Suivi collaboratif mono-objet dans un réseau de caméras

Quoc C. LE[1,3] Donatello CONTE[1,3] Moncef HIDANE[2,3] Pierre GAUCHER[1,3]

[1] Université de Tours
[2] INSA Centre Val de Loire
[3] Laboratoire d'Informatique Fondamentale et Appliquée de Tours (LIFAT)

quoccuong.le@etu.univ-tours.fr

1 Résumé

Le suivi d'objets est un problème majeur en vision par ordinateur. Malgré des avancées impressionnantes au cours de la dernière décennie, il constitue encore un très grand challenge. Les algorithmes récents de suivi mono-object se basent sur le flux capturé par une seule caméra et sont très sensibles à l'occultation.
Nous proposons ici une méthode qui permet l'utilisation d'un réseau de caméras avec des champs de vue qui se chevauchent afin de suivre plus précisément un objet. Notre travail se situe dans le cadre du filtrage particulaire dans lequel nous proposons un modèle d'apparence favorisant la parcimonie de la cible, dans un dictionnaire de prototypes dynamiques. Dans notre approche, chaque caméra réalise un suivi local, basé sur son propore flux ainsi que sur celui des autres caméras du réseau. Cela permet une robustesse accrue vis-à-vis des occultations. La méthode proposée est généralisable dans le contexte d'un réseau de caméras dynamiques et son efficacité est illustrée dans un contexte de vidéo-surveillance.

2 État de l'art

Le suivi d'objets se retrouve dans plusieurs applications comme la vidéo-surveillance, la sécurité ou la navigation autonome. Pendant l'acquisition de la vidéo, la scène et l'apparence de l'objet varient suite à des changements d'environnement, de luminosité, d'échelle et d'occultation. Ces variations affectent directement la robustesse des algorithmes de suivi [1].
Depuis plusieurs années, des a priori de parcimonie dans dictionnaires fixes ou appris sont utilisés dans les applications importantes, comme la reconnaissance de visage ou le suivi visuel [2]. Pour le suivi, l'approche consiste à échantillonner un certain nombre de candidats à l'aide d'un modèle aléatoire, puis de sélectionner celui qui minimise l'erreur, obtenue comme combinaison linéaire creuse de prototypes pré-calculés. Cet ensemble de prototypes, appelé dictionnaire, peut être mis à jour régulièrement afin de maintenir le modèle d'apparence à jour. Les développements ultérieurs de cette approche se sont concentrés sur la compensation de l'occultation partielle en ajoutant des prototypes triviaux [2, 3] ou sur des modèles d'apparence par morceaux [4, 5].
Des approches multi-caméras ont été proposées pour pouvoir appliquer le suivi visuel à une zone observée plus étendue [6, 7]. Presque tous les auteurs travaillent sous l'hypothèse que la topologie des caméras est connue et que le calibrage des caméras a été réalisé au préalable. Nous travaillerons sous les mêmes hypothèses.

3 Approche proposée

3.1 Filtre particulaire et parcimonie

Notre méthode est basé sur le framework de filtre particulaire [8]. Soit $\mathbf{x}_t$ l'état caché à l'instant t, $\mathbf{y}_t$ l'observation récupérée dans le séquence d'images, f_t l'évolution temporelle de $\mathbf{x}_t$, h_t la fonction de mesure et $\mathbf{v}_t$, $\mathbf{w}_t$ des bruits blancs indépendants. Le filtre Bayesien non-linéaire consiste à estimer la densité à posteriori $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ (où $\mathbf{y}_{1:t}$ dénote la séquence des mesures de temps 1 à t). Étant donné que cette densité difficile à calculer, nous suivons la technique des filtres particulaires pour résoudre ce problème. Dans notre proposition de suivi d'objets résolu par filtre particulaire, l'état caché est représente par $\mathbf{x}_t = (x_t, y_t, s_t, \alpha_t)$ où x_t, y_t sont les coordonnées en pixel, s_t est le paramètre d'échelle de la boîte englobante et α_t est le ratio hauteur/largeur de l'objet à l'instant t.
Dans l'étape de modélisation de l'apparence de l'objet, nous utilisons un a priori de parcimonie dans un dictionnaire de prototypes en nous inspirant à la démarche de [9]. En général, chaque candidat est présenté comme une combinaison parcimonieuse linéaire du modèle de la cible et des modèles triviaux (la base canonique). L'ensemble des modèles, qui s'appelle

dictionnaire, est mis à jour périodiquement. La partie trivial du dictionnaire, formée par des vecteur de la base canonique, sert à compenser les occultations partielles d'une cible. Les coefficients de représentation sont calculés en résolvant un problème aux moindres carrés pénalisé par la norme ℓ_1 et une contrainte de positivité.

