



HAL
open science

Estimation temps réel de l'altitude d'un drone à partir d'un capteur de stéréovision mixte

Damien Eynard, Pascal Vasseur, Cedric Demonceaux, Vincent Fremont

► **To cite this version:**

Damien Eynard, Pascal Vasseur, Cedric Demonceaux, Vincent Fremont. Estimation temps réel de l'altitude d'un drone à partir d'un capteur de stéréovision mixte. 17ème congrès francophone AFRIF-AFIA RFIA, Jan 2010, France. hal-00447613

HAL Id: hal-00447613

<https://hal.science/hal-00447613v1>

Submitted on 15 Jan 2010

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.

Estimation temps réel de l'altitude d'un drone à partir d'un capteur de stéréovision mixte.

Damien Eynard¹

Pascal Vasseur¹

Cédric Demonceaux¹

Vincent Frémont²

¹ Laboratoire MIS - Université de Picardie Jules Verne
33, rue Saint Leu - 80000 Amiens

² Laboratoire Heudiasyc - Université de Technologie de Compiègne
Centre de Recherches de Royallieu - BP 20529 - 60205 Compiègne

¹ {prénom.nom}@u-picardie.fr

² vincent.fremont@hds.utc.fr

Résumé

L'altitude est l'un des paramètres les plus importants à connaître pour un drone surtout pendant les phases de décollage et d'atterrissage. Dans ce papier nous présentons un nouveau capteur de stéréovision mixte composé d'une caméra fisheye et d'une caméra perspective pour l'estimation de l'altitude. Contrairement aux systèmes de stéréo classiques, basés sur l'appariement de points, nous proposons une approche de planesweeping pour l'estimation de l'altitude et la segmentation du sol. Par rapport au plan du sol, il existe une homographie entre les deux vues, le capteur étant calibré et l'altitude estimée par la caméra fisheye, l'algorithme consiste en la recherche de l'altitude qui vérifie cette homographie. Nous montrons que cette méthode est robuste et précise et l'implémentation GPU permet une estimation temps réel. Des résultats expérimentaux sur des séquences réelles montrent la validité de l'approche.

Mots Clef

Drone, plane-sweeping, capteur mixte, vision, fisheye.

Abstract

Altitude is one of the most important parameters to be known for an Unmanned Aerial Vehicle (UAV) especially during critical maneuvers such as landing or steady flight. In this paper, we present a new mixed stereoscopic vision system made of a fish-eye camera and a perspective camera for altitude estimation. Contrary to classical stereoscopic systems based on feature matching, we propose a plane sweeping approach in order to estimate the altitude and consequently to detect the ground plane. Since there exists an homography between the two views and the sensor being calibrated and the attitude estimated by the fish-eye camera, the algorithm consists then in searching the altitude

which verifies this homography. We show that this approach is robust and accurate and a GPU implementation permits a real time estimation. Experimental results on real sequences demonstrate the validity of the approach.

Keywords

UAV, plane-sweeping, capteur mixte, vision, fisheye.

1 Introduction

Les drones, appareils volants sans pilote, font l'objet d'une attention particulière cette dernière décennie du point de vue de l'autonomie. Cette notion inclue la capacité à réaliser des manoeuvres complexes telles que le décollage et l'atterrissage. Il est donc indispensable d'estimer précisément et en temps réel l'altitude, l'attitude et les vitesses de déplacement. Dans ce papier, nous proposons un nouveau système de stéréovision mixte, fisheye/perspectif, capable d'estimer de façon autonome à la fois l'altitude d'un drone, son attitude et le plan d'atterrissage.

L'altitude peut être estimée par différents moyens comme le "Global Positioning System" (GPS), l'altimètre (laser ou à pression), les radars ou encore la vision par ordinateur. Les GPS possèdent une précision verticale d'environ 25m à 50m. Ils sont sensibles aux interruptions de transmission dans les environnements urbains. Un autre type de capteur d'altitude sont les altimètres barométriques. Ces systèmes possèdent comme principal inconvénient d'être dépendants à la variation de pression extérieure. L'erreur peut ainsi varier de 6% à 7%. L'altimètre laser quant à lui est très précis mais nécessite certaines conditions de réflexion de la surface à mesurer. Enfin, les radars permettent d'estimer à la fois l'altitude et le plan du sol. Ce sont des capteurs actifs, principal inconvénient qui les rend détectables et gourmands en énergie.

