

Suivi d'objets pour une caméra embarquée dans un drone

Benoît LOUVAT, Laurent BONNAUD, Nicolas MARCHAND, Gerard BOUVIER

GIPSA-lab

961 rue de la Houille Blanche Domaine universitaire - BP 46 St Martin d'Hères cedex France

benoit.louvat@gipsa-lab.inpg.fr, laurent.bonnaud@gipsa-lab.inpg.fr,
nicolas.marchand@gipsa-lab.inpg.fr, gerard.bouvier@gipsa-lab.inpg.fr

Résumé – Cet article décrit une tâche d'asservissement visuel pour une caméra commandable en pan et tilt embarquée dans un drone. La tâche se résume à garder un objet sélectionné par l'utilisateur au centre de l'image quelques soient les mouvements du drone. Afin de réaliser cette tâche nous proposons un algorithme de suivi de primitives utilisant un système de prédiction basé sur l'analyse du mouvement global dans l'image. Cet algorithme permet une bonne robustesse aux très fortes perturbations dues à la transmission vidéo et fonctionne à une cadence proche de la cadence vidéo. La commande du système est basée sur une loi de commande à double boucle, qui assure une convergence rapide vers la position désirée. L'expérimentation en conditions réelles montre l'efficacité du système proposé.

Abstract – This paper deals with visual servoing for a pan and tilt camera embedded in a drone. Video is transmitted to the ground where images are processed on a PC, and turret controls are sent back to the drone. The objective is to track any fixed object on the ground without knowledge about shape or texture and to keep it centered in the image. In order to achieve this task an algorithm that combines feature-based and global motion estimation is proposed. This algorithm provides a good robustness to very strong video transmission noise and works at a frame rate close to 25 fps. The control of the system is based on a double closed loop, which achieves a fast convergence to the desired position. Experimentation in real conditions shows the effectiveness of the proposed scheme.

1 Introduction

Parmi les capteurs existants (ultrasons, infrarouges...) la caméra vidéo est celui qui délivre le plus d'informations exploitables pour le positionnement dans un environnement complexe. C'est donc tout naturellement que de nombreuses études se sont portées sur le couplage de la vision et de la commande d'actionneurs. Ce domaine de recherche permet de commander les mouvements d'un robot à partir d'informations visuelles. Le besoin d'extraire des informations fiables des images acquises implique l'utilisation d'un traitement d'image qui peut se révéler coûteux en temps de calcul. Cela vient en contradiction avec la nécessité de contrôler les mouvements du robot à une cadence élevée pour garantir une grande réactivité ainsi qu'une grande stabilité. C'est pourquoi la plupart des méthodes d'asservissement visuel ont un point commun : l'utilisation de marqueurs sur les objets. Ceci permet de simplifier au maximum les algorithmes d'extraction de l'information visuelle. Dans le cadre de notre système d'asservissement visuel, notre objectif a été de développer un algorithme autorisant le suivi d'objets, sans connaissance a priori sur leurs formes et sans segmentation préalable pour une caméra commandable en pan et tilt embarquée sur un drone. La tâche d'asservissement visuel se résume à garder un objet sélectionné par l'utilisateur au centre de l'image quelques soient les mouvements du drone. Les images sont acquises par la caméra embarquée dans le drone. Elles sont transmises au sol, puis traitées par l'algorithme d'estimation du mouvement. Les déplacements mesurés dans l'image permettent de calculer la

commande nécessaire au bon déplacement de la tourelle. Une liaison par modem radio assure la transmission des ordres à la tourelle.

L'article est structuré de la façon suivante. En premier lieu nous présenterons notre système d'asservissement visuel, ainsi que les algorithmes d'estimation du mouvement et de commande utilisés. Puis nous exposerons les résultats obtenus par la méthode proposée. Enfin, nous concluons.

2 Estimation du Mouvement

2.1 Présentation

De nombreuses études traitent de l'estimation du mouvement dans une séquence d'images, on peut les classer en trois catégories. La première catégorie est basée sur la définition explicite d'un modèle géométrique de l'objet à suivre [1]. Compte tenu des contraintes imposées par notre système, elles ne sont pas appropriées dans ce contexte. La deuxième grande classe de méthodes s'appuie sur l'extraction de primitives géométriques (points, lignes, contours,...)[2, 3] dans l'image et leur suivi. Le choix de bonnes primitives permet une estimation précise de la position de l'objet dans l'image. Leurs principaux avantages sont leur faible consommation en temps de calcul et leur précision. Par contre, elles sont très sensibles aux perturbations dans l'image. Dans le cas de notre tâche, des perturbations dues à la transmission vidéo endommagent fortement les images à traiter. La troisième classe de mé-

