Control Engineering Practice 18 (2010) 789–799

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Vision-based navigation of unmanned aerial vehicles

Jonathan Courbon [a,b,*], Youcef Mezouar [b], Nicolas Guénard [a], Philippe Martinet [b]

[a] CEA-List, 18 route du Panorama, BP6, F-92265 Fontenay Aux Roses, France
[b] LASMEA, 24 Avenue des Landais, 63177 Aubiere, France

ARTICLE INFO

Article history:
Received 31 March 2009
Accepted 15 March 2010
Available online 3 April 2010

Keywords:
UAV
Monocular vision
Visual navigation
Visual memory

ABSTRACT

This paper presents a vision-based navigation strategy for a vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV) using a single embedded camera observing natural landmarks. In the proposed approach, images of the environment are first sampled, stored and organized as a set of ordered key images (visual path) which provides a visual memory of the environment. The robot navigation task is then defined as a concatenation of visual path subsets (called visual route) linking the current observed image and a target image belonging to the visual memory. The UAV is controlled to reach each image of the visual route using a vision-based control law adapted to its dynamic model and without explicitly planning any trajectory. This framework is largely substantiated by experiments with an X4-flyer equipped with a fisheye camera.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Sarris (2001) establishes a list of civilian applications for UAVs including border interdiction, search and rescue, wild fire suppression, communications relay, law enforcement, disaster and emergency management, research, industrial and agricultural applications. 3D archaeological map reconstruction and image mosaicing may be added to this list. In order to develop such applications automatic navigation of those vehicles has to be addressed. While most of the current researches deal with the attitude estimation (Hamel & Mahony, 2006) or with the control of UAVs (Guénard, Hamel, & Eck, 2006), few works propose navigation strategies. In this area, the most popular sensor is the GPS receiver. In this case, the navigation task consists generally in reaching a series of GPS waypoints or on following a 3D trajectory. In Nikolos, Tsourveloudis, and Valavanis (2002), this trajectory is extracted from an elevation map with a genetic algorithm. Unfortunately, GPS data are not always available (for instance in indoor environment) or can be inaccurate (for instance in dense urban area where buildings can mask some satellites or when light GPS receiver are used). For those reasons, it is necessary to use other sensors. Employing a camera is very attractive to solve those problems because in places where the GPS is difficult to use such as city centers or even indoors, there are usually a lot of visual features. A navigation system based on vision could thus be a good alternative to GPS. Some techniques originally developed for ground vehicles have been transposed to the context of UAV navigation. For instance in Angeli, Filliat, Doncieux, and Meyer (2006), a 2D simultaneous localization and mapping (SLAM) technique is used. In Frew, Langelaan, and Stachura (2007), a bearing-only SLAM is proposed. Note that SLAM techniques only focus on the mapping and localization parts whereas the aim of the authors is here to build a complete navigation framework which includes mapping, localization and also control. Vision-based strategies have also been proposed to control the motions of UAVs. For instance, an homography-based control scheme is proposed in Hu, Dixon, Gupta, and Fitz-Coy (2006). However, this approach requires the camera to point to the ground which is supposed to be planar. In Guénard, Hamel, and Mahony (2007), an image-based control strategy using centroid of artificial landmarks (white blobs) with known positions is used for a positioning task. In Bourquardez and Chaumette (2007), a visual servoing scheme is proposed to align an airplane with respect to a runway in a simulated environment. In Chitrakaran, Dawson, Kannan, and Feemster (2006), leader–follower and visual trajectory following strategies based on homography decomposition are proposed and simulated. Note that in this approach, planar surfaces have to be observed by a pan-tilt camera. In the approach proposed in this paper, the camera is not restricted to observe planar surfaces and experiments have been conducted in real contexts without prior knowledge of the environment.

Many visual-memory based navigation strategies have been proposed for ground vehicles (for instance refer to Courbon, Mezouar, & Martinet, 2009; Goedemé, Tuytelaars, & Gool, 2004; Matsumoto, Ikeda, Inaba, & Inoue, 1999). Specific additional challenges are involved to apply those strategies for UAVs. First, aerial vehicles are underactuated rigid body objects moving in 3D while ground mobile robots are generally vehicle with non-holonomic kinematics moving on a locally plane world.

* Corresponding author at: CEA-List, 18 route du Panorama, BP6, F-92265 Fontenay Aux Roses, France.
E-mail address: jonathan.courbon@lasmea.univ-bpclermont.fr (J. Courbon).

0967-0661/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2010.03.004

Second, dynamic effects and external perturbations are important for small aerial vehicles while they can be generally neglected when dealing with ground vehicles. Finally, due to data transmission and shaky movements the video sequences sends from the UAV to the ground station is of bad quality which can impact the navigation strategy.

1.1. Method overview and paper structure

An overview of the proposed navigation framework is presented in Fig. 1. The method can be divided into three steps (1) visual memory building, (2) localization, (3) autonomous navigation.

In the first off-line step (visual memory building), a sequence of images is acquired during a human-guided navigation. It allows to derive paths driving the UAV from its initial to its goal locations. In order to reduce the complexity of the image sequences, only key views are stored and indexed on a visual path. The set of visual paths can be interpreted as a visual memory of the environment. Section 2 details more precisely this point.

In the second step, before the beginning of the motion, the localization of the robotic system is performed on-line. During this stage, no assumption about the UAV's position is made. The localization process consists in finding the image of the visual memory which best fits the current image. In this step, only the most similar view is sought and not the metric position of the robotic system. More details about the proposed hierarchical localization process are given in Section 3.

In the last stage (refer to Section 4), given an image of one of the visual paths as a target, the UAV navigation mission is defined as a concatenation of visual path subsets, called visual route. A navigation task then consists in autonomously executing a visual route, on-line and in real-time. This control, taking into account the model of the UAV, guides the vehicle along the reference visual route without explicitly planning any trajectory.

Experiments have been carried out with an X4-flyer equipped with a fisheye camera, navigating in an indoor environment. Results are presented in Section 5.

2. Visual memory and route building

The first step of the proposed framework consists in a learning stage to build the visual memory. The structure of the visual memory initially developed in the context of wheeled mobile robots (refer to Courbon et al., 2009 for more details) is recalled in this section.

2.1. Visual memory

The visual memory is composed of a set of images $\{\mathcal{I}_i | i \in \{1,2,\ldots,n\}\}$ connected to form a graph. Let $\mathcal{R}(\mathcal{O}_c,\mathbf{x}_c,\mathbf{y}_c,\mathbf{z}_c)$ be the body fixed frame attached to the center of mass of the

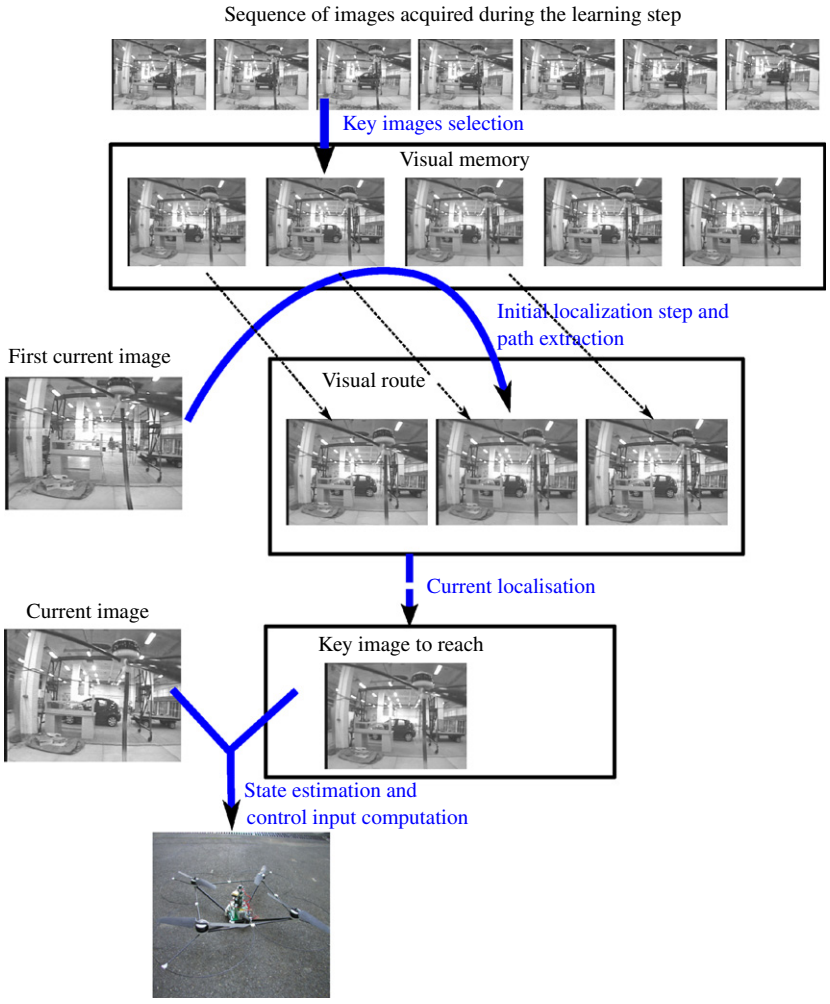

Fig. 1. Overview of the proposed vision-based framework.