L'utilisation de la vision par ordinateur a fortement augmenté cette dernière décennie pour l'estimation des paramètres des drones. Certains systèmes ont ainsi été proposés pour l'estimation de l'altitude. L'estimation d'altitude par vision présente de nombreux avantages. Tout d'abord la vision peut être utilisée pour d'autres tâches telles que l'évitement d'obstacles, la navigation ou encore la localisation. Ensuite les caméras sont des systèmes passifs, peu consommateurs d'énergie et peuvent fournir un nombre important d'images par seconde. La difficulté principale des systèmes de vision réside en la sélection d'une référence pour l'estimation de l'altitude. Dans [14], [28] et [29], les auteurs proposent de balayer le sol avec une caméra perspective afin d'estimer l'altitude à partir d'une mire fixée sur le sol. Ce type d'approche est intéressante puisqu'elle nécessite l'utilisation d'une seule caméra, fournit une pose complète et peut être utilisée en temps réel. Elle est cependant limitée à un environnement spécifique, muni de marques artificielles au sol. D'autres systèmes basés sur le flot optique se basent également sur une caméra monoculaire perspective comme [1], [4], [6] et [16]. Ces systèmes, inspirés des abeilles, permettent de déduire l'altitude, à partir du flot optique, en connaissant la vitesse de la caméra. [4] propose d'estimer le tangage pour corriger le flot optique. Cette méthode, peut conduire à un système instable. Un autre travail a été proposé, utilisant une seule caméra perspective. Dans [7], les auteurs utilisent une technique d'apprentissage. Une base de données est constituée à partir d'une méthode d'analyse de différents types de sols par champs de markov spatio-temporel. L'altitude est estimée à partir de l'analyse de l'image et du temps. L'apprentissage repose ensuite sur l'association de l'information texture du sol à une valeur d'altitude. Dans [27], un algorithme multi vues est proposé dans le but de calculer une carte numérique du sol. Celui-ci est basé sur une séquence d'images issues d'une seule caméra. [20] et [26] proposent d'utiliser des capteurs stéréoscopiques plutôt qu'une caméra unique. Ce procédé permet d'améliorer l'appariement entre les différentes vues. Dans ce travail, les auteurs développent une approche basée sur la mise en correspondance de points d'intérêt afin d'extraire la carte d'élévation du sol.

Dans ce papier, nous proposons également un capteur de stéréovision légèrement différent des systèmes mentionnés précédemment. Il est en effet constitué de deux caméras différentes : une fisheye et une perspective. L'avantage d'un champ de vue large pour les drones a déjà été démontré dans différents travaux tels que [19] pour la navigation ou [12] pour l'estimation de l'attitude. Ainsi, l'utilisation d'un capteur mixte permet d'obtenir un champ de vue large avec la caméra fisheye tandis que la perspective fournit une meilleure précision dans l'image.

Nous supposons le capteur calibré, le sol planaire et qu'il représente au moins la moitié de l'image perspective. Dans ce cas, il existe une homographie du plan du sol entre les deux images. De plus, nous pouvons estimer l'attitude du drone grâce à l'image omnidirectionnelle comme dans [12] et [11], et en déduire la normale au sol. Il reste donc à trouver l'altitude qui vérifie cette homographie (fig. 1, fig.4). Nous proposons alors un algorithme de planesweeping pour résoudre ce problème.

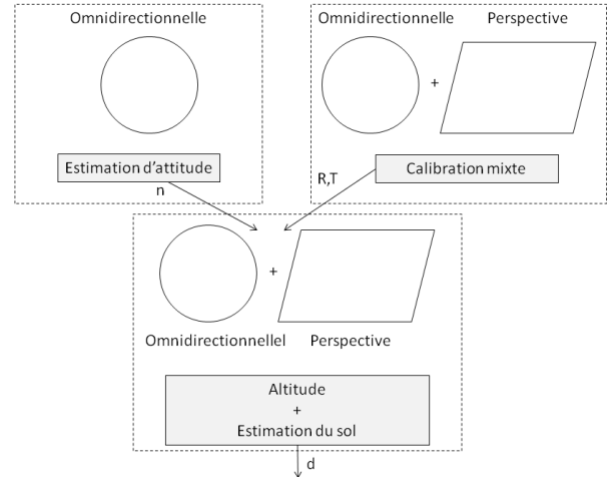


FIGURE 1 – Système mixte

Notre approche amène plusieurs contributions. Tout d'abord, le système est capable d'estimer de façon autonome l'altitude, l'attitude et de segmenter le sol sans l'aide d'aucun autre capteur. Ensuite, nous proposons une méthode sans mise en correspondance permettant de traiter des images de géométries différentes (planes et sphériques), méthode d'autant plus robuste que les approches classiques stéréoscopiques sont basées sur la mise en correspondance. Enfin, l'implémentation GPU permet une estimation temps réel de l'altitude et du sol.

