

Suivi temps réel robuste d'objets 3D avec connaissance a priori du modèle

Robust Real Time Tracking of 3D Objects with a priori knowledge of the model

Lucie Masson¹

Frédéric Jurie²

Michel Dhome¹

¹ LASMEA - UMR6602 CNRS

Université Blaise Pascal

63177 AUBIERE cedex

Prénom.NOM@lasmea.univ-bpclermont.fr

² INRIA

Laboratoire GRAVIR / projet LEAR

Unité de Recherche INRIA Rhône-Alpes

Frederic.Jurie@inrialpes.fr

Résumé

Le sujet de cet article est le suivi d'objets 3D rigides. Nous proposons un algorithme qui combine efficacité – l'algorithme est avant tout prévu pour pouvoir être exécuté en temps réel sur une architecture standard –, robustesse aux occultations et précision. Il a pour objectif de réaliser le suivi d'objets rigides dont la géométrie 3D ainsi que la texture des surfaces sont connues. La base de cet algorithme est l'utilisation de nombreuses occurrences d'un algorithme efficace de suivi planaire, dont les résultats sont combinés de façon robuste pour estimer la pose 3D de l'objet. Des résultats expérimentaux viennent prouver la validité de cette approche.

Mots Clef

suivi, temps réel, objets rigides

Abstract

In this article the problem of tracking rigid 3D objects is addressed. The contribution of the proposed approach is an algorithm which combines : efficiency (i.e. the algorithm is designed before all to be real time using standard architecture), robustness (occlusions are allowed) and accuracy. It is devoted to the tracking of 3D rigid objects, assuming that the 3D geometry as well as the texture of the surface is known. Such performances can be obtained through a two levels scheme : the core of the approach consists in an efficient 2D patch tracker whose results are combined robustly to compute the 3D object pose. This article provides experimental results proving the soundness of the proposed approach.

Keywords

tracking, real time, rigid objects

1 Introduction

Il existe de nombreux algorithmes permettant de suivre des objets dans des séquences vidéo. Il est possible de les classer en deux catégories : ceux basés sur les primitives, où l'objectif est de suivre certaines primitives dans les images sans utiliser de modèle de l'objet [1, 7], et ceux basés sur la connaissance d'un modèle connu – 2D ou 3D – de la cible [11, 4]. Notre algorithme appartient à la seconde catégorie.

L'objectif de l'algorithme proposé est d'estimer la pose de l'objet-cible – notée \mathbf{L} , et correspondant aux translations et rotations de l'objet – dans le système référentiel de la caméra à partir des images issues du flux vidéo. Une prédiction de \mathbf{L} est supposée disponible pour chaque nouvelle image. Les méthodes pour obtenir cette prédiction dépassent le cadre de cet article. On peut imaginer utiliser un filtre de Kalman, un filtre à particule, ou juste estimer que l'objet-cible se déplace suffisamment peu d'une image à l'autre pour que la position prédite à l'instant t se réduise à la position de l'objet à l'instant $t - 1$.

La surface de l'objet-cible est décomposée en petites surfaces carrées de centre et de normale connus. Ces « surfaces carrées », qui sont planaires et tangentes à la surface de l'objet, sont appelées des *patches*. Ces patches sont disposés sur le modèle 3D de l'objet. C'est leur répartition sur toute la surface de l'objet qui permet à l'algorithme d'être robuste aux occultations : quand l'objet est partiellement occulté, les patches qui ne sont pas occultés amènent suffisamment d'information pour rendre le suivi de l'objet-cible possible.

Plusieurs vues-clefs sont nécessaires afin d'apprendre la texture de chacun des patches. La pose de l'objet dans ces vues est donnée manuellement en associant certains

points du modèle avec leur projection dans l’image-clef. Les patches peuvent être répartis de façon homogène sur toute la surface ou centrés sur des points d’intérêt suivant le critère de Harris et Stephens [5], les patches étant placés sur les points où ce critère est maximum. Un résultat typique de cette sélection de la position des patches peut être observé à la figure 1.



FIG. 1 – Exemple de répartition des patches à la surface d’un cube.

Pendant le suivi, la pose de l’objet peut être calculée en mettant en correspondance les centres des patches avec leur projection dans l’image. Ces appariements 3D-2D sont utilisées comme point de départ d’un algorithme d’estimation robuste de pose.

