



HAL
open science

Real Time UAV Altitude, Attitude and Motion Estimation form Hybrid Stereovision

Damien Eynard, Pascal Vasseur, Cédric Demonceaux, Vincent Fremont

► **To cite this version:**

Damien Eynard, Pascal Vasseur, Cédric Demonceaux, Vincent Fremont. Real Time UAV Altitude, Attitude and Motion Estimation form Hybrid Stereovision. *Autonomous Robots*, 2012, 33 (1-2), pp.157-172. 10.1007/s10514-012-9285-0 . hal-00716093

HAL Id: hal-00716093

<https://hal.science/hal-00716093v1>

Submitted on 30 Apr 2018

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.

Real time UAV Altitude, Attitude and Motion Estimation from Hybrid Stereovision.

Damien Eynard and
Pascal Vasseur and
Cédric Demonceaux
and Vincent Frémont

Received: date / Accepted: date

Abstract Knowledge of altitude, attitude and motion is essential for an Unmanned Aerial Vehicle during critical maneuvers such as landing and take-off. In this paper we present a hybrid stereoscopic rig composed of a fisheye and a perspective camera for vision-based navigation. In contrast to classical stereoscopic systems based on feature matching, we propose methods which avoid matching between hybrid views. A plane-sweeping approach is proposed for estimating altitude and detecting the ground plane. Rotation and translation are then estimated by decoupling: the fisheye camera contributes to evaluating attitude, while the perspective camera contributes to estimating the scale of the translation. The motion can be estimated robustly at the scale, thanks to the knowledge of the altitude.

We propose a robust, real-time, accurate, exclusively vision-based approach with an embedded C++ imple-

mentation. Although this approach removes the need for any non-visual sensors, it can also be coupled with an Inertial Measurement Unit.

Keywords UAV · hybrid stereovision · motion · attitude · altitude

1 Introduction

Unmanned Aerial Vehicles (UAV) have received a lot of attention over the last decade, in relation to command and on-board computer vision. Interest in these topics has focused largely on increasing UAV autonomy, which includes maneuvers such as landing and take-off. In this context a fast, robust, accurate estimation of critical parameters such as altitude, attitude, motion and velocity is clearly crucial for the control loop.

A number of sensors have been used to estimate these parameters. Altitude can be determined by GPS, altimeter, radar, or laser. Attitude can be provided by Inertial Measurement Unit (IMU), and motion by GPS or radar. Raw Global Positioning Systems (GPS) have a poor accuracy both vertically (accuracy between 25 and 50 meters) and horizontally (less than 15 meters). Moreover, GPS is sensitive to interruptions in transmission in urban environments. Radar can estimate both altitude and velocity with greater accuracy, but it requires active sensors that consume power. Altimeters are very widely used, but they depend on pressure variations, which implies an accuracy error of between 6% and 7%. Laser altimeters, on the other hand, are highly accurate, but they have very specific requirements concerning reflecting surfaces. Finally, IMU can provide indications of velocity, acceleration, attitude and orientation, but it is subject to drift and error accumulation.

As an alternative to these sensors, vision-based systems, thanks to the growth of computational capacity, can now rapidly estimate all these parameters, as well as performing other visual tasks. Furthermore, cameras remain compact, passive systems with correspondingly low energy consumption, and can provide information at high rates (up to 200Hz).

Most of the works using computer vision for UAV are based on the optical flow analysis obtained with a single camera Barrows et al. (2001); Beyeler et al. (2006); Chahl et al. (2004); Green et al. (2003). Indeed, optical flow can be efficiently computed and consequently embedded for navigation purpose. These systems were bio-inspired by information processing mechanisms that have evolved in bees, and consist in deducing the altitude according to the optical flow, given knowledge of the camera's motion. Beyeler et al. (2006) is alone in proposing an estimation of pitch to cor-

D. Eynard
MIS Laboratory, 7, rue du moulin neuf - University of Picardie Jules Verne, Amiens, France
Tel.: +333-22-827663
Fax: +333-22-827618
E-mail: damien.eynard@u-picardie.fr

P. Vasseur
LITIS Laboratory, University of Rouen, Saint-Etienne-du-Rouvray, France
E-mail: pascal.vasseur@univ-rouen.fr

C. Demonceaux
Le2i Laboratory - UMR CNRS 5158, University of Burgundy, Le Creusot, France
E-mail: cedric.demonceaux@u-bourgogne.fr

V. Frémont
Heudiasyc Laboratory of University of Technology of Compiègne, France
E-mail: vincent.fremont@hds.utc.fr

rect the optical flow, without which an unstable system might result. An original approach using a single perspective camera is to be found in Cherian et al. (2009). The authors use a technique based on the learning of a mapping between texture information contained in a top-down aerial image and altitude values. This learning stage is performed over different kinds of terrain, and a spatio-temporal Markov Random Field is used. However, optical flow information is clearly not sufficient for a complete control of a UAV.

2 Related Work

As explained before, orientation and position parameters are essential for the control of the UAV and vision based methods can be very interesting because of the richness of the provided information. A hierarchy between these parameters can be established according to their importance in the control loop. The most important one deals with the attitude knowledge for the stabilization followed by the altitude for hovering and landing maneuvers and finally position and yaw angle for trajectory. Consequently, this ranking imposes the refreshing rate requirement for the control. In this work, even if we do not manage the control of the UAV, our aim is to propose a complete onboard vision system able to provide orientation and position parameters at the required frequency rate for a future loop control. Many works based on vision solely or vision fused with other sensors have been already proposed. A complete recent review is presented in Weiss et al. (2011). These works can be divided in two categories which are respectively based on cameras placed in the environment or based on embedded cameras on the UAV. The former consists generally in a fixed camera network that supervises completely a 3D scene. The most well known system is Vicon and has been successfully employed in UAV control Mellinger et al. (2011). Such systems are particularly efficient with a high measure frequency, very accurate with no drift and also robust. However, the fixed configuration does not allow any exploration of a new and unknown environment. The second category deals with onboard camera and can also be divided according to two sections. The first one is based on the use of a known pattern placed in the environment. In Garcia-pardo et al. (2000); Saripalli et al. (2002); Sharp et al. (2001), the authors make use of a downward-pointing perspective camera to estimate altitude according to a predefined pattern laid out on the ground. Recently, a circular pattern has been used in order to measure the pose of a UAV Eberli et al. (2011). This kind of approach is interesting, since it requires only one camera, provides a complete pose and can be used in real time.

This solution is the complementary of Vicon approach since the camera looks an *a priori* known environment while with Vicon, cameras of the environment tracks a known pattern placed on the UAV. The advantages and drawbacks are then similar. The other solution consists of onboard cameras that capture and treat data from an unknown and unstructured environment (no artificial landmark). In this way, it is possible to estimate orientation and position according to features detected in the environment. If some absolute features such as horizon, vertical direction or ground are used, it is then possible to perform the estimation in an absolute reference frame and to obtain absolute measures of attitude and altitude. For example to estimate the attitude from horizon line, Démonceaux et al. (2006) propose a method based on an omnidirectional view while Dusha et al. (2007) use a perspective camera. Thurrowgood et al. (2009) estimates the attitude using a fast histogram based method which can be adapted for any type of camera. For a complete absolute six degrees of freedom estimation, an initial geo-localized reference is then necessary. If used features have no particular identity, the orientation and position estimation is performed relatively to the structure formed by these features. Such methods (SLAM, SFM, Visual Odometry, ego-motion) are now well-known and many different algorithms have been already proposed successfully in Davison et al. (2007); Hartley and Zisserman (2004); Nistér (2004). Then, works of Li et al. (2008); Pless (2003) can determine the motion from non-overlapping cameras. Nevertheless they need a least of 14-points or 17-points to retrieve the motion that increases the computation time and noise sensitivity. However, if these methods are generally well adapted for mobile robots or vehicles, a direct transfer to micro-aerial vehicles is not possible and requires specific adaptations and developments Lee et al. (2010). Since most of the efficient methods are based on feature points, a possible adaptation consists in including some *a priori* knowledges or hypotheses in order to reduce the number of necessary points. In Kalantari et al. (2011), Fraundorfer et al. (2010), Naroditsky et al. (2011), the authors propose a relative pose estimation between two views by a three-plus-one algorithm, based on the correspondence of three points, given a common direction in the two views. This common direction can then be obtained by IMU for example. A recent work proposed also to use only one point for structure from motion Scaramuzza (2011). However, the used constraint for reducing the number of points is only valid for mobile robots.

The most recent and successful autonomous micro-aerial vehicle based on vision with adapted methods is described in Blösch et al. (2010); Weiss et al. (2011).