La suite du papier s'articule comme suit. Une première partie traite du système global et de sa modélisation. Ensuite, dans une seconde partie, l'algorithme d'estimation d'altitude et la segmentation du sol seront proposés. Enfin, nous présenterons dans une troisième partie les résultats expérimentaux sur des séquences réelles avec une évaluation quantitative de l'erreur et du temps de calcul.

2 Capteur de stéréovision mixte

2.1 Présentation générale du capteur

Nous proposons un système de stéréovision mixte omnidirectionnel/perspectif (fig. 2) capable d'estimer l'altitude et le plan du sol en temps réel ainsi que l'attitude. L'avantage d'un capteur omnidirectionnel est le large champ de vue alors que les inconvénients sont

la faible résolution et les fortes distorsions. Les avantages du fisheye comparé aux caméra catadioptriques sont les réductions de vibrations dues à l'absence de tige et la suppression du trou noir au centre de l'image. D'un autre côté, les caméra perspectives possèdent une bonne résolution mais un champ de vue limité. Le problème principal de notre système réside dans la mise en correspondance des deux vues fisheye et perspective. Plusieurs approches sont alors possibles :

- Tout d'abord, connaissant les paramètres intrinsèques de la caméra fisheye, il est possible de transformer l'image pour obtenir l'équivalent perspectif. On peut ensuite appliquer une méthode de mise en correspondance. Cependant, cette approche nécessite différents traitements tels que le matching et l'interpolation. L'implémentation temps réel est alors impossible.
- D'autres travaux récents proposent d'utiliser la sphère unitaire pour représenter les images omnidirectionnelles. On peut ensuite appliquer les algorithmes de mise en correspondance. Cependant, comme précédemment, cette solution ne peut être implémentée en temps réel.
- Enfin, nous proposons une approche de comparaison directe des images sans aucune mise en correspondance.

Comme l'altitude est calculée en fonction du sol, nous pouvons utiliser ce plan comme référence. Ainsi, nous allons démontrer qu'il existe une homographie entre l'image omnidirectionnelle et perspective. L'équation générale de l'homographie est $H = R - T \frac{n'}{d}$, où R et T définissent la transformation rigide entre les deux vues, n' est la normale au sol dans la première image et d est la distance entre le sol et la première caméra. Dans notre cas, d correspond à l'altitude. En conséquence, si nous pouvons trouver l'homographie entre les deux images, nous pouvons déduire d puisque R et T peuvent être estimés par calibration et n' peut être estimé par estimation de l'attitude. Ces méthodes sont basées sur la vision omnidirectionnelle [3], [10] et [11].

2.2 Modèles de caméra et calibration

Bien que les objectifs fisheye ne sont pas à point de vue unique [2], nous utilisons la sphère unitaire pour modéliser notre caméra [31]. Mei et Rives [25] proposent une méthode de calibration basée sur le modèle sphérique. Ce modèle est précis et permet de modéliser les distorsions radiales et tangentielles d'un objectif fisheye. Nous présentons dans la figure 3 le modèle sphérique du fisheye. La projection est divisée en deux étapes. Tout d'abord, un point du monde x_m est projeté sur la sphère unitaire x_s en passant par son centre S . Ensuite, ce point x_s est projeté sur le plan image en un point x_i en passant par le centre O . Le paramètre ξ définit la distance entre O et S . Ce paramètre est estimé pendant la phase de calibration. La calibration stéréo



FIGURE 2 – Caméras montées sur un banc stéréo

mixte est obtenue à partir des toolbox de Bouguet [5] et Mei [25].

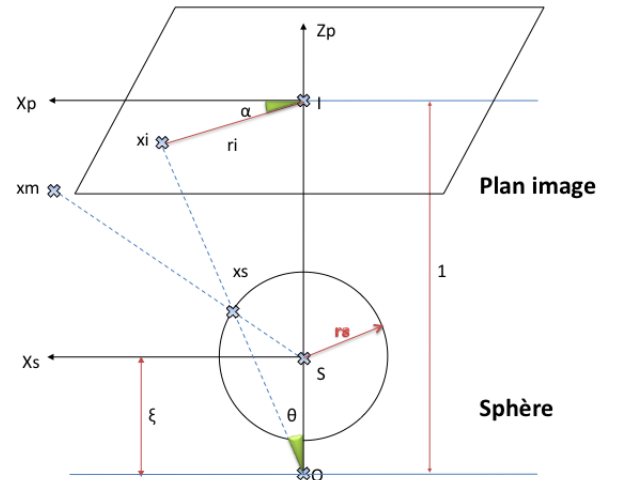


FIGURE 3 – Fisheye modélisé par une sphère.