thodes est fondée sur l'analyse du mouvement dans une région ou la globalité de l'image [4, 5], à l'aide de l'intensité lumineuse. Plus robustes que les méthodes précédentes, elles offrent une meilleure représentation du mouvement par l'utilisation de modèles prenant en compte des déformations 2D complexes (affine, projective,...). Leurs principaux inconvénients sont la nécessité d'une grande quantité d'informations pour converger vers une bonne estimation et leur temps de calcul important. Dans le contexte de notre tâche d'asservissement visuel, compte tenu de la taille des objets à suivre, leurs mouvements diffèrent du mouvement global dans l'image. Cela a pour conséquence de rendre ce type d'algorithmes imprécis.

Pour rendre notre système plus robuste tout en gardant une bonne précision sur la position de la cible, nous proposons un algorithme basé sur les deux précédentes classes de méthodes. L'algorithme développé est représenté figure 2. Il repose sur l'algorithme KLT [2] pour l'extraction de primitives points et sur l'algorithme RMRm [5] pour l'analyse du mouvement dans l'image.

2.2 Algorithme KLT

Pour l'extraction de points dans l'image et le suivi de ces points nous avons utilisé l'algorithme KLT, il a été développé par Kanadé et Lucas, on le retrouve dans de nombreuses tâches de suivi d'objets. L'extraction de points dans la première image se fait à l'aide du détecteur de Harris [6]. Pour le suivi, les points caractéristiques sont assimilés au centre d'une fenêtre de calcul W , en considérant le mouvement constant sur une petite région de l'image. Il existe plusieurs méthodes pour choisir cette fenêtre W et le modèle de déplacement à estimer [7]. Dans notre cas, pour des raisons de temps de calcul nous avons choisi une fenêtre de petite taille (7x7 pixels) et un simple modèle $\theta(p, t)$ de translation comme suit :

$$\forall p \in W \quad \theta(p, t) = d(t)$$

Où d est le déplacement inter-frame. Soit une image I à l'instant t et la même image à l'instant $t + \tau$ l'estimation du vecteur d revient à un problème d'optimisation où il faut minimiser le résidu suivant pour une fenêtre W :

$$E = \sum_{p \in W} [I(p, t + \tau) + \nabla I(p, t + \tau) \cdot d(t) - I(p, t)]^2$$

Où $\nabla I(p, t)$ est le gradient spatial au point p . Ce système est résolu en utilisant la méthode itératif de Newton-Raphson.

2.3 Algorithme RMR

Pour estimer un modèle de mouvement global dans l'image, nous avons utilisé l'algorithme RMRm développé par J-M Obodez [5]. Il permet à partir d'une estimation robuste multi-échelle de définir un modèle de mouvement 2D à plusieurs paramètres entre deux images. A présent, nous résumons brièvement l'algorithme RMRm : soit deux images successives acquises par la caméra. L'estimation du modèle θ_t correspond à la minimisation du critère suivant :

$$C(\theta_t) = \sum_p \rho(\nabla I(p, t) \cdot \omega_{\theta_t}(p) + I_t(p, t))$$

Avec $\omega_{\theta_t}(p)$ la vélocité au point p , $\nabla I(p, t)$ et $I_t(p, t)$ respectivement le gradient spatial et le gradient temporel au point p .

2.4 Algorithme Proposé

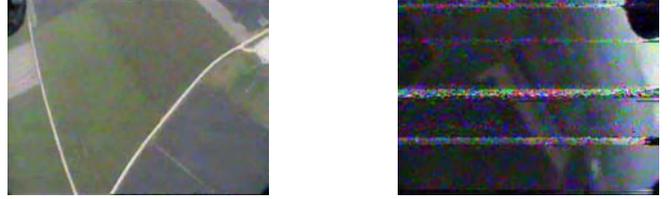


FIG. 1 – Qualité des images

Les principaux problèmes rencontrés pour estimer la position de la cible dans nos images, sont la taille des objets et la qualité très fluctuante des images, comme on peut le voir sur la figure 1. En considérant que la scène filmée par notre drone est plane nous proposons un algorithme robuste au problème de perturbation dans les images et d'une grande précision sur la localisation de la cible. La figure 2 resume notre algorithme. Il se décompose en trois phases.

Une première phase d'initialisation est nécessaire. Elle comporte l'extraction des points caractéristiques de la zone à suivre et l'initialisation du support pour l'estimation de mouvement global.

