



Suivi Adaptatif d'Objets Non Rigides Basé sur des Histogrammes de Couleur et une Sélection Automatique de Paramètres

Aude Jacquot, Peter Sturm, Olivier Ruch

► To cite this version:

Aude Jacquot, Peter Sturm, Olivier Ruch. Suivi Adaptatif d'Objets Non Rigides Basé sur des Histogrammes de Couleur et une Sélection Automatique de Paramètres. Journées ORASIS, May 2005, Fournol, France. 2005. <inria-00524405>

HAL Id: inria-00524405

<https://hal.inria.fr/inria-00524405>

Submitted on 25 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Suivi Adaptatif d'Objets Non Rigides Basé sur des Histogrammes de Couleurs et une Sélection Automatique de Paramètres

Adaptive Tracking of Non-Rigid Objects Based on Color Histograms and Automatic Parameter Selection

A. Jacquot^{1,2}

P. Sturm¹

O. Ruch²

¹ INRIA Rhône-Alpes
655, avenue de l'Europe
38334 Saint Ismier
prenom.nom@inrialpes.fr

² THALES Optronique
rue Guynemer
78283 Guyancourt Cedex
prenom.nom@fr.thalesgroup.com

Résumé

Une des principales difficultés du suivi d'objets dans une séquence vidéo réside dans la prise en compte des changements d'apparence (non seulement ceux de la cible, mais aussi ceux qui proviennent de la scène directement, les changements d'illumination par exemple). Le cadre Bayésien est très flexible et ses performances dans le domaine du suivi d'un ou plusieurs objets ont été prouvées. D'autre part, les histogrammes de niveaux de gris ou de couleurs permettent de suivre un objet avec une complexité calculatoire faible. Les algorithmes de suivi d'objets reposant sur des histogrammes de couleurs intégrés dans un cadre bayésien [5, 7] se sont montrés performants pour une application donnée (le suivi de visage par exemple), mais ne peuvent pas être généralisés facilement, les paramètres et l'initialisation de l'algorithme étant spécifiques à la séquence d'entrée.

Cet article présente une méthode basée sur des histogrammes de niveaux de gris combinés à un filtrage particulière qui permet de résoudre un certain nombre des problèmes traditionnels du suivi d'objets (occultations, changements d'apparence de l'objet, changements d'échelle ou d'illumination de la scène), et de s'adapter facilement à la séquence d'entrée (suivi de joueur de football, de structures dans des images aériennes, de visages). La nouveauté de l'algorithme réside dans sa capacité à fixer automatiquement tous les paramètres nécessaires au suivi.

Mots Clef

Filtrage particulière, histogrammes de couleurs, modèles d'apparence.

Abstract

One of the main difficulties in visual tracking is to take into account appearance changes (not only of the target but also of or due to the scene, illumination for example). The use of a Bayesian framework is very flexible and has

proven to be very efficient in visual tracking. Moreover, color or greylevel histograms allow to track an objet with a low computational cost. The recently proposed color-based trackers integrated in a probabilistic framework [7, 5] are efficient for a given application (face tracking for example) but can not be generalized easily, due to the initialization and the adjustment of the different tracker parameters that are dependent on the input sequence. This paper presents a method based on color integrated in a particle filter that allows to cope with some of the usual problems of visual tracking (occlusions, target appearance changes, changes in resolution or in illumination) and to adapt easily to different applications (tracking of structures in aerial imagery as well as football players). The novelty of the tracker is its ability to automatically regulate all the parameters needed for tracking, which makes it flexible and easily usable for different applications.

Keywords

Partile filtering, color histograms, appearance models.

1 Introduction

Quelque soit l'objet que l'on veut suivre, on a besoin d'un modèle pour le décrire : ce modèle peut contenir aussi bien de l'information a priori sur l'objet que de l'information extraite des images précédentes. Il peut être constitué de contours géométriques, de régions, de descripteurs globaux ou de tout autre type de primitives. Un des facteurs limitant les performances des algorithmes de suivi est le modèle d'apparence sélectionné pour modéliser l'objet à suivre (tout au long de l'article nous parlerons de modèle d'apparence pour désigner le modèle pris pour modéliser notre objet à suivre). Les méthodes basées sur la correspondance de régions ("template matching") ne permettent pas de prendre en compte les changements d'apparence de l'objets, et les méthodes basées sur l'estimation du mouvement permettent au modèle d'apparence de l'objet de changer rapidement, mais sont susceptibles de dévier de leur

cible.

