163, en raison de la similarité entre la cible et le fond. L'utilisation de l'espace HSV pour Mean shift n'est pas utile dans cette séquence parce que les couleurs prédominantes sont le bleu et le blanc. meanshift, TLD, MSLBP et MIL sont moins robustes et moins précis car ils dérivent dans quelques trames (trames 163 et 227). La figure 5.31(c), (d) montre que, tous les trackers sont robustes tout au long de la séquence. Dans la séquence Juice, lorsque la cible change la taille les trackers MSLPQ, MSLBP, MSBSIF, MS(HSV), Struck et MIL sont peu précis, car ils ne sont pas capables d'estimer l'échelle de la bôite de jus (trames 251 et 404). Cependant, dans la séquence Sunshade le tracker meanshift ne peut pas suivre le visage avec précision (trames 129 et 169), car il est sous l'effet de changement d'éclairage. Dans la séquence torus (figure 5.31 (e)), on peut constater que tous les trackers sont peu robustes et peu précis parce qu'ils dérivent en quelques trames, en raison du mouvement rapide de la cible et la rotation qui mène à changer sa forme.

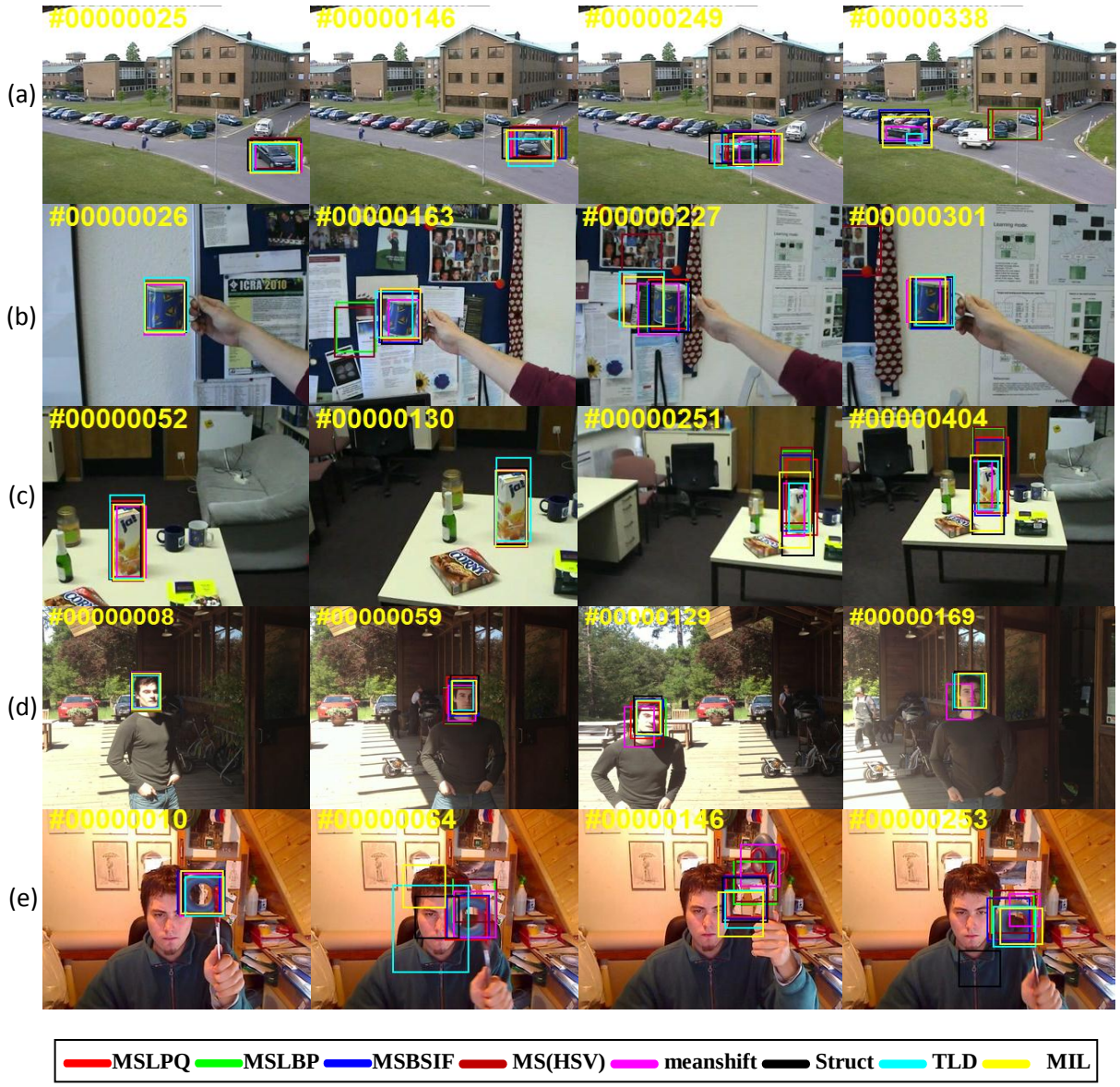


Figure 5.31 – Résultats de suivi de différents trackers sur quelques séquences des bases VOT2013 : (a) Car, (b) Cup, (c) Juice, (d) Sunshade et (e) Torus. Les index de trame sont affichés en haut à gauche de chaque.

5.5 Conclusion

L'objectif de ce chapitre est de montrer, d'abord la comparaison entre le tracker Mean shift et les trackers améliorés Camshift et Kalman Mean shift. Ensuite, l'influence des espaces de couleurs sur les performances de tracker Mean shift. Enfin, l'efficacité de l'utilisation les histogrammes conjoints proposés HSV-LPQ, HSV-LBP et HSV-BSIF pour représenter l'objet cible dans le tracker Mean shift au lieu de l'histogramme de couleur. Les résultats expérimentaux ont été évalués sur les bases de données OTB et VOT2013 suivant le protocole

d'évaluation défini dans [23], qui repose sur deux critères : l'erreur de localisation du centre CLE et le taux de recouvrement VOR. Le tracker traditionnel Mean shift et quatre trackers populaires ont été choisis pour la comparaison.

Les résultats de la comparaison ont montré que les trackers Camshift et Kalman Mean shift sont robustes et précis aux défis de changements d'échelle et d'occultation totale, respectivement. Cependant, la robustesse et la précision de ces trackers sont plus faibles à l'existence des deux ou trois inconvénients en même temps. De plus, Camshift et Kalman Mean shift ne traitent pas le principal inconvénient du Mean shift qui est la similarité entre l'objet et le fond.

L'étude de performance de l'influence des espaces de couleurs sur le tracker Mean shift a démontré que le choix d'un espace de couleur qui donne un histogramme de couleur plus robuste sur une séquence est très important et difficile, puisqu'il dépend de la capacité de distinguer entre l'objet et son fond et de la situation actuelle qui peut varier entre les trames. En effet, la performance globale sur les séquences couleurs des bases OTB détermine que l'espace HSV rend le tracker Mean shift plus robuste que l'espace RGB de 2%.