Pour chaque frame t, le filtre particulaire donne un ensemble de candidats noté $(\mathbf{y}_t^{(n)})_{n=1}^N \in (\mathbb{R}^d)^N$. L'approche présentée dans [2] modélise l'apparence de l'objet comme une *combinaison linéaire parcimonieuse* d'un ensemble de prototypes. Le dictionnaire est constitué de deux parties : le modèle de cible $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_p]$ et le modèle trivial $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_d]$. Soit $\mathbf{D} = [\mathbf{T}, \mathbf{E}]$, chaque candidat $\mathbf{y}^{(n)}$ est approximé par une combinaison linéaire dont les coefficients sont la solution de l'Eq. 1.

$$\left[\mathbf{a}_*^{(n)}, \mathbf{b}_*^{(n)}\right] = \underset{\mathbf{a},\mathbf{e}\geq 0}{\operatorname{argmin}} \left\| \begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix} \right\|_1 + \lambda \left\| [\mathbf{T}, \mathbf{E}] \begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix} - \mathbf{y}^{(n)} \right\|_2^2, \tag{1}$$

Parmi tous les candidats $\mathbf{y}^{(n)}$, celui dont l'erreur de reconstruction est plus faible sera sélectionné, $\mathbf{y}^* = \mathbf{y}_{i^*}$, où $i^* = \operatorname{argmin}_{1\leq n\leq N} \|\mathbf{T}\mathbf{a}_*^{(n)} - \mathbf{y}^{(n)}\|_2$.

L'ensemble du modèle de cible est mis à jour régulièrement pour s'adapter aux changements de luminosité ou de pose. Enfin, le poids d'importance de chaque particule est calculé via l'erreur de reconstruction à l'instant t. Ensuite, l'étape de re-échantillonage régénère les particules à l'instant $t+1$ sur la base des poids au dernier instant $\left\{\mathbf{x}_t^{(n)}, w_t^{(n)}\right\}$.

3.2 Méthode proposée de suivi par caméras collaboratives

Afin de gagner en robustesse dans le suivi d'un seul objet, nous proposons de travailler avec plusieurs caméras. Nous supposons que chaque caméra traite son propre flux vidéo, calcule les coordonnées de l'objet sur le plan image et le projette sur le plan commun Z. Nous supposons également que chaque paire de caméras peut échanger entre elles des coordonnées 2D ou 3D.

En détail, étant donnée une caméra C_k, tous les candidats sont encodés dans son dictionnaire local et celui dont l'erreur de reconstruction est plus faible sera pré-sélectionnée. Si le taux de parcimonie, c'est-à-dire le nombre d'atomes non nuls selectionnés, est supérieur à un seuil ϵ, il sera considéré comme la cible cherchée. Sinon, les candidats de l'objet sur la caméra C_k sont encodés, après une projection, sur un autre caméra $C_{k'}$ à partir du dictionnaire de $C_{k'}$. Si un candidat a un taux de parcimonie inférieur au seuil fixe, ce candidat, après une projection sera considéré comme la cible cherché sur la caméra C_k. Sinon l'objet est considéré comme étant sortie de la scène.

4 Résultats expérimentaux

Parmi les nombreux benchmarks qui ont été développés pour le suivi d'objets, PETS2009 [10] est très intéressant pour plusieurs raisons : (a) il est utilisé depuis de nombreuses années par la plupart des chercheurs qui proposent des algorithmes de suivi d'objets ; (b) les personnes sont filmés en utilisant 8 caméras avec différents points de vue, différentes distances par rapport aux objets, différents éclairages et conditions d'environnement ; (c) les caméras sont calibrées et synchronisées. Nous évaluons donc notre approche sur cette base.