The authors propose a monocular Simultaneous Localization and Mapping (SLAM) framework from which they extract pose parameters for a complete control of the six degrees of freedom. The use of a SLAM approach allows to correct the possible drift by loop closing. Since their system is based on a single wide field of view camera, the depth information can not be directly recovered. This is why they propose the initialization of the map scale by hand and maintain the consistency of this scale by fusing IMU data and camera poses through an EKF Nützi et al. (2011). The complete pose from the SLAM is computed at 15-30 Hz and the comparison of results with a VICON shows the effectiveness of the approach. In Artieda et al. (2009), a monocular vision sensor is also proposed in order to perform the visual SLAM in partially structured environments. Other sensors are just used in order to propose comparisons. In their work, the authors deal with a deep study of the different steps of the visual SLAM such as the calibration, the influence of depth and the image processing techniques for feature detection and matching for example. Their results are also shown to be globally satisfactory but the treatment is made off-line at an average of 12 FPS. A discussion of the scale estimation is proposed but no real results on this parameter are shown. Recently, a visual monocular SLAM algorithm has been proposed especially for MAVs by integrating a plane constraint Lee et al. (2011). This constraint allows to obtain an approximately constant time algorithm that has been tested in indoor environments. A work quite more closer to ours consists in using horizon and points of the ground in order to estimate the ego-motion of a UAV Oreifej et al. (2011). The use of the horizon line allows to linearize and disambiguate the planar flow, and consequently to obtain a unique solution for the UAV motion estimation. However, this work has only been tested on a balloon-based UAV with motion particularly different from a quad-rotor.

All these works propose very interesting results but they are based on a single camera and either are up to scale or require a manual initialization and the use of other sensors for maintaining this scale Weiss et al. (2011). In order to obtain a complete vision approach, we propose in this paper to use a stereovision sensor for the estimation of the orientation and position parameters at scale. Due to the limited field of view of the perspective cameras, it has been demonstrated that the use of omnidirectional cameras is more suitable for motion estimation Lee et al. (2000); Lhuillier (2008). Indeed, omnidirectional cameras can gather more information from the environment but are less accurate owing to their limited resolution. In order to have both the wide field of view and the accuracy, we then propose

an embedded hybrid stereovision sensor. However, classical stereovision based methods employ feature point matching between views, which generally constitutes a bottleneck in the whole process. In our case, this difficulty is compounded because of the heterogeneous nature of our images that prevents a direct comparison between points without any adaptation. Thus, in order to reduce this difficulty, a first solution consists in using the epipolar geometry between views and to limit the search area for each point to match. Recently, Puig et al. proposed an hybrid fundamental matrix and its use for hybrid matching Puig et al. (2008). However, their results show that very few points are positively matched and a complete and accurate pose estimation seems to be complicated. A second way for improving the results consists in adapting the point features to the geometry of the sensor. Many different works have been already proposed in order to adapt Harris corner detector Demonceaux et al. (2011) or SIFT descriptors Bastanlar et al. (2010), Lourenço et al. (2010). However, the principal drawback is the expensive computation time that totally prevents a real time implementation on an UAV. We then propose in this paper an hybrid stereovision approach that avoids point feature matching between hybrid views and which is based on a direct comparison of images by assuming that the planar ground surface is visible in both images.

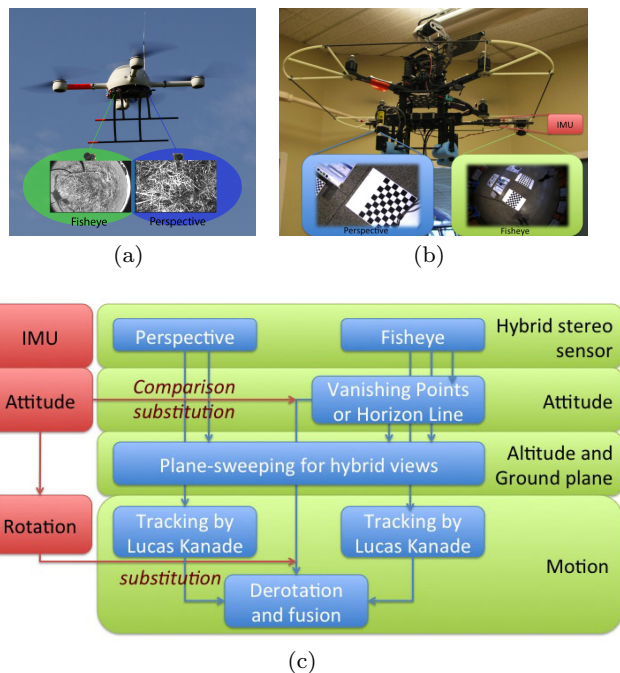


Fig. 1 (a) UAV with hybrid system outdoors. (b) Pelican in an indoor environment. (c) General method proposed.

The principal contributions of this paper are four-fold. The first is the hybrid sensor comprising fisheye and perspective views. Secondly, we have the use of correspondence-less methods. This leads to the third contribution, namely the real-time implementation of altitude, attitude and motion, adapted for future embedded applications. Finally, we have an estimation of the motion performed independently in each of the views, at a metric scale. The estimated motion is then merged and filtered by a Kalman filter.

The organization of the article is as follows. Part III deals with the hybrid sensor and its modeling. Part IV describes the estimation of attitude, altitude and motion. Part V is devoted to a quantitative evaluation of data obtained using a small UAV.

3 Hybrid sensor

3.1 Motivation

In this paper, we present a new approach for state (orientation and position) estimation of a MAV based on an hybrid stereovision sensor. Contrary to SLAM frameworks that provide simultaneously all the parameters, our method proposes a cascading approach that allows to obtain attitude, altitude and motion at different rates (see Fig.1(c)). Attitude is then first obtained from the omnidirectional camera by use of the horizon or vertical direction but can also be obtained by IMU if available. This parameter being the most important for flight stability, our system is able to provide it at 30Hz for the vanishing points method and 100Hz for the horizon line based method. The use of a stereovision sensor allows also to directly estimate the altitude at scale and consequently to avoid any manual initialization or any knowledge about the dimension of the scene or the use of an additional sensor for maintaining this information during the flight. We assume a consistent planar ground similarly than in Lee et al. (2011) which is reasonable in most of the cases. Therefore, we will show in the results that imperfect grounds have low impact on the altitude estimation and that little obstacles can be present in the image. In the same way, we show experiments in order to show the accuracy of the system according to the ratio between altitude and baseline.

Thus, our hybrid sensor thus combines nested methods.:

- First, we calibrate the hybrid system using Caron and Eynard (2011).
- Second, attitude is estimated either by vanishing point detection in urban and indoor environments

Demonceaux et al. (2007), or by the horizon fitting in other cases Demonceaux et al. (2006);

- Third, as a result of the previous steps, knowing that a homography exists between the two views, altitude estimation and ground plane segmentation are performed by plane-sweeping Eynard et al. (2010).
- Finally, motion is decoupled in rotation and translation. Rotation is known by attitude or IMU, and translation is estimated by a two-point algorithm in hybrid views Eynard et al. (2011). By knowing altitude previously estimated, the translation is provided at the metric scale.

By implementing these methods in C++ we show that our methods are real time and ready to be embedded in a UAV.

3.2 Modeling

Perspective Model

The perspective projection (Fig. 2(a)) models pin-hole cameras. Let $\mathbf{X} = (X Y Z)^T$ be a 3D point expressed in the camera frame. It is projected onto the image plane as $\mathbf{x} = (x y 1)^T$:

$$\mathbf{x} = pr(\mathbf{X}) \text{ with } \begin{cases} x = \frac{X}{Z} \\ y = \frac{Y}{Z} \end{cases}. \quad (1)$$

\mathbf{x} is the point on the normalized image plane and $\mathbf{u} = (u v 1)^T$, the pixel point, is obtained by the relation $\mathbf{u} = \mathbf{K}\mathbf{x}$. \mathbf{K} is the intrinsic matrix, knowing parameters $\gamma_p = \{p_x, p_y, u_0, v_0\}$:

$$\mathbf{K} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2)$$

The full perspective projection of a 3D point to the pixelic image plane is therefore $pr_{\gamma_p}(\mathbf{X}) = \mathbf{K}pr(\mathbf{X})$.

Spherical Model

Fisheye lenses cannot be classified as single view-point sensors Baker and Nayar (1999). Nevertheless, we can show that this modeling is a good approximation of Ying and Hu (2004). Mei and Rives (2007) have proposed a calibration method based on a unitary sphere. This model is particularly accurate, and allows the radial and tangential distortions caused by the fisheye lens to be modeled. Using the spherical model (Fig. 2(b)), a 3D point \mathbf{X} is first projected onto a unitary sphere, centered at $(0 0 0)^T$. The obtained point, \mathbf{x}_s , is then

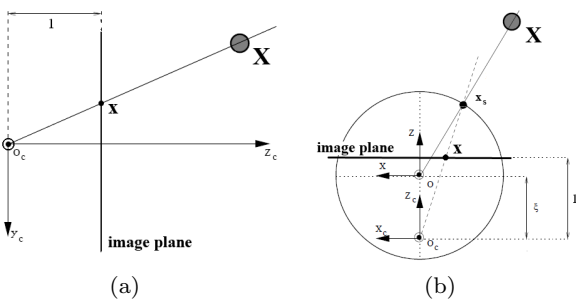


Fig. 2 (a) Perspective model used for pinhole cameras. (b) Spherical model used for omnidirectional cameras.

perspectively projected onto the image plane as \mathbf{x} , from the second center of projection $(0 \ 0 \ -\xi)^T$ (2(b)):

$$\mathbf{x}_s = \frac{\mathbf{X}}{\rho}, \quad \mathbf{x} = pr_{\xi}(\mathbf{X}) \quad \text{with} \quad \begin{cases} x = \frac{X}{Z+\xi\rho} \\ y = \frac{Y}{Z+\xi\rho} \end{cases}, \quad (3)$$

and $\rho = \sqrt{X^2 + Y^2 + Z^2}$. The image point is obtained from a 3D point, knowing the intrinsic parameters $\gamma_s = \{p_x, p_y, u_0, v_0, \xi\}$ and using $pr_{\gamma_s}(\mathbf{X}) = \mathbf{K}pr_{\xi}(\mathbf{X})$ (eq. 1).