2.3 Homographie entre une sphère et un plan

Dans cette partie, nous proposons de définir l'homographie qui relie un plan et une sphère sur lesquels un plan 3D a été projeté (fig. 4). Dans [17], l'homographie lie la projection d'un plan 3D sur deux plans. Dans le travail [24], l'homographie lie la projection d'un plan 3D sur deux sphères.

$$H = R - T \frac{n'}{d} \quad (1)$$

Considérons :

- X_p , un point du plan 3D projeté sur la vue perspective.
- X_s , un point du plan 3D projeté sur la sphère.
- X_s^* est la projection de X_s sur la perspective par homographie.

Pour un système à deux caméras perspectives, X_p^* est la projection de X_p sur la perspective par homographie.

Nous avons l'équation (2).

$$X_p^* \sim HX_p \quad (2)$$

Nous remplaçons ces deux plans de projection par une projection planeaire et une projection sphérique. De même, l'équation (2) est à un facteur d'échelle prêt :

$$\frac{X_p^*}{\|X_p^*\|} \sim \rho HX_p \quad (3)$$

Donc :

$$X_s^* \sim \rho HX_p \quad (4)$$

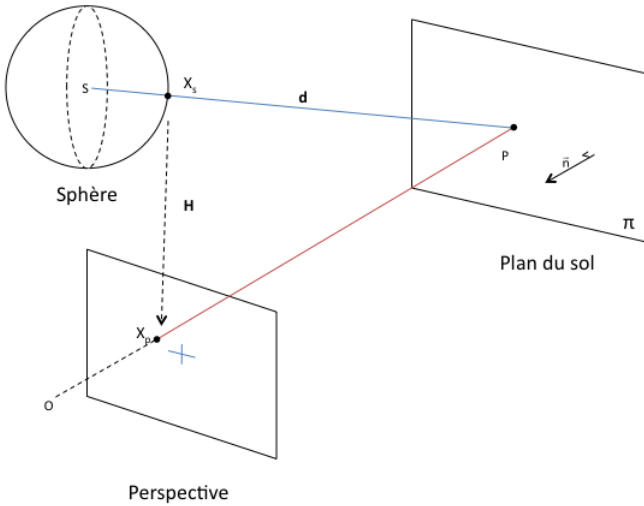


FIGURE 4 – Homographie sphere/plan.

L'homographie H dépend de R , T , l'attitude \vec{n} et l'altitude d . Tout d'abord, la transformation rigide entre les caméras définie par la rotation R et la translation T est obtenue par calibration. Ensuite, la normale \vec{n} au plan du sol est obtenue par [3], [10], [11] ou par centrale inertielle. Enfin, nous estimerons l'altitude d par planesweeping.

3 Estimation de l'altitude et segmentation du sol

Nous allons estimer l'altitude d par planesweeping. Dans une première partie, nous présenterons une méthode de planesweeping adaptée au cas mixte perspectif/omnidirectionnel pour estimer à la fois l'altitude et

segmenter le plan du sol. Dans une seconde partie, nous présenterons l'algorithme développé.

3.1 Planesweeping

La notion de planesweeping a été introduite par Collins [8]. Tout d'abord, une vue de référence est définie. Pour chaque normale et pour chaque distance à un plan 3D, nous comparons chaque image transformée (eq. 5) par homographie à l'image de référence.

$$I_s^*(p, d) = I_s(H_p) = I_s((R - T \frac{\vec{n}'}{d})p) \quad (5)$$

La meilleure estimation de l'homographie H est obtenue lorsque la moyenne globale de la différence entre l'image de référence et l'image transformée est minimum. Nous choisissons dans notre application d'utiliser la vue perspective comme référence. En effet, il est plus facile de manipuler un voisinage sur une modélisation planeaire que sphérique pour la segmentation du sol. Ainsi l'image obtenue par la caméra fisheye est projetée dans l'espace de la caméra perspective par l'homographie H . Ces deux images peuvent ensuite être comparées par soustraction (fig. 5).

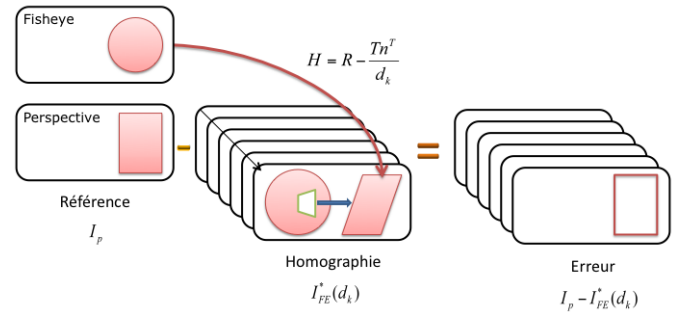


FIGURE 5 – Plane-sweeping entre deux vues mixtes.