Il est possible de diviser la description de l’algorithme en deux parties : la mise en correspondance 3D-2D et l’estimation robuste de pose. Ces deux parties vont être décrites dans les deux sections qui suivent. Des résultats expérimentaux seront présentés dans la dernière section.

2 Appariements 3D-2D

Cette section a pour but de décrire la partie de l’algorithme chargée de la mise en correspondance des centres 3D des patches avec leur projection dans l’image.

Une estimation de la pose de l’objet – notée \mathbf{L} – est supposée connue au début de cette étape. La texture de chaque patche a été apprise sur les vues-clef. La pose étant supposée connue, il est possible de calculer la transformation homographique qui explique les coordonnées des quatre coins des patches projetés dans l’image.

Cette homographie va permettre de comparer la texture dans l’image avec la texture du patche apprise précédemment. Si la pose estimée est identique à la pose courante, les deux textures vont être identiques. Si la pose est légèrement différente, les textures seront différentes. L’image de différence, qui correspond à la différence entre les deux textures, est représentative de l’erreur lors de l’estimation de la pose. Nous avons montré dans [8] comment calculer

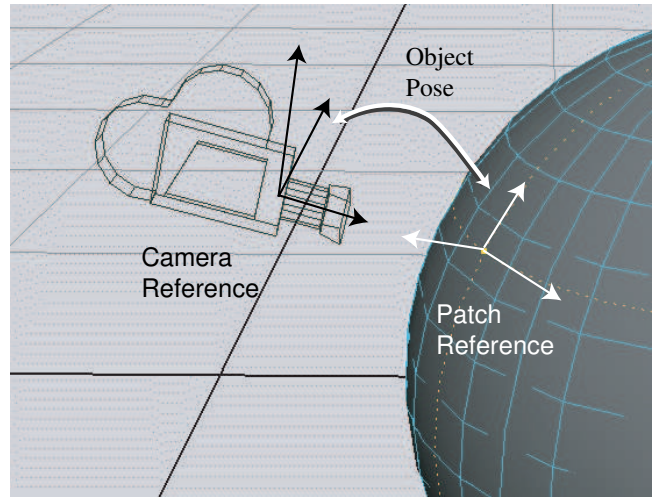


FIG. 2 – Référentiel des patches et référentiel de la caméra.

efficacement et précisément la pose d’un patche courant à partir d’une image de différence.

Cependant, lorsque l’image de différence est petite – ce qui est le cas lorsque l’on utilise de « petits patches » –, tous les paramètres de déplacement ne peuvent pas être estimés correctement. C’est pourquoi seuls quatre paramètres de déplacement seront calculés pour chaque patche : les deux translations 2D, la rotation 2D suivant le plan tangent à la surface et le facteur d’échelle. On utilise la même approche que celle proposée dans [8]. Une fois ces paramètres estimés, la transformation correspondante est appliquée au centre du patche pour le reprojeter dans l’image. Puis, connaissant les coordonnées de la projection du centre du patche dans l’image, nous obtenons l’appariement entre les coordonnées 3D du centre du patche dans le référentiel de la caméra et les coordonnées 2D de sa projection dans l’image.

En appliquant cette stratégie à chacun des patches visibles on obtient un ensemble d’appariements 3D-2D, qui va permettre d’estimer la pose de l’objet-cible.

2.1 Méthode de suivi planaire utilisant l’image de différence

Nous n’avons jusqu’à présent que survolé l’algorithme utilisé pour estimer la transformation planaire d’un patche à partir des images de texture – celle apprise et celle dans l’image courante –, algorithme décrit plus précisément dans [8]. Son principe consiste à considérer qu’au voisinage de la position initiale du patche, il existe une relation linéaire entre les paramètres de déplacement et l’image de différence. Cet algorithme se décompose en deux phases, une phase d’apprentissage et une phase de suivi proprement dit.

Phase d’apprentissage : La texture du patche est décrite par n points (p_1, p_2, \dots, p_n) , où $p_i = (x, y)$. Ils forment le vecteur \mathbf{P} . Ces coordonnées sont exprimées dans le réfé-

rentiel du patche. Les p_i points sont sélectionnés de façon à respecter 2 contraintes : être répartis sur toute la surface du patche, et être positionnés là où le gradient de l'image est élevé. Dans le cas où l'on utilise de « petits patches », cette dernière contrainte est moins importante et l'on peut se contenter de répartir de façon homogène les points sur le patche.

