

UAV Attitude Computation by Omnidirectional Vision in Urban Environment

Cédric Demonceaux, Pascal Vasseur and Claude Pégard

Abstract—Attitude is one of the most important parameters for a UAV during a flight. Attitude computation methods based on vision generally use the horizon line as reference. However, the horizon line becomes an inadequate feature in urban environment. We then propose in this paper an omnidirectional vision system based on straight lines (very frequent in urban environment) that is able to compute the roll and pitch angles. The method consists in finding bundles of horizontal and vertical parallel lines in order to obtain an absolute reference for the attitude computation. We also develop here a new and efficient method for line extraction and bundle of parallel line detection. An original method of horizontal and vertical plane detection is also provided. We show experimental results on different images extracted from video sequences.

I. INTRODUCTION

The attitude of an autonomous or tele-operated UAV is a very important parameter during a flight. However, inertial sensors which are useful for this measurement, are sensitive to drift and expensive if a high accuracy is required. The use of vision sensors becomes then an interesting option which allows not only to compute the attitude but also to perform higher level tasks (see figure 1(a)). In this case, most of the proposed approaches are based on the extraction of the horizon line from the image in order to compute the roll and pitch angles [4] [5] [6] [11] [13]. Thus, in [6] and [11], the authors use a perspective camera and the horizon is considered as a straight line. They define a criterion based on color and on texture which has to be optimized in order to separate the ground from the sky. In [14], an infrared camera is used in order to find the horizon. With this sensor, the authors are able to treat images in which steep reliefs are present. In [13], a stereoscopic system detects a known beacon on the ground for a complete pose determination. However, such an approach radically limits the evolution space of the UAV. In [4] and [5], attitude computation is performed from catadioptric images. These images present three advantages in comparison with perspective images. First, the horizon line is always present in the image, even if partially. This characteristic ensures a possible estimate for every image. Second, the horizon can be detected accurately even in the presence of occultations. Finally, contrary to the perspective case which supposes that the altitude is known, the pitch can be estimated without prior knowledge. In this way, the attitude can be completely and accurately

computed with a catadioptric image. In both approaches, the estimation of the angles is performed according from the horizon line extracted in [4] thanks to a robust method based on color difference between the sky and the ground and in [5] thanks to an optimization function also based on the color. However, all these methods provide good results only in a free space where the horizon is largely visible. Indeed, in the case of a low altitude flight in urban environment, you quickly realize that these methods are inadequate because the horizon is almost non-existent, as shown in figure 1(b). Thus, in [9] the authors propose to use accelerometers in association with omnidirectional vision in order to compute attitude. Another approach without any addition of supplementary sensors consists in using adequate features for urban environment such as straight lines. Indeed, if the system is able to detect horizontal and vertical lines in the image, it is then able to compute roll and pitch angles. The use of a perspective camera permits a fast detection of lines but does not ensure the collect of enough significant information for an accurate estimation. In this way, we propose in this paper a system based on a central catadioptric sensor which displays several advantages for our problem. First, the field of view ensures the presence of a large number of lines. Next, geometric characteristics of central sensors allow to compute directly and accurately the roll and the pitch by using the sphere of equivalence [7]. Finally, the sphere of equivalence allows to find horizontal and vertical planes which are useful for the attitude computation. The use of omnidirectional vision in urban environment for a six degree of freedom robot is not really new. Indeed, in [8], the authors propose an omnidirectional sensor associated with two perspective cameras for navigation in a urban canyon. In [3], an algorithm of structure from motion based on parallel lines is proposed. The approach presented in this paper contains two distinct contributions. First, we propose a new method of line detection valid for any calibrated central catadioptric sensor. We show that this method allows a fast and accurate detection of the lines in the image. This permits to subsequently detect bundles of parallel lines by using antipodal vanishing points. The second contribution concerns the vertical and horizontal planes of the scene thanks to the detection of the sky in the image. Finally, the estimation of the attitude is naturally deduced from these planes.