Cet article présente un algorithme de suivi utilisant des distributions de couleurs intégrées dans un cadre probabiliste. On choisit les histogrammes comme modèle d'apparence, car ils présentent de nombreux avantages : ils sont robustes aux occultations partielles, invariants aux rotations et sont peu coûteux en calculs. D'autre part, les filtres particuliers se sont montrés performants dans le cadre d'occultations. Quelques algorithmes de suivi basés sur des histogrammes combinés à des approches probabilistes ont été présentés récemment ; les résultats obtenus sont bons pour l'application pour laquelle ils ont été présentés, mais ne peuvent pas être adaptés facilement à un autre type d'application. D'autres techniques proposées par Bradski (le "Camshift" [2]), Comaniciu (le "Mean Shift" [3]) ou plus récemment Zivkovic (qui propose une extension de l'algorithme du Mean Shift dans [10]) n'utilisent pas de probabilités mais font une recherche déterministe de la région pour laquelle le contenu colle au mieux au modèle de référence. Ces méthodes ont les mêmes limitations que les précédentes. La nouveauté de notre algorithme réside dans l'intégration de certains critères qui permettent de sélectionner automatiquement le nombre adapté de classes de l'histogramme pour le suivi de l'objet, et une nouvelle manière de mettre à jour le modèle. Notre algorithme est robuste aux occultations, aux changements d'illumination ainsi qu'aux changements d'apparence de l'objet, et s'est montré performant aussi bien dans le cadre du suivi d'objets pris à partir d'une caméra embarquée que dans celui du suivi de structures statiques à partir de séquences aériennes.

Le plan de l'article est le suivant : dans la partie 2, nous décrivons brièvement le principe du filtrage particulière, l'utilisation des histogrammes de couleurs et la façon dont les deux sont combinés dans notre algorithme. La partie 3 décrit nos contributions : le gain de l'information spatiale apporté par la division de la région d'intérêt en quartiers, la mise à jour du modèle, et la sélection automatique du nombre de classes du modèle et des paramètres nécessaires au suivi. Enfin, l'évaluation de nos contributions et les résultats du suivi effectués sur différents types de séquences sont présentés dans la partie 4.

2 Intégrer couleurs et probabilités pour faire du suivi

L'objectif de cette partie est de présenter les bases du suivi d'objets intégrant des histogrammes de couleurs dans un cadre probabiliste [7, 5].

2.1 Rappels sur le filtrage particulière

On se place dans le cadre Bayésien pour suivre des objets lorsque les densités de probabilité a posteriori $P(X_t | Z_t)$ et le modèle d'observation $P(Z_t | X_t)$ ne sont pas nécessairement gaussiennes. L'objet suivi est caractérisé par son vecteur d'état X_t , et les observations du temps $t = 0$ jusqu'au temps t sont définies par le vecteur

Z_t .

L'idée du filtrage particulière est d'approcher la distribution de probabilité de l'état de l'objet par un ensemble d'échantillons associés à des poids. Chaque échantillon, aussi appelé particule, est un élément qui représente un état hypothétique de l'objet s , auquel on associe un poids π , représentant sa vraisemblance par rapport au modèle. On peut écrire l'ensemble de la façon suivante :

$$S = \left\{ \left(s^{(i)}, \pi^{(i)} \right), i = 1, \dots, n \right\} \text{ où } \sum_{i=1}^n \pi^{(i)} = 1$$

Par la suite on utilisera indifféremment les termes *échantillon* et *particule*.

L'évolution de l'ensemble est obtenu en propageant chacun des échantillons selon un modèle de mouvement. On attribue ensuite un poids à chaque particule en fonction des observations, et on détermine l'état moyen du système à chaque étape par :

$$E[S] = \sum_{i=1}^n \pi^{(i)} s^{(i)} \quad (1)$$

Un des avantages du filtrage particulière est que l'on modélise l'incertitude : en sortie de l'algorithme, il n'y a pas une solution unique, mais un ensemble d'échantillons représentant la densité de probabilité du vecteur d'état dans l'image ; cela en fait une approche robuste en cas de bruit ou d'occultation.

2.2 Les histogrammes de couleurs comme choix du modèle

Comme il a été dit précédemment, nous avons choisi d'utiliser les distributions de couleurs pour modéliser l'objet pour de nombreuses raisons : invariance aux rotations, robustesse aux occultations partielles, et à l'aspect rigide/non rigide de l'objet à suivre. On suppose que les distributions sont discrétisées en K classes (voir la partie 3 pour la sélection automatique de K). Dans notre approche, l'objet est modélisé par une ellipse (de nombreuses approches utilisent des rectangles mais il nous a paru plus intéressant d'utiliser une ellipse pouvant tourner) ; on peut calculer les histogrammes dans l'espace RGB (ou dans n'importe quel autre espace) des couleurs, ou plus simplement en niveaux de gris, en fonction de la séquence d'entrée.

Pour limiter la perte d'information spatiale due aux histogrammes, Nummiaro [7] et Pérez [9] ont attribué des poids différents aux pixels de l'ellipse. Leur raisonnement est le suivant : plus les pixels sont loin du centre de l'ellipse, plus on leur associe un poids faible. La fonction de poids correspondante est donnée par l'expression suivante :

$$k(x) = \begin{cases} 1 - x^2 & \text{si } x \leq 1 \\ 0 & \text{sinon} \end{cases}$$

où x est la distance du pixel au centre de l'ellipse.

D'autres fonctions de poids peuvent être utilisées : Comaniciu, par exemple dans [3], utilise le noyau d'Epanechnikov.