En pratique, le patche est découpé en baquets et un nombre donné de points est choisi dans chaque baquet. Cette sélection peut par exemple se faire en tirant pseudo-aléatoirement des points dans chaque baquet, et en ne conservant que ceux ayant le gradient le plus élevé. L'intensité lumineuse en chacun de ces points est stockée dans le vecteur $\mathbf{V}(0)$.

Soit $\mu(t) = (\mu_1(t), \mu_2(t), \dots, \mu_p(t))$ le vecteur de paramètres décrivant la position du patche à l'instant t dans le référentiel du patche. Ces paramètres correspondent aux paramètres d'une fonction f qui amène les points de la position courante à la position estimée.

On applique m « petites perturbations » à la position initiale du patche.

Soit $\delta\mathbf{V}(i) = \mathbf{V}(0) - \mathbf{V}(i)$ le vecteur des différences d'intensité, calculé en faisant la différence entre le vecteur des intensités à la position initiale et le vecteur à la nouvelle position, après la $i^{\text{ième}}$ perturbation.

Soit $\delta\mu(i)$ le vecteur décrivant la $i^{\text{ième}}$ perturbation.

Nous considérons que la relation entre le déplacement et les valeurs d'intensité est linéaire¹. Cela signifie qu'il existe une matrice constante \mathbf{H} qui vérifie :

$$\delta\mu(i) = \mathbf{H} \cdot \delta\mathbf{V}(i) \quad (1)$$

La phase de tracking consiste donc à déterminer la matrice \mathbf{H} en résolvant le système suivant :

$$\begin{pmatrix} \delta\mu_0(0) & \dots & \delta\mu_m(0) \\ \vdots & & \vdots \\ \delta\mu_0(p) & \dots & \delta\mu_m(p) \end{pmatrix} = \mathbf{H} \cdot \begin{pmatrix} \delta\mathbf{V}_0(0) & \dots & \delta\mathbf{V}_m(0) \\ \vdots & & \vdots \\ \delta\mathbf{V}_0(n) & \dots & \delta\mathbf{V}_m(n) \end{pmatrix}$$

Phase de suivi : Nous calculons le vecteur de différence d'intensité $\delta\mathbf{V}$ comme expliqué précédemment. Ce vecteur permet d'obtenir $\delta\mu$, une estimation des paramètres de déplacement dans le référentiel initial, en utilisant l'équation (1).

Cette équation n'étant valable qu'au voisinage de la position initiale du patche, on change de référentiel pour se placer dans le référentiel du patche, en utilisant

$$\mathbf{F}(\mu(t)) = \mathbf{F}(\mu(t-1)) \circ \mathbf{F}(\delta\mu(t)),$$

$\mathbf{F}(\mathbf{v})$ étant la matrice 3×3 obtenue à partir du vecteur de paramètres de transformation \mathbf{v} .

On peut vite s'apercevoir que la complexité algorithmique de cette approche est très faible : uniquement une multiplication matricielle et plusieurs opérations mathématiques simples (addition, etc.).

C'est pour cette raison que cet algorithme est très efficace.

En plus de son efficacité, cet algorithme a aussi l'avantage d'être précis ; la régression donnée en éq. 1 permet d'obtenir une précision allant jusqu'au sub-pixellique dans le calcul des paramètres de déplacement. Des résultats sont disponibles dans la partie « résultats expérimentaux » de cet article.

3 Estimation de la pose de l'objet

Une fois que les appariements 3D-2D ont été obtenus grâce au suivi planaire décrit à la section précédente, il est temps de calculer la pose de l'objet-cible. Cette section décrit comment la calculer à l'aide d'un estimateur robuste.

Pour chaque patche, la phase d'appariement (section 2) nous donne les paramètres de déplacement du patche (rotation 2D, translations, facteur d'échelle) dans le référentiel du patche. Ces paramètres permettent de calculer la position du centre de chaque patche dans l'image, ce qui nous donne les appariements 3D-2D. Ces appariements sont notés (x, y, X, Y, Z) , avec (x, y) la position du centre du patche dans l'image et (X, Y, Z) les coordonnées 3D du centre du patche dans le référentiel de l'objet.