robot (refer to Fig. 5). Without loss of generality, it is supposed that the camera frame coincides with the robot frame. For control purpose, the authorized motions between two connected images are assumed to be limited to those of the considered UAV. Hypothesis 2.1 formalizes these constraints.

Hypothesis 2.1. Given two frames $\mathcal{R}_i$ and $\mathcal{R}_j$ respectively associated to the vehicle when two successive key images $\mathcal{I}_i$ and $\mathcal{I}_j$ of the memory were acquired, there exists an admissible path (Υ) from $\mathcal{R}_i$ to $\mathcal{R}_j$ for the UAV.

Moreover, the vehicle is controllable from $\mathcal{I}_i$ to $\mathcal{I}_j$ only if the hereunder Hypothesis 2.2 is respected.

Hypothesis 2.2. Two successive key images $\mathcal{I}_i$ and $\mathcal{I}_j$ contain a set $\mathcal{P}_i$ of matched visual features, which can be observed along the path (Υ) performed between $\mathcal{R}_i$ and $\mathcal{R}_j$ and which allows the computation of the control law.

If Hypotheses 2.1 and 2.2 are verified then an edge connects the two configurations of the vehicle's workspace related to the two corresponding images of the visual memory. In case of an omnidirectional vehicle like the X4-flyer, if the UAV is able to be controlled from $\mathcal{R}_i$ to $\mathcal{R}_j$, it is able to be controlled from $\mathcal{R}_j$ to $\mathcal{R}_i$. The visual memory is then structured as a graph with undirected edges linking images.

2.2. Visual route

A visual route describes the vehicle's mission in the sensor space. Given two key images of the visual memory $\mathcal{I}_s^*$ and $\mathcal{I}_g$, corresponding respectively to the starting and goal locations of the vehicle in the memory, a visual route is a set of key images which describes a path from $\mathcal{I}_s^*$ to $\mathcal{I}_g$. The starting image $\mathcal{I}_s^*$ is the closest key image to the first image $\mathcal{I}_s$ acquired on-line. $\mathcal{I}_s^*$ is extracted from the visual memory during the localization step detailed in Section 3.

2.3. Keyframes selection

The keyframes selection process can be splitted into three stages:

1. *Image pre-processing*: Considered UAV sequences are affected by noise due to the video transmission system. This noise is usually characterized by white stripes or severe black and white disturbances. These corrupted frames cause obvious problems in features detection. A simple but effective technique was developed to eliminate them. It is based on two criteria:
   - *White stripes detection*. Firstly, the left border is checked, vertically, looking for a white pixel. If a white pixel is found, then it is checked if the whole line is white as well. If at least three white stripes are detected, the frame is deleted.
   - *Similarity of consecutive frames*. The distance between two consecutive frames is measured. If it is too high, it means that the second frame is corrupted and thus the image is eliminated. The Kullback–Leibler distance, or mutual entropy, on the histograms of the two frames:

$$d(\mathrm{p},\mathrm{q}) = \sum_i \mathrm{p}(i)\log\frac{\mathrm{p}(i)}{\mathrm{q}(i)}$$

     where p and q are the histograms of the frames is used. The threshold is fixed on 0.2.
2. *Key-image selection*: The first image of the video sequence is selected as the first key frame $\mathcal{I}_1$. A key frame $\mathcal{I}_{i+1}$ is then chosen so that there are as many video frames as possible between $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ while there are at least M common interest points matched between $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$. The image matching process will be detailed in Section 2.4.
3. *Manual verification*: Some remaining images with poor quality (see Fig. 2 for example) are manually rejected.

Note that the first stage of this process is also employed during the autonomous navigation to eliminate corrupted frames.

2.4. Feature matching

A central clue for implementation of the proposed framework relies on efficient point matching. This process takes place in all steps of the proposed navigation framework. It allows key image selection during the learning stage (in step 2) and it is also used during the localization step and during the autonomous navigation. A similar process to the one proposed in Royer, Lhuillier, Dhome, and Lavest (2007) and successfully applied for the metric localization of autonomous vehicles in outdoor environment is used. Interest points are detected in each image with Harris corner detector (Harris & Stephens, 1988). For an interest point P_i at coordinates $[x\ y]^\top$ in image $\mathcal{I}_i$, a region of interest (ROI) is defined in image $\mathcal{I}_{i+1}$. This ROI is a rectangle of center of the point of coordinates $[x\ y]^\top$. For each interest point P_{i+1} inside the ROI in image $\mathcal{I}_{i+1}$, a score between the neighborhoods of P_i and P_{i+1} is computed using a zero normalized cross correlation. The point with the best score that is greater than a certain threshold is kept as a good match and the unicity constraint is used to reject matches which have become impossible. This method is illumination invariant and its computational cost is small.

3. Localization in a memory of wide field of view images

The output of the learning process is a data set of images (*visual memory*). The first step of the autonomous navigation process is the self-localization of the vehicle in the visual memory. In this step, the robot is assumed to be situated nearby the situation where a key image was acquired. The localization consists in finding the image of the memory which best fits the current image by comparing pre-processed and on-line acquired

Fig. 2. Frames corrupted by noise: (a) white stripes, (b) black and white disturbances, (c) remaining image (manually rejected).

images. In this paper, the authors particularly focus on a method suitable when the data set consists in omnidirectional images. Omnidirectional cameras are usually intended as a vision system providing a huge field-of-view. Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by opportunely combining mirrors and conventional cameras, or employing purely dioptric fisheye lenses (Baker & Nayar, 1999). As first demonstrated in Barreto (2006) and exploited in robotic applications in Courbon, Mezouar, Eck, and Martinet (2007), images acquired by those sensors have a similar behaviour. In the experiments detailed in Section 5, a fisheye camera is employed.

The efficiency of a visual localization method can be measured by means of: (1) accuracy of the results, (2) memory needed to store data and (3) computational cost. The main objective is to optimize the localization process under those criteria. Two main strategies exist to match images: the image can be represented by a single descriptor (*global approaches*) (Gaspar, Winters, & Santos-Victor, 2000; Matsumoto et al., 1999) or alternatively by a set of descriptors defined around visual features (*landmarks-based* or *local approaches*) (Goedemé et al., 2005; Murillo, Guerrero, & Sagüés, 2007). In those last methods, some relevant visual features are extracted from the images. A descriptor is then associated to each feature neighbourhood. The robustness of the extraction and the invariance of the descriptor are one main issue to improve the matching process. In one hand, local approaches are generally more accurate but have a high computational cost (Murillo et al., 2007). On the other hand, global descriptors speed up the matching process at the price of affecting the robustness to occlusions. A hierarchical approach is proposed in Murillo et al. (2007): a first selection is done using a global descriptor while the final localization results from local descriptors.

In this paper, a hierarchical approach for localization in a database of omnidirectional images is proposed. The computational efficiency is ensured in a first step by defining a well suited global descriptor which allows to select a set of candidate images. Local descriptors are then exploited to select only the best image and thus to ensure accuracy.