3.3 Calibration

The hybrid calibration is based on Virtual Visual Servoing work, which is in turn based on the calibration of perspective cameras Marchand and Chaumette. Caron and Eynard (2011) simultaneously estimate the projection parameters and relative poses between N cameras of N models composing the stereo rig. They have demonstrated that calibration results for standard cameras are similar to Bouguet's toolbox or Mei's toolbox using same points in the image. This method is implemented in the Hyscas software Hyscas (2011). We use this software to calibrate our hybrid model. Then, the extrinsic parameters obtained from hybrid calibration are the rotation \mathbf{R}_c and the translation \mathbf{t}_c between the two cameras.

In the case of attitude estimation using IMU, the IMU and fisheye calibration is estimated by the InerVis software Ine. The matrices provided by IMU are directly corrected in order to be expressed in the fisheye view.

3.4 Attitude

We can distinguish three main approaches for estimating attitude using two sensors.

Vision in urban environments: attitude and orientation can be estimated in catadioptric views Bazin

et al. (2008); Démonceaux et al. (2007). These works are based on vanishing points resulting from the projection of vertical and horizontal lines in 3D space onto the unitary sphere. The algorithm consists of first detecting edges (using Canny for instance) in the omnidirectional image. Then a step of split and merge keeps long edges by deleting short lines. Once lines have been extracted, their normals are computed from the unitary sphere, which is known by calibration. Finally, using a voting approach, the three main directions are computed from both vertical and horizontal lines. Views are modeled by the proposition of Barreto and H. (2001).

Vision in natural environments: Démonceaux et al. (2006) also proposes estimating attitude via a projection of the horizon line onto a catadioptric view. This method is able to estimate pitch and roll. By using one of Kalantari et al. (2009); Montiel and A.-J. (2006); Scaramuzza and Siegwart (2008) methods, it is also possible to retrieve the heading.

IMU: attitude can be obtained from Euler angles.

In our experiments we estimate attitude using either a fisheye view or IMU.

3.5 Hardware

The hybrid stereo rig is composed of two IDS uEye cameras with M12 lenses, the first is a perspective lens while the second is a fisheye. The stereo baseline is 32 cm. Each camera provides 752×480 RGB images, and the two cameras are located at opposite extremities of the rig, pointing downwards. Frame grabbers are triggered by hardware, and ring-buffer acquisition allows continuous data transfer and non-stop frame acquisition.

An XSens IMU is attached to the rig to determine the attitude ground truth. An embedded board with an Atom CPU receives and records images as well as IMU data. Images are sent and the UAV acquisition controlled via a wifi interface. The Atom board, the cameras and IMU are embedded on an MD4-1000 UAV (see Fig.1(a)). For processing, we use a Macbook Pro with Core 2 Duo at 2.4 GHz.

4 Altitude

Plane-sweeping was introduced by Collins (1996), proposed in real time by Gallup et al. (2007) and then adapted to hybrid views in Eynard et al. (2010). First a reference view has to be defined. Then, for each normal \mathbf{n} and for each distance to a 3D plane, the different warped images are compared (Eq. 4) to the reference image by homography. Let \mathbf{I} and \mathbf{I}_{ref} be respectively the current and the reference image, $\mathbf{I}(\mathbf{x})$ the intensity

of the normalized pixel \mathbf{x} . Then, \mathbf{I}^* is the homography of \mathbf{I} by the homography matrix \mathbf{H} . For a pixel \mathbf{x} at the altitude d we get:

$$\mathbf{I}^*(\mathbf{x}, d) = \mathbf{I}(\mathbf{H}_{\mathbf{x}, d}) = \mathbf{I}\left(\left(\mathbf{R}_c - \mathbf{t}_c \frac{\mathbf{n}^T}{d}\right)\mathbf{x}\right) \quad (4)$$

The best estimation of the homography \mathbf{H} corresponds to the minimum global error of the difference between the warped image and the reference view $|\mathbf{I}^* - \mathbf{I}_{ref}|$. In our application we extend this aspect. We take the perspective view as the reference \mathbf{I}_{ref} . Neighborhood manipulations are easier to perform and faster than on a sphere. Images \mathbf{I} obtained via the fisheye camera are projected onto the sphere and then onto the reference plane by homography, and are denoted \mathbf{I}^* . Since the cameras are pointing towards the ground, the centered working surface on the fisheye is subject to fewer distortions and gives a better resolution than at the edge of the image. Finally, plane-sweeping can be performed (see Fig. 3(a)).

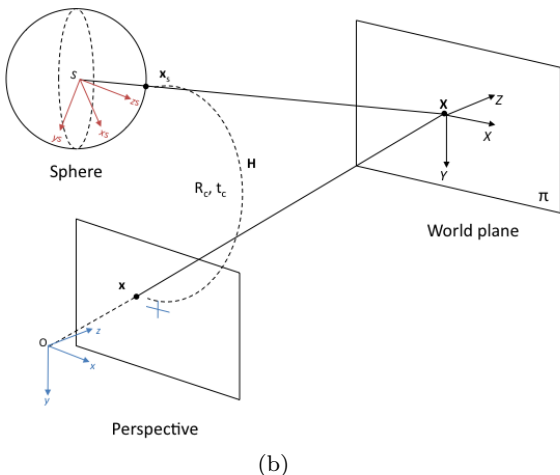
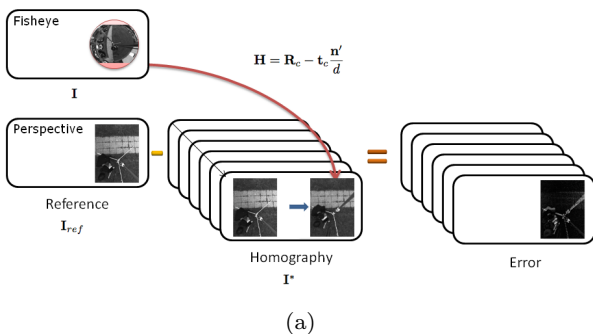


Fig. 3 (a) Mixed plane-sweeping. (b) Sphere/plane homography.

We propose estimating the altitude d and segmenting the ground plane by mixed plane-sweeping, with

$\mathbf{R}_c, \mathbf{t}_c, \mathbf{n}$ known by calibration and attitude estimation. We first present the homography used in our hybrid case, before describing the mixed plane-sweeping algorithm.

4.1 Sphere to plane homography

Given a mixed stereo rig modeled by a plane and a unitary sphere, we shall now define the homography of the 3D ground plane that exists between two views (see Fig.3(b)). In Hartley and Zisserman (2004), a homography links two projections of a 3D plane onto two planes. In Mei et al. (2006), homography relates two projections of a 3D plane onto two spheres. The homography is defined by the following equation:

$$\mathbf{H} = \mathbf{R}_c - \mathbf{t}_c \frac{\mathbf{n}^T}{d} \quad (5)$$

We obtain $\mathbf{x}^* \sim \mathbf{H}^{-1}\mathbf{x}$ for a homography between two planar projections of a 3D plane. These two planar projections are then replaced by a planar and a spherical projection, giving Eq.6.

$$\mathbf{x} \sim \mathbf{x}_s^* = \rho' \mathbf{H} \mathbf{x}_s \quad (6)$$

Homography \mathbf{H} depends on $\mathbf{R}_c, \mathbf{t}_c$, the attitude \mathbf{n} and the altitude d . $\mathbf{R}_c, \mathbf{t}_c$ and \mathbf{n} being known, the next step is to obtain the estimation of altitude that minimizes the error defined by Eq.7. It will be remarked that in this step we can extract the ground plane using the mask \mathbf{G} . All the points in the perspective view satisfying \mathbf{H} can be considered as belonging to the ground plane (Eq.8). In the following subsection we present the algorithm for segmenting the ground plane and estimating the altitude d .

$$\hat{d} = \operatorname{argmin}_d \left(\sum_{\mathbf{x}} |\mathbf{I}_{ref}(\mathbf{x}) - \mathbf{I}^*(\mathbf{x}, d)| \right) \quad (7)$$

$$\mathbf{G} = \left\{ \mathbf{x}, \frac{\sum_{\mathbf{x}} |\mathbf{I}_{ref}(\mathbf{x}) - \mathbf{I}^*(\mathbf{x}, \hat{d})|}{\sum_{\mathbf{x}} (\mathbf{I}_{ref}(\mathbf{x}))} < thres \right\} \quad (8)$$

4.2 Algorithm

Altitude is determined by an exhaustive search. Let $[d_{min}, d_{max}]$ be the altitude range in the estimation. For each iteration, the best altitude \hat{d} is estimated from this range, then the mask \mathbf{G} is updated (see Algorithm 1). Pixels corresponding to the ground plane are in white in figures 8(c) and 10(e). In order to obtain a real-time method we estimate the altitude at time t using the altitude at $(t - 1)$.