La pose de l'objet est calculée est utilisant l'algorithme itératif développé par Lowe [10] et modifié par Dhome *et al.* [3]. Cet algorithme est rapide et précis ; moins de 10 itérations sont généralement nécessaires pour estimer la pose. C'est pourquoi cet algorithme est bien adapté à notre application. En pratique, quand une prédiction de la pose de l'objet est connue, la convergence est obtenue en moins de 3 ou 4 itérations.

Malheureusement, l'algorithme de Lowe n'est pas robuste aux faux appariements. Comme nous voulons que notre algorithme puisse supporter les occultations, il est nécessaire de supprimer les faux appariements – ou du moins de ne pas les prendre en compte – lors de la phase de calcul de pose. Cela peut être fait en ajoutant un M-estimateur de Tuckey [6] à l'algorithme de calcul de pose.

Soit $P_{i,t} = (X, Y, Z)$ les coordonnées 3D du point i à l'instant t , et $P_{i,t+1}$ ses nouvelles coordonnées à l'instant $t+1$. L'algorithme itératif de Lowe calcule les deux matrices $R_{\alpha,\beta,\gamma}$ et $T_{u,v,w}$ qui vérifient

$$P_{i,t+1} = R_{\alpha,\beta,\gamma} \cdot P_{i,t} + T_{u,v,w}$$

Soit $p_{i,t+1} = (x, y)$ les coordonnées dans l'image du point $P_{i,t+1}$. Considérons $p_{i,t+1} = (x, y)$ comme l'intersection du plan de l'image avec deux plans d'« interprétation », quasi-horizontaux et quasi-verticaux, définis par les vecteurs normaux $N_{i,t+1}^1$ et $N_{i,t+1}^2$. L'algorithme de Lowe minimise le critère suivant pour les n centres des patches :

$$E = \sum_{i=1}^n \sum_{j=1}^2 [distance(P_{i,t+1}, N_{i,t+1}^j)]^2$$

Si nous appelons $\mathcal{F}(\alpha, \beta, \gamma, u, v, w, N_{i,t+1}^j, P_{i,t})$ la distance de $\tilde{P}_{i,t} = R_{\alpha,\beta,\gamma} \cdot P_{i,t} + T_{u,v,w}$ au plan défini par $N_{i,t+1}^j$, nous

²Les plans passant par le centre optique de la caméra et respectivement par la ligne et la colonne de l'image qui caractérisent le point $P_{i,t+1}$

¹à condition d'être au voisinage de la position initiale, bien sûr

pouvons écrire :

$$E = \sum_{i=1}^n \sum_{j=1}^2 \mathcal{F}(\alpha, \beta, \gamma, u, v, w, N_{i,t+1}^j, P_{i,t})$$

Notons $\mathbf{L} = (\alpha, \beta, \gamma, u, v, w)$ le vecteur décrivant une pose (rotations et translations). Soit $\mathbf{L}_k = (\alpha_k, \beta_k, \gamma_k, u_k, v_k, w_k)$ le vecteur décrivant la pose à l'itération k de l'algorithme de Lowe.

Pour résoudre le problème avec la méthode de Newton-Raphson on effectue un développement à l'ordre un :

$$-\mathcal{F}(\mathbf{L}_k, N_{i,t+1}^j, P_{i,t}) = \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{i,t+1}^j, P_{i,t})}{\delta \alpha} \Delta \alpha + \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{i,t+1}^j, P_{i,t})}{\delta \beta} \Delta \beta + \dots + \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{i,t+1}^j, P_{i,t})}{\delta w} \Delta w$$

Ou, en utilisant une notation matricielle, avec \mathbf{F} un vecteur de dimension $2n$ et \mathbf{D} une matrice de dimension $2n \times 6$:

$$\mathbf{F} = \begin{bmatrix} -\mathcal{F}(\mathbf{L}_k, N_{1,t+1}^1, P_{1,t}) \\ -\mathcal{F}(\mathbf{L}_k, N_{1,t+1}^2, P_{1,t}) \\ \vdots \\ -\mathcal{F}(\mathbf{L}_k, N_{n,t+1}^1, P_{n,t}) \\ -\mathcal{F}(\mathbf{L}_k, N_{n,t+1}^2, P_{n,t}) \end{bmatrix}$$