3.1. Global descriptor

The first step is based on a geometrical image representation derived from surface interpolation. Images have first their histogram equalized in order to be more robust to illumination changes. Pixels are seen as a discrete 3D surface $\mathcal{S}$ with the grey level as the third coordinate (refer to Fig. 3):

$$\mathcal{S}: \begin{cases} [0,1,\ldots,N]\times[0,1,\ldots,M]\mapsto[0,255] \\ (u,v)\rightarrow\mathcal{S}(u,v) \end{cases}$$

The interpolation consists in locally approximating this surface $\mathcal{S}(u,v)$ by a surface $f(s,t), s\in[0;1], t\in[0;1]$. Note that it is necessary to have control points at the same positions in order to compare descriptors of different images. Moreover, regular positions ensure a better interpolation. In that aim, the use of the triangular mesh vertices represented in Fig. 3(a) as control points and the altitude $\mathbf{z}$ of the control points of the approximated surface as descriptors are proposed. This triangular mesh is generated as proposed in Persson and Strang (2004). Node locations are computed by solving for equilibrium in a truss structure using piecewise linear force–displacement relations. The proposed global descriptor is thus the interpolation by a cubic function of the image surface at the node locations defined previously. The required computational cost is low and interpolation errors are small.

3.2. First selection and local descriptor

Descriptor $\mathbf{z}_c$ (respectively $\mathbf{z}_i$) is computed for the current image $\mathcal{I}_c$ (respectively for the memorized image $\mathcal{I}_i$). The global distance d_i^{global} between those two images is the L_1 distance between $\mathbf{z}_c$ and $\mathbf{z}_i$. Kept candidate images are such that $d_i^{\text{global}}/\min_i d_i^{\text{global}} \leq t$ where the threshold $t\geq 1$ allows to not reject the images which have a distance similar to the minimal distance. The output of this first stage is a small amount of candidate images.

A local approach is proposed to select the best candidate since only few images are involved (i.e in this case the computational cost is low). With this aim, a classical local approach based on the zero normalized cross correlation (ZNCC) between patches around Harris corners is employed since the computational cost is much lower than SIFT or SURF based approaches whereas similar accuracy is obtained with images corresponding to close viewpoints (Courbon, Mezouar, Eck, & Martinet, 2008). In this stage, the local distance between two images is simply chosen as $d_i^{\text{local}}=1/n$ where n is the number of matched features. The final result of the localization is the image $\mathcal{I}_k$ such that $d_k^{\text{local}}=\min_i(d_i^{\text{local}})$.

This hierarchical method has been compared to state-of-the-art techniques in Courbon et al. (2008). The obtained results show that the proposed method is a good compromise between accuracy, amount of memorized data and computational cost.

4. Route following

When starting the autonomous navigation task, the output of the localization step provides the closest image $\mathcal{I}_s^*$ to the current image $\mathcal{I}_s$. A visual route Ψ connecting $\mathcal{I}_s^*$ to the goal image $\mathcal{I}_g$ is then extracted from the visual memory. As previously explained, the visual route is composed of a set of key images:

$$\Psi=\{\mathcal{I}_1,\mathcal{I}_2,\ldots,\mathcal{I}_{n-1},\mathcal{I}_n\}$$

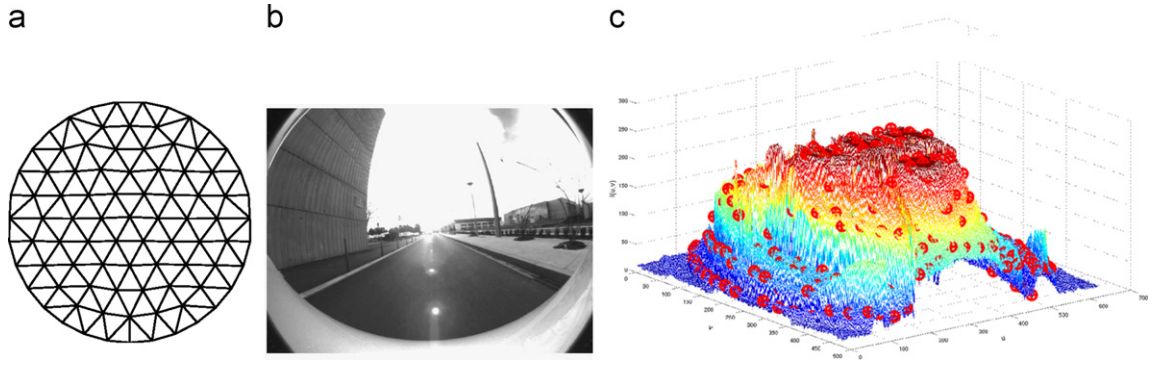

Fig. 3. Given the control point positions in the plane (a), the image (b) is seen as a surface and interpolated (c).

where $\mathcal{I}_1 = \mathcal{I}_s^*$, $\mathcal{I}_n = \mathcal{I}_g$ and n is the number of images of the path. The next step is to automatically follow this visual route using a vision-based control scheme.

To design the controller, described in the sequel, the key images of the reference visual route are considered as consecutive waypoints to reach in the sensor space. The control problem is thus formulated as a position control to guide the underactuated robot along the visual route. The computation of the control input requires the design of the control law and the estimation of the state of the vehicle from the current image and the desired key image (refer to Section 4.4) as presented in Fig. 4.

4.1. Vehicle modelling

In this section, the equations of motion for a UAV in quasi-stationary flight conditions are briefly recalled following Hamel, Mahony, Lozano, and Ostrowski (2002). Let $\mathcal{R}_{\text{in}}(\mathcal{O}_{\text{in}},\mathbf{e}_1,\mathbf{e}_2,\mathbf{e}_3)$ be the inertial frame attached to the earth, relative to a fixed origin assumed to be Galilean and $\mathcal{R}_c(\mathcal{O}_c,\mathbf{x}_c,\mathbf{y}_c,\mathbf{z}_c)$ be the frame attached to the UAV (refer to Fig. 5) with $\mathcal{O}_c$ the gravity center.

The position of $\mathcal{O}_c$ with respect to the inertial frame $\mathcal{R}_{\text{in}}$ is denoted $\mathbf{p}$. The orientation of the airframe is given by a rotation $\boldsymbol{\Theta} : \mathcal{R}_c \to \mathcal{R}_{\text{in}}$. Let $\mathbf{v}$ (respectively $\boldsymbol{\Omega}$) be the linear (resp. angular) velocity of the center of mass expressed in the inertial frame $\mathcal{R}_{\text{in}}$ (resp. in $\mathcal{R}_c$). The geometry of the robot is supposed to be perfect. The control inputs to send to the vehicle are: T, a scalar input termed thrust or heave, applied in direction $\mathbf{z}_c$ and $\boldsymbol{\Gamma} = [\Gamma_1\ \Gamma_2\ \Gamma_3]^\top$ (the control torques relative to the Euler angles). Let m denotes the mass of the airframe, g the gravity constant and let $\mathbf{I}$ be the 3×3 constant inertia matrix around the centre of mass, expressed in $\mathcal{R}_c$. Newton's equations of motion yield the following dynamic model for the motion of a rigid object:

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v} \\ m\dot{\mathbf{v}} = -T\boldsymbol{\Theta}\mathbf{e}_3 + mg\mathbf{e}_3 \\ \dot{\boldsymbol{\Theta}} = \boldsymbol{\Theta}\,\mathrm{sk}(\boldsymbol{\Omega}) \\ \mathbf{I}\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega}\times\mathbf{I}\boldsymbol{\Omega} + \boldsymbol{\Gamma} \end{cases} \tag{1}$$

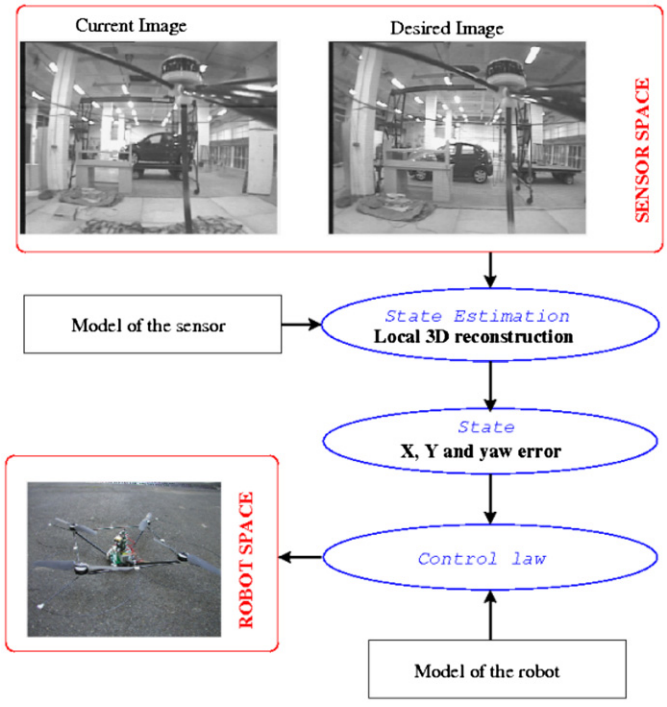

Fig. 4. Visual route following process.