The estimation is performed in two steps:

Algorithm 1 Altitude and ground plane segmentation algorithm - initialization

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

```

{Initialization}
 $a_0 = \widehat{d}_{-1} = d_{min}$ 
 $b_0 = \widehat{d}_0 = d_{max}$ 
 $G = \{pePixels\}$ 
while  $|\widehat{d}_k - \widehat{d}_{k-1}| > \Delta d$  do
  {Estimation of the best altitude}
   $\widehat{d}_{k+1} = \operatorname{argmin}_{d \in \{\frac{b_k - a_k}{s-1}t + a_k; t \in [0, s-1]\}} \sum_{\mathbf{x} \in G} |I_{ref}(\mathbf{x}) - I^*(\mathbf{x}, d_k)|$ 
  {Estimation of the mask of inliers/outliers}
   $G = \{\mathbf{x}, \frac{\sum_{\mathbf{x}_1 \in W_x} |I_{ref}(\mathbf{x}_1) - I^*(\mathbf{x}_1, \widehat{d})|}{\sum_{\mathbf{x}_1 \in W_x} I_{ref}(\mathbf{x}_1)} < thres\}$ 
  {Estimation of the new range depending on sampling}
   $a_{k+1} = \widehat{d}_{k+1} - \frac{b_k - a_k}{s-1}$ 
   $b_{k+1} = \widehat{d}_{k+1} + \frac{b_k - a_k}{s-1}$ 
   $k = k + 1$ 
end while
Return  $\widehat{d}$ 

```

- Initialization: we estimate the best altitude from within a wide altitude range.
- During flight we use the altitude estimated in the initialization phase to obtain a narrower range in Eq.10) with r_d in Eq.9. This range depends on the vertical velocity v_v of UAV (about $\pm 5m/s$) and the hardware's computational power in frames per second denoted fps .

$$r_d = \frac{v_v}{fps} 10^3 (mm) \quad (9)$$

$$d_t \in [d_{t-1} - r_d, d_{t-1} + r_d] \quad (10)$$

We now show how calibration, attitude, estimated altitude and the ground plane segmentation, once obtained, can be used to estimate autonomously the motion using the hybrid views, without any feature matching between hybrid views.

5 Motion estimation

To estimate the motion we need the calibration ($\mathbf{R}_c, \mathbf{t}_c$), the altitude d at the meter scale, the attitude and the ground plane segmentation \mathbf{G} . The attitude can be determined either by IMU or by vision using vanishing points in urban environments, or the horizon line in natural environments. In the case of natural environments the UAV's heading has to be computed *a posteriori*, using such methods as Kalantari et al. (2009); Montiel and A.-J. (2006); Scaramuzza and Siegwart (2008). Since

altitude, attitude, ground plane and calibration are determined by vision or IMU, we decouple the rotation and the translation of the motion. Below we present an algorithm for estimating the translation from two sets of points (Fig. 4(b)), that is to say points located on the ground plane \mathbf{G} and points located randomly in the environment with depth either known or unknown. Our algorithm is based on the technique of point tracking in hybrid views for estimating the translation from two views. The estimated translation \mathbf{t} is then filtered and smoothed by a Kalman filter.

5.1 Motion of the stereo rig

In each image, tracking is performed by the method given in Bouguet (2000). We define \mathbf{x}_t a tracked point in the image acquired at time t . Each point is related by a rotation \mathbf{R}_t relative to the world reference \mathbf{X}_w estimated by the IMU (see Eq.11).

$$\mathbf{x}_t = {}^t\mathbf{R}_w \mathbf{X}_w \iff \mathbf{X}_w = {}^t\mathbf{R}_w^{-1} \mathbf{x}_t \quad (11)$$

Then, for a couple of points in the image ($\mathbf{x}_t; \mathbf{x}_{t+1}$) that indicate motion between times t and $t+1$, we can express the point \mathbf{x}_t from the frame t to the frame $t+1$ by Eq.12. We thus obtain a rotation corresponding to motion (eq.13).

$$\mathbf{x}_{t+1} = {}^{t+1}\mathbf{R}_w {}^t\mathbf{R}_w^{-1} \mathbf{x}_t \quad (12)$$

$$\mathbf{R}_{t+1} = {}^{t+1}\mathbf{R}_w {}^t\mathbf{R}_w^{-1} \quad (13)$$

The points are related by a motion composed of the rotation \mathbf{R}_{t+1} previously defined and the translation \mathbf{t}_{t+1} (Fig.4(a), Eq.14). The distance of the image point to the 3D point is defined by the altitude d (see Fig.4(a)) known by plane-sweeping in the case of a point belonging to the ground plane.

$$\mathbf{x}_{t+1} = d \mathbf{R}_{t+1} \mathbf{x}_t + \mathbf{t}_{t+1} \quad (14)$$

5.2 Motion from ground points

As mentioned previously, plane-sweeping is used to estimate both the altitude and the segmentation of the ground plane \mathbf{G} . Knowledge of these parameters entails knowledge of tracked points in the two views belonging to the ground plane, together with their depth (see Fig.4(a)). Then, the motion is estimated at the metric

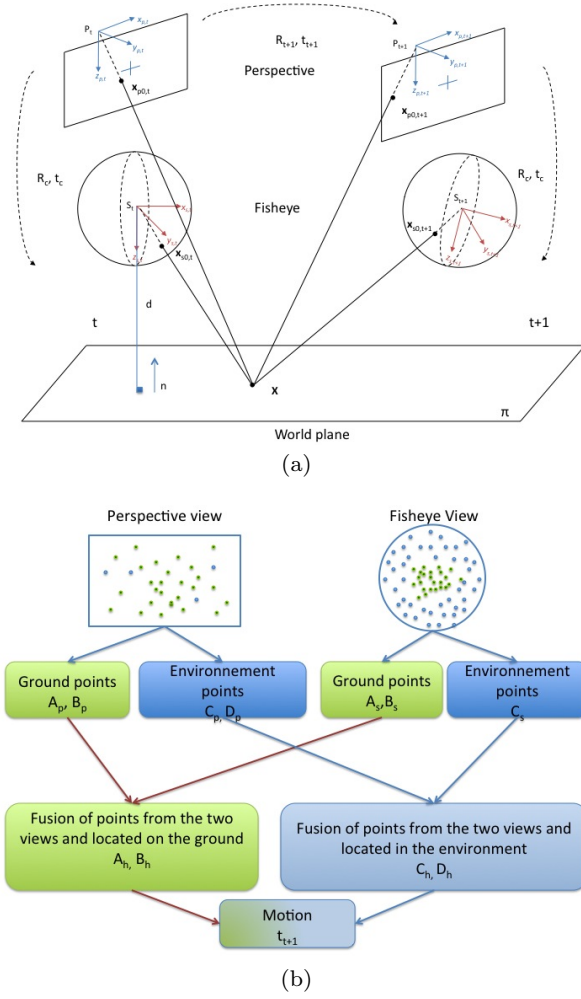


Fig. 4 fig. 4(a) 3D points located on the ground plane and in the environment. Fig. 4(b) Combining points located in the environment and the ground plane from hybrid views.

scale and defined as follows, with \times the cross product of two vectors:

$$(\mathbf{x}_{t+1} \times \mathbf{t}_{t+1}) = -d(\mathbf{x}_{t+1} \times \mathbf{R}_{t+1}\mathbf{x}_t) \quad (15)$$

We now show how motion can be estimated in each view and then in the mixed view. First, in the spherical an referential case, we obtain the matrices \mathbf{A}_s Eq.16 and \mathbf{B}_s Eq.17. Secondly, in the perspective case, images are considered to be rectified by the rotation \mathbf{R}_c , and we obtain the matrices \mathbf{A}_p Eq.16 and \mathbf{B}_p Eq.18. In the hybrid case \mathbf{A}_s and \mathbf{A}_p are concatenated in \mathbf{A}_h , and \mathbf{B}_s and \mathbf{B}_p are concatenated in \mathbf{B}_h . Finally, we obtain Eq.22.

$$\mathbf{A}_{s/p} = \begin{bmatrix} 0 & -x_{t+1} & x_{t+1} \\ z_{t+1} & 0 & -x_{t+1} \\ -y_{t+1} & x_{t+1} & 0 \end{bmatrix} \quad (16)$$

$$\mathbf{B}_s = -d_s (\mathbf{x}_{t+1} \times \mathbf{R}_{t+1}\mathbf{x}_t)^T \quad (17)$$

$$\mathbf{B}_p = -d_p ((\mathbf{x}_{t+1} \times \mathbf{R}_{t+1}\mathbf{x}_t) + \mathbf{x}_{t+1} \times ((\mathbf{R}_{t+1} - \mathbf{I}_d)\mathbf{t}_c))^T \quad (18)$$

5.3 Motion from environment points

Concerning points located randomly in the environment, without possessing any knowledge of their depth we can extend to the mixed view the motion estimation proposed by Bazin et al. (2010) for the spherical view (see Fig.4(a)). The translation \mathbf{t}_{t+1} from two points is defined as in Eq.19. As mentioned previously, perspective points and spherical points are concatenated in order to estimate the translation \mathbf{t}_{t+1} for obtaining the relation in Eq.23.