$$\mathbf{D} = \begin{pmatrix} \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{1,t+1}^1, P_{1,t})}{\delta \alpha} \Delta \alpha & \dots & \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{1,t+1}^1, P_{1,t})}{\delta w} \Delta w \\ \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{1,t+1}^2, P_{1,t})}{\delta \alpha} \Delta \alpha & \dots & \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{1,t+1}^2, P_{1,t})}{\delta w} \Delta w \\ \vdots & \vdots & \vdots \\ \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{n,t+1}^1, P_{n,t})}{\delta \alpha} \Delta \alpha & \dots & \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{n,t+1}^1, P_{n,t})}{\delta w} \Delta w \\ \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{n,t+1}^2, P_{n,t})}{\delta \alpha} \Delta \alpha & \dots & \frac{\delta \mathcal{F}(\mathbf{L}_k, N_{n,t+1}^2, P_{n,t})}{\delta w} \Delta w \end{pmatrix}$$

On peut voir que chaque itération de l'algorithme correspond à la résolution du système matriciel suivant :

$$\mathbf{F} = \mathbf{D} \cdot \Delta \mathbf{L}$$

Finalement, pour chaque itération, nous remettons à jour le vecteur \mathbf{L} :

$$\mathbf{L}_{k+1} = \mathbf{L}_k + \Delta \mathbf{L}^3$$

Après ce bref résumé de l'algorithme itératif de Lowe, il reste à introduire la robustesse dans le calcul de la pose. Nous avons pour cela ajouté un M-estimateur tel qu'expliqué par Comport *et al.* [2]. Cela consiste à ajouter un terme de pondération à l'expression

$$\mathbf{W} \cdot \mathbf{F} = \mathbf{W} \cdot \mathbf{D} \cdot \Delta \mathbf{L}$$

avec

$$\mathbf{W} = \begin{pmatrix} w_0 & 0 & 0 & \dots & \dots & \dots \\ 0 & w_0 & 0 & \dots & \dots & \dots \\ 0 & 0 & w_1 & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & w_n & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & w_n \end{pmatrix}$$

³puisque'il s'agit d'une transformation géométrique, la notation + est bien sûr abusive...

Le calcul des w_i se fait de la façon suivante :

$$w_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma}$$

où ψ est la fonction d'influence et $\delta_i = \Delta_i - \text{Med}(\Delta)$, Δ étant le vecteur des résidus. σ représente la valeur de l'écart-type du bruit sur les « bonnes » mesures, calculée par $\sigma = 1.4826 \cdot \text{Med}(|\delta|)$.

La fonction d'influence de l'estimateur de Tuckey est donnée par :

$$\psi(u) = \begin{cases} u(C^2 - u^2)^2 & \text{si } |u| \leq C \\ 0 & \text{sinon} \end{cases}$$

avec C le facteur de proportionnalité de Tuckey de valeur $C = 4.6851$.

On constate qu'avec cet estimateur les appariements erronés obtiennent un poids nul, et sont donc complètement ignorés lors du calcul de pose. C'est ce qui permet à l'algorithme de supporter les occultations et de n'estimer la pose qu'à partir de correspondances de points significatives.

4 Résultats expérimentaux

Cet algorithme a été implanté sur un ordinateur de bureau relié à une caméra numérique SONY permettant d'obtenir un flux vidéo de 30 images par seconde sur le bus FireWire (IEEE1394).

Il a été testé sur des séquences réelles et artificielles. Pour des raisons de simplicité, un modèle cubique a été choisi. On dispose 16 patchs sur chaque face du cube, chaque patche ayant sa texture décrite par un vecteur \mathbf{V} de 100 éléments. L'apprentissage se fait sur 4 niveaux de perturbation (soit 4 matrices \mathbf{H} , voir équation 1). Le niveau de perturbation le plus élevé est peu précis mais permet de tolérer des déplacements importants, le niveau le plus faible est extrêmement précis. En les utilisant l'un après l'autre, le moins précis en premier, on obtient un suivi stable et précis.

Avec ces paramètres, les deux étapes de suivi (estimation des appariements 3D-2D et calcul de la pose) demandent 20ms par image, l'appariement étant la phase la plus longue, approximativement 0.5ms par patche.

Comme on peut le voir aux figures 3 et 4, l'algorithme est robuste aux occultations : malgré les doigts qui cachent une partie de l'objet, le suivi continue.