La validation expérimentale de l'efficacité des nos histogrammes conjoints couleur-texture a montré des résultats très satisfaisants sur les bases de données utilisées OTB et VOT2013. En fait, les trackers proposés MSLPQ, MSLBP et MSBSIF sont plus robustes et précis que le tracker traditionnel Mean shift pour tous les défis. Aussi, ils sont rapides comme Mean shift, car les descripteurs de texture utilisés sont très rapides et la méthode de combinaison est très simple. Nos résultats montrent que MSLPQ, MSLBP et MSBSIF ont atteint une grande amélioration des performances du KMS de l'ordre de 13.6%, 12.7% et 11% pour le score de précision et de l'ordre de 7.8%, 7.7% et 7.3% pour le score de réussite, respectivement, pour les bases OTB, et pour la base VOT2013 de l'ordre de 8.2%, 2.5% et 7.8% pour le score de précision et de l'ordre de 7.4%, 4.7% et 6.2% pour le score de réussite, respectivement. Comparées aux trackes de l'état de l'art, les trackers proposés ont atteint de bonnes performances avec Struck et très élevées avec les autres tracker, pour les bases OTB. Mais pour la base VOT2013, le tracker TLD est plus robuste que les trackers proposés, bien qu'ils ont obtenu de bonnes performances. De plus, l'histogramme conjoint qui utilise les informations de textures extraits par le descripteur LPQ a atteint des performances élevées à celle des autres histogrammes, c-à-d MSLPQ est plus robuste que MSLBP et MSBSIF.

Conclusion

Le problème de détection et suivi d'objet en temps réel dans des séquences d'images est, depuis ces dernières décennies, un thème de recherche très actif dans le monde de la vision par ordinateur. Le problème du suivi d'objet peut s'exprimer en termes de détection de l'objet au sein de chaque image. Le suivi visuel est une sorte d'analyse de mouvement au niveau de l'objet, composé en deux composantes principales: la représentation d'objet et la localisation de l'objet. A ce jour, il n'y a aucun tracker capable de maîtriser toutes les situations difficiles pouvant apparaître lors du suivi d'un objet : changements d'illumination, d'échelle, occultations, mouvement de la caméra, déformation d'objet, la similarité entre l'objet et le fond, etc.

Dans cette thèse, nous nous sommes intéressés au problème du suivi d'objets mobiles dans une séquence d'images en utilisant seulement les informations de couleurs. Nous avons présenté un état de l'art et une classification des méthodes de suivi d'objets selon les techniques et les approches utilisées, ceci nous a permis de choisir l'algorithme Mean shift qui adéquate à la problématique. Puis, nous avons étudié le processus de suivi d'objets par le tracker Mean shift et l'influence des espaces couleurs sur les performances de ce tracker. Enfin, nous avons proposé une nouvelle méthode combinant les caractéristiques de couleur et de texteur pour représenter le modèle d'objet cible, ceci rend le tracker Mean shift plus robuste et plus précis, en particulier en présence de flou de mouvement et de similarité entre l'objet et le fond.

Mean Shift est un algorithme basé sur l'information couleur afin de construire le modèle d'apparence de l'objet cible. La procédure de suivi par cet algorithme est de maximiser la similarité d'apparence itérativement en comparant les histogrammes de l'objet modèle et une fenêtre autour de la position estimée d'objet candidat. Mean shift, utilise l'histogramme de couleurs quantifié par l'espace de couleur RGB pour représenter la distribution des couleurs du modèle cible et les candidats cibles. L'exploitation de l'information couleur pour le suivi visuel est un défi difficile. Pour capturer l'information chromatique, plusieurs espaces de

couleurs utilisées dans la littérature. L'utilisation seulement de l'information de couleurs pour représenter l'objet cible rend l'algorithme Mean shift incapable de détecter l'information spatiale qui est perdue. De plus, ne peut pas distinguer entre l'objet cible et le fond, lorsque la cible a une apparence similaire.

Notre premier objectif de cette thèse était d'étudier dans quelle mesure il est possible de comprendre l'influence des espaces de couleurs sur la robustesse et la précision du tracker Mean shift. Dans notre étude nous avons essayé de comprendre l'influence des espaces de couleurs sur les performances du tracker Mean shift à travers les informations intrinsèques de ce tracker : le coefficient de Bhattacharyya et la carte de rétroprojection qui est une carte de probabilité de l'image. Pour évaluer la qualité du modèle d'apparence du tracker Mean shift nous avons appliqué les espaces les plus utilisés dans le suivi d'objets sur des séquences d'images qui contiennent plusieurs difficultés (changements d'illumination, variation d'échelle, occultation, déformation, flou de bougé, similarité entre objet et fond, objets similaires, etc.). Cette étude montre qu'il n'existe pas un seul espace de couleur approprié pour tous les défis. Par ailleurs, le choix de l'espace colorimétrique peut être très important et très difficile, car l'espace couleur approprié dépend de la situation actuelle qui peut varier entre les trames et les informations de couleur de la cible et du fond. La comparaison des résultats de suivi en utilisant plusieurs espaces de couleurs sur des séquences difficiles (base de donnes OTB) a démontré que l'espace HSV donne de meilleures resultats que les autres, c'est-à-dire, donne un modèle d'apparence plus robuste que les autres.

Le second et principal objectif de cette thèse était de proposer une nouvelle approche qui a pour but d'améliorer l'efficacité et la robustesse de tracker Mean shift par la combinaison des informations couleurs et spatiales, afin de construire un modèle d'apparence plus robuste. Nous avons proposé une méthode de représentation robuste de l'objet cible contre la diversité des facteurs difficiles tels que; flou de mouvement, changements d'illuminations et la similarité entre objet et fond ou objets similaires. Les caractéristiques utilisées pour exploiter les informations spatiales sont les caractéristiques de texture. Pour cela, trois parmi les descripteurs locaux les plus efficaces LBP, LPQ et BSIF ont été utilisés afin de fournir plus de puissance discriminative pour les objets cibles. L'image d'entrée de l'objet est transformée tout d'abord en niveau de gris et l'image de texture a été extraite à travers les descripteurs mentionnés précédents. Ensuite, nous avons utilisé l'espace de couleur HSV au lieu de RGB utilisé dans le tracker Mean shift traditionnel, car HSV possède un certain degré d'invariance contre les changements d'illumination et donne une représentation robuste de la cible comme

nous avons pu le voir dans l'étude des espaces de couleurs. Enfin, nous avons combiné les composantes HSV et l'image de texture de l'objet pour construire les histogrammes conjoints HSV couleur-LPQ texture, HSV couleur-LBP texture et HSV couleur-BSIF texture. De cette manière, l'information locale est représentée à partir de l'objet cible d'une façon discriminante. Les résultats expérimentaux démontrent que les algorithmes proposées MSLPQ, MSLBP et MSBSIF ont atteint de meilleures performances pour tous les défis par rapport à l'algorithme de suivi de Mean shift original, car ils sont capable de tirer parti des caractéristiques de couleur et de texture. De plus, ils sont rapides comme Mean shift, car les descripteurs de texture utilisés sont très rapides et la méthode de combinaison est très simple. Les resultats d'évaluations des différentes combinaisons de couleur HSV et les textures LBP, LPQ et BSIF, montrent que le modèle d'apparence de l'objet cible qui se construit par la couleur HSV et la texture LPQ est plus robuste que les autres puisqu'il plus discriminant et insensible au flou. En comparaison à des autres trackers de l'état de l'art, nos trackers MSLBP, MSLPQ et MSBSIF ont obtenu de meilleures performances pour les deux bases OTB et de bonnes performances pour la base VOT2013.