3.2 Algorithme

Dans ce papier, le planesweeping est utilisé pour une recherche exhaustive de l'altitude. A chaque itération, la meilleure altitude \hat{d} est estimée à partir de l'ensemble $d \in [d_{min}, d_{max}]$. Le masque G est ensuite mis à jour (algorithm 1). Dans notre algorithme, le masque G correspond au plan du sol segmenté. Les pixels appartenant au sol sont représentés en blanc dans les figures 6(c) and 7(c).

Afin d'obtenir une méthode temps réel, nous proposons d'estimer l'altitude au temps t en utilisant les données issues de $t - 1$. Cette estimation se décompose en deux phases :

- Initialisation : nous estimons la meilleure altitude à partir d'un ensemble d'altitudes assez large (algorithm 1).
- Pendant le vol : l'altitude estimée à l'étape d'initialisation est réutilisée afin d'obtenir un espace de

Algorithm 1 Algorithmes d'estimation d'altitude et segmentation du sol - initialisation

Estimation($d_{min}, d_{max}, s, \Delta d$)

```

{Initialisation}
 $a_0 = \widehat{d}_{-1} = d_{min}$ 
 $b_0 = \widehat{d}_0 = d_{max}$ 
 $G = \{p \in Pixels\}$ 
while  $|\widehat{d}_k - \widehat{d}_{k-1}| > \Delta d$  do
  {Estimation de la meilleure altitude}
   $\widehat{d}_{k+1} = \operatorname{argmin}_{d \in \{\frac{b_k - a_k}{s-1} t + a_k; t \in [0, s-1]\}} (\sum_{p \in G} |I_P(p) - I_S^*(p, d_k)|)$ 
  {Estimation du masque d'inliers/outliers}
   $G = \{p \in Pixels, \frac{|\sum_{p_1 \in W_p} |I_P(p_1) - I_S^*(p_1, \widehat{d})|}{\sum_{p_1 \in W_p} I_P(p_1)} < thres\}$ 
  {Estimation du nouvel espace de recherche en fonction de l'échantillonnage}
   $a_{k+1} = \widehat{d}_{k+1} - \frac{b_k - a_k}{s-1}$ 
   $b_{k+1} = \widehat{d}_{k+1} + \frac{b_k - a_k}{s-1}$ 
   $k = k + 1$ 
end while
Return  $\widehat{d}$ 

```

recherche plus petit $d_t \in [d_{t-1} - r_d, d_{t-1} + r_d]$ (algorithme 2) avec r_d dans l'équation (6). v_v représente la vitesse ascensionnelle maximum du drone et fps le nombre d'estimations par seconde fonction du matériel utilisé. Cet ensemble dépend de la vitesse verticale du drone (environ $\pm 5m/s$) et de la puissance de calcul.

Algorithm 2 Algorithmes d'estimation d'altitude et segmentation du sol - aspect temporel

$\widehat{d} = \mathbf{Estimation}(d_{t-1} - r_d, d_{t-1} + r_d, step, \Delta d)$

$$r_d = \frac{v_v}{fps} 10^3 \quad (6)$$

Nous allons présenter dans la section qui suit les résultats de notre algorithme capable d'estimer l'altitude et de segmenter le sol en temps réel à partir d'un système stéréo mixte.

4 Résultats expérimentaux

4.1 Caméra fisheye

Les résultats qui suivent sont obtenus à partir d'une caméra Sony XCD-V50CR C-mount avec un capteur 1/3" et un objectif fisheye Fuji de focale 1.8mm. L'angle de vue d'une image fisheye dépend non seulement de la lentille mais aussi du couple lentille/caméra. Sur une caméra 1/3", l'angle de vue observé dans l'image est de $154^{\circ}08'$ horizontalement

et $115^{\circ}27'$ verticalement. Une fois l'ensemble caméra/objectif correctement adapté à notre application, l'étape suivante consiste à l'analyse du temps d'exécution de l'algorithme.

4.2 Estimation d'altitude et segmentation du sol

Nous présentons dans cette partie les résultats obtenus avec notre algorithme. Les acquisitions ont été réalisées en milieu extérieur à partir d'un mât pneumatique. L'altitude réelle est estimée par un télémètre laser et l'altitude estimée par notre algorithme est calculée a posteriori. Les résultats obtenus par ces tests montrent une très bonne précision de notre algorithme sur sol dégagé (tab. 1). Plus notre système prend de l'altitude moins l'estimation est précise. Ce phénomène s'explique par la diminution de la résolution en fonction de l'altitude.

Dans une scène avec obstacles, on s'aperçoit que la zone d'inliers du masque G diminue lorsque les obstacles apparaissent sur le sol dans l'image. L'erreur augmente donc mais reste tout de même tolérable (tab. 1).