$$(\mathbf{R}_{t+1}\mathbf{x}_t \times \mathbf{x}_{t+1})^T \cdot \mathbf{t}_{t+1} = 0 \quad (19)$$

5.4 Combining 3D points from mixed views

Note: In this part, we adapt the generic formulation to the case of perspective case. $\mathbf{x}_{p_i,t}$ denotes the i^{th} point tracked in the perspective view at time t , and $\mathbf{x}_{s_i,t}$ denotes the i^{th} point tracked in the sphere at the time t .

$$\mathbf{C}_h = \begin{bmatrix} \mathbf{R}_{t+1}\mathbf{x}_{s_0,t} \times \mathbf{x}_{s_0,t+1} \\ \vdots \\ \mathbf{R}_{t+1}\mathbf{x}_{s_m,t} \times \mathbf{x}_{s_m,t+1} \\ \mathbf{R}_{t+1}\mathbf{x}_{p_0,t} \times \mathbf{x}_{p_0,t+1} \\ \vdots \\ \mathbf{R}_{t+1}\mathbf{x}_{p_n,t} \times \mathbf{x}_{p_n,t+1} \end{bmatrix} \quad (20)$$

Considering perspective images rectified by \mathbf{R}_c known by calibration, we get:

$$\mathbf{D}_h = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ (\mathbf{R}_{t+1}\mathbf{x}_{p_0,t} \times \mathbf{x}_{p_0,t+1}) \cdot (\mathbf{R}_{t+1}\mathbf{t}_c - \mathbf{t}_c) \\ \vdots \\ (\mathbf{R}_{t+1}\mathbf{x}_{p_n,t} \times \mathbf{x}_{p_n,t+1}) \cdot (\mathbf{R}_{t+1}\mathbf{t}_c - \mathbf{t}_c) \end{bmatrix} \quad (21)$$

$$\mathbf{A}_h \mathbf{t}_{t+1} = \mathbf{B}_h \quad (22)$$

$$\mathbf{C}_h \mathbf{t}_{t+1} = \mathbf{D}_h \quad (23)$$

We have already presented a generic formulation for translation estimation, either from ground points or from environment points. When translating from planar points, the translation is estimated at the metric scale. However, the main drawback in the case of 3D motion is the pixel projection noise sensitivity. When translating from environment points, on the other hand, motion estimation has the advantage of being more robust to noise than on the plane, but the estimation is performed up to scale. One of the contributions of this paper is thus the combination of the two methods to increase both accuracy and robustness. The first method is defined by Eq. 22, and the second by Eq. 23. \mathbf{C}_h is defined by Eq. 21. Concatenating the matrices \mathbf{A}_h and \mathbf{C}_h yields \mathbf{A}_{fus} , and concatenating \mathbf{B}_h and \mathbf{D}_h gives us \mathbf{B}_{fus} . Finally we obtain the $\mathbf{A}_{fus} \mathbf{t}_{t+1} = \mathbf{B}_{fus}$ solved by least squares.

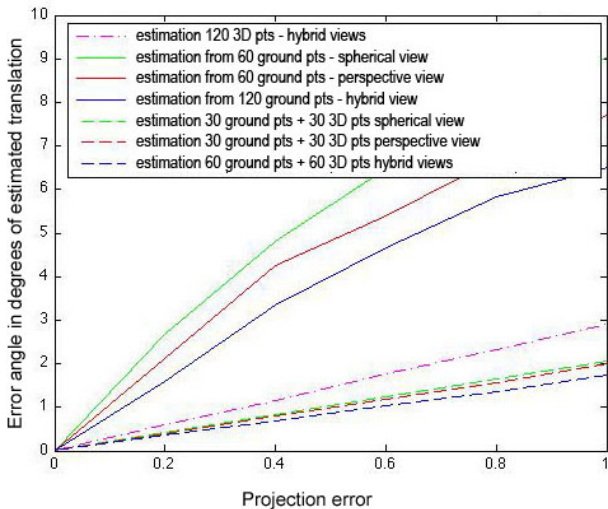


Fig. 5 Comparison of Ground Points, 3D Points and 3D + Ground Points. Dashed lines represent estimation from 3D of fused points while plain lines represent estimation from planar points. A smaller slope corresponds to a low sensitivity of the corresponding algorithm to the noise.

Figure 5 illustrates the advantages of combining 3D points and ground points. The introduced error is a random angle, applied to view-lines whichever the projection model used. The angle error is computed between the real vector of translation and the estimated vector of translation.

5.5 Kalman filtering

Once the motion has been estimated at the metric scale we observe certain discontinuities and brutal variations. In order to reduce bad estimations and to refine the trajectory, we have chosen to use a linear Kalman filter Kalman (1960). The considered state is simply the translation vector of the ego-motion i.e. $X_k = \mathbf{t}_{t+1}$, and is modeled as a linear Gaussian system given by Eq. 24:

$$\begin{aligned} X_{k+1} &= \mathbf{E}X_k + V_k \\ Y_k &= \mathbf{O}X_k + W_k \end{aligned} \quad (24)$$

where $\mathbf{E} = \mathbf{I}d_3$ is the linear state transition model and $\mathbf{O} = \mathbf{I}d_3$ the observation model, assuming constant speed during a sample interval. The vectors V_k and W_k correspond respectively to the error model and the observation noise. They are assumed to be additive and white zero-mean Gaussian, with user-defined covariance matrices. In order to retain the dynamic nature of the measurement, more uncertainty is given to the measures, i.e. 10^6 and 10^2 for the model. Given the above considerations, the Kalman filter consists in predicting the translation vector \mathbf{t}_{t+1} and then obtaining a refined value using an update step when a new observation is available. We therefore obtain the translation vector of the ego-motion and its estimated accuracy, from all past observations up to the current time.

6 Results

In this section we present the results of the different methods. First, a ground truth comparison is used to validate the effectiveness of each method and also its accuracy, then each method is tested on UAV image sequences. Finally, we analyze the performance of the algorithm.

6.1 Attitude

First, attitude can be provided either by vision or by IMU. For the case of vision, two types of methods are analyzed: in urban environments and natural environments. Results are compared to an IMU.

Attitude estimation by vanishing points

We employ the method of Demonceaux et al. (2007) and Bazin et al. (2008). The error introduced by this method does not exceed 3° . Our algorithm is insensitive to low attitude errors. For example, with real images, an error of 5° for the roll and the pitch will introduce respectively an error of 0.87% and 0.39% (fig. 6(b))

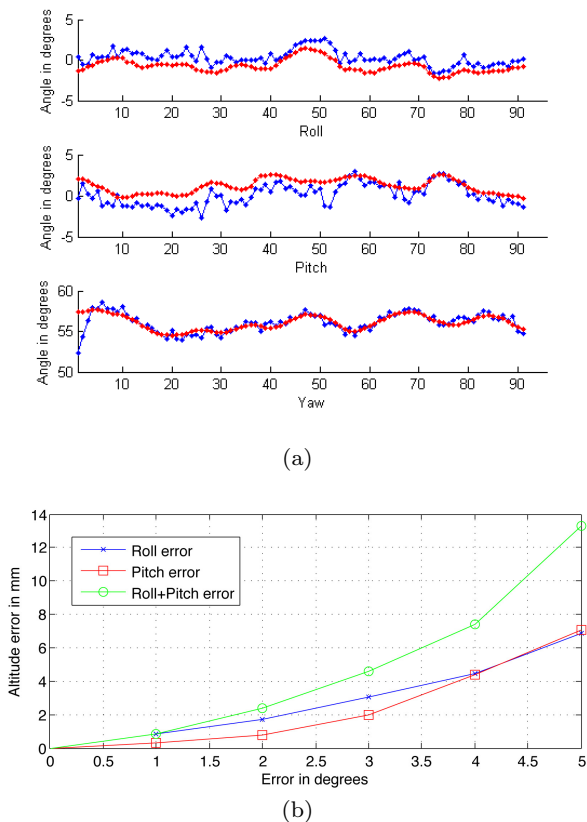


Fig. 6 (a) Comparison between attitude estimated by IMU (red) and by vision (blue). (b) Influence of attitude error (pitch and roll) on the estimation of altitude.

for altitude estimation. This method, which has been tested on our fisheye lens, is as good as a catadioptric lens at estimating attitude. In the works presented here, attitude can be estimated in real time (around 30 Hz).

Indoor experiments were carried out using a Pelican UAV, where vanishing points can be determined. Fig.6(a) and Table 1 present the results from vision-based method (blue line) and IMU (red line). Notice that the red curve is smoothed by the IMU Kalman filter. Finally, for the yaw, the absolute offset between the vision estimation and the IMU is removed by : $y_{i,vp} = y_{i,vp} - \bar{y}_{vp} + \bar{y}_{imu}$, with $y_{i,vp}$ the i^{th} value estimated by vanishing points, \bar{y}_{imu} the mean of yaw angles estimated by IMU.