Perspectives

Les résultats obtenus dans cette thèse nous permettent d'ouvrir les perspectives suivantes:

- Déterminer une approche qui sélectionne automatiquement l'espace colorimétrique le plus approprié afin d'améliorer la qualité de suivi du tracker Mean shift. Le choix de l'espace de couleur peut être fait automatiquement comme une pré-étape avant de construire le modèle d'apparence d'un objet, mais il est plus difficile, car l'espace couleur approprié dépend de la situation actuelle qui peut varier entre les trames, et de la capacité de distinguer l'objet de son fond.
- Intégrer les caractéristiques de couleur et les caractéristiques plus complexes de l'image telles que les caractéristiques de forme HOG (Histogram of Oriented Gradients) et les caractéristiques des réseaux convolutifs appelées "deep features", dans le cadre de suivi par Mean shift. Récemment, l'utilisation des caractéristiques calculées par des réseaux profonds est devenue très populaire à cause de leur grande capacité de représenter des objets.
- Introduire des méthodes de sélection de caractéristiques discriminantes pour construire un modèle d'apparence de l'objet cible. Ces méthodes basées sur les algorithmes d'optimisation telle que : optimisation par essaim de particules (Particle Swarm

Optimization, PSO), algorithme génétique (Genetic Algorithms, GA) et les nouvelles méthodes méta heuristiques.

- Développer l'algorithme de suivi Mean shift et le rendre le plus intelligent; adaptable à la variation d'apparence de l'objet, détecter l'objet cible après l'occultation complexe et plus distinctif entre l'objet et le fond, surtout lorsqu'ils sont similaries. Pour atteindre cet objectif, il faut améliorer les trois points essentiels : 1) Développer un modèle d'apparence d'objet adaptable au contexte de la scène. 2) La proposition des mécanismes pour la détection d'objet après l'occultation. 3) L'utilisation des plusieurs caractéristiques afin de résoudre le problème de la similarité de l'objet et le fond.

Productions Scientifiques

Publications Internationales:

- Medouakh. S, Boumehraz. M, Mean shift based object tracking: the effect of color space, in courrier du savoir, université de Biskra, Algérie, N0 21, pp : 67-74 , 2016.

- Medouakh. S, Boumehraz. M, Terki. N, Improved object tracking via joint color-LPQ texture histogram based mean shift algorithm, signal, image and video processing (SIViP), vol. 12, pp : 583-590, 2018.

Bibliographie

[1] Comaniciu, D., Ramesh,V.,Meer, P.: Kernel based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25(2), 564–577 (2003)

[2] Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. 77(1–3), 125–141(2008)

[3] Mei, X., Ling, H.: Robust visual tracking using l1 minimization.In: IEEE 12th International Conference on Computer Vision, pp. 1436–1443 (2009)

[4] Kwon, J., Lee, K.M.:Visual tracking decomposition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2010, 1269–1276 (2010).

[5] Zhang, S., Yao, H., Zhou, H., Sun, X., Liu, S.: Robust visual tracking based on online learning sparse representation. Neurocomputing 100, 31–40 (2013).

[6] Hare, S., Saffari, A., Torr, P.H.: Struck: structured output tracking with kernels. In: IEEE International Conference on Computer Vision (ICCV), pp. 263–270 (2011)

[7] Babenko, B., Yang, M.H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern. Anal. Mach. Intell. 33(8), 1619–1632 (2011)

[8] Kalal, Z.,Mikolajczyk, K.,Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. 34(7), 1409–1422 (2012)

[9] Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37(3), 583–596 (2015)

[10] Ma, C., Yang, X., Zhang, C., Yang, M-H.: Long-term correlation tracking. In: CVPR (2015)

[11] Lan, X.Y., Yuen, P.C., Challappa, R.: Robust MIL-based feature template learning for object tracking. In: The Thirty First AAAI Conferences on Artificial Intelligence (AAAI), pp 4118–4125 (2017)

[12] Porikli, F., Yilmaz, A.: Object Detection and Tracking, Video Analytics for Business Intelligence, Springer Berlin Heidelberg, vol. 409, pp. 3-41 (2012).

[13] Wang, Q., Fang, J., Yuan, Y.: Multi-cue based tracking. Neurocomputing 131, 227–236 (2014)

[14] Ning, J., Zhang, L., Zhang, D., Wu, C.: Robust object tracking using joint color-texture histogram. Int. J. Pattern Recognit. Artif. Intell. 23(07), 1245–1263, (2009).

[15] Lan, X.Y., Ma, A. J., Yuen, P.C.: Multi-cue visual tracking using robust feature-level fusion based on joint sparse representation. In: CVPR, pp 1194–1201, (2014).

[16] Dou, J., Li, J.: Robust visual tracking based on joint multi-feature histogram by integrating particle filter and mean shift. Optik Int. J. Light Electron Opt. 126, 1449–1456 (2015).

[17] Lan,X.Y.,Zhang, S.,Yuen, P.C.: Robust joint discriminative feature learning for visual tracking. In: IJCAI, pp. 3403–3410 (2016).

[18] Yuan, Y., Xiong, Z., Wang, Q.: An incremental framework for video-based traffic sign detection, tracking, and recognition. IEEE Trans. Intell. Transp. Syst. 18(7), 1918–1929 (2017).

[19] Xiaorong, P., Zhihu, Z.: A more robust mean shift tracker on joint color-CLTP histogram. Int. J. Image Gr. Signal. Process. 4(12), 34–42 (2012).

[20] Tavakoli, R.H., Moin, M.S., Heikkila, J.: Local similarity number and its application to object tracking. Int. J. Adv. Robot. Syst. 10, 184 (2013).

[21] Yilmaz, A., Javed, O., and Shah, M.: Object tracking: A survey, ACM Computing Surveys (CSUR), vol. 38, no. 4, p. 13 (2006).

[22] Cannons, K.: A review of visual tracking. Technical report, York University CSE (2008).

[23] Y.Wu, J. Lim, and M.-H. Yang.: Online object tracking: A benchmark. In CVPR (2013).

[24] Li, X.; Hu, W.M.; Shen, C.H.; Zhang, Z.F.; Dick, A.; Hengel, A.V.D.: A Survey of Appearance Models in Visual Object Tracking. ACM Trans. Intell. Syst. Technol, 4, 58 (2013).