Des séquences générées artificiellement ont été utilisées pour évaluer la précision de notre algorithme ; dans ce cas, la vérité terrain est connue. Dans ces séquences, le cube a été suivi sans, puis avec M-estimateur, et la précision des résultats a été mesurée. Comme critère de précision, nous avons utilisé la médiane des résidus dans l'image : pour chaque sommet de l'objet, la distance entre sa position re-projetée et sa véritable position dans l'image est calculée, et la valeur conservée comme critère est la médiane de ces distances.

La figure 5 montre la valeur de cette médiane dans le cas d'une séquence artificielle de 30 images perturbées par un



FIG. 3 – Suivi d'un cube occulté (1)



FIG. 4 – Suivi d'un cube occulté (2)

bruit gaussien. Bien évidemment, sans la robustesse apportée par le M-estimateur l'objet-cible est perdu rapidement (ici à l'image 18). Avec le M-estimateur, non seulement la précision du résultat est sub-pixellique, c'est à dire que la distance médiane entre les sommets du cube et leur projection dans l'image est inférieur à 1 pixel, mais cette distance médiane est relativement constante au cours du temps. Cela est dû au fait que le M-estimateur ne prend pas en compte les appariements jugés mauvais.

Des extraits d'une séquence réelle sont visibles à la figure 6.

5 Conclusion et perspectives

Dans cet article nous avons décrit un algorithme efficace, précis et robuste de suivi d'objet 3D rigide.

La précision et l'efficacité sont obtenus grâce à l'utilisation d'un algorithme de suivi planaire fournissant des appariements 3D-2D avec un coût algorithmiquement faible. La robustesse est obtenue par l'utilisation d'un M-estimateur au sein d'un algorithme de calcul de pose standard.

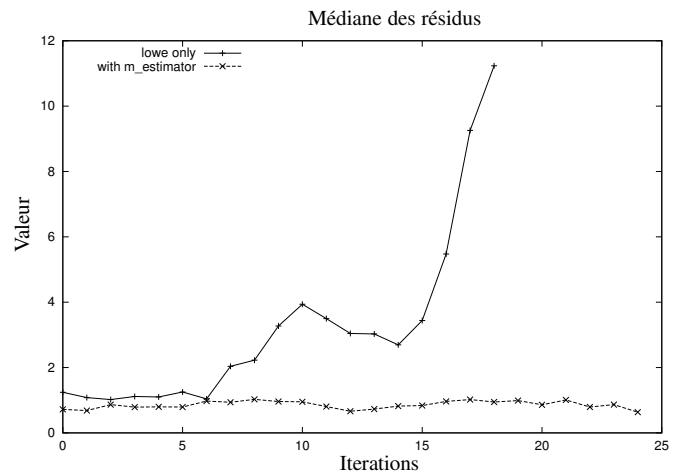


FIG. 5 – Médiane des résidus, en pixels, pour une séquence artificielle

Des améliorations pourraient être ajoutées à notre algorithme. Dans notre implantation, la texture est apprise une fois pour toutes. On pourrait imaginer l'apprendre en ligne pour construire le modèle de l'objet plus facilement.

Une autre limitation est la zone de convergence relativement restreinte des patches de suivi : leur zone de convergence dépend de leur taille dans l'image, et comme ils sont de taille réduite leur zone de convergence est elle aussi réduite. Pour résoudre ce problème nous avons couplé notre algorithme avec un algorithme simple de reconnaissance de formes basé sur l'appariement de points d'intérêt, qui prend le relais si l'objet s'est déplacé trop vite dans l'image, et permet d'initialiser très facilement le suivi. Même s'ils sont généralement plus gourmands en temps de calcul, les algorithmes de reconnaissance ont l'avantage de pouvoir supporter n'importe quel déplacement important. Un article récent de Vincent Lepetit [9] a d'ailleurs montré qu'il était possible de suivre un objet en utilisant des techniques de reconnaissance couplées à un algorithme de classification efficace.