On calcule une distribution de couleur ou de niveau de gris $p_x = \{p_x^{(j)}\}_{j=1,\dots,K}$ à un pixel donné x par :

$$p_x^{(j)} = C \sum_{x_i \in E} k \left(\frac{\|x - x_i\|}{\sqrt{l_x^2 + l_y^2}} \right) \delta(h(x_i) - j) \quad (2)$$

où δ est la fonction delta de Kronecker, E est l'ensemble des pixels contenus dans l'ellipse, l_x et l_y les demi axes de l'ellipse, $h(x_i)$ attribue une des K classes de l'histogramme d'une couleur donnée au pixel x_i et C est le facteur de normalisation, qui assure que $\sum_{j=1}^K p_x^{(j)} = 1$. L'expression de C est donnée par :

$$C = \left[\sum_{x_i \in E} k \left(\frac{\|x - x_i\|}{\sqrt{l_x^2 + l_y^2}} \right) \right]^{-1}$$

La ressemblance entre deux distributions p et q est donnée par le coefficient de Bhattacharyya :

$$\rho[p, q] = \sum_{j=1}^K \sqrt{p^{(j)} q^{(j)}} \quad (3)$$

Pour deux distributions identiques, on a $\rho = 1$, ce qui correspond à une correspondance parfaite. On utilise la distance de Bhattacharyya $d = \sqrt{1 - \rho[p, q]}$ dans notre algorithme pour évaluer la distance entre deux histogrammes. Notons que d'autres mesures de distances entre histogrammes existent [11], et que la distance de Bhattacharyya présente le meilleur compromis entre complexité calculatoire et performances.

2.3 Comment combiner les histogrammes de couleurs et le filtrage particulaire ?

On veut suivre une région d'intérêt dans le plan image. Nous avons choisi de paramétrer cette région par une ellipse, définie par :

$$s = \{x, y, \dot{x}, \dot{y}, \theta, l_x, l_y, \dot{l}_x, \dot{l}_y\}$$

où x et y représentent les coordonnées du centre de l'ellipse, \dot{x} et \dot{y} les vitesses en x et y du centre de l'ellipse, θ l'orientation de l'ellipse, l_x et l_y les demi axes de l'ellipse, et \dot{l}_x et \dot{l}_y les vitesses de l_x et l_y . Notons que les paramètres de l'ellipse peuvent varier indépendamment les uns des autres ce qui rend ce modèle flexible.

Pour propager une particule, nous utilisons un modèle de mouvement au premier ordre :

$$s_t = A s_{t-1} + b_{t-1} \quad (4)$$

où b_t est une variable aléatoire Gaussienne et A une matrice dont le but est de décrire un objet se déplaçant à vitesse

constante en x, y, l_x et l_y . A s'exprime donc de la manière suivante :

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

L'algorithme fonctionne de la manière suivante : nous calculons la distribution du modèle dans la première image, et nous initialisons l'ensemble des particules (l'ellipse est initialisée à la main sur la première image). Puis, pour chaque image de la séquence, les particules sont propagées en utilisant le modèle de mouvement défini précédemment. Pour chaque particule, nous calculons la distance de Bhattacharyya entre la distribution du modèle et celle de l'état hypothétique correspondant à cette particule. Cette distance est utilisée pour calculer le poids π associé à la particule. Nous pouvons remarquer que les poids associés à chacune des particules favorisent les particules dont les distributions sont ressemblantes à celles du modèle. Les poids sont calculés en utilisant l'équation :

$$\pi^{(i)} = \exp(-\beta d^{(i)}) \quad (5)$$

pour chaque particule i de l'ensemble, où β est une constante fixée, et $d^{(i)}$ représente la distance de Bhattacharyya entre la $i^{ème}$ particule et le modèle de l'objet.

La dernière étape consiste à rééchantillonner les particules pour assurer la bonne évolution de l'ensemble des particules, et de déterminer l'état moyen de l'objet. Lors de cette étape de rééchantillonnage, les particules sont éliminées ou dupliquées selon leur poids : plus leur poids est élevé, plus on duplique la particule. Dans la littérature, différentes méthodes existent pour rééchantillonner un ensemble de particules, et on peut se référer à [6] pour plus de détails ; nous avons choisi d'utiliser un rééchantillonnage systématique pour sa simplicité. La procédure est la suivante : on divise l'intervalle $[0, 1]$ en n segments. Puis on génère une variable aléatoire uniforme U sur $[0, \frac{1}{n}]$; on définit $U_1 = U$ et $U_i = U_{i-1} + \frac{1}{n}$ pour $i = 2, \dots, n$. Si U_i appartient au $j^{ème}$ segment, alors on pose $\Xi^i = X^j$, où Ξ^i est un élément du nouvel échantillon. L'avantage de cette méthode est que seulement $O(n)$ tests de comparaison sont nécessaires pour constituer le nouvel échantillon (Ξ^1, \dots, Ξ^n).

3 Nos contributions

De nombreuses approches reposant sur des histogrammes de couleurs intégrés à un cadre probabiliste ont été proposées pour le suivi d'objets (soumis ou non à des déformations). Mais aucune de ces approches n'est assez

flexible pour régler les paramètres du suivi automatique de manière à le rendre utilisable facilement pour diverses applications. Nous proposons dans cette partie des critères permettant de régler les paramètres du suivi automatiquement.