Une deuxième phase correspond au suivi de la cible. Elle comporte deux parties :

- Une première partie correspond à la prédiction de la position des points à suivre dans l'image courante. Pour cela une estimation du mouvement global dans l'image est effectuée. Dans [8] il est démontré qu'un modèle de type quadratique est le plus adapté pour décrire le mouvement d'une scène en 2D. Pour des raisons de temps de calcul, le modèle que nous avons retenu est le modèle affine. Pour les mêmes raisons de temps de calcul, l'analyse du mouvement global se fait sur un support plus petit que l'image entière. Ce support permet aussi de limiter les effets de bord. Cette étape correspond à l'analyse du mouvement global dans l'image.

- Une deuxième partie correspond à la recherche des points à l'aide de la prédiction précédente pour raffiner l'estimation de la position de la cible dans l'image. Pour cela, l'algorithme Newton-Raphson est initialisé avec l'estimation globale fournie par RMR. Cette étape correspond à l'analyse du mouvement local dans l'image. Les deux étapes de suivi permettent un gain en robustesse tout en gardant une grande précision. Les estimations globale (RMR) et locale (KLT) sont calculées à l'aide d'une analyse multi-résolution.

Une dernière phase consiste à tester la pertinence des points retrouvés et à extraire de nouveaux points le cas échéant. Une vérification est faite sur le déplacement de chacun des points par rapport au déplacement moyen. Si le déplacement d'un point dépasse un seuil, il est éliminé. L'extraction de nouveaux points se fait dans le cas de la perte complète des points précédents. La prédiction de

mouvement donnée par l'estimation globale est utilisée afin de positionner la zone de l'image où extraire de nouveaux points.

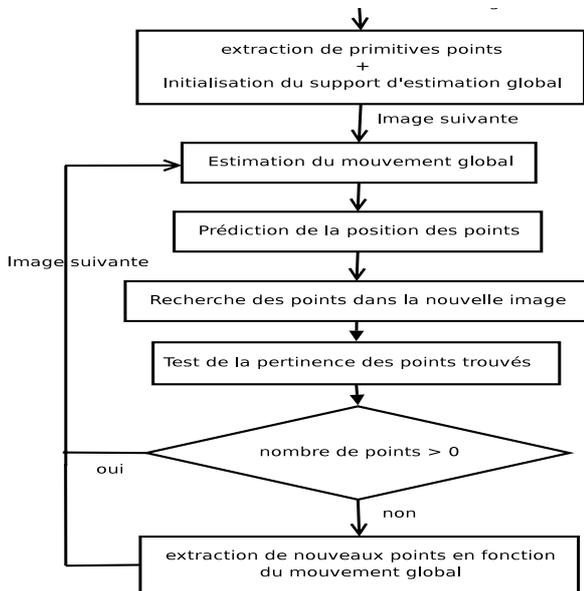


FIG. 2 – Algorithme d'estimation du mouvement

3 Loi de Commande

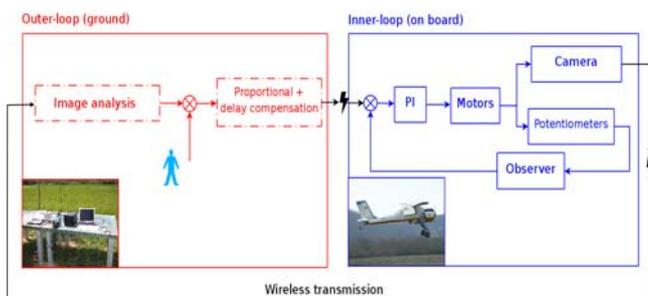


FIG. 3 – Boucle de commande

Dans la plupart des tâches d'asservissement visuel, la partie contrôle se résume à un gain proportionnel [9]. En effet, beaucoup de travaux s'intéressent à la partie cinématique et la modélisation entre les différents actionneurs du robot et de la caméra. Par contre très peu d'études prennent en compte la dynamique de ces actionneurs. Dans notre tâche d'asservissement visuel, le mouvement des objets peut être très rapide, à cause des mouvements du drone. Ce qui nous oblige à prendre en compte la dynamique de notre système. Pour compléter notre tâche de suivi d'objet, une loi de commande basée sur une double boucles est proposée afin de centrer l'objet dans l'image. La figure 3 détaille cette loi de commande. elle est composée de deux boucles.

Une boucle interne embarquée dans le drone permet la régulation des vitesses en pan et tilt imposées par la boucle externe. Les vitesses angulaires des moteurs ne peuvent pas être mesurées directement. La seule mesure disponible

est la position angulaire. Afin de réguler nos moteurs en vitesse nous avons construit un observateur permettant d'estimer les vitesses angulaires.