In the following of the paper, we first describe the line detection algorithm with a presentation of the image formation model that we use. In section III, the detection of bundles of parallel lines is presented. Horizontal and vertical planes extraction is described in section IV with a new sky

This work has been supported by GIP-ANR, CaViAR Project
C. Demonceaux, P. Vasseur and C. Pégard are with Robotics, Power Engineering and Control Center (CREA - EA 3299), University of Picardie Jules Verne, 7 Rue du Moulin Neuf, 80000 Amiens, France
FirstName.Name@u-picardie.fr

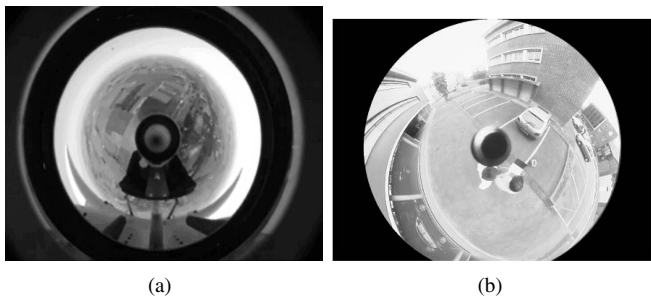


Fig. 1. Omnidirectional images captured from a UAV, (a) High altitude (horizon line is largely visible), (b) Low altitude in urban environment (horizon line is almost non-existent).

detection algorithm and the attitude computation. We present experimental results in section V followed by a conclusion and perspectives in section VI.

II. LINE DETECTION

A. Introduction

Line detection in catadioptric images is not a trivial problem. Indeed, the projection of any 3D real line is a conic that may be degenerate. Thus, in the case of an uncalibrated sensor, it is necessary to estimate five parameters for each line while only two parameters are necessary for a calibrated sensor. Nevertheless, even in the calibrated case the detection is still difficult and requires important computation times. Thus, in [12] and [15], the authors propose a detection based on an adaptation of the Hough transform. These methods permit a good detection but with important computation times for our application. In [1], the approach, valid uniquely for paracatadioptric sensors, consists in determining the conic equation in giving only two points on the line. This method, useful in the uncalibrated case, does not present possible extensions to the other central sensors and does not propose any automatic selection of the two initial points. In this paper, our method is valid for any central catadioptric sensor and also uses two points for each line. However, the selection of these points is totally automatic and the detection is based on a prediction/verification algorithm.

B. Algorithm Overview

The algorithm is composed of four main steps which respectively consist in detecting the edges in the image, in chaining the connected edge pixels, in selecting significant chains and finally in verifying if each chain is the projection of a real 3D line. Edge detection is performed by a Canny edge detector and we only keep chains of edgels (pixels of the edges) with a length greater than $NbPixels$. The third step aims to verify if the detected chains do not contain ruptures of direction. In this way, we compute the difference of gradient orientation for each pair of connected pixels of each chain. If two consecutive pixels have a difference of orientation greater than a threshold, the chain is then divided. The final aim is to obtain chains which are probably the projection of a single 3D edge. The last operation consists then in verifying which chains are the images of 3D lines.

C. Line Projection Model

As mentioned previously, we suppose that the sensor is calibrated. To do this, we use the method proposed in [2]. Thus, we first propose in this section the image formation model that we use in our algorithm. The projection of a 3D point is performed by the way of the unitary sphere (figure 2). Thus, we define oriented projective ray P_1 passing by 3D point x_w and the center of the sphere. This ray intersects the surface of the sphere in x_s . We then consider oriented projective ray P_2 passing by x_s and a point situated on the z-axis between the center of the sphere and the north pole. This point is at distance ξ from the center of the sphere and only depends on the mirror geometric characteristics. P_2 intersects the plane at infinity in point x_i . Finally, homography H defined between the plane at infinity and the catadioptric image plane projects point x_i into point x_c . H includes intrinsic parameters of the camera, possible rotations between the sphere frame and the camera frame, and finally the parameters of the mirror. According to this model, we can develop the projection of a 3D line into the catadioptric image plane (figure 3). We consider plane Π_R which contains the real 3D line and the center of the sphere. This plane intersects the sphere and then defines a great circle onto its surface. The set of oriented projective rays passing through the points of the great circle and point O_2 then define a cone which intersects the plane at infinity into conic C_i . Finally, homography H transforms C_i into conic C_c in the catadioptric image plane. In the plane at infinity, we know that the equation of conic C_i is equal to :