3.1 Le choix du nombre adapté de classes d'un histogramme

Le nombre de classes nécessaires pour calculer les histogrammes est un paramètre important et doit être déterminé automatiquement. La discrétisation en un nombre de classes trop important ne permet pas d'être robuste aux changements d'illumination de la scène ou aux changements d'apparence de l'objet, et la plupart du temps l'algorithme dérive au fur et à mesure de la séquence. À l'inverse, un nombre de classes trop faible ne permet pas de discriminer efficacement l'objet du reste de la scène, et le suivi échoue. Cette intuition/remarque a été confirmée par nos expériences, comme le montrent les résultats de notre évaluation présentés dans la partie 4.

Dans la plupart des approches existantes, le nombre de classes paraît être choisi arbitrairement au début de la séquence et reste fixé tout au long du suivi. Rien n'indique qu'une telle partition est optimale étant donné la densité que l'on veut estimer. S'il était possible de trouver la partition optimale, l'algorithme serait plus robuste.

De nombreuses recherches dans le domaine de l'estimation et de la sélection de modèles ont été réalisées dans le but de résoudre le problème suivant : comment déterminer le modèle optimal à partir des données disponibles ? On considère qu'un modèle est d'autant meilleur qu'il est peu complexe et fidèle aux données. On parle donc de modèle optimal au sens de ce compromis à faire (entre complexité et fidélité aux données). Généralement les méthodes obtenues sont basées sur des considérations asymptotiques, et ne marchent pas bien dans le cas où les données sont de petites tailles. De plus, nombreuses d'entre elles font l'hypothèse d'information a priori sur la densité à estimer. Récemment, Birgé et Rozenholc [1] ont généralisé l'estimateur d'Akaike. Le théorème d'Akaike est une mesure statistique pour la sélection de modèle qui stipule que si deux modèles représentent aussi bien les données, le modèle le plus simple sera le meilleur prédicteur. Dans ce qui suit, nous résumons brièvement leur méthode pour déterminer le nombre optimal de classes pour nos histogrammes. Pour les détails théoriques sous-jacents, on peut se référer à [1].

L'objectif est de trouver un estimateur pour notre histogramme \hat{f} basé sur une partition $\{I_1, \dots, I_K\}$ de $[0, 1]$ en K intervalles de même longueur. X_1, X_2, \dots, X_n sont n échantillons de la densité inconnue f que l'on cherche à estimer. K est donné par :

$$K = \arg \max_K (L_n(K) - \text{pénalité}(K)) \quad (6)$$

où $L_n(K)$ est le log de la vraisemblance de l'histogramme avec K classes, donné par :

$$L_n(K) = \sum_{j=1}^K M_j \log\left(\frac{KM_j}{n}\right) \quad \text{avec} \quad M_j = \sum_{i=1}^n \mathbf{1}_{I_j}(X_i)$$

où $\mathbf{1}_{I_j}$ est la fonction indicatrice définie par :

$$\mathbf{1}_{I_j}(x) = \begin{cases} 1 & \text{si } x \in I_j \\ 0 & \text{sinon} \end{cases}$$

La fonction de pénalité est donnée par :

$$\text{pénalité}(K) = K - 1 + (\log(K))^{2.5} \quad \text{pour } K \geq 1$$

Cette approche est donc un exemple typique de méthode de sélection de modèle, mettant en évidence le compromis à faire entre la complexité du modèle et sa fidélité aux données.

3.2 Incorporer de l'information spatiale

Le problème avec l'utilisation d'histogrammes est que l'information spatiale est perdue, à l'inverse des approches de type "template matching" qui utilisent toute l'information spatiale. Comme il a été dit précédemment, attribuer des poids différents aux pixels selon la distance qui les sépare du centre de l'ellipse permet de compenser cette perte d'information. Pour conserver davantage la relation spatiale existante entre les pixels voisins de la région d'intérêt, nous avons décidé de diviser notre ellipse en 4 quartiers, et de procéder pour chaque quartier de la même façon que pour l'ellipse entière décrite précédemment. La division de l'ellipse accroît la robustesse du suivi puisque l'on dispose de 4 mesures de similarité entre une hypothèse et le modèle, qui sont facilement combinables ; cela permet une meilleure discrimination de l'objet par rapport au reste de la scène.

Un autre avantage de la division de l'ellipse est de pouvoir utiliser le critère pour la sélection automatique du nombre de classes à chacun des quartiers. Le nombre de classes peut être différent dans chacun des quartiers de l'ellipse selon la quantité de données disponible (un quartier de l'ellipse contenant une région homogène n'a pas besoin d'autant de classes qu'un quartier très texturé).

Enfin, la division de l'ellipse en quartiers permet de détecter et/ou gérer les occultations partielles plus facilement : si le coefficient de Bhattacharyya est mauvais pour un des quartiers mais très bon pour les autres, alors on détecte une occultation partielle. Pour évaluer l'état moyen de l'objet, on combine les 4 mesures en calculant la valeur médiane.