[25] Q. Wang, F. Chen, W. Xu, M. Yang, An experimental comparison of online object tracking algorithms, in: Proceedings of SPIE: Image and Signal Processing, 2011, pp. 1-11.

[26] Zhong, W., Li, H., Yang, M.: Robust object tracking via sparsity-based collaborative model. In: IEEE Conference Computer Vision and Pattern Recognition, pp. 1838–1845 (2012)

[27] Wren, C. R., Azarbayejani, A., Darrell, T., Pentland, A. P., Pfinder.: Real-Time Tracking of the Human Body. IEEE. Transactions on Pattern Analysis and Machine Intelligence, 19(7): 780-785 (1997).

[28] Stauffer, C., Grimson, W. E. L.: Learning Patterns of Activity Using Real-time Tracking, IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), 747–757 (2000).

[29] Lucas, B. D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In IJCAI, pp, 674-679 (1981).

[30] Viola, P., Jones, M.: Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR 2001, IEEE Computer Society, vol 1, pp 511–518 (2001).

[31] Dalal, N. and Triggs, B.: Histograms of oriented gradients for human detection. IEEE International Conference on Computer Vision and Pattern Recognition 1, 886–893 (2005).

[32] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. TPAMI (2010).

[33] Black, M. J. Jepson, A. D.: EigenTracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision, 26(1):63–84 (1998).

[34] Babenko, B., Yang,M. H., Belongie,S.: Visual tracking with online multiple instance learning, in: CVPR (2009).

[35] Zhu, S., Yuille, A,: Region competition: unifying snakes, region growing, and bayes/mdl for multiband image segmentation. IEEE Trans. Patt. Analy. Mach. Intell. 18, 9, 884–900 (1996).

[36] Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. Int. J. Comput. Vision 46, 3, 223–247 (2002).

[37] Elgammal, A.,Duraiswami, R.,Harwood, D., Anddavis, L.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of IEEE 90, 7, 1151–1163 (2002).

[38] Cootes, T. F., Edwards, G. J., Taylor, C. J.: Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681–685 (2001).

[39] Mughadam, B. and Pentland, A. Probabilistic visual learning for object representation. IEEE Trans. Patt. Analy. Mach. Intell. 19, 7, 696–710 (1997).

[40] Avidan, S.: Support vector tracking. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 184–191 (2001).

[41] Paschos, G.: Perceptually uniform color spaces for color texture analysis: an empirical evaluation. IEEE Trans. Image Process. 10, 932–937 (2001).

[42] Van, S. K., Gevers, T., Snoek, C.: Evaluation color descriptors for object and scene recognition. IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 9, (2010).

[43] Yang, H., Shao, L., Zheng, F.,Wang, L., Song, Z.: Recent advances and trends in visual tracking: a review. Neurocomputing 74, 3823–3831 (2011).

[44] Isard, M., Blake, A.: Condensation, conditional density propagation for visual tracking. International journal of computer vision, vol. 29, no. 1, pages 5-28 (1998).

[45] Fang, H., Kim, J.W., Jang, J.W.: A Fast Snake Algorithm for Tracking Multiple Objects. (2011).

[46] Lowe, D.G.: Object recognition from local scale-invariant features. In Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, volume 2, pages 1150_1157. IEEE (1999).

[47] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, no. 10, pp 1615-1630 (2005).

[48] Haralick, R., Shanmugam, B., and Dinstein, I.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. 33, 3, 610–622 (1973).

[49] Manjunath, Ma, B. W.: Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8), 837–842 (1996).

[50] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 972–987 (2002).

[51] Lun Zhang, Rufeng Chu, Shiming Xiang, Shengcai Liao, and Stan Z Li. Face detection based on multi-block lbp representation. In Advances in biometrics, Pp 11–18. Springer, (2007).

[52] Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. Trans. Img. Proc. 19(6) :1635–1650 (2010).

[53] Sun, D., Roth, S., and Black, M.: Secrets of optical flow estimation and their principles, Proc. IEEE Conf. On Computer Vision and Pattern Recognition, San Francisco, CA (2010).

[54] Moravec, H.: Visual mapping by a robot rover. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). 598–600 (1979).

[55] Harris, C. and Stephens, M. A.: combined corner and edge detector. In 4th Alvey Vision Conference. 147–151 (1988).

[56] Shi, J. and Tomasi, C.: Good features to track. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 593–600 (1994).

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

[58] Bay, H., Tuytelaars, T. et Gool, L. V.: SURF : Speeded up robust features. Dans Proceedings of 9th European Conference on Computer Vision, pages 404–417 (2006).

[59] Wren, C., Azarbayejani, A., Darrell, T. et Pentland, A.P., Pfinder : Realtime tracking of the human body. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pages 780–785, 1997.

[60] Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition - CVPR 1999, volume 2, pages 246–252 (1999).

[61] Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04), Washington, DC, USA. IEEE Computer Society, Vol. 02, pp 28–31 (2004).

[62] Bouwmans, T. and El Baf, F.: Modeling of dynamic backgrounds by type-2 fuzzy gaussians mixture models. MASAUM Journal of of Basic and Applied Sciences, 1(2) :265–276 (2009).

[63] Zhao, Z., Bouwmans, T., Zhang, X., and Fang, Y.: A fuzzy background modeling approach for motion detection in dynamic backgrounds. In Multimedia and Signal Processing, pp. 177–185 (2012).

[64] Oliver, N.M., Rosario, B., Pentland, A.P.: A Bayesian Computer Vision System for Modeling Human Interactions. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pages 831–843 (2000).

[65] Dong, Y. and DeSouza, G. N.: Adaptive learning of multi-subspace for foreground detection under illumination changes, Computer Vision and Image Understanding, vol. 115, no. 1, pp. 31–49 (2011).

[66] Comanicu, Meer, D. P.: Mean shift : A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, pp 603–619 (2002).

[67] Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function with applications in pattern recognition. IEEE Transactions on Information Theory, IT-21, pages 32–40 (1975).

[68] Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell., 17(8):790–799 (1995).

[69] Wu, Z. and Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its applications to image segmentation. IEEE Trans. Patt. Analy. Mach. Intell. 11, 1101–1113 (1993).

[70] Shi, J. and Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Patt. Analy. Mach Intell. 22, 8, 888–905 (2000).

[71] Kass, M., Witkin, A., and Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vision 1, 321–332 (1988).

[72] Caselles, V., Kimmel, R., and Sapiro, G.: Geodesic active contours. In IEEE International Conference on Computer Vision (ICCV). 694–699 (1995).

[73] Ronfard, R.: Region based strategies for active contour models. Int. J. Comput. Vision 13, 2, 229–251 (1994).

[74] Yilmaz, A., Li, X., and Shah, M.: Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Patt. Analy. Mach. Intell. 26, 11, 1531–1536 (2004).

[75] Freund, Y. and Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Computat. Learn. Theory. 23–37 (1995).