$$C_i = \begin{bmatrix} n_x^2(1 - \xi^2) - n_z^2\xi^2 & n_x n_y(1 - \xi^2) & n_x n_z \\ n_x n_y(1 - \xi^2) & n_y^2(1 - \xi^2) - n_z^2\xi^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{bmatrix} \quad (1)$$

with (n_x, n_y, n_z) the vector, which describes the normal to plane P_w which contains the 3D line. We obtain the equation of conic C_c in image plane thanks to the following relation :

$$C_c = H^{-T} C_i H^{-1} \quad (2)$$

It appears that if the normal to the plane which contains the 3D line is known, we can easily obtain the equation of the conic in the image plane. Thus, if two points of the great circle are known, we obtain the normal vector by the cross product of these two points expressed in frame O_1 . Since the sensor is calibrated, any point of the image plane can be back-projected on the sphere by :

$$x_i = H^{-1} x_c \quad (3)$$

and then by computing the intersection between the sphere and the projective ray defined by x_i and the center of the sphere.

The algorithm of line detection then consists in selecting two significant points for each chain of edges and in back-projecting them onto the sphere. Two points are considered as significant if they are close to the endpoints of the chain

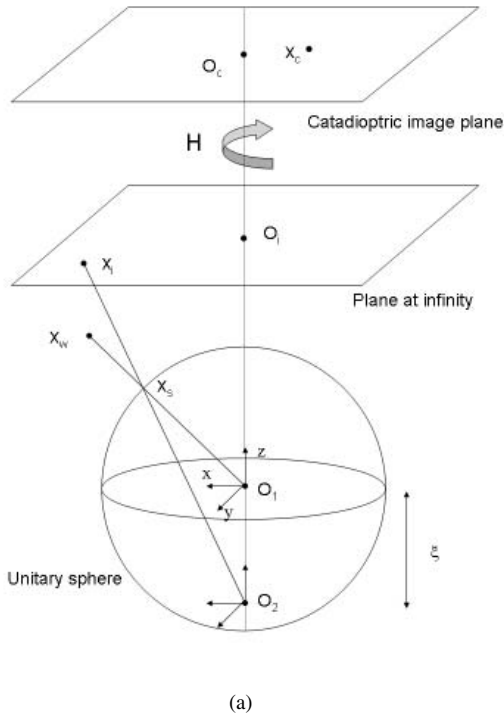


Fig. 2. Image formation model. Example of projection via the unitary sphere for a 3D point.

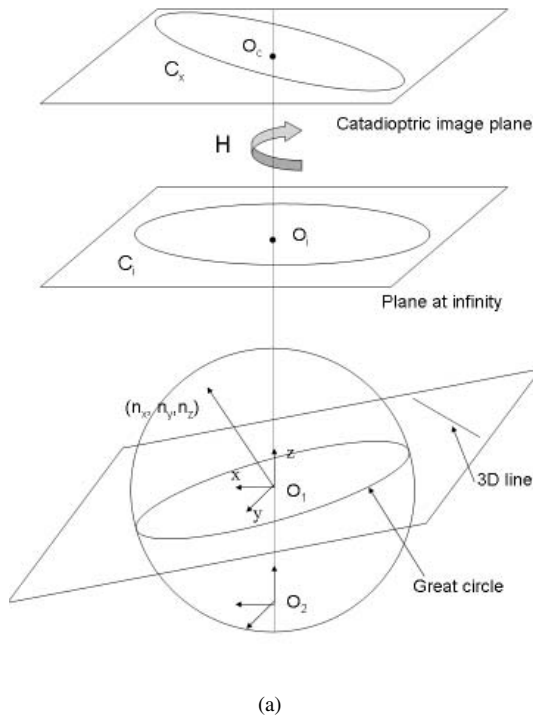


Fig. 3. Projection of a 3D line via the unitary sphere into the catadioptric image plane.