3.3 La mise à jour du modèle

La couleur apparente d'un objet peut varier au cours du temps ; cela peut être dû à des changements d'illumination de la scène, à des modifications des paramètres de

la caméra ou au mouvement de l'objet. Pour faire face à ces changements d'apparence, on a besoin de mettre à jour le modèle. Le filtrage particulière a déjà été utilisé avec des modèles statiques [4] ou adaptatifs [8]. La plupart du temps, le modèle est mis à jour à chaque fois que la probabilité de l'état moyen de l'objet calculé est au-dessus d'un seuil fixé arbitrairement à l'initialisation. Le risque de cette méthode est qu'elle peut dériver progressivement de l'objet initial.

L'idée que nous proposons est la suivante : pourquoi ne pourrait-on pas mettre à jour le modèle lorsqu'il n'est plus bon ? Le modèle devrait être mis à jour lorsque l'apparence du modèle a suffisamment changé. Ainsi, nous utilisons le critère suivant : si l'état moyen est en dessous d'un seuil π_T (voir le paragraphe suivant pour la détermination de ce seuil), alors le modèle est mis à jour en utilisant l'équation suivante :

$$p_{E[S_t]} \leq \pi_T \Rightarrow q_t^j = (1 - \alpha)q_{t-1}^j + \alpha p_{E[S_t]}^j \quad (7)$$

Fixer le seuil π_T Ce seuil dépend de plusieurs paramètres plus ou moins connectés : la taille de l'ellipse et le nombre de classes de l'histogramme. En général, plus l'ellipse est grande, plus le nombre de classes est grand. Le coefficient de Bhattacharyya est alors plus faible, et le seuil pour la mise à jour du modèle doit être plus bas lui aussi. Pour régler ce seuil automatiquement, on fait l'hypothèse qu'au début de la séquence (les premières images), l'état moyen de l'objet est bien estimé, et on fixe le seuil de manière empirique : $\pi_T = \rho - c$ où c est une constante.

Le schéma global de l'algorithme est donné Figure 1.

-
1. Initialisation :
 - On sélectionne l'ellipse dans la première image
 - On détermine automatiquement le nombre de classes de chaque histogramme selon l'équation (6)
 - On calcule la distribution du modèle dans chaque quartier de l'ellipse avec l'équation (2)
 - On initialise l'ensemble des particules (n copies identiques de l'ellipse)
 2. Pour chaque nouvelle image :
 - On propage l'ensemble des particules en utilisant le modèle dynamique selon l'équation (4)
 - Pour chaque particule S_t , on calcule
 - La distribution de couleur avec l'équation (2)
 - Le coefficient de Bhattacharyya avec (3)
 - Le poids avec l'équation (5)
 - On estime l'état moyen selon l'équation (1)
 - On rééchantillonne (voir Paragraphe 2.3)
 - On met à jour le modèle si besoin avec (7)
-

FIG. 1 – L'algorithme.

4 Expérimentations

L'objectif de cette partie est d'évaluer les performances de notre algorithme. Tout d'abord, nous mettons en évidence l'apport de la sélection automatique du nombre de classes et de la division de l'ellipse en quartiers, puis nous présentons les résultats expérimentaux de notre algorithme testé sur trois séquences très différentes par le type d'objet à suivre et par le type de difficultés rencontrées dans le suivi (en termes de changement d'échelle, vitesses des objets dans la scène, caméra statique ou en mouvement...).

4.1 Evaluation des contributions

L'objectif de cette partie est de prouver que les critères mis en place pour la sélection automatique du nombre de classes des histogrammes et pour prendre en compte l'information spatiale (perdue par l'utilisation des histogrammes) améliorent les performances du suivi. Nous avons mis en place des expérimentations pour valider nos contributions.

La description des séquences Nous avons utilisé trois séquences pour valider notre algorithme.

- *Le suivi de visage* : La séquence présente une personne entrant et bougeant dans une pièce (<http://www.ee.oulu.fi/~mikak/tracking/FaceColor.html>). C'est une séquence de 500 images de 384x288 pixels. La taille des demi axes de l'ellipse dans la première image est de 12 et 18 pixels.
- *Le suivi de voiture* : La séquence est une des séquences de tests de la base de PETS 2001 (<ftp://pets2001.cs.rdg.ac.uk/>) dans le contexte d'une application d'assistance à la conduite. La séquence contient 500 images.
- *Le suivi de joueur de football* : La séquence est tirée d'un match de football, et contient 200 images de 889x676 pixels. La taille des demi axes de l'ellipse dans la première image est de 26 et 15 pixels.

Pour le critère de sélection automatique du nombre de classes des histogrammes nous avons exécuté notre algorithme plusieurs fois avec la sélection automatique du nombre de classes et pour un nombre de classes fixé, dans le cadre du tracking effectué avec l'ellipse entière. Nous utilisons la valeur moyenne pour la présentation des résultats.