TABLE 1 – Estimation de l'altitude et du sol avec et sans obstacle - Paramètres de l'algorithme : pas $s = 6$, seuil $thres = 25$

Sol	Alt. réelle	Alt. estimée	Erreur
\emptyset	2187mm	2200mm	0.59%
\emptyset	3244mm	3250mm	0.18%
obstacles (low contrast)	3244mm	3488mm	7.52%
\emptyset	4072mm	4200mm	3.14%
\emptyset	5076mm	5202mm	2.48%
obstacles	4080mm	4440mm	8.82%

La segmentation du plan donne de bons résultats. Dans le cas d'un sol sans obstacles, le mât où les caméras sont fixées est représenté par des outliers (fig. 6(c)). Pour une image composée d'un plan principal et de murs (fig. 7(c)), la zone détectée d'inliers est de 31%. Sur l'image réelle, la proportions d'inliers réel est de 54%. Afin d'obtenir une bonne estimation, il est nécessaire d'avoir au moins 50% d'inliers dans l'image. Ces résultats intéressants possèdent toutefois quelques limites. En effet les inconvénients des algorithmes de mise en correspondance restent présents dans notre méthode. Lorsque le sol ou les outliers d'une scène sont faiblement texturés ou homogènes, la segmentation outlier/inlier est plus difficile. Un autre aspect à analyser est la variation de luminosité entre la caméra perspective et fisheye. Ce problème est fréquemment rencontré en stéréovision. De plus dans le cas mixte, le fisheye ne possède pas le même champ de vue que la

perspective. La luminosité moyenne résultant des deux images est donc différente.

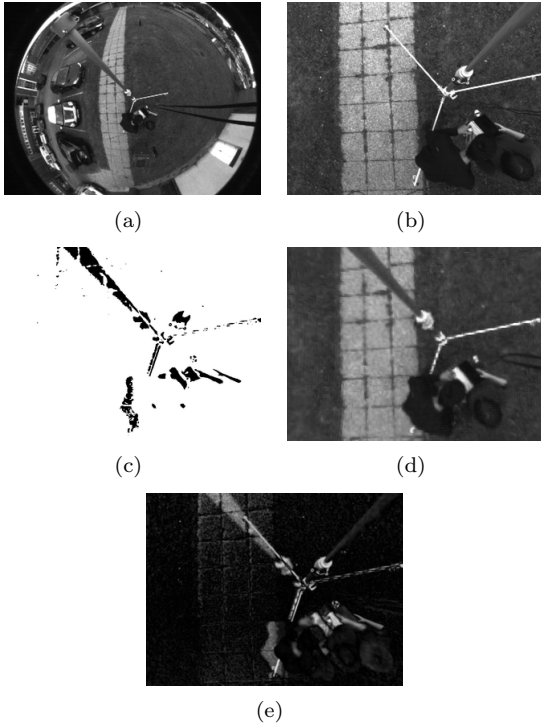


FIGURE 6 – Segmentation du plan et estimation d'altitude - 4.8% d'inliers - Vue fisheye (a), vue perspective (b), segmentation du sol (c), homographie sphère vers plan (d), différence entre la référence et l'homographie (e)

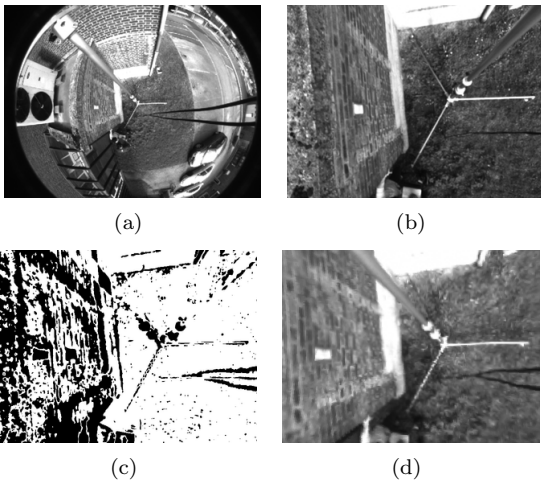


FIGURE 7 – Segmentation du plan et estimation d'altitude avec obstacles - 31% d'inliers - Vue fisheye (a), vue perspective (b), segmentation du sol (c), homographie sphère vers plan (d)