Type	Roll	Pitch	Yaw
Error max	3.97°	2.43°	5°
Mean of absolute error	1.32°	1.96°	0°
Std of error	1.65°	2.07°	0.78°

Table 1 Attitude Comparison for IMU vs. the Vision Method

Attitude estimation from the horizon line

We compared the attitude provided by horizon line detection to IMU. For the image in Fig.7(a) we obtain an error of less than 1°. Fig.7(b) represents the ground/sky segmentation after detection of the horizon line. The C++ real-time algorithm provides an average of 80Hz after the first initialization, on offboard processor.

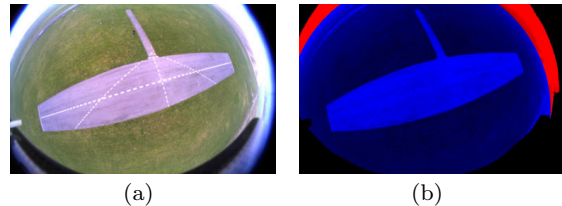


Fig. 7 (a) View from the UAV's fisheye. (b) Segmented ground/sky.

6.2 Altitude

Once the calibration and attitude have been estimated we proceed to estimate the altitude. We compare our algorithm to a laser telemeter, and UAV altitude estimation validates the effectiveness of this approach.

6.2.1 Comparison with the ground truth

We present two cases of experimental results where real altitude is estimated by a laser telemeter. The error ϵ between actual and estimated altitude is expressed as $\epsilon = \frac{(\hat{d} - d_{gt})}{d_{gt}}$, with d_{gt} the ground truth of the altitude.

- Case 1: two cameras with a 447mm baseline are fixed on a pneumatic telescopic mast. Altitude and ground plane estimation are performed offline on a GPU.
- Case 2: two micro cameras with a 314mm baseline are embedded on a compact UAV. Altitude estimation is performed online by CPU processing.

In the first experiment we observe an accurate estimation of altitude on a free ground plane (tab. 2), with an error between 0.18% and 3.14%. When there are obstacles on the ground plane, we observe a higher error, between 7.52% and 8.82%. The higher the altitude, the less accurate the estimation because of the decrease in resolution as altitude increases. Moreover, the accuracy depends on the size of the baseline. This fact does not impact our application negatively, given that accuracy is most critical during the two phases of

landing and take-off, i.e. where the UAV is close to the ground plane.

Type	Grnd truth	Estim. altitude	Error
Free Ground	2187mm	2200mm	0.59%
Free Ground	3244mm	3250mm	0.18%
Gnd + obstacles (low contrast)	3244mm	3488mm	7.52%
Free Ground	4072mm	4200mm	3.14%
Free Ground	5076mm	5202mm	2.48%
Gnd + obstacles	4080mm	4440mm	8.82%

Table 2 Altitude ground plane estimation with and without obstacles - Algorithm parameters for this test: $s = 6$, $thres = 25$

The second aspect of our algorithm is the segmentation of the ground plane, which is well estimated for contrasted areas. In case of a plane without obstacles, the pneumatic telescopic mast, where cameras are fixed, is well represented by outliers (dark on the image) (Fig. 8(c)). For an image composed of a dominant ground plane and walls, the ground plane is segmented as inliers, while walls are segmented as outliers. Our algorithm allows inliers/outliers to be segmented globally when estimating the dominant ground plane for our application.

The aspect of our algorithm in greatest need of improvement is where there are poorly textured planes and reflections which depend on camera orientation. When the ground plane or outliers (walls, objects) are homogeneous or poorly textured, the segmentation of outliers/inliers becomes difficult.

6.2.2 UAV Altitude Estimation

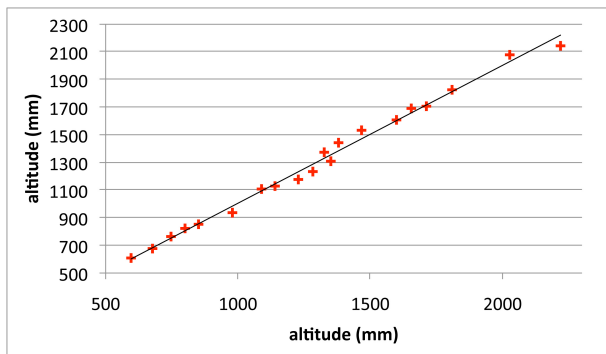


Fig. 9 Altitudes (cross) are estimated by plane-sweeping and the line represents measurements from the laser range finder based on himself, thus ideal measures.

For the second experiment we implemented our system on a small quadrotor UAV. Micro cameras embed-

ded on the UAV are linked to an external laptop to perform online altitude estimation. We tested the accuracy by comparing altitudes estimated by plane-sweeping to altitudes estimated by laser telemeter. Fig.9 shows that altitude is well estimated for the range of altitudes corresponding to the landing and take-off phases of a UAV. The mean error is 2.41%.

6.2.3 UAV Ground Plane Segmentation

By estimating the altitude from the two views (Fig. 10(a),10(b) and the homography (Fig. 10(c)), the ground plane (Fig. 10(e)) is segmented using the error (Fig. 10(d)) and becomes the mask \mathbf{G} . Figure 10(e) shows the good segmentation of planar (white pattern) and non-planar objects (cables in black) after the threshold.

6.2.4 Performances

First we developed the algorithm on GPU with brook+ for ATI that gives a real-time (30 Hz) frame rate in order to estimate altitude and to segment the ground plane. The algorithm was tested on ATI 4850 with an E8400 3Ghz CPU.

We then implemented the algorithm on a CPU with a good deal of optimizing and without segmenting the ground plane. With this implementation we obtained $min : 80 \text{ Hz}$, $mean : 180 \text{ Hz}$, $max : 250 \text{ Hz}$, that is to say values higher than the video frame rate, which meant that we could use our algorithm for online processing. The platform for our tests is a Macbook Pro with a C2D P8400 2.26Ghz CPU. A demonstration can be seen at Eynard (2010). We use a stereo rig with uEye cameras and get the normal of the ground plane with an IMU. During this demonstration, the system is able to estimate altitude in real time with robustness and accuracy.

An embedded version of our algorithm was exported to the ARM of a Gumstix Overo Fire with OMAP3530 ARM @600Mhz based processor. With this implementation we get a frame rate of around 5 Hz, which is not enough for real-time applications, but relatively interesting given the power/size ratio.

By developing these algorithms on GPU, CPU and an embedded board we obtained interesting results. For simultaneous ground plane segmentation and altitude estimation, results are real time and can be implemented on a UAV with GPU. For altitude estimation only, computation is faster and can be implemented on smaller quadrotor UAV.

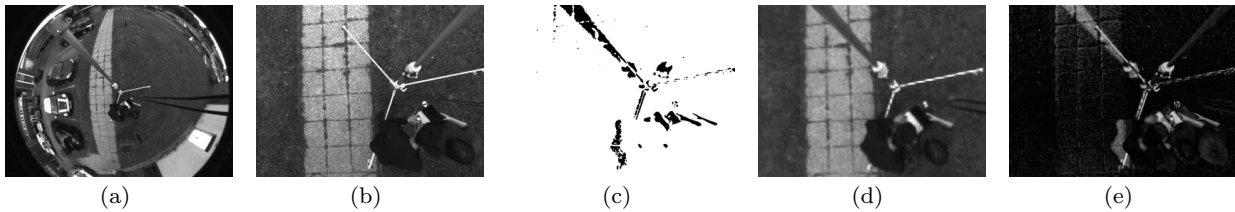


Fig. 8 Altitude and ground plane segmentation - 4.8% of inliers - Fisheye view (a), perspective view (b), ground plane segmentation (c), sphere to plane homography (d), reference and homography comparison (e)

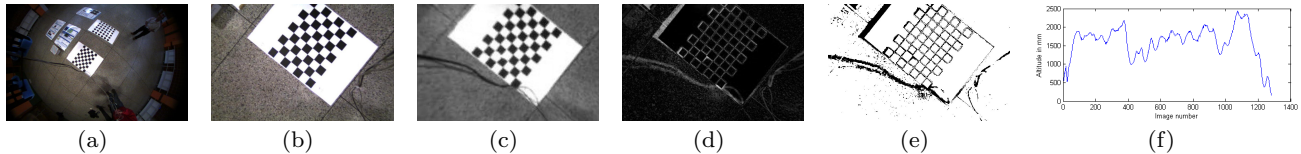


Fig. 10 (a) embedded view of the Pelican UAV (from left to right: fisheye and perspective views) - Est. altitude 1912mm. (b) Homography of fisheye view to the reference (c) Error (d) Ground plane segmentation \mathbf{G} (e) Altitude of the trajectory estimated by plane-sweeping.

6.3 Motion

We propose estimating motion using a linear method, namely least squares (LS). This method is robust to Gaussian noise, but sensitive to outliers Malis and Marchand (2005). For this reason, outliers are rejected by the RANSAC method Hartley and Zisserman (2004). This method was first validated on robot arm performing a loop, as we now describe.