Le Tableau 1 fait la synthèse des résultats obtenus : le nombre indique le numéro de l'image pour laquelle le suivi échoue, et les nombres en gras correspondent aux résultats obtenus pour la sélection automatique du nombre de classes. Pour juger de la réussite ou de l'échec du suivi, nous avons établi auparavant une vérité terrain pour chacune des séquences en conservant dans un fichier les coordonnées d'un suivi réussi. La détection d'un échec dans le suivi de l'objet se fait lorsque, dans une image, la taille de l'ellipse est trop grande ou le centre de l'ellipse trop éloigné de la vérité terrain. Les résultats montrent que fixer

# de classes	Football	Indoor	Plaque
2	16	19	31
4	87	23	46
6	107	111	55
8	109	125	41
10	95	172	60
12	118	169	94
14	90	168	92
16	121	164	84
18	117	132	93
20	111	217	61
22	75	108	78
24	88	135	62
26	72	90	63
28	86	107	73
30	76	78	74
32	69	85	82
34	102	70	61
36	92	92	86
38	102	76	69
40	87	59	91

TAB. 1 – Performances de l’algorithme pour le suivi de l’ellipse entière avec différent nombres de classe.

arbitrairement un nombre de classes n’est pas une bonne solution pour suivre des objets divers. Il est donc préférable d’utiliser le critère de sélection automatique du nombre de classes et avoir un résultat proche du meilleur que de fixer un nombre de classes de manière aléatoire et risquer un mauvais suivi.

Pour la division de l’ellipse en quartiers nous avons effectué deux suivis pour chacune des séquences avec le critère de sélection automatique du nombre de classes : le premier utilisant l’ellipse entière et le second utilisant la division de l’ellipse en quartiers.

Les résultats sont présentés sous forme de courbes dans la Figure 2. Ces courbes montrent que l’utilisation des 4 quartiers améliore beaucoup les performances du suivi.

4.2 Résultats obtenus sur les séquences

Nous avons testé notre algorithme de suivi sur trois séquences : les résultats montrent que notre approche permet de l’utiliser pour des applications diverses, sans avoir besoin d’initialiser les paramètres du suivi à chaque fois. 3 ensembles de résultats sont fournis ci dessous ; on utilise des séquences en niveaux de gris pour nos expérimentations.

1. *Le suivi de visage* : La séquence présente une personne entrant et bougeant dans une pièce. Il y a de nombreux changements d’apparence à prendre en compte, la personne se retournant complètement au cours de la séquence. Les résultats montrent que la mise à jour du modèle est performante et permet

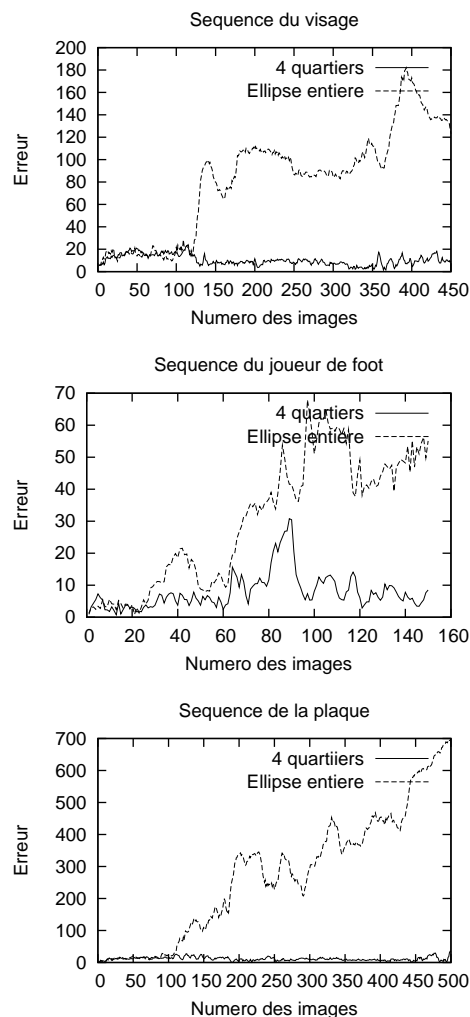


FIG. 2 – Influence de la division de l’ellipse en quartiers pour chacune des 3 séquences.

de prendre en compte les changements d’apparence de l’objet. D’autre part, l’algorithme est capable de suivre la personne, même si celle-ci se déplace à des allures différentes au cours de la séquence.

2. *Le suivi de voiture* : La séquence se place dans le contexte d’une application d’assistance à la conduite. Nous avons fait deux types de tests, les résultats sont présentés sur les Figures 4 and 5. Les difficultés de la séquence sont : la capacité à prendre en compte les mouvements rapides de l’objet tout autant que ceux de la caméra, les différents points de vue que l’on a de l’objet à suivre, des changements d’échelle et les rotations de l’objet d’intérêt dans le plan de l’image. Les résultats montrent que l’algorithme est performant tout au long de la séquence.

Pour la première expérimentation, nous avons suivi la voiture entière, et nous avons obtenu des résultats équivalents à ceux obtenus par [7]. Les demi axes de l’ellipse dans la première image sont de 40 et 50



FIG. 3 – Résultats du suivi de visage : images 1, 140, 395, 412 et 449 de la séquence.