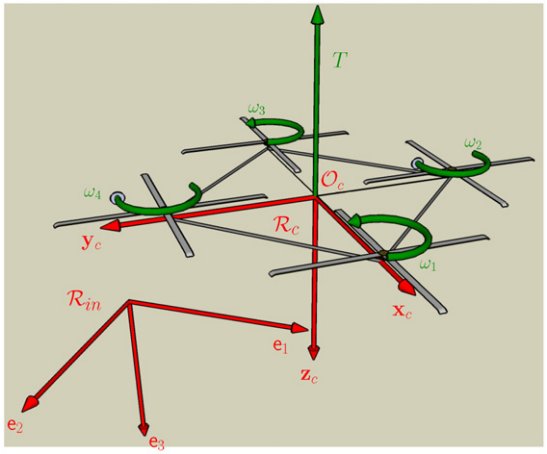

Fig. 5. The four rotors generating the collective thrust.

4.2. Control objective

Let $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ be two consecutive key images of a given visual route to follow and $\mathcal{I}_c$ be the current image. $\mathcal{R}_{i+1} = (\mathcal{O}_{i+1},\mathbf{x}_{i+1},\mathbf{y}_{i+1},\mathbf{z}_{i+1})$ is the frame attached to the vehicle when $\mathcal{I}_{i+1}$ was stored and $\mathcal{R}_c = (\mathcal{O}_c,\mathbf{x}_c,\mathbf{y}_c,\mathbf{z}_c)$ is the frame attached to the vehicle in its current location. The hand-eye parameters (i.e. the rigid transformation between $\mathcal{R}_c$ and the frame attached to the camera) are supposed to be known. According to Hypothesis 2.2, the state of a set of visual features $\mathcal{P}_i$ is known in the images $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$. The state of $\mathcal{P}_i$ is also assumed available in $\mathcal{I}_c$ (i.e. $\mathcal{P}_i$ is in the camera field of view). The visual task to achieve is to drive the state of $\mathcal{P}_i$ from its current value to its value in $\mathcal{I}_{i+1}$ which is equivalent to drive $\mathcal{R}_c$ to $\mathcal{R}_{i+1}$. In the case of a quadrotor, rotational dynamic and translational dynamic are coupled (refer to (1)) and a translation is obtained by inclining the UAV. During autonomous navigation, it is not required to have the same pitch and roll angles (i.e. the same translational velocity) than in the learning step. The task to achieve is thus defined as the regulation to zero of the position error $\tilde{\mathbf{p}}$ (i.e. the position of $\mathcal{O}_{i+1}$ in $\mathcal{R}_c$) and the yaw error $\tilde{\theta}$ of $\mathcal{R}_{i+1}$ with respect to $\mathcal{R}_c$. The control scheme designed to realize this objective is presented in Section 4.3. Geometrical relationships between two views acquired with a camera under the generic projection model (which includes conventional, catadioptric and some fisheye cameras) are exploited to enable a partial Euclidean reconstruction from which $\tilde{\mathbf{p}}$ and $\tilde{\theta}$ are derived (Section 4.4).

4.3. Control design

The positioning task described in the previous section is realized using a control scheme composed of three loops (refer to Fig. 6). The outer loop consists in assigning the desired translational velocity and to assure that the system remains in quasi-stationary flight conditions. The intermediate loop ensures the convergence of the translational velocity $\mathbf{v}$ to the desired velocity $\mathbf{v}_d$ by assigning the desired matrix $\boldsymbol{\Theta}_d$. Finally, the control torques $\boldsymbol{\Gamma}$ are assigned in order to have the rotational matrix $\boldsymbol{\Theta}$ converging to this desired matrix $\boldsymbol{\Theta}_d$ in the inner loop. This control scheme assures that the tilt angle is limited to small-angle and that the velocity is bounded in order that the UAV remains in quasi-stationary flight conditions. Stability analysis of the embedded controller is detailed in Guénard, Moreau, Hamel, and Mahony (2008). The experimental system and gain adjustments ensure a quick convergence of the rotation to the

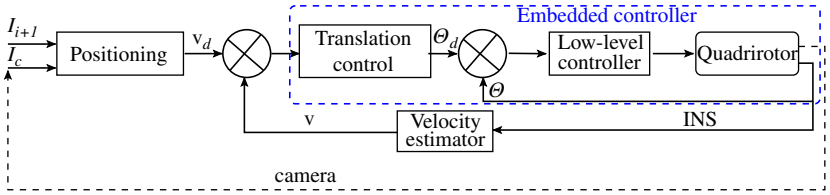

Fig. 6. Control loops.

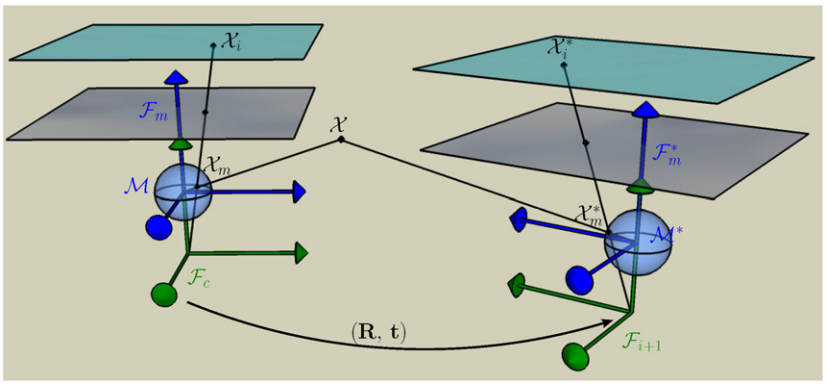

Fig. 7. Geometry of two views.

desired rotation. For the translational dynamic, associated gains are smaller. Thus, considering that the dynamic of $\mathbf{v}$ is slow compared to the dynamic of $\boldsymbol{\Theta}$, $\boldsymbol{\Theta}_d$ changes slowly. Coupling terms between the two loops can be neglected and thus it is assured that $\boldsymbol{\Theta} \rightarrow \boldsymbol{\Theta}_d$ and $\mathbf{v} \rightarrow \mathbf{v}_d$.

The position error $\tilde{\mathbf{p}}$ and the velocity error $\tilde{\mathbf{v}}$ are defined by the following equations:

$$\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p_d} \tag{2}$$

$$\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}_d \tag{3}$$

where $\mathbf{p_d}$ is the constant desired position ($\dot{\mathbf{p}}_\mathbf{d} = 0$). The vectorial function noted $\text{sat}_\varepsilon(\mathbf{x})$ represents the saturation of each component of the vector $\mathbf{x}$ to ε: $\text{sat}_\varepsilon(x_i) = x_i$ if $|x_i| \leq \varepsilon$ and $\text{sat}_\varepsilon(x_i) = \varepsilon \ \text{sign}(x_i)$ if $|x_i| > \varepsilon$. As a consequence, the relation $\mathbf{x}^\top \text{sat}_\varepsilon(\mathbf{x}) > 0$ exists for all $\mathbf{x} \neq \mathbf{0}$.

Theorem 4.1. *The control input defined by*

$$\mathbf{v}_d = -\kappa \, \text{sat}_\varepsilon(\tilde{\mathbf{p}}) \tag{4}$$

with κ small compared to the translational dynamic gains, is stabilizing and assures that the system stays in quasi-stationary flight conditions.

Note that ε depends on the quasi-stationary flight limit conditions on the translational velocity. The proof is given in Appendix.

