Une version modifiée de l'Ensemble Tracking

Thomas PENNE[1] Vincent BARRA[2] Christophe TILMANT[3] Thierry CHATEAU[3]

[1] Prynel / Fédération TIMS
[2] LIMOS
[3] LASMEA

Prynel, RD 974 Corpeau, 21190 MEURSAULT France
tpenne@prynel.com

Résumé

Considérant le suivi comme un problème de classification binaire, l'algorithme Ensemble Tracking de Shaï Avidan permet de localiser un objet dans une séquence vidéo grâce à un classifieur entrainé pour distinguer les pixels du fond des pixels de l'objet. Nous introduisons ici une nouvelle approche pour la sélection des exemples d'apprentissage ainsi qu'une technique de modularisation de l'algorithme permettant au système de travailler sur des espaces de caractéristiques homogènes.

Mots Clef

Suivi de piétons, apprentissage supervisé, Adaboost.

Abstract

Ensemble Tracking algorithm from Shaï Avidan considers tracking as a binary classification problem. It allows to locate an object in a video sequence thanks to a classifier trained to distinguish background pixels from object ones. We introduce here a new learning samples selection approach as well as a method to make the system modular, allowing it to work on homogeneous feature spaces.

Keywords

Pedestrian tracking, supervised learning, Adaboost.

1 Introduction

Domaine de recherche privilégié de la vision par ordinateur, le suivi visuel a pour but la génération de la trajectoire d'un objet au cours du temps. Les deux phases majeures du suivi que sont la détection de l'objet et la mise en correspondance de ses instances peuvent être réalisées séquentiellement ou parallèlement. Lors d'une utilisation parallèle, comme dans l'algorithme Ensemble Tracking (ET) [1], la région de l'objet est obtenue par mise à jour itérative de sa position et des informations contenues dans les images précédentes. Nous proposons ici une version modifiée de cet algorithme capable de suivre en temps-réel un piéton dans l'image.

2 L'Ensemble Tracking

2.1 Travaux antérieurs

Le suivi d'objets comporte deux phases. La première est la détection d'objet qui regroupe 4 grandes catégories d'algorithmes : la détection de points [4] qui repère les points d'intérêt d'une image (Harris, KLT, SIFT), la soustraction de fond [4] qui construit un modèle de représentation de la scène, la segmentation [4] qui partitionne l'image en régions similaires (Mean Shift) et l'apprentissage supervisé [4] qui construit une fonction de mise en correspondance des données avec les sorties (étiquettes) souhaitées (boosting adaptatif [3], SVM). La seconde phase est la mise en correspondance des instances de l'objet qui regroupe elle aussi plusieurs familles d'algorithmes : le filtrage [4] qui permet la mise en correspondance des points d'intérêt d'une image à l'autre (filtres à particules ou de Kalman), le suivi de silhouette [4] qui utilise l'information codée à l'intérieur de la région objet et le suivi par noyau [4] qui recherche l'objet par comparaison à un modèle construit dans les images précédentes.
Seul le suivi par noyau permet un suivi temps-réel robuste des piétons. L'ET vient se classer dans cette catégorie mais diffère des travaux précédents par le fait qu'il ne se concentre pas seulement sur l'objet mais il se concentre aussi sur le fond. L'ET combine des classifieurs faibles construits par la méthode Adaboost [3] sur les données d'apprentissage pour obtenir un classifieur fort capable de discerner le fond de l'objet.

2.2 Algorithme de base

Le principe de base de l'ET 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. Etant donnés des examples étiquetés fond/objet sur la première image d'une séquence vidéo, l'ET construit un ensemble de classifieurs faibles. Pour cet apprentissage l'auteur utilise, pour chaque pixel, un vecteur de caractéristiques constitué des trois composantes couleur R, G et B et d'un histogramme d'orientations du gradient (8 directions). Un clas-

sifieur fort est ensuite calculé via l'algorithme Adaboost sur cet ensemble. Dans chaque image suivante de la séquence, le classifieur fort est utilisé pour fabriquer une carte de confiance représentant le taux de confiance accordé à chaque pixel vis à vis de son appartenance à l'objet. L'algorithme Mean Shift analyse ensuite cette carte et récupère la nouvelle position de l'objet. Enfin l'ensemble des classifieurs faibles est mis à jour : l'algorithme ne conserve que les meilleurs, met à jour leur pondération et en entraine de nouveaux afin de maintenir un nombre constant de classifieurs dans l'ensemble.