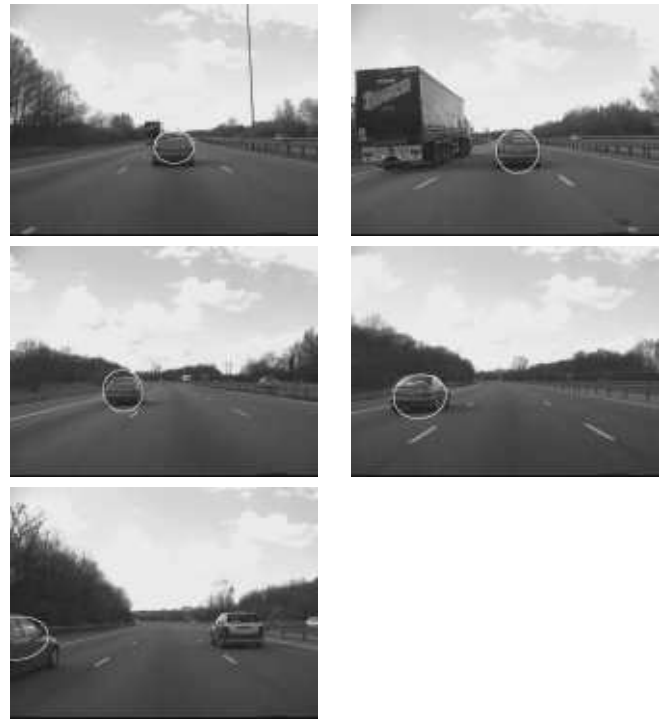


FIG. 4 – Résultats du suivi de voiture : images 1, 187, 347, 462 et 563 de la séquence.

pixels.

Pour la seconde expérimentation, nous avons suivi la plaque d'immatriculation de la voiture uniquement. Les demi axes de l'ellipse dans la première image sont de 12 et 18 pixels. La difficulté supplémentaire réside donc dans la taille de l'objet suivi. Les résultats présentés en Figure 5 montrent que notre algorithme est capable de suivre efficacement des objets même de petite taille.

3. *Le suivi de joueur de football* : La séquence est tirée d'un match de football ; les difficultés de cette séquence sont les mouvements rapides des joueurs et les occultations de certains joueurs par d'autres joueurs sur le terrain. Dans l'image 78, un joueur est en train de tomber et un autre tente d'attrapper le ballon, ce qui provoque une occultation importante du joueur. Notre algorithme reste performant dans ce cas.

5 Conclusions et discussion

Ces résultats mettent en évidence que notre système est capable de suivre :

- un objet soumis à d'importants changements d'apparence (des changements d'aspect ou d'orientation) ;
- un objet dans une scène soumise à des changements d'échelle et d'illumination ;
- un objet se déplaçant à des vitesses variées comme illustré par la séquence avec le joueur de football.
- un objet déformable (le joueur de football par exemple) ;



FIG. 5 – Résultat du suivi de la plaque : images 1, 187, 347, 417, 520 de la séquence, et un zoom fait sur l’image 417.

Nous avons également testé notre algorithme sur des séquences aériennes. Les résultats montrent que notre algorithme est robuste pour des applications variées.

L’algorithme proposé utilise un critère qui permet de détecter le nombre optimal de classes pour les histogrammes dans le but d’effectuer un suivi robuste de divers objets. D’autre part, nous avons mis en place des critères permettant à l’algorithme de fonctionner automatiquement quelque soit l’application. Notre approche est une étape vers un suivi entièrement automatique et adaptatif. L’algorithme est basé sur des distributions de couleurs intégrées à un cadre probabiliste.

En ce qui concerne la complexité de notre algorithme, elle est linéaire par rapport au nombre de particules utilisées, et linéaire aussi par rapport à la taille de l’ellipse. Les opérations sont simples et ne nécessitent pas de gros calculs, ce qui permet de rendre l’algorithme implémentable pour faire du temps réel : la propagation des particules est une multiplication de matrice, le remplissage des histogrammes correspond au nombre de pixels de l’ellipse, la comparaison d’histogrammes est linéaire par rapport au nombre de classes des histogrammes, et l’opération de rééchantillonnage est en $O(nbparticules)$.

Remarquons que nous avons divisé l’ellipse en quartiers ; il serait intéressant de mettre en place un critère similaire à celui utilisé pour sélectionner le nombre de classes, dans le but de trouver le meilleur compromis entre la quantité d’information spatiale dans le modèle et la flexibilité