6.3.1 Comparison with the ground truth

The system is presented in Fig.11(b), and images are processed offline. In each view the number of tracked points is between 50 and 200. In 11(c), we compare our algorithm with the ground truth, up to the micrometer, obtained by the robot arm. The trajectory is a square of side 330 mm, with a total motion of 1320 mm. All trajectories are expressed within the same coordinate system of the robot arm. First we estimate the altitude by plane-sweeping (Fig. 10(f)). Then the motion is estimated using the derotation obtained from IMU (blue line) or from the ground truth (green line). In the blue case, the error is 19.084 mm in XYZ, or 1.45%. In the green case, the error is 15.865 mm in XYZ, or 1.20%.

6.3.2 UAV motion estimation

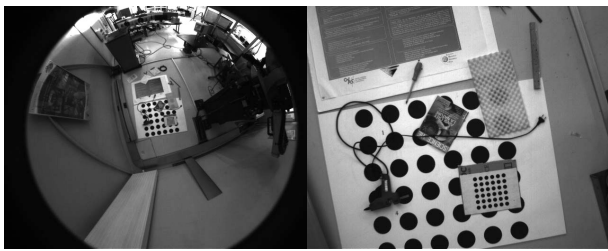
Once it had been validated, we tested our algorithm on a quadrotor with two uEye cameras with images processed offline. This validation was performed both indoors and outdoors. A Xsens IMU provides the attitude and the rotation of the motion while altitude is estimated via plane-sweeping. In each view, the number

of tracked points is between 50 and 200. We assume that the perspective camera is pointing at the ground and looking at planar points, while the fisheye observes environment points. In an indoor environment, Fig.12(a) presents the final 3D trajectory of the motion estimation. Raw data sensitive to noise and bad estimations are shown in red, while data filtered and smoothed by Kalman filter are shown in blue. In other experiment, Fig.13(a) presents the final trajectory in green, superimposed to an external view.

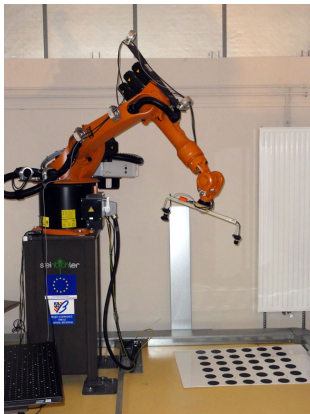
In the outdoor environment the ground texture, mainly grass, means that the plane-sweeping algorithm cannot give accurate results, because of ambiguities in the search range. The poorly textured ground, the high altitude (15 m), and the short baseline make altitude estimation difficult. In this case we hand altitude estimation over to a pressure altimeter, embedded on the UAV. Fig.14(a) shows the motion estimated by GPS filtered by the altitude provided by the pressure altimeter. We then estimate the motion using embedded IMU to estimate the attitude and the rotation of the motion.

6.3.3 Performances

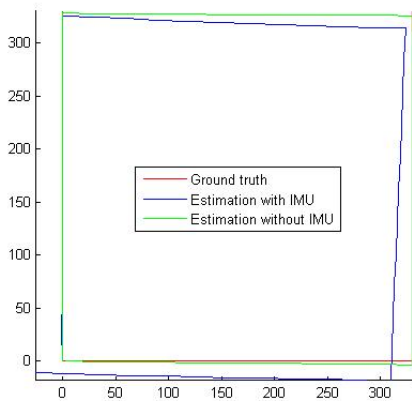
The algorithm was implemented in C++ and tested on a Macbook Pro. Estimating rotation by IMU and with tracking points on two images of 752×480 , we could estimate the translation up to 24 Hz. The Lucas Kanade tracker has been adapted Sanahuja (2010) for the OMAP 3530 DSP, that allows real-time tracking for one image per Gumstix.



(a)



(b)



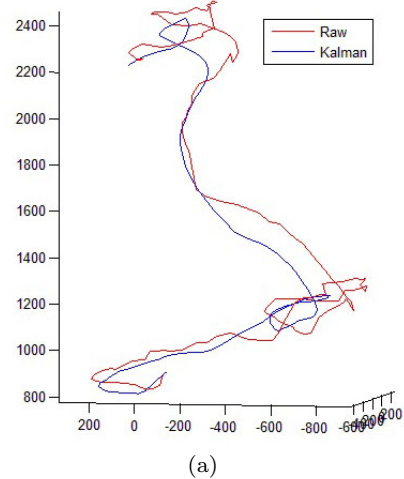
(c)

Fig. 11 Loop of a robot 11(c) with hybrid and merged views, Fig.11(a) fisheye and perspective views of the mobile robot.

7 Conclusions and future works

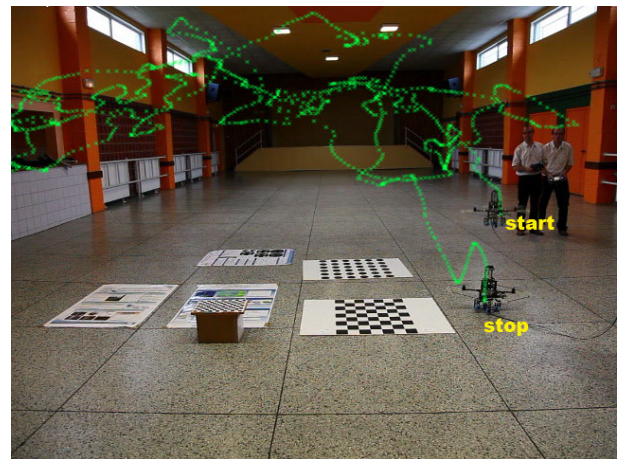
This paper presents a hybrid stereo system composed of fisheye and perspective views, modeled respectively by a sphere and a plane, which is able to estimate navigation parameters for a UAV without the use of any other sensor. Navigation parameters are attitude, altitude, ground plane segmentation and motion, assuming calibrated cameras.

Projecting the horizon line can determine the attitude, as well as vanishing points obtained from lines in



(a)

Fig. 12 3D trajectory of the UAV.

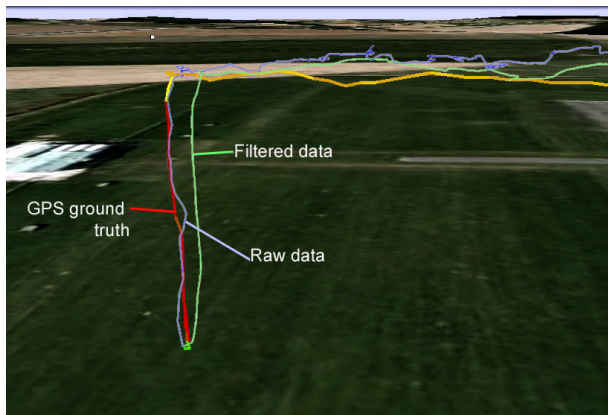


(a)

Fig. 13 Motion in indoor environment with Pelican.

an urban environment. The projections of the ground plane onto the sphere and the plane are related by a homography which allows both altitude and ground plane to be estimated using plane-sweeping. Once the altitude has been estimated and the ground plane segmented, tracked points on the ground provide the information for metric translation, while environment points give an accurate estimation of translation up to scale. By combining points from mixed views, and by merging two sets of points, translation is estimated accurately using the metric information, and smoothed by Kalman filter.

The methods we propose are real-time methods that do not require matching between hybrid views. We show in this article that our sensor is able to solve the main drawbacks of conventional sensors by using a compact, low power-consumption, passive, accurate, robust vision sensor. Furthermore, once navigation parameters have been estimated, this system is available to perform



(a)

Fig. 14 Taking-off in natural environment with MD4-1000. The red line represents GPS ground truth. The purple line is the estimated motion and green line the estimated motion filtered by Kalman filter (for more details, please refer to corresponding video).

other visual tasks such as surveillance and environment discovery.

Perspectives of this work will be the development of a new hardware architecture able to grab stereo images and provide all the parameters in parallel, as well as integrating the estimated parameters into the control loop.

8 Acknowledgments

This work is supported by the European FEDER (Fonds Européen de Développement Régional) and Région Picardie Project ALTO (Automatic Landing and Take-Off).