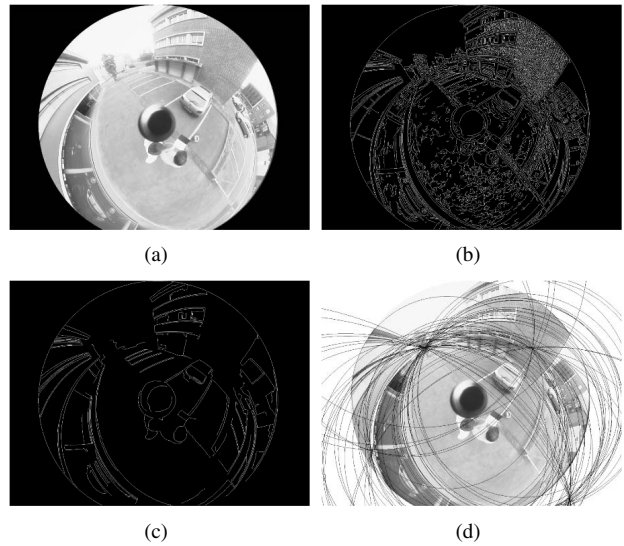


Fig. 4. (a) Original image, (b) edge detection, (c) chains of edgels (d) detected lines

in avoiding the endpoints themselves because they are more sensitive to noise. Once the points have been projected on the sphere, we compute the normal of the plane including the 3D real line. The equation of the conic into the plane at infinity can then be computed with equation 1 and finally the equation of the conic into the image is obtained by equation 2. In fact, this step corresponds to the prediction of the position of the projection of the 3D line into the image. The verification then consists in counting the number of edgels of the chain which belong to the predicted conic. In this way, we consider that an edgel is on the conic if the error computed by $x_c^T C_x x_c$ is less than a tolerance. The chain is considered as the projection of a 3D line if and only if at least 90% of its edgels verify the previous tolerance. Figure 4 presents different steps of the detection algorithm.

III. PARALLEL LINE BUNDLE DETECTION

Once the lines have been detected in the image, we have to determine which lines are respectively horizontal and vertical. In this way, we first aim to find bundles of parallel lines and we use the following property :

Property : If d_1 and d_2 are two parallel lines with unit vector \vec{u} , then their equivalent great circles C_1 and C_2 on the unitary sphere intersect into two antipodal points I_1 and I_2 . These points have the property of being independent from the position of d_1 and d_2 in the 3D scene and only depend on direction \vec{u} . We then have $\vec{I}_1 \vec{I}_2 = \vec{u}$ (figure 5).

Proof : Consider P_1 and P_2 , planes which contain respectively d_1 and d_2 and also O_1 . It is obvious that these planes intersect into line d which includes O_1 . This demonstrates that two great circles C_1 and C_2 intersect into two antipodal points I_1 and I_2 .

Let note \vec{n}_1 (respectively \vec{n}_2) be the normal of plane P_1 (respectively P_2). However, $d_1 \in P_1$ and $d_2 \in P_2$, then $\vec{n}_1 \cdot \vec{u} = \vec{n}_2 \cdot \vec{u} = 0$ and $\vec{u} = \vec{n}_1 \times \vec{n}_2$. In the same

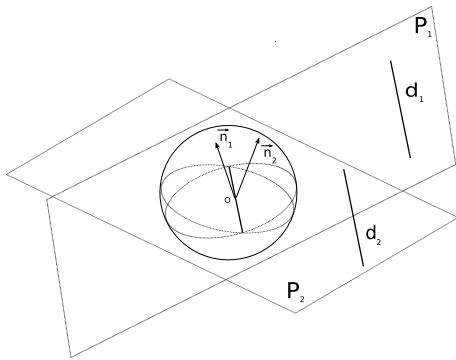


Fig. 5. Parallel line projection onto the surface of the unitary sphere.

way, $\overrightarrow{I_1 I_2} \in P_1 \cap P_2$, then $\overrightarrow{I_1 I_2} \cdot \overrightarrow{n_1} = \overrightarrow{I_1 I_2} \cdot \overrightarrow{n_2} = 0$ and $\overrightarrow{I_1 I_2} = \overrightarrow{n_1} \times \overrightarrow{n_2} = \overrightarrow{u}$.