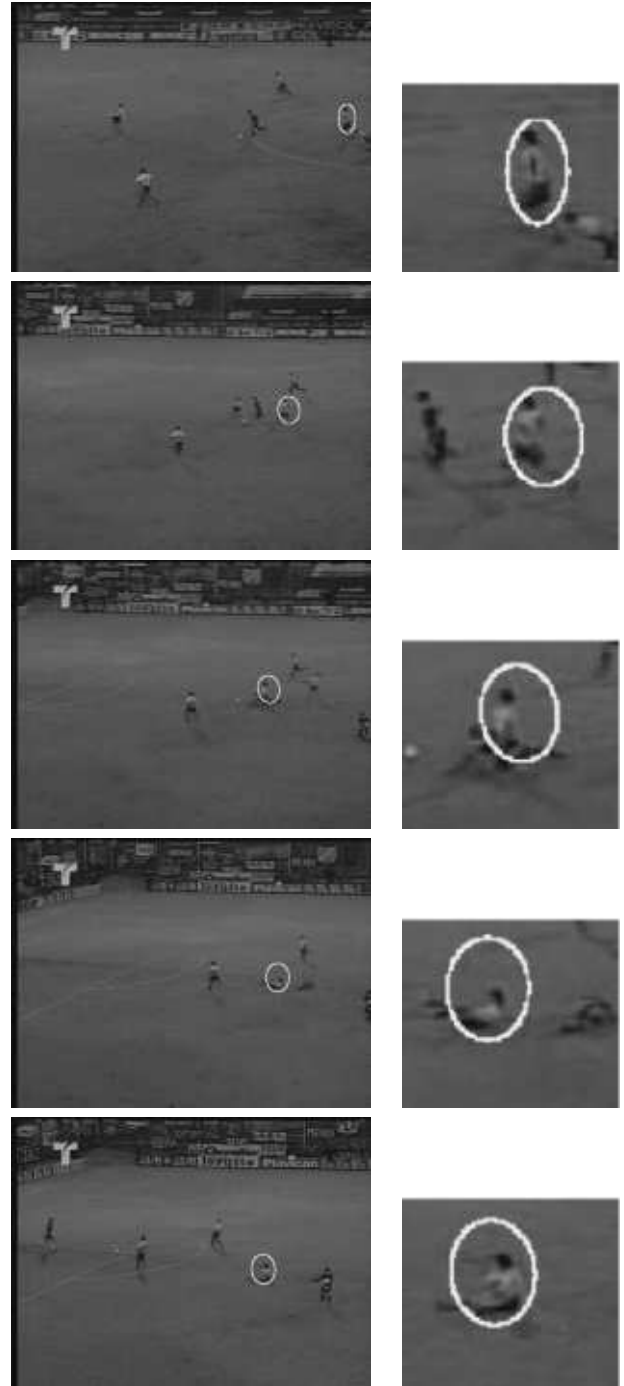


FIG. 6 – Résultats obtenus pour le suivi de joueur de football : images 1, 55, 76, 97 et 141 de la séquence. Dans l’image 76, le joueur blanc occulte le noir.

des histogrammes. D'autre part, il serait utile de pouvoir mettre à jour le nombre de classes au cours de la séquence si nécessaire (par exemple, si l'objet grossit dans l'image, on pourrait mettre à jour le modèle avec plus de classes pour prendre en compte l'information supplémentaire). Reste à signaler que la mise à jour du modèle en utilisant les distributions de couleurs lors de l'état courant peut poser problème, notamment dans le cas où un objet d'apparence similaire se trouve dans le voisinage.

Enfin, on pourrait facilement étendre l'approche au suivi de plusieurs objets. Si les objets sont de même types (et présentent les mêmes histogrammes), notre approche convient parfaitement : les particules se répartiront naturellement sur les différents objets ; dans le cas où les histogrammes des objets sont différents, il suffit d'initialiser différents objets et de lancer l'algorithme pour chacun des objets.

Ce travail a fait l'objet d'une présentation d'un poster au workshop IEEE Motion 2005 à Breckenridge, Colorado en Janvier 2005 et a été publié [12].

Références

- [1] L. Birgé, Y. Rozenholc, "How many bins should be put in a regular histogram", *Technological Report, Laboratoire Probabilités et Modèles Aléatoires, Université Pierre et Marie Curie*, Paris, France, PMA-721, 2002.
- [2] G.R. Bradski, "Computer vision face tracking as a component of a perceptual user interface", *In Workshop on Applications of Computer Vision*, pp 214-219, Princeton, NJ, 1998.
- [3] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 25, no 5, 2003.
- [4] T. Heap, D. Hogg, "Wormholes in Shape Space : Tracking through Discontinuous Changes in Shape", *Sixth International Conference on Computer Vision*, Bombay, India, 1998.
- [5] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Vol I, pp 415-422, 2001.
- [6] F. Legland, "Filtrage particulière", *Proceedings 19ème Colloque GRETSI sur le Traitement du Signal et des Images*, Vol I, pp 1-8, Paris, 2003.
- [7] K. Nummiaro, E. Koller-Meier, L. Van Gool, "Color Features for Tracking Non-Rigid Objects", *ACTA Automatica Sinica*, 2003.
- [8] K. Nummiaro, E. Koller-Meier, L. Van Gool, "Object Tracking with an Adaptive Color-Based Particle Filter", *Image and Vision Computing*, 2002.
- [9] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, "Color-based probabilistic tracking", *In European Conference on Computer Vision*, LNCS 2350, pp 661-675, Copenhagen, Denmark, 2002.

- [10] Z. Zivkovic, B. Krose, "An EM-like algorithm for color-histogram-based object tracking", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2004.
- [11] S. H. Cha, S. N. Srihari, "On measuring the distance between histograms", *Pattern Recognition*, Vol 35, no 6, pp 1355-1370, June 2002.
- [12] A. Jacquot, P. Sturm, O. Ruch, "Adaptive Tracking of Non-Rigid Objects Based on Color Histograms and Automatic Parameter Selection", *IEEE Workshop on Motion and Video Computing*, pp 103-109, Breckenridge, Colorado, January 2005.