4.4. State estimation from the generic camera model

In this work, the unified model described in Geyer and Daniilidis (2003) is used since it allows to formulate state estimations that are valid for visual sensors having a single viewpoint (that is, there exists a single center of projection, so that, every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction). In other words, it encompasses all sensors in this class (Geyer & Daniilidis, 2003): perspective and catadioptric cameras. A large class of fisheye cameras are also concerned by this model (Barreto, 2006; Courbon et al., 2007).

The unified projection model consists in a central projection onto a virtual unitary sphere followed by a perspective projection onto the image plane (Geyer & Daniilidis, 2003). This generic model is parametrized by ξ describing the type of sensor and by a matrix $\mathbf{K}$ containing the intrinsic parameters.

Let $\mathcal{X}$ be a 3D point and $\mathbf{R}$ and $\mathbf{t}$ the rotational matrix and the translational vector between the current and the desired frames. Let $\mathbf{x}_m$ (respectively $\mathbf{x}_m^*$) be the coordinates of the projection of $\mathcal{X}$ onto the unit sphere linked to the current frame $\mathcal{F}_c$ (resp. to $\mathcal{F}_{i+1}$) (refer to Fig. 7). The epipolar plane contains the projection centers $\mathcal{O}_c$ and $\mathcal{O}_{i+1}$ and the 3D point $\mathcal{X}$. $\mathcal{X}_m$ and $\mathcal{X}_m^*$ clearly belong to this plane. The coplanarity of those points leads to the relation:

$$\mathbf{x}_m^\top \mathbf{E} \mathbf{x}_m^{*\top} = 0 \tag{5}$$

where $\mathbf{E} = \mathbf{R} \, \text{sk}(\mathbf{t})$ is the essential matrix (Svoboda & Pajdla, 2002). In Eq. (5), $\mathbf{x}_m$ (respectively $\mathbf{x}_m^*$) corresponds to the coordinates of the point projected onto the sphere, in the current image $\mathcal{I}_c$ (respectively in the desired key image). Those coordinates are obtained from the coordinates of the point matched in the first and second images in two steps:

Step 1: The 2D projective point $\underline{\mathbf{x}} = [x \ y \ 1]^\top$ is obtained from the coordinates $\underline{\mathbf{x}}_i = [u \ v \ 1]^\top$ of the point in the image after a plane-to-plane collineation $\mathbf{K}^{-1}$: $\underline{\mathbf{x}} = \mathbf{K}^{-1} \underline{\mathbf{x}}_i$.

Step 2: $\mathbf{x}_m$ can be computed as a function of the coordinates in the image and the sensor parameter ξ:

$$\mathbf{x}_m = (\eta^{-1} + \xi) \begin{bmatrix} x & y & \dfrac{1}{1 + \xi \eta} \end{bmatrix}^\top \tag{6}$$

with

$$\begin{cases} \eta = \dfrac{-\gamma - \xi(x^2 + y^2)}{\xi^2 (x^2 + y^2) - 1} \\ \gamma = \sqrt{1 + (1 - \xi^2)(x^2 + y^2)} \end{cases}$$

The essential matrix $\mathbf{E}$ between two images can be estimated using five couples of matched points as proposed in Nistér (2004) if the camera calibration (matrix $\mathbf{K}$ and parameter ξ) is known. Outliers are rejected using a random sample consensus (RANSAC) algorithm (Fischler & Bolles, 1981). From the essential matrix, the

Fig. 8. Quad-rotor UAV used in the proposed experiments.

Fig. 11. Key image Im-4 to reach (Exp. 1).

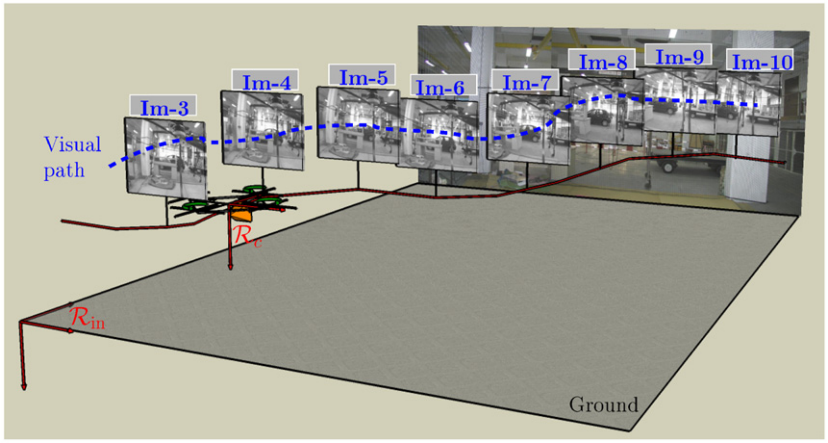

Fig. 9. Some images of the visual memory *Drone I* of the UAV.

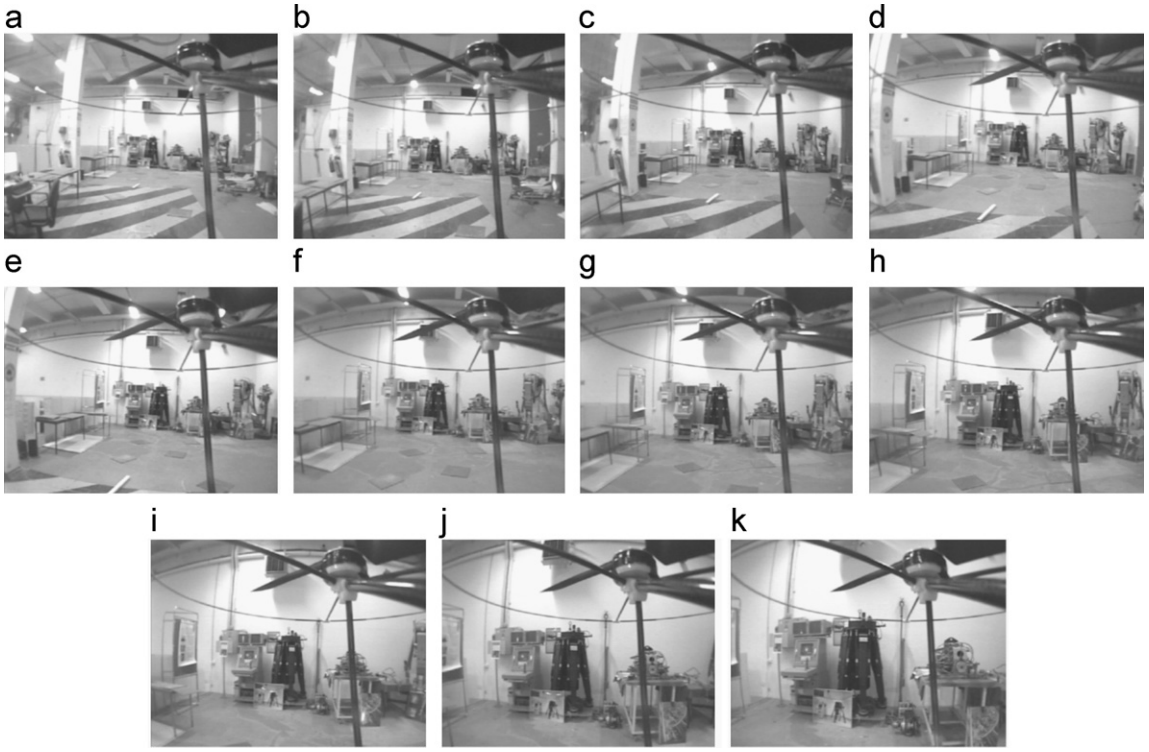

Fig. 10. Images of the sequence *Drone II*. (a) $\mathcal{I}_1$. (b) $\mathcal{I}_2$. (c) $\mathcal{I}_3$. (d) $\mathcal{I}_4$. (e) $\mathcal{I}_5$. (f) $\mathcal{I}_6$. (g) $\mathcal{I}_7$. (h) $\mathcal{I}_8$. (i) $\mathcal{I}_9$. (j) $\mathcal{I}_{10}$. (k) $\mathcal{I}_{11}$.