This property has then two consequences :

- The set of parallel lines with unit vectors \overrightarrow{u} provides a set of great circles which intersect into two antipodal points I_1 and I_2 , and $\overrightarrow{I_1 I_2} = \overrightarrow{u}$.
- Consider two bundles of parallel lines with unit vector \overrightarrow{u} and \overrightarrow{v} . Notice I_1^i and I_2^i the intersection points of each bundle ($i = 1, 2$). If these two bundles are perpendicular, we obtain $\overrightarrow{I_1^1 I_2^1} \cdot \overrightarrow{I_1^2 I_2^2} = 0$

Thus, in order to detect bundles of parallel lines, we use an exhaustive algorithm which computes the intersection points of each pair of lines. If at least three lines have the same intersection points, we consider them as a bundle. Finally, we verify which bundles are perpendicular (figure 6).

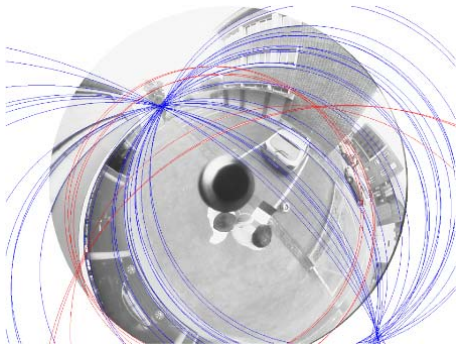


Fig. 6. Two orthogonal bundles of parallel lines.

In the best case, if the scene is made of several buildings such as in figure 7, we obtain three perpendicular bundles. Thanks to their unit vectors $\overrightarrow{u_1}, \overrightarrow{u_2}, \overrightarrow{u_3}$, these bundles define an orthogonal frame with respectively one vertical direction and two horizontal directions. Otherwise, if only two perpendicular bundles are detected, we obtain the frame with $\overrightarrow{u_1}, \overrightarrow{u_2}, \overrightarrow{u_3} = \overrightarrow{u_1} \times \overrightarrow{u_2}$.

The attitude can then be estimated thanks to this orthogonal frame. However, it is necessary to determine which vector of this frame represents the vertical unit vector beforehand. This is the aim of the next section.

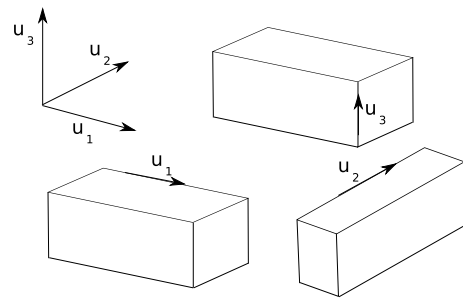


Fig. 7. Scene with three buildings and the obtained orthogonal frame.

IV. ATTITUDE COMPUTATION

Once the orthogonal bundles of the 3D scene are obtained, we can not differentiate the vertical and horizontal directions. However, in order to estimate the attitude of the UAV, it is necessary to know the vertical direction in the scene. For this purpose, we propose to detect the sky in the image in order to find the vertical line bundle.

A. Sky detection

The sky detection in the images is a subject which has often been treated for outdoor robotics [9] but also for image orientation [10]. Most of these systems use color and/or texture information in order to manage the different appearances of the sky (clear, gray, cloudy, ...). In our approach, in order to obtain a fast algorithm, we propose to only use the brightness information. With that aim, we suppose that the sky represents the brightest part of the image, which is a reasonable assumption. The second suggested hypothesis is that the sky is almost always in the part close to the omnidirectional image border. Except if the camera is turned upside down, this assumption is always true and we will see that even this case can be easily resolved. The algorithm can then be described in the following way:

- 1) Finding the point of maximum brightness in the image.
- 2) Considering a circle centered on the image center and with a radius $R_{max} - 5$, where R_{max} is the radius of the omnidirectional image perimeter (see figure 8(a))
- 3) Normalizing by the maximum brightness, the brightness of the pixels which belong to this circle.
- 4) Binarizing the brightness of the pixels on the circle respecting a threshold equal to 0.95 in order to preserve only the points of high brightness (see figure 8(b)).
- 5) Eliminating arcs of circle with a length lower than 10 pixels having a binarized brightness equal to 255.
- 6) Seeking the longest arc of circle whose points have a null binarized brightness. This arc of circle is considered as the part without sky. The complementary