References

- J. Artieda, J. M. Sebastián, P. Campoy, J. F. Correa, I. F. Mondragón, C. Martínez, and M. Olivares. Visual 3-d slam from uavs. *Journal of Intelligent and Robotic Systems*, 55(4-5):299–321, 2009.
- S. Baker and S. K. Nayar. A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35:175–196, 1999. ISSN 0920-5691. URL <http://dx.doi.org/10.1023/A:1008128724364>. 10.1023/A:1008128724364.
- J. Barreto and A. H. Issues on the geometry of central catadioptric image formation. *International Conference On Pattern Recognition*, 2001.
- G. Barrows, C. Neely, and K. Miller. Optic flow sensors for mav navigation. 195:557–574, 2001.
- Y. Bastanlar, A. Temizel, Y. Yardimci, and P. F. Sturm. Effective structure-from-motion for hybrid camera systems. In *ICPR'10*, pages 1654–1657, 2010.
- J. Bazin, I. Kweon, C. Démonceaux, and P. Vasseur. Uav attitude estimation by vanishing points in catadioptric image. In *IEEE International Conference on Robotics and Automation 2008 (ICRA'08)*, pages 2743–2749, Pasadena, CA, May 2008. IEEE.
- J. Bazin, C. Démonceaux, P. Vasseur, and I. Kweon. Motion estimation by decoupling rotation and translation in catadioptric vision. *Computer Vision and Image Understanding*, Volume 114, Issue 2(0):254–273, February 2010.
- A. Beyeler, C. Mattiussi, J. christophe Zufferey, and D. Floreano. Visionbased altitude and pitch estimation for ultra-light indoor aircraft. In *in IEEE International Conference on Robotics and Automation ICRA'06*, pages 2836–2841, 2006.
- M. Blösch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based mav navigation in unknown and unstructured environments. *IEEE International Conference on Robotics and Automation (ICRA 2010)*, 2010.
- J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm, 2000. URL http://robots.stanford.edu/cs223b04/algo_tracking.pdf.
- G. Caron and D. Eynard. Multiple camera types simultaneous stereo calibration. In *IEEE Int. Conf. on Robotics and Automation, ICRA'11*, Shanghai, China, May 2011.
- J. S. Chahl, M. V. Srinivasan, and S.-W. Zhang. Landing strategies in honeybees and applications to uninhabited airborne vehicles. *I. J. Robotic Res.*, 23(2): 101–110, 2004.
- A. Cheriany, J. Andersh, V. Morellas, N. Panikolopoulos, and B. Mettler. Autonomous altitude estimation of a uav using a single onboard camera. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems, IROS'09*, pages 3900–3905, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3803-7. URL <http://portal.acm.org/citation.cfm?id=1732643.1732689>.
- R. Collins. A space-sweep approach to true multi-image matching. In *IEEE Computer Vision and Pattern Recognition*, pages 358–363, June 1996.
- A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, 2007.
- C. Démonceaux, P. Vasseur, and C. Pégard. Robust attitude estimation with catadioptric vision. In

- IEEE/RSJ International Conference on Intelligent Robots and Systems 2006 (IROS'06)*, pages 3448–3453, Beijing, China, October 2006. IEEE.
- C. Demonceaux, P. Vasseur, and C. Pégard. Uav attitude computation by omnidirectional vision in urban environment. In *IEEE International Conference on Robotics and Automation 2007 (ICRA '07)*, pages 2017–2022, Roma, Italy, April 2007. IEEE.
- C. Demonceaux, P. Vasseur, and Y. D. Fougerolle. Central catadioptric image processing with geodesic metric. *Image Vision Comput.*, 29(12):840–849, 2011.
- D. Dusha, W. W. Boles, and R. Walker. Fixed-wing attitude estimation using computer vision based horizon detection. In *12th Australian International Aerospace Congress*, pages 1–19, Melbourne Australia, 2007. URL <http://eprints.qut.edu.au/6852/>.
- D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart. Vision based position control for mavs using one single circular landmark. *Journal of Intelligent and Robotic Systems*, 61(1-4):495–512, 2011.
- D. Eynard, 2010. URL <http://www.youtube.com/watch?v=ubXzf0eLud4>.
- D. Eynard, P. Vasseur, C. Demonceaux, and V. Fremont. Uav altitude estimation by mixed stereoscopic vision. In *IEEE Int. Conf. on Intelligent RObots and Systems, IROS'10*, Taipei, Taiwan, October 2010.
- D. Eynard, P. Vasseur, C. Demonceaux, and V. Fremont. Uav motion estimation using hybrid stereoscopic vision. In *IAPR Conf. on Machine Vision Applications, MVA '11*, Nara, Japan, June 2011.
- F. Fraundorfer, D. Scaramuzza, and M. Pollefeys. A constricted bundle adjustment parameterization for relative scale estimation in visual odometry, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509733.
- D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions, 2007.
- P. J. Garcia-pardo, G. S. Sukhatme, and J. F. Montgomery. Towards vision-based safe landing for an autonomous helicopter. 2000.
- W. E. Green, P. Y. Oh, K. Sevcik, and G. Barrows. Autonomous landing for indoor flying robots using optic flow. In *in ASME International Mechanical Engineering Congress and Exposition*, pages 1347–1352, 2003.
- R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- Hyscas. Hybrid stereoscopic calibration software. In <http://www.hyscas.com>, 2011.
- M. Kalantari, A. Hashemi, F. Jung, and J.-P. Guédon. A new solution to the relative orientation problem using only 3 points and the vertical direction. *CoRR*, abs/0905.3964, 2009.
- M. Kalantari, A. Hashemi, F. Jung, and J.-P. Guédon. A new solution to the relative orientation problem using only 3 points and the vertical direction. *Journal of Mathematical Imaging and Vision*, 39(3):259–268, 2011.
- R. E. Kalman. A new approach to linear filtering and prediction problems. 1960. URL <http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>.
- G. H. Lee, M. Achtelik, F. Fraundorfer, M. Pollefeys, and R. Siegwart. A benchmarking tool for mav visual pose estimation. In *ICARCV*, pages 1541–1546. IEEE, 2010.
- G. H. Lee, F. Fraundorfer, and M. Pollefeys. Mav visual slam with plane constraint. In *ICRA*, pages 3139–3144, 2011.
- J. Lee, S. You, and U. Neumann. Large motion estimation for omnidirectional vision. *Omnidirectional Vision, Workshop on*, 0:161, 2000. doi: <http://doi.ieeecomputersociety.org/10.1109/OMNVIS.2000.853824>.
- M. Lhuillier. Automatic scene structure and camera motion using a catadioptric system. *Computer Vision and Image Understanding*, 109:186–203, 2008. doi: 10.1016/j.cviu.2007.05.004.
- H. Li, R. I. Hartley, and J.-H. Kim. A linear approach to motion estimation using generalized camera models. In *CVPR'08*, pages –1–1, 2008.
- M. Lourenço, J. P. Barreto, and A. Malti. Feature detection and matching in images with radial distortion. In *ICRA*, pages 1028–1034, 2010.
- E. Malis and E. Marchand. Méthodes robustes d'estimation pour la vision robotique. In *Journées nationales de la recherche en robotique, JNRR'05*, Guidel, France France, 2005. URL <http://hal.inria.fr/inria-00351893/en/>.
- E. Marchand and F. Chaumette. Virtual visual servoing: A framework for real-time augmented reality. In *EUROGRAPHICS 2002 Conference Proceedings*, volume 21(3), pages 289–298, Saarebrün, Germany Germany. URL <http://hal.inria.fr/inria-00352096/en/>.
- C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, April 2007.
- C. Mei, S. Benhimane, E. Malis, and P. Rives. Homography-based tracking for central catadioptric cameras. In *IN IROS*, 2006.

- D. Mellinger, N. Michael, M. Shomin, and V. Kumar. Recent advances in quadrotor capabilities. In *ICRA*, pages 2964–2965, 2011.
- J. M. M. Montiel and D. A.-J. A visual compass based on slam. In *In Proc. Intl. Conf. on Robotics and Automation, 2006(accepted)*. Available: <http://pubs.doc.ic.ac.uk/visual-compass-slam>, 2006.
- O. Naroditsky, X.-S. Zhou, J. Gallier, S.-I. Roumeliotis, and K. Daniilidis. Two efficient solutions for visual odometry using directional correspondence. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 2011. URL <http://www.cis.upenn.edu/~kostas/mypub.dir/oleg2011pami-revised.pdf>. Under review.
- D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.
- G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *Journal of Intelligent and Robotic Systems*, 61(1-4):287–299, 2011.
- O. Oreifej, N. da Vitoria Lobo, and M. Shah. Horizon constraint for unambiguous uav navigation in planar scenes. In *ICRA*, pages 1159–1165, 2011.
- R. Pless. Using many cameras as one. In *In CVPR*, pages 587–593, 2003.
- L. Puig, J. Guerrero, and P. Sturm. Matching of omnidirectional and perspective images using the hybrid fundamental matrix. In *Proceedings of the Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras, Marseille, France, oct 2008*. URL <http://perception.inrialpes.fr/Publications/2008/PGS08>.
- G. Sanahuja. Commande et localisation embarquée d’un drone aérien en utilisant la vision, January 2010. URL http://www.hds.utc.fr/~gsanahuj/dokuwiki/lib/exe/fetch.php?id=frAccueil&cache=cache&media=fr:these_guillaume_sanahuja.pdf.
- S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2799–2804, 2002.
- D. Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*, pages 1–12, 2011. ISSN 0920-5691. URL <http://dx.doi.org/10.1007/s11263-011-0441-3>. 10.1007/s11263-011-0441-3.
- D. Scaramuzza and R. Siegwart. Correcting vehicle heading in visual odometry by using image appearance. In *Proc. of The First International Workshop on Omnidirectional Robot Vision*, November 2008.
- C. S. Sharp, O. Shakernia, and S. S. Sastry. A vision system for landing an unmanned aerial vehicle. In *IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea*, pages 1720–1727, 2001.
- S. Thurrowgood, D. Soccol, R. J. D. Moore, D. P. Bland, and M. V. Srinivasan. A vision based system for attitude estimation of uavs. In *IROS*, pages 5725–5730. IEEE, 2009.
- S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments. *J. Field Robotics*, 28(6):854–874, 2011.
- X. Ying and Z. Hu. Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, volume 3021 of *Lecture Notes in Computer Science*, pages 442–455. Springer Berlin / Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-24670-1_34. 10.1007/978-3-540-24670-1_34.