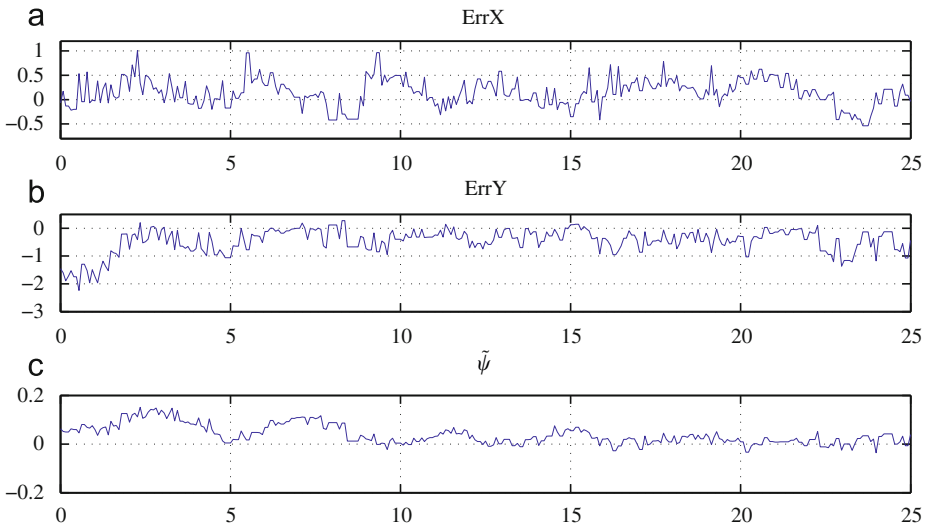

Fig. 12. (a) *ErrX* (expressed in meters), (b) *ErrY* (expressed in meters) and (c) yaw error (expressed in rad) vs. time (in seconds) (Exp. 1).

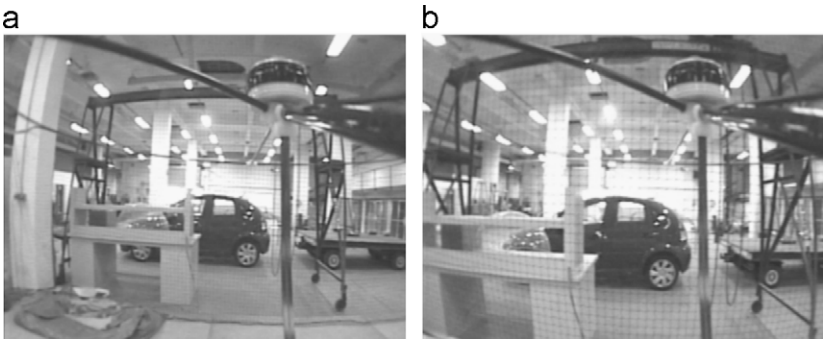

Fig. 13. Key images to successively reach (Exp. 2). (a) Key image Im-12. (b) Key image Im-11.

camera motion parameters (that is the rotation $\mathbf{R}$ and the translation $\mathbf{t}$ up to a scale) can be determined (refer to Hartley & Zisserman, 2000). Finally, the input of the control law (4), i.e. $\tilde{\mathbf{p}}$ and $\tilde{\theta}$ can be computed straightforwardly from $\mathbf{t}$ and $\mathbf{R}$. In the experimentation proposed in Section 5, the scale factor is roughly estimated. Let $s \in \Re^{+*}$ be the scale factor. The control input:

$$v_d = -\kappa \, \mathrm{sat}_\varepsilon(s\tilde{\mathrm{p}}) = -\kappa s \, \mathrm{sat}_\varepsilon(\tilde{\mathrm{p}}) \tag{7}$$

is stabilizing and assures that the system stays in quasi-stationary flight conditions if κs is small compared to the translational dynamic gains.

5. Experimental results

In this section, the results obtained with an experimental platform are discussed. The UAV used for the experimentation (refer to Fig. 8) is a quadrotor designed by the CEA. It is a vertical take off and landing (VTOL) vehicle ideally suited for stationary and quasi-stationary flight (Guénard et al., 2007).

5.1. Experimental set-up

The X4-flyer is equipped with a digital signal processing (DSP), running at 150 MIPS, which performs the control algorithm of the orientation dynamics and filtering computations. For orientation dynamics, an embedded high gain controller running at 166 Hz independently ensures the exponential stability of the orientation towards the desired one. The translational velocities in x_c and y_c directions are estimated from the INS measurements. Those information are quickly diverging thus they are readjusted thanks to the optical flow measured on the ground with a second embedded camera, using a fuzzy logic approach.[1] The control along the axis $\mathbf{z}_c$ is thus not considered here. The embedded camera used for navigation has a field of view of 120° and is pointing forward. It transmits 640×480 pixels images at a frequency of 12.5 fps to a laptop using RTAI-Linux OS with a 2 GHz Centrino Duo processor via a wireless analogical link. Vision algorithms are implemented in C++ language in the laptop. The state required by the control law is computed on this laptop and is sent to the ground station by an ethernet connection. Desired orientation and desired thrust are generated on the ground station and sent to the UAV.

5.2. Learning step

During a first learning step, the UAV is manually controlled along an approximately linear path situated in the $(\mathbf{x}_c,\mathbf{y}_c)$ plane and at 45° from the $\mathbf{x}_c$- axis direction of the UAV and images are acquired by the embedded camera pointing forward ($\mathbf{x}_c$ direction, refer to Fig. 9).

Key images are selected as explained in Section 2.3. It results to a single sequence (called *Drone I* in the sequel) containing 12 key images (refer to Fig. 9). In addition, to experiment a local servoing

[1] A patent by N. Guénard (CEA-LIST) is currently in registration about this approach.

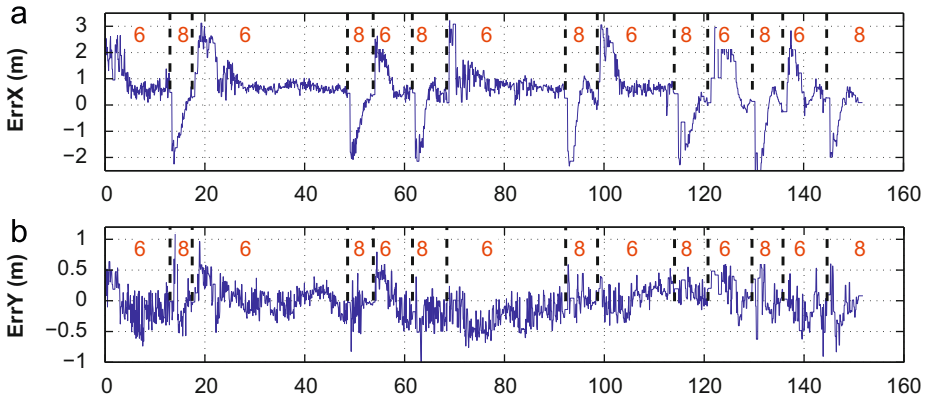

Fig. 14. (a) *ErrX* (m) and (b) *ErrY* vs. time (s) (Exp. 2).

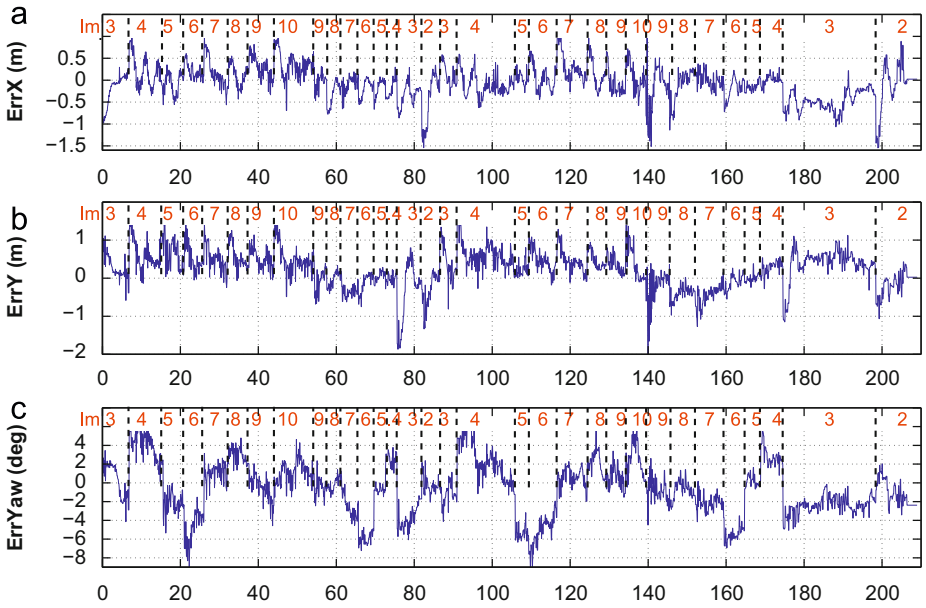

Fig. 15. (a) *ErrX* (m), (b) *ErrY* (m) and (c) yaw error (rad) vs. time (s) (Exp. 3).