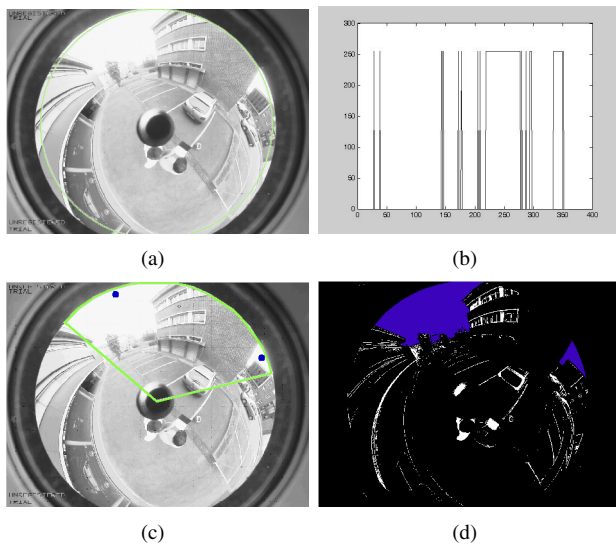


Fig. 8. (a) Original image with the circle of points used for sky detection, (b) binarized histogram of the circle, (c) probable position of the sky with the initial seeds for the region growing (d) detected sky after region growing process

arc of circle is then considered as the image part where the sky is located (see figure 8(c)).

- 7) Using a region growing process by using initial seeds determined from the complementary arc of circle previously detected (see figure 8(d)).

In the case where the sky is not detected, we apply the same algorithm in considering an inner circle, rather than an outer circle as described previously. However, this happens only when the UAV is flying upside down.

B. Horizontal Plane Detection

The trihedron $\vec{u}_1, \vec{u}_2, \vec{u}_3$ permits to define 3 planes, \mathcal{P}_{ij} which contain O_1, \vec{u}_i and $\vec{u}_j, i \neq j$. These 3 planes define 3 different partitions of the sphere.

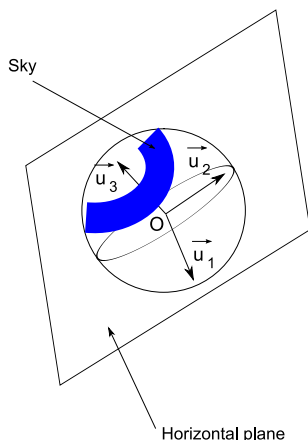


Fig. 9. Horizontal plane detection.

The horizontal plane is in fact plane \mathcal{P}_{ij} which partitions the spherical image without separating points of the sky (Fig 9). However, let us note that in few cases, it can happen that two \mathcal{P}_{ij} planes can be candidate. In this case, we cannot find the horizontal plane and we need to make a tracking of this plane in the sequence in order to clear up this ambiguity.

Once the horizontal plane has been detected, the vertical direction of the scene is simply vector \vec{u}_i which is not contained in the horizontal plane.

C. Pitch and roll estimation

Finally, knowing the vertical direction of scene $\vec{N} = (n_x, n_y, n_z)$, we deduce roll ρ and pitch ψ angles as follows :

$$\psi = \frac{n_x}{|n_x|} \times \arccos\left(\frac{|n_z|}{\sqrt{n_x^2 + n_z^2}}\right) \quad (4)$$

$$\rho = \frac{-n_y}{|n_y|} \times \arccos\left(\frac{|n_z|}{\sqrt{n_y^2 + n_z^2}}\right) \quad (5)$$

V. EXPERIMENTAL RESULTS

In this section, we show results on different images taken with a paracatadioptric sensor. We present in figures 10(a), 10(b), 11(a), 11(b) the detected lines on these images. Figures 10 11, (c) to (f) present two orthogonal bundles (in red lines and in blue lines) of each scene. We can see that vertical and horizontal lines are correctly detected. Moreover, we can note that in each case, the horizontal plane is detected without any ambiguity because the sky represents an important part of the image. This property is often checked, it is one of the main advantages when we use omnidirectional sensors.

Since we do not have ground truth values of the roll and pitch angles for these images, we only propose a visual validation of the results. Nevertheless, these results show an interesting stability between the consecutive images and appear to be coherent with the images. For the first image (Fig 10 left row), we obtain $\psi = 19.7^\circ$ and $\rho = 15.2^\circ$, for the second image (Fig 10 right row), we obtain $\psi = -37.5^\circ$ and $\rho = 17.8^\circ$, third image (Fig 11 left row) $\psi = 37.5^\circ$ and $\rho = -20^\circ$. Let us note that the image (Fig 11 right row), corresponds to the sensor placed horizontally and we obtain : $\psi = 2^\circ$ and $\rho = -1^\circ$.