Une boucle externe permet la régulation de la position de l'objet au centre de l'image. Les retards induits par les transmissions électroniques et l'algorithme d'estimation du mouvement sont compensés dans le calcul de la vitesse envoyée à la boucle interne.

4 Résultats

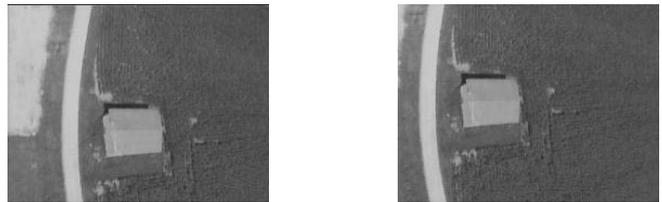


FIG. 4 – Déplacement de 30 pixels d'une maison

Tous les tests de validation de l'algorithme ont été réalisés sur des images issues de séquences filmées par le drone. Une première validation consistant à tester les performances de l'algorithme proposé par rapport aux algorithmes KLT et RMRm utilisés seuls est représentée sur les figures 5 et 7. Pour des raisons de reproductibilité des tests (le drone ne survolant jamais exactement la même zone) et l'ignorance du déplacement réel des objets dans l'image, la comparaison des performances des différents algorithmes a été réalisée sur un mouvement synthétique de translation (le déplacement d'une maison représenté figure 4) ayant une grande amplitude. Ceci afin de simuler les diverses amplitudes de déplacements auxquelles nous sommes confrontés durant les vols. Lors des tests, nous avons extrait 4 points sur la maison dans la première image puis nous avons lancé une recherche de ces points dans la deuxième image en faisant varier le déplacement de la maison. La figure 5 montre l'erreur de l'estimation de mouvement pour le déplacement d'une maison variant de 1 à 65 pixels. Le gain apporté par l'algorithme proposé par rapport à un algorithme de suivi de primitives simple comme KLT est important. En effet, notre algorithme permet une bonne estimation du mouvement même lors de grands déplacements alors que KLT échoue dès que le déplacement dépasse 20/40 pixels. La figure 6 montre le nombre de points retrouvés dans la seconde image en fonction de l'amplitude du déplacement de la maison. L'algorithme KLT perd des points à partir d'un déplacement de 20 pixels et perd la totalité des points pour un déplacement de 40 pixels et plus. Ce résultat correspond à la figure précédente où KLT donnait une mauvaise estimation à partir de 20 à 40 pixels. Par contre, avec l'algorithme proposé les points sont retrouvés quelque soit l'amplitude de déplacement de la maison. La figure 7 montre le gain en précision apporté par notre algorithme sur RMRm pour un déplacement de 30 pixels de la maison. Les résultats montrent l'utilité de KLT. Il corrige l'erreur d'estimation fournie par RMRm. La combinaison proposée par notre

algorithme permet donc de corriger les erreurs respectives de KLT et RMRm. Cette correction permet aussi de régler de façon optimale le nombre d'itérations de RMRm et de KLT afin d'obtenir une estimation robuste et précise tout en limitant le temps de calcul. Une deuxième validation a été de tester le comportement de l'algorithme dans la boucle d'asservissement visuel. Une vidéo du résultat est disponible [10]. La cadence de traitement est de l'ordre de 20 images par seconde sur un P4 3.2GHz avec RMRm réglé à 4 itérations maximum et KLT à 10 itérations maximum. La robustesse de l'algorithme est démontrée par la qualité du suivi. Quand l'image est endommagée la prédiction permet de retrouver les points et donc de poursuivre la tâche de suivi.