[76] Viola, P., Jones, M., and Snow, D.: Detecting pedestrians using patterns of motion and appearance. In IEEE International Conference on Computer Vision (ICCV). 734–741 (2003).

[77] Vapnik, V.: The Nature of Statistical Learning Theory, N-Y, springer-verlag edition, (1995).

[78] Papageorgiou, C., Oren, M., Poggio, T.: A General Framework for Object Detection, Proc. Sixth IEEE Int'l Conf. Computer Vision, pp. 555-562 (1998).

[79] Tomasz, M., Abhinav, G. Alexei, E.: Ensemble of exemplar-SVMs for object detection and beyond. in Proc. ICCV (2011).

[80] Hu, W., Tan, T., Wang, L., and Maybank, S.: A survey on visual surveillance of object motion and behaviors. IEEE Trans. on Syst., Man, Cybern. C, Appl. Rev 34, 3, 334–352 (2004).

[81] He, S., Yang, Q., Lau, R., Wang, J., Yang, M.-H.: Visual tracking via locality sensitive histograms. in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 2427–2434 (2013).

[82] Sethi, I. and Jain, R. Finding trajectories of feature points in a monocular image sequence. IEEE Trans. Patt. Analy. Mach. Intell. 9, 1, 56–73 (1987).

[83] Salari, V. and Sethi, I. K.: Feature point correspondence in the presence of occlusion. IEEE Trans. Patt. Analy. Mach. Intell. 12, 1, 87–91 (1990).

[84] Veenman, C., Reinders, M., and Backer, E.: Resolving motion correspondence for densely moving points. IEEE Trans. Patt. Analy. Mach. Intell. 23, 1, 54–72 (2001).

[85] Shafique, K., and Shah, M.: A non-iterative greedy algorithm for multi-frame point correspondence. In IEEE International Conference on Computer Vision (ICCV). 110–115 (2003).

[86] Scovanner, P., and Tappen, M.: Learning Pedestrian Dynamics from the Real World. In The International Conference on Computer Vision (ICCV). (2009).

[87] Kalman, R. E.: A new approach to linear filtering and prediction problems. Journal of Fluids Engineering, 82(1):35–45 (1960).

[88] Broida, T. and Chellappa, R.: Estimation of object motion parameters from noisy images. IEEE Trans. Patt. Analy. Mach. Intell. 8, 1, 90–99 (1986).

[89] Beymer, D. and Konolige, K.: Real-time tracking of multiple people using continuous detection. In IEEE International Conference on Computer Vision (ICCV) Frame-Rate Workshop (1999).

[90] Ponsa, D., López, A., Serrat, J., Lumbreras, F., Graf, T. : Multiple vehicle 3d tracking using an unscented kalman. In Intelligent Transportation Systems, pp : 1108-1113. IEEE, (2005).

[91] Robert, K.: Night-time traffic surveillance: A robust framework for multi-vehicle detection, classification and tracking. In Advanced Video and Signal Based Surveillance (2009).

[92] Gordon, N. J., Salmond, D. J., Smith, A. F. M.: Novel approach to nonlinear / non–Gaussian Bayesian state estimation. IEE Proceedings, Part F, 140(2) :107–113, Apr (1993).

[93] Kitagawa, G. Monte carlo filter and smoother for non-gaussian nonlinear state space models. Journal of computational and graphical statistics, 5(1):1–25 (1996).

[94] Arnaud, E., Memin, E. et Cernuschi-Frias, B. Conditional filters for image sequence-based tracking-application to point tracking. Image Processing, IEEE Transactions on, 14(1) : 63-79, (2005).

[95] Chang, Y. L. and Aggarwal, J. K.: 3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory. In Proceedings of the IEEE Workshop on Visual Motion , pages 268–273. IEEE (1991).

[96] Rasmussen, C. and Hager, G. D.: Probabilistic data association methods for tracking complex visual objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6) :560–576 (2001).

[97] Reid, D. B.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control , 24(6) :843–854 (1979).

[98] Cox, I. and Hingorani, S.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Trans. Patt. Analy. Mach. Intell. 18, 2, 138–150 (1996).

[99] Murty, K. G.: Letter to the editor an algorithm for ranking all the assignments in order of inreasing cost. Operations Researh, 16(3) : 682–687 (1968).

[100] Birchfield, S.: Elliptical head tracking using intensity gradients and color histograms. In IEEE International Conference on Computer Vision and Pattern Recognition, Washington, DC, USA (1998).

[101] Schweitzer, H., Bell, J. W. and Wu, F.: Very fast template matching. In European Conference on Computer Vision, pages 358–372, London, UK (2002).

[102] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., et al.: The visual object tracking vot2015 challenge results. In: ICCV2015 Workshops, Workshop on visual object tracking challenge (2015).

[103] Santner, J., Leistner, C., Saffari, A., Pock, T. and Bischof, H.: PROST: Parallel robust online simple tracking. In CVPR (2010).

[104] Tomasi, C. and Kanade, T.: Detection and tracking of point features. Technical Report CMU-CS-91-132, School of Computer Science, Carnegie Mellon University, USA (1991).

[105] Baker, S. and Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision, 56(3):221–255 (2004).

[106] Takacs, G., Chandrasekhar, V., Tsai, S., Chen, D., Grzeszczuk, R., Girod, B.: Unified real-time tracking and recognition with rotation-invariant fast features. In Computer Vision and Pattern Recognition (CVPR), pp 934_941. IEEE (2010).

[107] Vojir, T and Jiri, Matas. : The Enhanced Flock of Trackers. Registration and Recognition in Images and Videos - Studies in Computational Intelligence, Springer (2014).

[108] Comaniciu. D, Ramesh. V, Meer. P.: Real-time tracking of nonrigid objects using mean shift. IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp. 142_149 (2000).

[109] Elgammal, A., Duraiswami, R. and Davis, L. S.: Probabilistic tracking in joint feature-spatial spaces, 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2003).

[110] Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 798–805. IEEE (2006).

[111] Yang, C., Duraiswami, R., Davis, L.: Efficient mean-shift tracking via a new similarity measure. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, vol. 1, pp. 176–183 (2005).

[112] Tavakoli, H. R., Moin, M. S., and Heikkila, J.: Local similarity number and its application to object tracking. Int J Adv Robot Syst, 10(184):1-6 (2013).

[113] Karavasilis, C. Nikou, and A. Likas.: Visual Tracking by Adaptive Kalman Filtering and Mean Shift. Artiécial Intelligence: Theories, Models and Applications, pp. 153,162, (2010).

[114] Zhou, T., Yan, Y. Video target tracking based on mean shift algorithm with kalman filter. In: Natural Computation (ICNC), 10th International Conference on. IEEE; pp. 980–984 (2014).

[115] Tang D. and Zhang, Y.J.: Combining Mean-Shift and Particle Filter for Object Tracking," In Sixth Int. Conf. on Image and Graphics (ICIG), pp. 771-776 (2011).