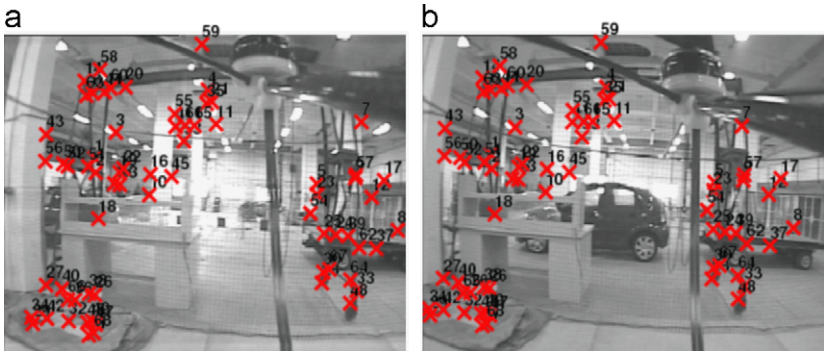

Fig. 16. Robustly matched features between the current image (a) and the image to reach (b; Im-5).

(Exp. 2), a new edge connecting two images is added into the visual memory. Those two images are such that the second key image is approximately situated at 1.5 m along the $\mathbf{x}_c$- axis back to the first image.

During a second learning step, a sequence is acquired in a 15-m long straight line in the direction $\mathbf{x}_c$ of the UAV. The visual memory built (called *Drone II*) contains in this case 11 key images (refer to Fig. 10).

5.3. Goal reaching (Exp. 1)

This section deals with the vision-based control of the UAV in order to reach the key image Im-4 drawn in Fig. 11. The robot is manually guided to an initial position approximately situated at 1.5 m at the right of the frame attached to the key image and similarly oriented. The robot is then automatically controlled in order to reach the key image. A mean of 73 robust matches for

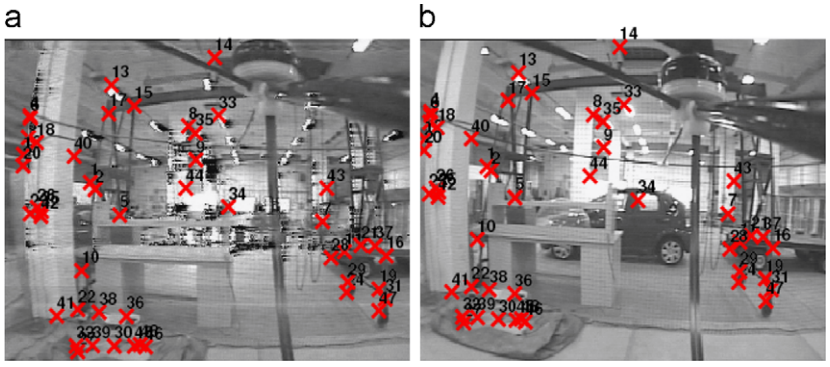

Fig. 17. Robustly matched features between the current image (a) and the image to reach (b; Im-4).

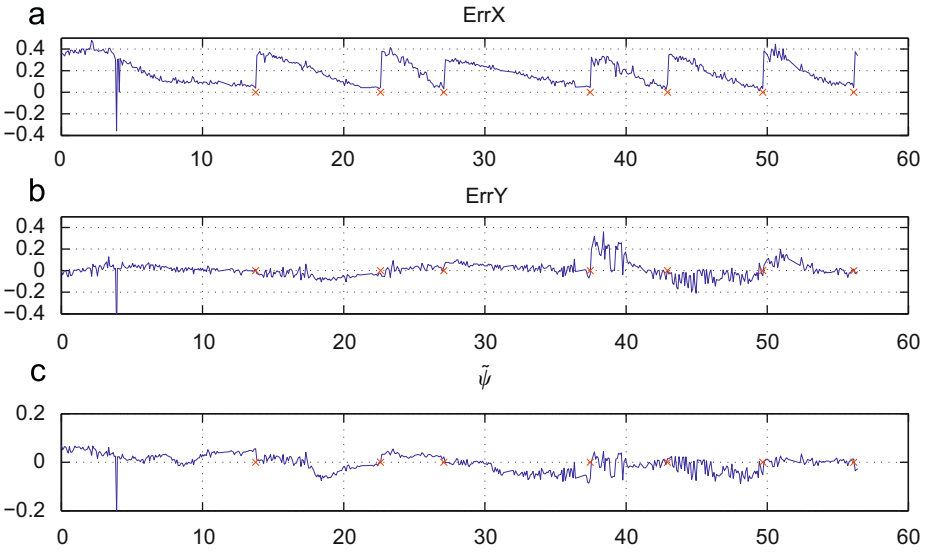

Fig. 18. Position errors *ErrX* (m) and *ErrY* (m) and yaw error $\tilde{\psi}$ (rad) vs. time (s) (Sequence *Drone II*).

each frame has been found during this experimentation. The mean computational time during the on-line navigation is 94 ms/image. Errors in translation (noted *ErrX* and *ErrY*), expressed in meters, and error in yaw angle, expressed in radian versus time (in seconds) are reported in Fig. 12. *ErrX*, *ErrY* and yaw angle errors are converging to zero. The remaining noise is caused by the mechanical vibrations of the body frame during the flight, the lost of quality in images after the transmission, the partial 3D reconstruction errors and by the asynchronous sensors' data. Moreover, oscillations may come from an error in translational velocity estimation. Nevertheless, the navigation task is correctly achieved.

5.4. Succession of two images (Exp. 2)

In this experiment, the two key images Im-12 and Im-11 of *Drone I* are defined as targets (refer to Fig. 13).

When the first target is reached, the key image 2 is set as the new target. When the key image 2 is reached, the key image 1 is set as the new target and so on (7 times). The two images are approximately situated in the direction of the vehicle. Translations thus occur mainly along the $\mathbf{x}_c$- axis direction. Results are reported in Fig. 14. In the figures vertical dotted lines denote that a key image is reached and the number on top of the axis represents the number of the key image to reach. After each change of desired key image, error in axis $\mathbf{y}_c$ and yaw angles are converging to zero. Error in axis $\mathbf{x}_c$ is also converging. A static error in axis $\mathbf{x}_c$ remains due to errors in velocity estimation. Future works will deal with this point.

5.5. Waypoints following (Exp. 3)

The visual path to follow is set manually as the sequence: Im: 3–4–5–6–7–8–9–10–9–8–7–6–5–4–3–2–3–4–5–6–7–8–9–10–9–8–7–6–5–4–3–2. A key image is assumed to be reached when the distance from the origin of the current frame to the origin of the desired frame in the $(\mathbf{x}_c,\mathbf{y}_c)$ plane is under a fixed threshold. Results are drawn in Fig. 15. Even if errors in axis $\mathbf{x}_c$ and $\mathbf{y}_c$ and in yaw angle are not exactly regulated to zero, the vehicle successfully follows the visual path.

Samples of robust matching between the current image and the desired key image are represented in Fig. 16 (68 matched points) and Fig. 17 (48 matched points). In Fig. 17, the current image has a low quality. Despite this fact, many points have been matched and the visual path has been successfully followed.

5.6. Drone II (Exp. 4)

The UAV is manually teleoperated nearby an image of the sequence *Drone II*. The localization step lasts 380 ms (35 ms/image) and the initial image found is $\mathcal{I}_3$ (refer to Fig. 10). The visual path extracted to reach $\mathcal{I}_{10}$ contains eight key images.

The autonomous navigation is stopped after reaching the key image $\mathcal{I}_9$. Position and yaw errors are represented in Fig. 18.

Firstly, note that the errors *ErrX* and *ErrY* are well regulated to zero for each key image. At time $t=3.9$ seconds, the quality of the image is very poor leading to an inaccurate estimation of the camera displacement as it can be observed in Fig. 18. Note that in this case, control inputs are filtered for a safer behaviour of the UAV.