Références

- [1] COMANICIU (D.), RAMESH (V.) et MEER (P.). « Real-time tracking of non-rigid objects using mean shift ». dans *CVPR00*, p. II : 142–149, 2000.
- [2] COMPORT (M. A. et Pressigout), MARCHAND (E.) et CHAUMETTE (F.). « Une loi de commande par asservissement visuel robuste aux mesures aberrantes ». dans *14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, RFIA'04*, Toulouse, France, janvier 2004.
- [3] DHOME (M.), RICHTIN (M.) et LAPRESTE (J.-T.), « Determination of the attitude of 3d objects from a single perspective view », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, n° 12, p. 1265–1278, 1989.

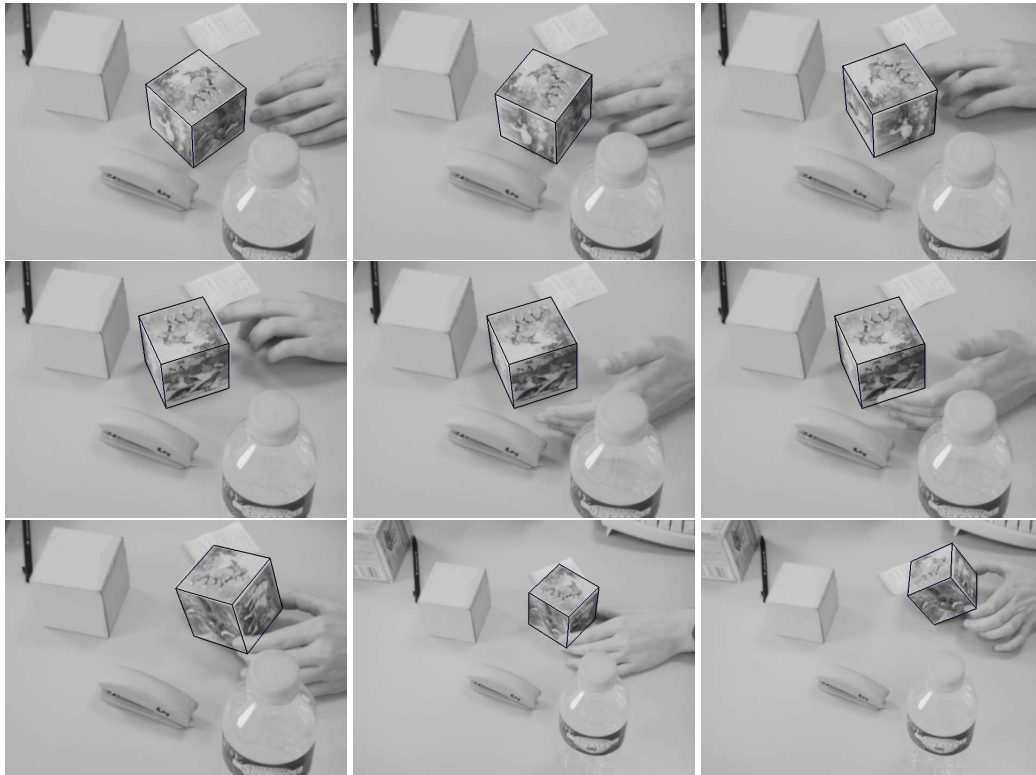


FIG. 6 – Extraits d'une séquence réelle de suivi

- [4] DRUMMOND (T.) et CIPOLLA (R.), « Real-time tracking of complex structures with on-line camera calibration », *IVC*, vol. 20, n° 5-6, p. 427–433, mars 2002.
- [5] HARRIS (C.) et STEPHENS (M.). « A combined corner and edge detector ». dans *Proceeding of the 4th Alvey Vision Conference*, p. 147–151, 1988.
- [6] HUBER (P.). *Robust Statistics*. Wiley, 1981.
- [7] JEPSON (A. D.), FLEET (D.) et EL-MARAGHI (T.). « Robust on-line appearance models for visual tracking ». dans *IEEE Conference on Computer Vision and Pattern Recognition*, p. pp. 415–422, 2001.
- [8] JURIE (F.) et DHOME (M.). « Real time template matching ». dans *Proc. IEEE International Conference on Computer vision*, p. 544–549, Vancouver, Canada, juillet 2001.
- [9] LEPETIT (V.), PILET (J.) et FUA (P.). « Point matching as a classification problem for fast and robust object pose estimation ». dans *Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [10] LOWE (D.). *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
- [11] LOWE (D.), « Robust model-based motion tracking through the integration of search and estimation », *International Journal of Computer Vision*, vol. 8, n° 2, p. 113–122, 1992.