4.3 CPU et GPU

Avec l'arrivée du calcul GPU (GPGPU), deux types d'implémentations d'algorithmes sont possibles. Les algorithmes parallèles sur GPU et ceux séquentiels sur des CPU traditionnels. Récemment, en terme de GPGPU, Nvidia a développé CUDA et ATI CAL/Brook+. Ces langages simples et basés sur le C présentent l'avantage d'être rapidement implémentables. Un algorithme de planesweeping a été implémenté en GPU dans [15]. Nous avons choisi de développer notre algorithme sur ATI avec Brook+. Les tests et résultats ont été obtenus avec un PC dont la configuration est un CPU Intel E8400, 4Go de RAM et un GPU ATI 4850 512Mo. Dans notre algorithme, le calcul d'homographies, et des masques d'inliers/outliers ont été développés en GPU. Cette partie nécessite en effet un nombre important de calculs en parallèle d'où notre choix de l'architecture GPU. Avec un CPU traditionnel, mono-core, chaque pixel d'une image est calculé un par un. Avec le GPU, tous les pixels d'une image sont traités en parallèle (avec une limite de taille d'image maximum qui dépend du matériel utilisé). Pour comparer le CPU au GPU, nous avons testé le calcul de 100 homographies. D'après nos résultats, le GPU est 45x plus rapide qu'un CPU pour cette tâche. Nos temps de calcul ne comprennent pas les temps de transfert entre la mémoire centrale et la mémoire graphique.

En développant ces algorithmes par GPU, nous obtenons des résultats intéressants. En effet, l'estimation d'altitude et du sol sont réalisés simultanément et en temps réel (tab. 2). Notre algorithme est ainsi implémentable sur un drone. Enfin, concernant uniquement l'estimation de l'altitude, le temps de calcul est légèrement meilleur comme le montre (tab. 3).

TABLE 2 – Estimation de l'altitude avec segmentation du sol - images par seconde $\sim 21Hz$

Sol	Altitude réelle	Altitude estimée	erreur
Obstacle	4080mm	3850mm	5.64%
\emptyset	2187mm	2220mm	1.51%
\emptyset	3244mm	3280mm	1.11%

TABLE 3 – Altitude estimation only - framrate $\sim 32Hz$

Sol	Altitude réelle	Altitude estimée	erreur
Obstacle	4080mm	3809mm	6.64%
\emptyset	2187mm	2217mm	1.37%
\emptyset	3244mm	3215mm	0.89%

5 Conclusions et perspectives

Nous avons présenté dans ce papier un système stéréoscopique mixte. Les deux vues sont reliées par une homographie qui permet d'estimer à la fois l'altitude et de segmenter le sol en utilisant du planesweeping. En comparaison aux algorithmes de mise en correspondance, la planesweeping est une approche différente, basée sur la comparaison globale d'images. Cet algorithme teste un ensemble d'altitudes. L'altitude la plus probable est celle correspondant à l'erreur globale est minimum. Nous avons implémenté cet algorithme en GPU, aspect principal pour le traitement temps réel. Des résultats préliminaires ont été obtenus. Nous observons que l'altitude et la segmentation du sol sont correctement estimés. Pour illustrer ces résultats, une vidéo est disponible sur <http://damien.eyn.free.fr/RFIA10/Planesweeping.avi>. Compte tenu des résultats, nous orienterons nos perspectives vers une implémentation de l'algorithme sur un drone. D'autres travaux futur viseront à améliorer l'efficacité de notre algorithme sur les surfaces faiblement texturées.

6 Remerciements

Ce travail est soutenu par le projet ALTO de la région Picardie.