[116] Iswanto, I. A., Li, B.: Visual Object Tracking Based on Mean-shift and Particle-Kalman Filter. Procedia Computer Science, Volume 116, pp. 587-595 (2017).

[117] Bradski, G.R.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal (1998).

[118] Allen, John G., Richard YD Xu, and Jesse S. Jin.: Object tracking using camshift algorithm and multiple quantized feature spaces. Proceedings of the Pan-Sydney area workshop on Visual information processing. Australian Computer Society, Inc (2004).

[119] Nouar, O.D., Ali, G., Raphael, C.: Improved object tracking with Camshift algorithm. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 657-660. IEEE, Piscataway (2006).

[120] Wang, Q., Chen, F., Xu, W., and Yang, M.: Object tracking via partial least squares analysis. IEEE Transactions on Image Processing 21, 10, 4454–4465 (2012)

[121] Li, X., Hu, W., Zhang, Z., Zhang, X., Luo, G.: Robust visual tracking based on incremental tensor subspace learning. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, (2007).

[122] Wen, J., Gao, X.B., Li, X., Tao, D.C.: Incremental learning of weighted tensor subspace for visual tracking, in: IEEE International Conference on Systems, Man, and Cybernetics, pp: 3788-3793 (2009).

[123] Yang, H., Song, Z., Chen, R.: An incremental PCA-HOG descriptor for robust visual hand tracking, in: ISVC (2010).

[124] Shirazi, S., Harandi, MT., Lovell, BC.: Object tracking via non-Euclidean geometry: a Grassmann approach. In: Proceedings of the IEEE winter conference on applications of computer vision, Steamboat Springs, pp.901–908 (2014).

[125] Shirazi, S., Sanderson, C., McCool, C. and Harandi, M. T.: Bags of affine subspaces for robust object tracking. In International Conference on Digital Image Computing: Techniques and Applications (DICTA) (2015).

[126] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1):267–288 (1996).

[127] Mei, X., Ling, H., Wu, Y., Blasch, E. and Bai, L.: Minimum error bounded efficient tracker with occlusion detection. In CVPR (2011).

[128] Wang, Q., Chen, F., Xu, W. and Yang, M-H.: Online discriminative object tracking with local sparse representation. In IEEE Workshop on Application of Computer Vision (WACV) (2012).

[129] Jia, X., Lu, H. and Yang, M-H.: Visual tracking via adaptive structural local sparse appearance model. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012).

[130] Zhang, T., Ghanem, B., Liu, S. and Ahuja, N.: Robust visual tracking via multi-task sparse learning. In CVPR (2012b).

[131] Zhang, T. Ghanem, B. Liu, S. and Ahuja, N.: Low-rank sparse learning for robust visual tracking. In ECCV (2012a).

[132] Hua, Y.: Towards robust visual object tracking : proposal selection and occlusion reasoning, PhD thesis, Université Grenoble Alpes (2016).

[133] Avidan, S.: Support vector tracking, IEEE Transactions On Pattern Analysis And Machine Intelligence, PAMI, 26(8):1064-1072 (2004).

[134] Avidan, S.: Ensemble tracking. IEEE Transactions On Pattern Analysis And Machine Intelligence, PAMI, 29(2):261–271 (2007).

[135] Penne, T., Barra, V., Tilmant, C. and Château, T. : Une version modifiée de l'ensemble tracking. ORASIS'09 - Congrès des jeunes chercheurs en vision par ordinateur (2009).

[136] Zhu, X.: Semi-Supervised Learning Literature Survey, Computer Sciences TR 1530 University of Wisconsin, Madison. Last modified on July 19 (2008).

[137] Grabner, H., Leistner, C., and Bischof, H.: Semi-supervised on-line boosting for robust tracking. In European Conference on Computer Vision. 234–247 (2008).

[138] Liu, S., Zhang, T., Cao, X., Xu, C.: Structural Correlation Filter for Robust Visual Tracking. CVPR: 4312-4320 (2016).

[139] Ma, C., Yang, X., Zhang, C. and Yang, M.-H.: Long-term correlation tracking. In CVPR, (2015b).

[140] Chen, Z., Hong, Z. and Tao, D. An experimental survey on correlation filter-based tracking. arXiv preprint arXiv:1509.05520 (2015).

[141] Bolme, D. S., Beveridge, J. R., Draper, B. and Lui, Y. M.: Visual object tracking using adaptive correlation filters. In CVPR (2010).

[142] Henriques, J. F., Caseiro, R., Martins, P. and Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In ECCV (2012).

[143] Henriques, J. F., Caseiro, R., Martins, P. and Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(3):583–596 (2015).

[144] Danelljan, M., Hager, G., Khan, F. and Felsberg, M.: Accurate scale estimation for robust visual tracking. In BMVC (2014).

[145] Li, Y. and Zhu, J.: A scale adaptive kernel correlation filter tracker with feature integration. In ECCV Workshop on Visual Object Tracking Challenge (2014).

[146] Huttenlocher, D. P., Noh, J.J., and Rucklidge, W.J., Tracking non-rigid objects in complex scenes, Proc. IEEE,. pp. 93-101 (1993).

[147] Ma, L., Liu, J., Wang, J., Cheng, J., Lu, H.: A improved silhouette tracking approach integrating particle filter with graph cuts. In: IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1142–1145. IEEE, New York (2010).

[148] Li, B., Chellappa, R., Zheng, Q., & Der, S.: Model-based temporal object verification using video. IEEE Transactions on Image Processing, 10(6), 897–908 (2001).

[149] Kang, J., Cohen, I. and Medioni, G.: Object reacquisition using invariant appearance model, Proceedings of the Int. Conf. on Pattern Recognition, pp.759-762 (2004).

[150] Sato, K. and Aggarwal, J.: Temporal spatio-velocity transform and its application to tracking and interaction, Computer Vision and Image Understanding, vol.96, issue.2, pp.100-128 (2004).

[151] Cai, Z., Wen, L., Lei, Z., Vasconcelos, N., Li, S.Z.: Robust deformable and occluded object tracking with dynamic graph. IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. 23, 5497–5509 (2014).

[152] Das, A. J., Saikia, N., & Sarma, K. K. : Object Classification and Tracking in Real Time: An Overview. Emerging Technologies in Intelligent Applications for Image and Video Processing, pp: 250-295 (2016).

[153] Terzopoulos, D. and Szeliski, R.: Tracking with Kalman snakes, in. Active Vision. Cambridge, MA: MIT Press, pp. 3–20 (1992).

[154] Maccormick, J. and Blake, A.: Probabilistic exclusion and partitioned sampling for multiple object tracking. Int. J. Comput. Vision 39, 1, 57–71 (2000).

[155] Chen, Y., Rui, Y., and Huang, T.: Jpdaf based hmm for real-time contour tracking. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 543–550 (2001).

[156] Bertalmio, M., Sapiro, G., and Randall, G.: Morphing active contours. IEEE Trans. Patt. Analy. Mach. Intell. 22, 7, 733–737 (2000).