VI. CONCLUSION AND PERSPECTIVES

In this paper, we have developed a new method of attitude estimation for a UAV in urban environment by catadioptric vision. We propose to detect bundles of parallel lines in the image in order to detect horizontal and vertical planes. The attitude can then be easily estimated. The main contributions deal with a new line detection algorithm for calibrated central catadioptric sensor and with an original approach in order to detect the horizontal plane and the vertical direction based on the sky detection in the image. Experimental results display very promising results which have to be compared with measures obtained from an inertial sensor.

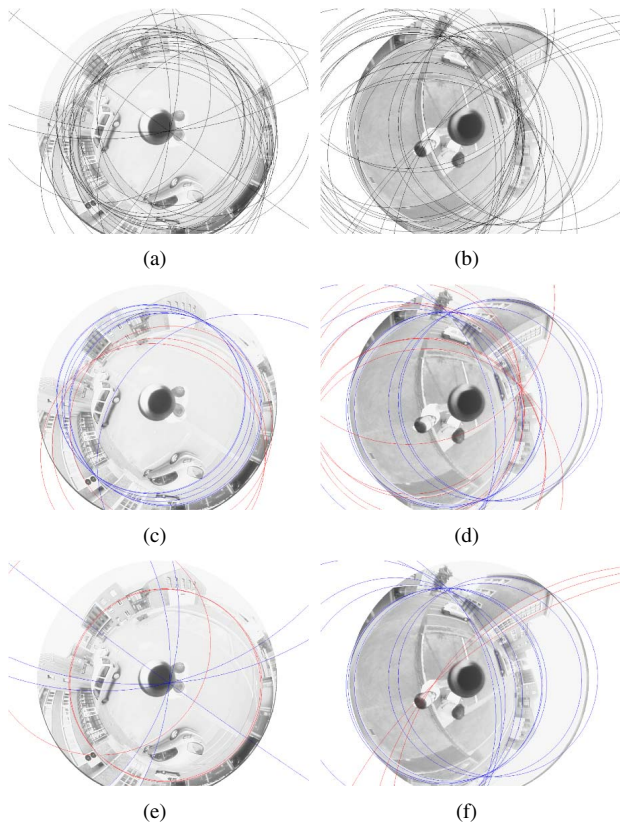


Fig. 10. Attitude estimation, left row : first image, right row : second image. (a),(b) Line detection, (c)-(f) orthogonal bundles

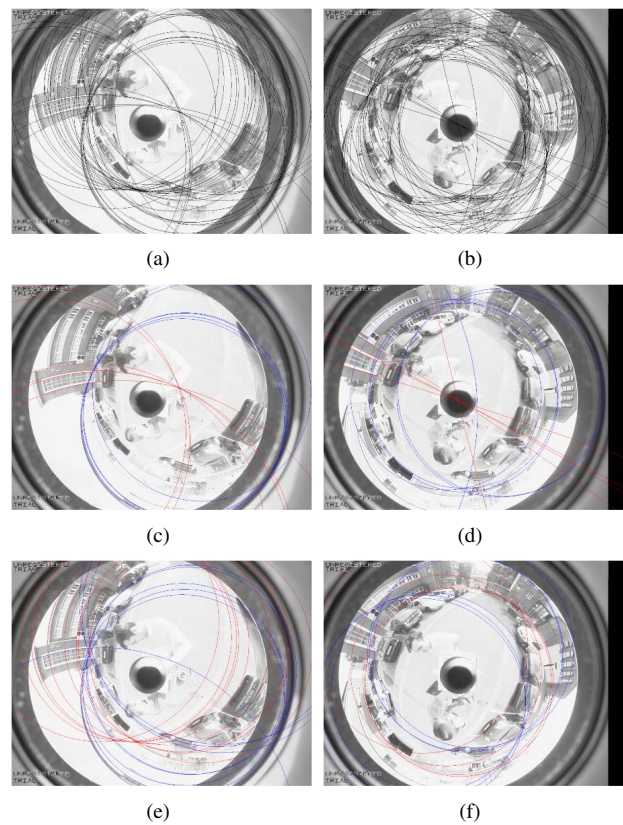


Fig. 11. Attitude estimation, left row : third image, right row : fourth image. (a),(b) Line detection, (c)-(f) orthogonal bundles