Références

- [1] G. Barrows, C. Neely, K. Miller, "Optic flow sensors for MAV navigation," In Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications, ser. Progress in Astronautics and Aeronautics, T. J. Mueller, Ed. AIAA, 2001, vol. 195, pp. 557-574.
- [2] S. Baker, S. K. Nayar, "A Theory of Single-Viewpoint Catadioptric Image Formation," International Journal of Computer Vision, 1999.
- [3] J-C Bazin, I. Kweon, C. Demonceaux, P. Vasseur, "UAV Attitude Estimation by Vanishing Points in Catadioptric Images", In Proceedings of IEEE International Conference on Robotics and Automation, 2008.
- [4] A. Beyeler, C. Mattiussi, J.C. Zufferey, D. Floreano, "Vision-based Altitude and Pitch Estimation for Ultra-light Indoor Microflyers," In Proceedings of IEEE International Conference on Robotics and Automation, 2006.
- [5] J.-Y. Bouguet, www.vision.caltech.edu/bouguetj/
- [6] J. Chahl, M. Srinivasan, H. Zhang, "Landing strategies in honeybees and applications to uninhabited airborne vehicles," In The International Journal of Robotics Research, vol. 23, no. 2, pp. 101-110, 2004.
- [7] A. Cherian, J. Andersh, V. Morellas, N. Papanikolopoulos, B. Mettler, "Autonomous Altitude Estimation Of A UAV Using A Single Onboard Camera," In Proceedings of IEEE International Conference on Intelligent and Robotic Systems, 2009.
- [8] R. T. Collins, "A space-sweep approach to true multi-image matching," In Proceedings of IEEE Computer Vision and Pattern Recognition, 1996.
- [9] J. Courbon, Y. Mezouar, L. Eck, P. Martinet, "A Generic Fisheye camera model," In Proceedings of IEEE International Conference on Intelligent and Robotic Systems, 2007.
- [10] C. Demonceaux, P. Vasseur, C. Pégard, "Robust Attitude Estimation with Catadioptric Vision," In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.
- [11] C. Demonceaux, P. Vasseur, C. Pégard, "Omnidirectional vision on UAV for attitude computation," In Proceedings of IEEE International Conference on Robotics and Automation, 2006.
- [12] C. Demonceaux, P. Vasseur, C. Pégard, "UAV Attitude Computation by Omnidirectional Vision in Urban Environment," In Proceedings of IEEE International Conference on Robotics and Automation, 2007.
- [13] D. Gallup, J-M. Frahm, P. Mordohai, Y. Qingxiong, M. Pollefeys, "Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions," In Proceedings of IEEE Computer Vision and Pattern Recognition, 2007.
- [14] P. Garcia-Padro, G. Sukhatme, J. Montgomery, "Towards visionbased safe landing for an autonomous helicopter," In Robotics and Autonomous Systems, 2000.
- [15] I. Geys, T. P. Koninckx, L. V. Gool, "Fast Interpolated Cameras by combining a GPU based Plane Sweep with a Max-Flow Regularisation Algorithm," In Proceedings of IEEE 3D Processing, Visualization and Transmission, 2004.
- [16] W. Green, P. Oh, K. Sevcik, G. Barrows, "Autonomous landing for indoor flying robots using optic flow," in ASME International Mechanical Engineering Congress and Exposition, vol. 2, 2003, pp. 1347-1352.
- [17] R. Hartley, A. Zisserman, "Multiple View Geometry in computer vision," Cambridge 2nd edition, 2003.
- [18] J. Hoffmann, M. Jungel, M. Lotzsch, "Vision Based System for Goal-Directed Obstacle Avoidance," In 8th International Workshop on RoboCup, 2004.
- [19] S. Hrabar, G. Sukhatme, "Omnidirectional vision for an autonomous helicopter," In Proceedings of IEEE International Conference on Robotics and Automation, 2004.

- [20] I.K. Jung, S. Lacroix, "High resolution terrain mapping using low altitude aerial stereo imagery," In Proceedings of IEEE International Conference on Computer Vision, 2003.
- [21] Y. Kim, H. Kim, "Layered ground floor detection for vision-based mobile robot navigation," In Proceedings IEEE International Conference on Robotics and Automation, 2004.
- [22] B. Liang, N. Pears "Visual Navigation using Planar Homographies," In Proceedings of IEEE International Conference on Robotics and Automation, 2002.
- [23] Y. Ma, S. Soatto, J. Kosecka, S. Shankar Sastry, "An invitation to 3D Vision," Springer, 2003.
- [24] C. Mei, S. Benhimane, E. Malis, P. Rives, "Homography-based Tracking for Central Catadioptric Cameras," In Proceedings of IEEE Intelligent Robots and Systems, 2006.
- [25] C. Mei, P. Rives, "Single View Point Omnidirectional Camera Calibration from Planar Grids," In Proceedings of IEEE International Conference on Robotics and Automation, 2007
- [26] M. Meingast, C. Geyer, S. Sastry, "Vision Based Terrain Recovery for Landing Unmanned Aerial Vehicles," In Proceedings of IEEE International Conference on Decision and Control, 2004.
- [27] M. Sanfourche, G. Besnerais, S. Foliguet, "Height estimation using ariel side looking image sequences," In ISPRS, Vol.XXXIV, Part3/W8, Munich, 2003.
- [28] S. Saripalli, J. Montgomery, G. Sukhatme, "Vision based autonomous landing of an unmanned aerial vehicle," In Proceedings of IEEE International Conference on Robotics and Automation, 2002.
- [29] C. Sharp, O. Shakernia, S. Sastry, "A vision system for landing an unmanned aerial vehicle," In Proceedings of IEEE International Conference on Robotics and Automation, 2001.
- [30] P. Sturm, "Mixing Catadioptric and Perspective Cameras," In Proceedings of IEEE Omnidirectional Vision,2002
- [31] X. Ying, Z. Hu, "Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model," In European Conference on Computer Vision, 2004.
- [32] J. Zhou, B. Li, "Robust Ground Plane Detection with Normalized Sequences from a Robot Platform," In Proceedings of IEEE International Conference on Image Processing, 2006