[157] Mansouri, A.: Region tracking via level set pdes without motion computation. IEEE Trans. Patt. Analy. Mach. Intell. 24, 7, 947–961 (2002).

[158] Yilmaz, A. Li, X. and Shah, M.: Object contour tracking using level sets. In. Proceedings of the ACCV (2004).

[159] Yilmaz. A.: Object tracking and activity recognition in video acquired using mobile cameras. Ph. D. Dissertation, College of. Engineering and Computer Science, University of Central Florida, Orlando, Florida (2004).

[160] Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. In: IEEE International Conference on Computer Vision, Barcelona, vol. 117, no. 10, pp. 81–88 (2011).

[161] Duffner, S., Garcia, C.: PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 2480–2487. IEEE (2013).

[162] Comaniciu, D., Meer, P.: Robust Analysis of Feature Spaces : Color Image Segmentation, IEEE Conf. Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico, pp. 750-755 (1997).

[163] Collins, R.T.: Mean-shift blob tracking through scale space, in: Proc. IEEE Intl.Conf. on Comput. Vision Pattern Recognition, Madison, WI, USA, pp.234–240 (2003).

[164] Zivkovic, Z., Krose, B.: An EM-like algorithm for color histogram-based object tracking. Computer vision and pattern recognition, 2004. Proc. IEEE Comput. Soc. Conf. Vol. 1, I-798– I-803 (2004).

[165] Ning, J., Zhang, L., Zhang, D.,Wu, C.: Scale and orientation adaptive mean shift tracking. Comput. Vis. IET 6(1), 52–61 (2012).

[166] Vojir, T., Noskova, J., Matas, J.: Robust scale-adaptive mean-shift for tracking: in image analysis. Berlin Heidelberg: Springer, 652-663 (2013).

[167] Birchfield, S. T., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. Comput. Vis. Pattern. Recognit. CVPR 2005. IEEE computer society conference on. IEEE, Vol. 2, 1158– 1163 (2005).

[168] ZHAO, J., QIAO, W., G-ZUN. M,: An approach based on mean shift and kalman filter for target tracking under occlusion, article. University, Baoding 071002, China College of Economics, Hebei University, Baoding 071002, China, 12-15 (2009).

[169] Yang, Y., Jia, Y.X., Rong, C.Z., Zhu, Y., Wang, Y., Yue, Z.J., Gao, Z.X.: Object tracking based on corrected background-weighted histogram mean shift and kalman filter. Adv. Mate. Res. 765, 720–725 (2013).

[170] Khan, Z.H., Gu, I.Y.H., Backhouse, A.G.: Robust visual object tracking using multi-mode anisotropic mean shift and particle filters. Circuits. Syst. Video Technol. IEEE Trans. 21(1), 74–87 (2011).

[171] Yao, A., Lin,X.,Wang, G.,Yu, S.: A compact association of particle filtering and kernel based object tracking. Pattern. Recognit. 45(7), 2584–2597 (2012).

[172] Phadke, G., Velmurugan, R.: Improved mean shift for multi-target tracking. In Performance evaluation of tracking and surveillance (PETS), IEEE International Workshop on. IEEE, 37–44, (2013).

[173] Peng, N.S., Yang, J., Liu, Z., Mean shift blob tracking with kernel histogram filtering and hypothesis testing, Pattern Recognition Letters, 26(5), 605–614 (2005).

[174] Kailath, T. The divergence and bhattacharyya distance measures in signal selection. Communication Technology, IEEE Transactions on, 15(1):52-60 (1967).

[175] Chen, X., Zhang, M., Ruan, K., Xu, G., Sun, S., Gong, C., Min, J., Lei, B.: Improved mean shift target tracking based on selforganizing maps. SIViP 8, S103–S112 (2014)

[176] Emami, E., & Fathy, M. (2011, November). Object tracking using improved camshaft algorithm combined with motion segmentation. In Machine Vision and Image Processing (MVIP), 7th Iranian (pp. 1-4). IEEE (2011).

[177] Xiu, C., Wei, S., Wan, R., Cheng, Y., Luo, J., & Tian, H.: CamShift tracking method based on target decomposition. Mathematical Problems in Engineering (2015).

[178] Hidayatullah, P., Hubert, K.: CAMSHIFT improvement on multi-hue and multi-object tracking. Electrical Engineering and Informatics (ICEEI), International Conference on. IEEE (2011).

[179] Horn, B. K. P.: Robot vision. MIT Press (1986).

[180] Freeman, W. T., Tanaka, K., Ohta, J. and Kyuma, K.: Computer Vision for Computer Games. Int. Conf. On Automatic Face and Gesture Recognition, pp.100-105 (1996).

[181] Shehan, F., Tmja, C.: Mean Shift Kalman Object Tracking for Video Surveillance. Conference, 19TH Eru symposium (2013).

[182] Kalman, R. E.: A new approach to linear filtering and prediction problems. Journal of Fluids Engineering, 82(1) : 35–45 (1960).

[183] Almanza. L. D, Détection et suivi d'objets mobiles perçus depuis un capteur visuel embarqué. In PhD thesis, University of Toulouse (2011).

[184] Birgé, L. & Rozenholc, Y. How many bins should be put in a regular histogram. ESAIM: Probability and Statistics, 10: 24-25 (2006).

[185] Hunt, R.W.G.: Measuring color, 2nd edition, Ellis Horwood Series in Applied Science and Industrial Technology (1991).

[186] Ford, Adrian, and Alan Roberts. "Colour space conversions.: Westminster University, London (1998).

[187] Colantoni, P. et al. : Color space transformations. Technical report (2004), http://faculty.kfupm.edu.sa/ICS/lahouari/Teaching/colorspacetransform-1.0.pdf

[188] Fairchild, M.D.: Color Appearance Models, 2nd Ed, John. Wiley & Sons, Ltd (2005).

[189] Ibraheem, N.A. et al, “Understanding color models: A review”, ARPN Journal of Science and Technology, vol. 2, no. 3, pp. 265-275, April 2012.

[190] Gritzman, A.D., Rubin, D.M., Pantanowitz, A.: Comparison of colour transforms used in lip segmentation algorithms, Signal Image Video Processing., vol 9, pp: 947-957 (2015).

[191] Trémeau, A., Fernandez-Maloigne, C., Bonton, P. : Image Numérique couleur : de l’acquisition au traitement. Cours et applications. Dunod, Paris (2004).

[192] Ohta, Y., Kanade, T., Sakai, T.: Color information for region segmentation. Computer Graphics and Image Processing, 13(1) :222–241 (1980).

[193] Van, S. K. E., Gevers, T., Snoek, C. G.: Evaluating color descriptors for object and scene recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1582–1596 (2010).

[194] Blasch, P., Ling, H.: Encoding color information for visual tracking: algorithms and benchmark. IEEE Transactions on Image Processing, 24(12), 5630–5644 (2015).

[195] Danelljan, M., Shahbaz Khan, F., Felsberg, M. and Van de Weijer, J.: Adaptive color attributes for real-time visual tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition (2014).