6. Conclusion

A complete framework for autonomous navigation for an unmanned aerial vehicle which enables a vehicle to follow a visual path obtained during a learning stage using a single camera and natural landmarks has been proposed. The robot environment is represented as a graph of visual paths, called visual memory from which a visual route connecting the initial and goal images can be extracted. The vehicle can then be driven along the visual route thanks to a vision based control law which takes into account the dynamic model of the robot. Furthermore, the state of the robot, required for the control law computation, is estimated using a generic camera model valid for perspective, catadioptric as well as a large class of fisheye cameras. Experiments with an X4-flyer equipped with a fisheye camera have shown the validity of the proposed approach.

From a practical point of view, this navigation scheme is planned to be tested as soon as possible in outdoor environments. Future research works will be devoted first to improve the velocity estimator. Besides, another goal will be to robustly estimate the velocity using only the embedded camera employed for the navigation task. The second point is the improvement of the control law in order to be more robust to external perturbations such as the wind. Other perspectives include the study of a fully automatic scheme to build the visual memory, and the improvement of this navigation scheme in order to realize navigation tasks along paths which have not been realized during the learning stage.

Appendix

Proof of Theorem 4.1. Consider the storage function:

$$S=\tfrac{1}{2}\|\tilde{\mathbf{p}}\|^2 \tag{8}$$

Taking into account Eqs. (1) and the control input (4), the time derivative of S is $\dot{S}=\tilde{\mathbf{p}}^\top\mathbf{v}$. This equation may be written as

$$\dot{S}=-\kappa\tilde{\mathbf{p}}^\top \mathrm{sat}_\varepsilon(\tilde{\mathbf{p}})+\tilde{\mathbf{p}}^\top\tilde{\mathbf{v}} \tag{9}$$

The term $\tilde{\mathbf{p}}^\top\tilde{\mathbf{v}}$ acts as a perturbation on the position stabilization. If κ is chosen small compared to the translational dynamic gains then $\mathbf{v}_d$ is slowly changing and $\mathbf{v}$ tends to $\mathbf{v}_d$ faster than the convergence of $\mathbf{p}$ to $\mathbf{p}_d$. In this condition, $\tilde{\mathbf{v}}$ tends to zero and then $\dot{S}=-\kappa\tilde{\mathbf{p}}^\top \mathrm{sat}_\varepsilon(\tilde{\mathbf{p}})$. This function is definite negative which assures the convergence of $\mathbf{p}$ to $\mathbf{p_d}$. □

References

Angeli, A., Filliat, D., Doncieux, S., & Meyer, J.-A. (2006). 2D simultaneous localization and mapping for micro aerial vehicles. In *European micro aerial vehicles* (*EMAV* 2006), Braunschweig, Germany.

Baker, S., & Nayar, S. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, *35*(2), 1–22.

Barreto, J. (2006). A unifying geometric representation for central projection systems. *Computer Vision and Image Understanding*, *103*(3), 208–217. (special issue on Omnidirectional vision and camera networks).

Bourquardez, O., & Chaumette, F. (2007). Visual servoing of an airplane for alignment with respect to a runway. In *IEEE international conference on robotics and automation*, *ICRA*'07 (pp. 1330–1335), Rome, Italy.

Chitrakaran, V., Dawson, D., Kannan, H., & Feemster, M. (2006). *Assisted autonomous path following for unmanned aerial vehicles*. Technical Report, Clemson University CRB Technical Report, CU/CRB/2/27/06, March.

Courbon, J., Mezouar, Y., Eck, L., & Martinet, P. (2007). A generic fisheye camera model for robotic applications. In *IEEE/RSJ international conference on intelligent robots and systems*, *IROS*'07, San Diego, CA, USA, pp. 1683–1688, October 29–November 2.

Courbon, J., Mezouar, Y., Eck, L., & Martinet, P. (2008). Efficient hierarchical localization method in an omnidirectional images memory. In *IEEE international conference on robotics and automation*, *ICRA*'08 (pp. 13–18), Pasadena, CA, USA, May.

Courbon, J., Mezouar, Y., & Martinet, P. (2009). Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Transactions on Intelligent Transportation Systems*, *10*(3), 392–402.

Fischler, M., & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*, 381–395.

Frew, E., Langelaan, J., & Stachura, M. (2007). Adaptive planning horizon based on information velocity for vision-based navigation. In *AIAA guidance, navigation and controls conference*, Hilton Head, South Carolina, USA, August.

Gaspar, J., Winters, N., & Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera. In *VisLab-TR 12/2000—IEEE transaction on robotics and automation* (Vol. 16, pp. 890–898), December.

Geyer, C., & Daniilidis, K. (2003). Mirrors in motion: Epipolar geometry and motion estimation. In *International conference on computer vision, ICCV* 03 (pp. 766–773) Nice, France, October.

Goedemé, T., Tuytelaars, T., & Gool, L. J. V. (2004). Fast wide baseline matching for visual navigation. In *Computer vision and pattern recognition* (Vol. 1, pp. 24–29) Washington, DC, June–July.

Goedemé, T., Tuytelaars, T., Vanacker, G., Nuttin, M., Gool, L. V., & Gool, L. V. (2005). Feature based omnidirectional sparse visual path following. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1806–1811), Edmonton, Canada, August.

Guénard, N., Hamel, T., & Eck, L. (2006). Control laws for the tele-operation of an unmanned aerial vehicle known as X4-flyer. In *IEEE/RSJ international conference on intelligent robots and systems, IROS*'06 (pp. 3249–3254) Beijing, China, October.

Guénard, N., Hamel, T., & Mahony, R. (2007). A practical visual servo control for a unmanned aerial vehicle. In *IEEE international conference on robotics and automation, ICRA*'07 (pp. 1342–1348), Rome, Italy, April.

Guénard, N., Moreau, V., Hamel, T., & Mahony, R. (2008). Synthesis of a controller for velocity stabilization of an Unmanned Aerial Vehicle known as X4-Flyer through roll and pitch angles. *European Journal of Automated Systems (RS-JESA)*, *42*(1), 117–138.

Hamel, T., & Mahony, R. (2006). Attitude estimation on SO(3) based on direct inertial measurements. In *IEEE international conference on robotics and automation, ICRA*'06 (pp. 2170–2175), Orlando, FL, May.

Hamel, T., Mahony, R., Lozano, R., & Ostrowski, J.-N. (2002). Dynamic modelling and configuration stabilization for an X4-flyer. In *15th International federation of automatic control symposium, IFAC*'2002, Barcelona, Spain, July.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Alvey conference* (pp. 189–192).

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge University Press. ISBN: 0521623049.

Hu, G., Dixon, W., Gupta, S., & Fitz-Coy, N. (2006). A quaternion formulation for homography-based visual servo control. In *IEEE international conference on robotics and automation, ICRA*'06 (pp. 2391–2396), Orlando, FL, May.

Matsumoto, Y., Ikeda, K., Inaba, M., & Inoue, H. (1999) Visual navigation using omnidirectional view sequence. In *IEEE/RSJ international conference on intelligent robots and systems, IROS*'99 (Vol. 1, pp. 317–322), Kyongju, Korea, October.

Murillo, A., Guerrero, J., & Sagüés, C. (2007). Topological and metric robot localization through computer vision techniques. In *ICRA'07, workshop: from features to actions—unifying perspectives in computational and robot vision*, Rome, Italy, April.

Nikolos, I., Tsourveloudis, N., & Valavanis, K. (2002). Evolutionary algorithm based 3-D path planner for UAV navigation. In *10th Mediterranean conference on control and automation—MED* 2002, Lisboa, Portugal, July.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), 756–770.

Persson, P.-O., & Strang, G. (2004). A simple mesh generator in MATLAB. *SIAM Review*, *46*, 329–345.

Royer, E., Lhuillier, M., Dhome, M., & Lavest, J.-M. (2007). Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, *74*, 237–260. (special joint issue on Vision and robotics).

Sarris, Z. (2001). Survey of UAV applications in civil markets. In *9th Mediterranean conference on control and automation* (pp. 1–11) Dubrovnik, Croatia, June.

Svoboda, T., & Pajdla, T. (2002). Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, *49*(1), 23–37.