En ce qui concerne la mesure d'évaluation, nous utilisons le score de chevauchement moyen (AOS) ainsi que les indices *missing objects (MO)* et *false positive (FP)*. Un algorithme de suivi performant doit avoir un score AOS élevée et des indices MO et FP faibles.

Nous avons comparé notre méthode à des algorithmes bien connus, y compris **MIL** [11], **TLD** [12], **KCF** [13]. Tous ces algorithmes ont été implémentés dans la bibliothèque OpenCV. Nous avons également sélectionné un sous-ensemble des algorithmes les plus performants basés sur le benchmark [14] (disponible sur le site http ://www.visual-tracking.net) : **STRUCK** [15], **SCM** [16] et **ASLA** [4].

Le résultat du tableau 1 démontre que l'introduction de plusieurs caméras améliore les performances du suivi. La configuration avec plusieures caméras collaboratives présente, en général, les meilleurs résultats.

5 Conclusion

Nous avons présenté un nouvel algorithme de suivi en ligne multi-caméra mono-objet robuste. Notre méthode repose sur un ensemble de caméras synchronisées et calibrées. Notre hypothèse de travail est que chaque caméra est capable de mettre en oeuvre un tracker local et de communiquer un ensemble de coordonnées 3D aux autres caméras. Une validation et une comparaison numériques ont été effectuées sur des données réelles.

Références

[1] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking : An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, 2014.

Algorithme \ *Vue*	AOS			MO			FP		
	v1	v5	v8	v1	v5	v8	v1	v5	v8
MIL	0.159	0.315	0.100	0.0	0.016	0.009	0.0	0.0	0.0
TLD	0.171	0.182	0.119	0.038	0.016	0.011	0.0	0.0	0.0
BOOSTING	0.039	0.260	0.175	0.0	0.016	0.009	0.0	0.0	0.161
STRUCK	0.321	0.282	0.175	0.0	0.0	0.007	0.0	0.0	0.0
KCF	0.138	0.018	0.013	0.001	0.016	0.009	0.0	0.0	0.0
ASLA	0.478	0.541	0.506	0.0	0.0	0.0	0.0	0.108	0.047
SCM	0.498	0.394	0.479	0.0	0.0	0.0	0.075	0.282	0.194
L1	0.553	0.433	0.340	0.0	0.0	0.0	0.0	0.256	0.064
Cam157	**0.632**	**0.575**	–	0.0	0.0	–	0.0	0.008	–
Cam168	0.630	–	0.566	0.0	–	0.0	0.0	–	0.045
Cam178	**0.636**	–	**0.616**	0.0	–	0.0	0.002	–	0.0
Cam1678	0.573	–	0.604	0.0	–	0.0	0.0	–	0.030
Cam135678	0.612	**0.611**	**0.611**	0.0	0.0	0.0	0.0	0.0	0.0

TABLE 1 – Résultats sur les séquences 'PETS09 _S2L1' sur plusieurs vues. Les indices considérés sont l'AOS, le taux MO et le taux FP. Le gris foncé indique la meilleure valeur et le gris clair la deuxième meilleure valeur.

[2] X. Mei and H. Ling, "Robust visual tracking using ℓ 1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1436–1443.

[3] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *Computer vision and pattern recognition (CVPR), 2011 IEEE conference on*. IEEE, 2011, pp. 1257–1264.

[4] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1822–1829.

[5] S. Rousseau, P. Chainais, and C. Garnier, "Dictionary learning for a sparse appearance model in visual tracking," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4506–4510.

[6] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 267–282, 2008.

[7] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[8] A. Doucet, N. De Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[9] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 11, pp. 2259–2272, 2011.

[10] J. Ferryman and A. Shahrokni, "Pets2009 : Dataset and challenge," in *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. IEEE, 2009, pp. 1–6.

[11] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.

[12] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[14] D. Wang, H. Lu, and M.-H. Yang, "Online object tracking with sparse prototypes," *IEEE transactions on image processing*, vol. 22, no. 1, pp. 314–325, 2013.

[15] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck : Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[16] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1838–1845.