[196] Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. Proc. European Conf. Computer Vision, pp. 661–675 (2002).

[197] Chen, H.-T., Liu, T.-L.: Trust-region methods for real-time tracking. Proc. IEEE Int’l Conf. Computer Vision, pp. 717–722 (2001).

[198] Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1940–1947 (2012).

[199] Zhang, J., Ma, S., Sclaroff, S. : MEEM: Robust tracking via multiple experts using entropy minimization. In ECCV (2014).

[200] Van, W. J., Schmid, C., Verbeek, J., and Larlus, D.: Learning color names for real-world applications,” IEEE Trans. Image Processing, vol. 18, no. 7, pp. 1512–1523 (2009).

[201] Medouakh. S, Boumehraz. M, Terki. N.: Improved object tracking via joint color-LPQ texture histogram based mean shift algorithm, Signal, Image and Video Processing (SIViP), vol. 12, pp:583-590 (2018).

[202] Bousetouane, F. : Lynda Dib et Hichem Snoussi. Improved mean shift integrating texture and color features for robust real time object tracking. The Visual Computer, vol. 29, pages 155_170 (2013).

[203] Phadke, G., Velmurugan, R.: Mean LBP and modified fuzzy Cmeans weighted hybrid feature for illumination invariant meanshift tracking. SIViP 11(4), 665–672 (2017).

[204] Rezazadegan, H., Shahram, M., Heikkila, J.: Local similarity number and its application to object tracking”, International Journal of Advanced Robotic Systems, 10(184), 1-7 (2013).

[205] Chen, L., Huang, Q., Pang, L., Su, F.: A Robust Tracking Combined with Texture Feature and Background-Weighted Color Histogram, Proceedings of the International Conference on Communications, Signal Processing and Systems, pp 751-759 (2015).

[206] Wouwer, G., Scheunders, P. and Dyck, D.: Statistical texture characterization from discrete wavelet representations, IEEE Trans. Imag. Process. 8(4), 592–598 (1999).

[207] M. Pietikäinen, T. Ojala and Z. Xu, Rotation-invariant texture classification using feature distributions, Patt. Recogn. 33(1), 43–52 (2000).

[208] Gotlieb, C. C. and Kreyszig, H. E.: Texture descriptors based on co-occurrence matrices, Comput. Vis. Graph. Imag. Process. 51(1), 70–86 (1990).

[209] Ojala, T., Valkealahti, K., Oja, E. and Pietikäinen, M.: Texture discrimination with multi-dimensional distributions of signed gray level differences, Patt. Recogn. 34(3), 727–739 (2001).

[210] Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE. Trans. Pattern. Anal. Mach. Intell. 24(7), 971–987(2002).

[211] Tan, X. and Triggs, B.: Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, IEEE Trans. on Image Processing, 19(6): pp. 1635-1650 (2010).

[212] Ojansivu, V., Heikkila, J.: Blur insensitive texture classification using local phase quantization. In: Proc. Image and Signal Processing ICISP, Cherbourg-Octeville, pp. 236-243 (2008).

[213] Heikkila, J., Ojansivu, V.: Methods for local phase quantization in blur-insensitive image analysis. In: Proc. International Workshop Local and Non-Local Approximation in Image Processing, pp. 104-111(2009)

[214] Ojala, T., Pietikäinen, M. and Harwood, D.: A comparative study of texture measures with classification based on feature distributions, Pattern Recognition, vol. 29, pp. 51-59 (1996).

[215] Hadid, A., Ylioinas, J. and Lopez, M. B.: Face and texture analysis using local descriptors: A comparative analysis. in Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on,, pp. 1-4 (2014).

[216] M. Bennamoun, Y. Guo, and F. Sohel, "Feature selection for 2D and 3D face recognition," Encyclopedia of electrical and electronics engineering. Book Chapter, pp. 1-54, 2015.

[217] Liu, L., Long, Y. P., Fieguth, W., Lao, S. and G. Zhao: BRINT: Binary rotation invariant and noise tolerant texture classification, Image Processing, IEEE Transactions on, vol. 23, pp. 3071-3084 (2014).

[218] Kannala, J. and Rahtu, E. Bsif: Binarized statistical image features, in Pattern Recognition (ICPR), 21st International Conference on, pp. 1363-1366 (2012).

[219] Pflug. A, Paul. P.N, Busch .C.: A comparative study on texture and surface descriptors for ear biometrics, Security , International Carnahan Conference, Technology (ICCST), Italy (2014).

[220] Hyvärinen, A., Hurri, J. and Hoyer, P. O.: Natural Image Statistics: A Probabilistic Approach to Early Computational Vision vol. 39: Springer Science & Business Media (2009).

[221] Ford, A., Roberts, A.: Colour Space Conversions. Westminster University, London (1998).

[222] Fei-Fei, L., Fergus, R. and Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4): 594–611 (2006).

[223] Everingham, M., Van Gool, L., Williams, C. K., Winn, J. and Zisserman, A.: The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision, 88(2):303–338 (2010).

[224] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252 (2015).

[225] Leang, I . Fusion en ligne d'algorithmes de suivi visuel d'objet. These de doctorat. Université Pierre et Marie Curie. France (2016).

[226] Collins, R., Zhou, X. et Teh, S. K. : An open source tracking testbed and evaluation web site. In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, volume 35 (2005).

[227] Ferryman, J. et Ellis, A. :Pets2010 : Dataset and challenge. In Advanced Video and Signal Based Surveillance (AVSS), Seventh IEEE International Conference on, pages 143–150. IEEE (2010).

[228] Smeulders, A. W., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A. and Shah, M.: Visual tracking: An experimental survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7):1442–1468 (2014).

[229] Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Porikli, F., Cehovin, L., Nebehay, G., Fernandez, G., Vojir, T. et al. : The visual object tracking VOT2013 challenge results. In ICCV Workshop on Visual Object Tracking Challenge (2013).

[230] Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., Nebehay, G., Vojir, T., Fernandez, G., Lukezic, A., Dimitriev, A. et al. : The visual object tracking VOT2014 challenge results. In ECCV Workshop on Visual Object Tracking Challenge (2014).

[231] Wu, Y., Lim, J., Yang, M. H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9): 1834-1848 (2015).

[232] Felsberg, M., Berg, A., Hager, G., Ahlberg, J., Kristan, M., Matas, J., Leonardis, A., Cehovin, L., Fernandez, G., Vojir, T., Nebehay, G. et Pflugfelder, R.: The thermal infrared visual object tracking vot-tir2015 challenge results. In The IEEE ICCV Workshops (2015).

[233] Cehovin, L., Kristan, M. et Leonardis, A. : Is my new tracker really better than yours ?. In IEEE Winter Conference on Applications of Computer Vision, pages 540–547. IEEE (2014).

[234] Kristan, M., Leonardis, A., Matas, J. et al. : The visual object tracking vot2016 challenge results. Proceedings of the European Conference on Computer Vision Workshops, 1-45 (2016).