3 Modification de la méthode

3.1 Modularisation

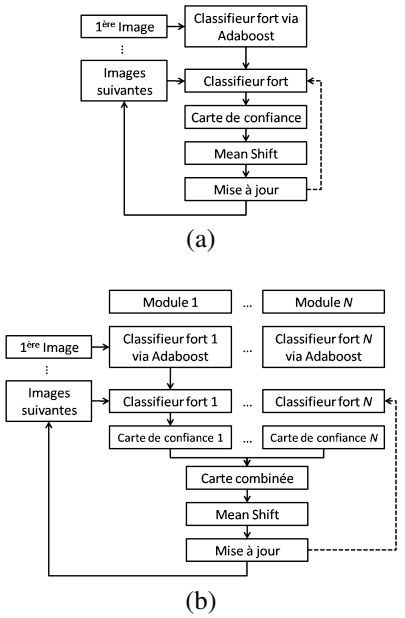

FIG. 1 – Comparaison des algorithmes. (a) Ensemble Tracking. (b) Ensemble Tracking modulaire.

La modification majeure de la méthode concerne sa décomposition en modules de caractéristiques. Chaque module utilise, indépendamment des autres, un vecteur de caractéristiques distinct calculé sur un espace homogène. L'algorithme s'en trouve modifié comme suit (cf. Figure 1) : on initialise via Adaboost, sur une première image, un classifieur fort par module en calculant, pour chaque pixel, le vecteur correspondant à l'espace de caractéristiques du module courant. Sur les images suivantes de la séquence, chaque module (classifieur fort) construit une carte de confiance relativement à son hyperplan propre. Les différentes cartes de confiance sont alors combinées relativement au score de classification de chaque classifieur fort. On obtient au final une carte de confiance unique qui est ensuite analysée par l'agorithme Mean Shift. Une fois la position de l'objet trouvée, chaque module met à jour son classifieur fort de la même manière que dans l'algorithme de S. Avidan.

Cette modularisation permettrait au système de réduire l'erreur Bayesienne globale induite par l'utilisation d'un espace de caractéristiques hétérogène. Aux vues des bons résultats obtenus avec les ondelettes de Haar dans [2], il est par ailleurs envisageable d'intégrer au système un module "de spécialisation" sans mise à jour entrainé sur une base de piétons de taille conséquente afin de garantir un suivi de personnes plus robuste. La modularisation permettrait également d'optimiser le rapport efficacité/temps en adaptant le nombre de modules aux performances souhaitées.

3.2 Optimisation

Outre la modularisation, nous proposons également une nouvelle approche de sélection des exemples d'apprentissage. Afin de rendre la zone spatiale d'apprentissage dynamique, nous suggérons d'effectuer le tirage des exemples en respectant une Gaussienne multivariée dont la matrice de covariance est construite de façon à tirer autant de pixels dans le rectangle objet qu'à l'extérieur (cf. Figure 2). Ce type d'échantillonage permet de régulariser l'apprentissage du fond et d'anticiper sur les changements brutaux qui peuvent intervenir.

FIG. 2 – Sélection des exemples d'apprentissage suivant une distribution Gaussienne multivariée.

4 Conclusion

Nous avons proposé ici une modification permettant de rendre l'ET plus robuste sur un type d'objet donné. Nous avons suggéré également une méthode de sélection des exemples d'apprentissage qui, combinée à une utilisation judicieuse de la modularisation, devrait permettre une optimisation des temps de calcul du sytème.

Références

[1] S. Avidan, *Ensemble Tracking,* IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), Vol. 29(2), pp. 261-271, 2007.

[2] T. Chateau, V. Gay-Belille, F. Chausse and J.T. Lapresté, *Real-Time Tracking with Classifiers,* Workshop on Dynamic Vision 2006, pp. 218-231, 2006.

[3] Y. Freund and R.E. Schapire, *A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting,* Computational Learning Theory : Eurocolt 95, pp. 23-37, 1995.

[4] A. Yilmaz, O. Javed and M. Shah, *Object Tracking : A Survey,* ACM Computing Surveys, Vol. 38(4), Article 13, 2006.