In our future work, we propose to improve the line detection method by including the chain division directly in the prediction/verification process in order to avoid the heuristic step. In a more interesting way, we plan to use direction information of the parallel lines in order to obtain the orientation of the 3D planes and thus to obtain a 3D estimation of the scene.

REFERENCES

- [1] J.P. Barreto and H. Araujo, "Direct Least Square Fitting of Paracatadioptric Line Images", *In Proc. IEEE Workshop on Omnidirectional Vision (OMNIVIS03)*, Madison, WI, USA, pp. 78, 2003.
- [2] J.P. Barreto, "General Central Projection Systems: Modeling, Calibration and Visual Servoing", PhD Thesis, University of Coimbra, September 2003.
- [3] M. Bosse and R. Rikoski and J. Leonard and S. Teller, "Vanishing Points and 3D Lines from Omnidirectional Video", *in The Visual Computer, Special Issue on Computational Video*, Vol. 19, N° 6, pp. 417-430, 2003.
- [4] C. Demoncaux and P. Vasseur and C. Pégard, "Omnidirectional Vision on UAV for Attitude Computation", *In Proc. IEEE International Conference on Robotics and Automation (ICRA06)*, Orlando, FL, USA, 2006.
- [5] C. Demoncaux and P. Vasseur and C. Pégard, "Robust Attitude Estimation with Catadioptric Vision", *In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS06)*, Beijing, China, 2006.
- [6] S.M. Ettinger and M.C. Nechyba and P.G. Ifju and M. Waszak, "Vision-guided Flight Stability and Control for Micro Air Vehicles", *In Advanced Robotics*, Vol. 17, N°7, pp. 617-640, 2003.
- [7] Geyer, C., Daniilidis, K.: Catadioptric Projective Geometry. In *International Journal of Computer Vision (IJCV)*, Vol. 45, N°3, pp. 223-243, 2001.
- [8] S.E. Hrabar and G. Sukhatme, "Omnidirectional Vision for an Autonomous Helicopter", *In Proc. IEEE International Conference on Robotics and Automation (ICRA03)*, Taipei, Taiwan, pp.558-563, 2003.
- [9] S. Li, "Estimating Head Orientation Based upon Sky-ground Representation", *In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05)*, Edmonton, Canada, pp. 580-585, 2005.
- [10] J. Luo and M. Boutell, "Automatic Image Orientation Detection via Confidence-Based Integration of Low-Level and Semantic Cues", *in IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 27, N°5, pp. 715-726, 2005.
- [11] S. Todorovic and M.C. Nechyba and P.G. Ifju, "Sky/Ground Modeling for Autonomous MAV Flight", *In Proc. IEEE International Conference on Robotics and Automation (ICRA03)*, Taipei, Taiwan, pp. 1422-1427, 2003.
- [12] P. Vasseur and E. M. Mouaddib, "Central Catadioptric Line Detection", *In Fifteenth British Machine Vision Conference (BMVC)*, Kingston University, London, September 7-9, 2004.
- [13] L.K. Wang and S.C. Hsieh and E.C.W. Hsueh and F.B. Hsiao and K.Y. Huang, "Complete Pose Determination for Low Altitude Unmanned Aerial Vehicle Using Stereo Vision", *In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05)*, Edmonton, Canada, pp. 316-321, 2005.
- [14] J.H. Woo and I.S. Kweon and G.S. Kim and I.C. Kim, "Robust Horizon and Peak Extraction for Vision-based Navigation", *In Proc. IAPR Workshop on Machine Vision Applications (MVA05)*, Tsukuba Science City, Japan, 2005.
- [15] X. Ying and Z. Hu, "Catadioptric Line Features Detection using Hough Transform", *In Proc. of International Conference on Pattern Recognition (ICPR04)*, Cambridge, United Kingdom, pp. 839-842, 2004.