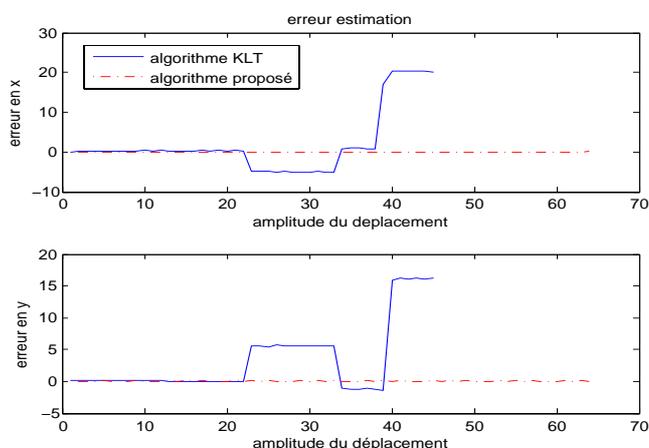


FIG. 5 – Estimation en fonction de l'amplitude de déplacement

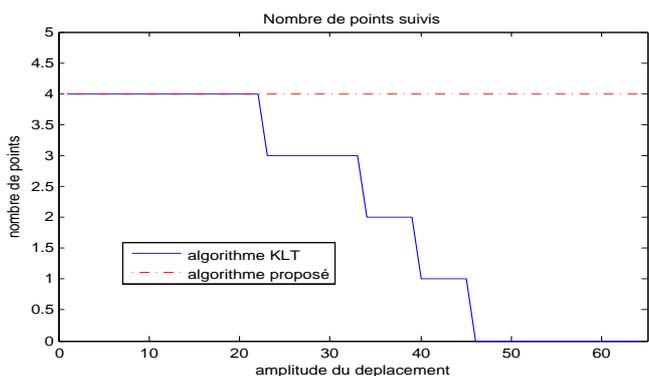


FIG. 6 – Nombre de points en fonction de l'amplitude de déplacement

5 Conclusions

Nous avons proposé un algorithme d'estimation de mouvement pour le suivi d'objet. Cet algorithme respecte les contraintes inhérentes à notre système. Il est précis tout en étant assez robuste pour permettre une bonne estimation lors de perturbations dans la transmission vidéo. Il autorise aussi l'estimation de grands déplacements. Enfin, son temps de calcul est approprié pour une tâche d'asservissement visuel. La robustesse aux problèmes de transmission

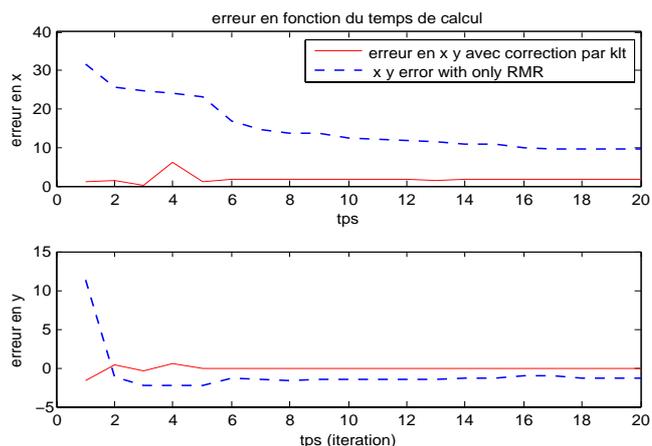


FIG. 7 – Estimation en fonction du nombre d'itérations de RMRm

vidéo n'est pas complètement résolue. En effet, lorsque les images sont trop fortement perturbées, le suivi échoue. Pour augmenter la robustesse de notre estimation nous pourrions détecter les pixels endommagés et les éliminer du support d'estimation. D'autres améliorations sont envisagées, comme le suivi de cible mobile ou la reconnaissance d'objet permettant de retrouver les cibles perdues lors du suivi.

Références

- [1] E. Marchand, P. Bouthemy, and F. Chaumette, "A 2D-3D model-based approach to real-time visual tracking," *Image and Vision Computing*, vol. 19, no. 13, 2001.
- [2] T. Kanade and C. Tomasi, "Detection and tracking of point features," Tech. Rep. CMU-CS-91-132, Carnegie Mellon University, Apr. 1991.
- [3] M. Berger, "How to track efficiently piecewise curved contours with a view to reconstructing 3d objects," in *Proc. of the Int. Conf. on Pattern Recognition, ICPR'94*, (Jerusalem, Israel), pp. 32–36, Oct. 1994.
- [4] G. Hager and K. Toyama, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1025–1039, Oct. 1998.
- [5] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric models applied to complex scenes," tech. rep., IRISA, Jan. 1994.
- [6] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proceedings of The Fourth Alvey Vision Conference*, pp. 147–151, 1988.
- [7] M. Okutomi and T. Kanade, "A locally adaptive window for signal matching," *Int. Journal of Computer Vision*, vol. 7, pp. 143–162, Jan. 1992.
- [8] M. Subbarao and A.-M. Waxman, "Closed-form solutions to image flow equations for planar surface in motion," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 2-3, pp. 208–228, 1986.
- [9] A. Cretul and F. Chaumette, "Visual servoing based on image motion," *Int. Journal of Robotics Research.*, vol. 20, no. 11, pp. 857–877, 2001.
- [10] "Demonstration video." Internet address : <http://film.drone.free.fr/>.