The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Visual navigation of a quadrotor aerial vehicle

Jonathan Courbon, Youcef Mezouar, Nicolas Guenard and Philippe Martinet

***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 and stored as a set of ordered key images (visual path) and organized providing 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 a X4-flyer equipped with a fisheye camera.**

I. Introduction

In [1], Sarris establishes a list of civilian applications for UAVs including border interdiction, search and rescue, wild fire suppression, communications relay, law enforcement, distaster 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 [2] and with the control of UAVs [3], few works propose navigation strategies. The most popular sensor for navigation of UAVs is the GPS receiver. In this case, the navigation task consists generally to reach a series of GPS waypoints. 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). For those reasons, it is necessary to use other sensors. The use of camera is very attractive to solve those problems because in place 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 vision-based navigation systems originally developed for ground vehicles have been transposed to the context of UAV navigation. For instance in [4], 2D Simultaneous Localization And Mapping (SLAM) techniques is used. In [5], a bearing-only SLAM is proposed. Vision based control scheme have been also employed to drive the UAV during the autonomous navigation step. An homography-based control scheme is proposed in [6]. However, this approach requires the camera to point to the ground supposed planar. In [7], an image-based control strategy using points as visual features is used for a positioning task. The centroid of artificial landmarks (white blobs) are used as visual features. In [8], a visual servoing scheme is proposed to align an airplane with respect to a runway in a simulated environment.

J. Courbon and N. Guenard are with CEA, LIST, F-92265 Fontenay-Aux-Roses, France firstname.lastname@cea.fr

J. Courbon, Y. Mezouar and P. Martinet are with LASMEA, F-63177 Aubière, France firstname.lastname@lasmea.univ-bpclermont.fr

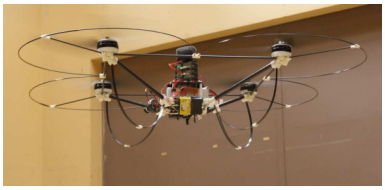

Fig. 1. Quad-rotor UAV used in our experiments

In this paper, we propose to control the VTOL UAV by successively reaching visual waypoints. Our framework does not need to recover the position of the UAV with respect to a reference frame. We suppose that images of the environment have been acquired during a human-guided navigation. These images are first sampled and stored as a set of key images (visual path) and organized providing a visual memory of the environment. The visual route to be followed by the UAV is then defined as a concatenation of visual path subsets connecting the initial and target images. 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.

The concept of visual memory is briefly described in Section II. The control strategy is presented in III-B. Experiments with a quadrotor UAV (see Figure 1) capable of quasistationary flight developed at CEA (French Atomic Energy Commission) equipped with a fisheye camera are finally presented in Section IV.

II. Visual memory and route building

In [9], navigation approaches using a memorization of images of the environment acquired with an embedded camera are ranked among mapless navigation systems. The first step of our framework consists on a learning stage to build the visual memory which structure is recalled in this section (refer to [10] for more details).

A. Visual memory

The visual memory is considered as a set of images $\{\mathcal{I}_i \mid i \in \{1, 2, \ldots, n\}\}$ connected to form a graph. Let $\mathcal{R}$ be the body fixed frame attached to the center of mass of the robot. Without loss of generality, we suppose that the camera frame is confounded 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. The following 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 ψ 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 $\mathcal{P}_{ij}$ of matched visual features, which can be observed along a path performed between $\mathcal{R}_i$ and $\mathcal{R}_j$ and which allows the computation of the control law.

If Hypothesis 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 [1].
The visual memory is thus structured as a graph with edges connecting key images.

B. 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 $\mathcal{I}_g$. $\mathcal{I}_s^*$ is the closest key image to the current image $\mathcal{I}_s$. The image $\mathcal{I}_s^*$ is extracted from the visual memory during a localization step. During this stage, no assumption about the vehicle's position is made. The localization process consists of finding the image which best fits the current image in the visual memory. With this aim, we use a hierarchical process combining global and local descriptors we have developped in a previous work [11].

C. Image matching

A central clue for implementation of our framework relies on efficient point matching. Interest points are detected in each image with Harris corner detector. Interest points between two images are then matched by using a Zero Normalized Cross Correlation score. This method is almost invariant to illumination changes and its computational cost is small. Note that matching has been preferred to tracking because it is more robust when images' information may be partially lost (this case frequently occurs when images are transmitted by wireless link as in our experiments).

III. Route following

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

[1]Depending on the vehicle, an edge is directed or not. 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 $\mathcal{R}_i$.

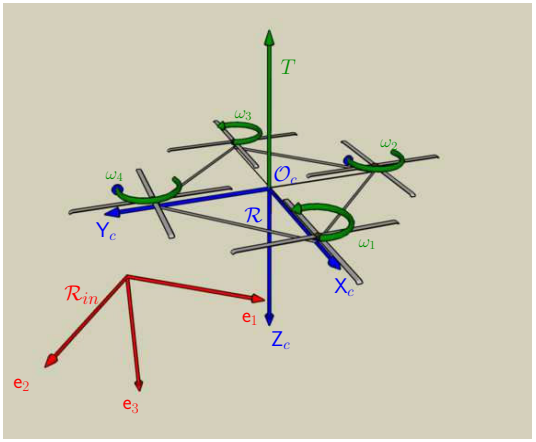

Fig. 2. The four rotors generating the collective thrust.

A. Model and assumptions

1) Vehicle Modelling: In this section, we derive equations of motion for a UAV in quasi stationary flight conditions following [12]. Let us define the inertial frame attached to the earth, relative to a fixed origin $\mathcal{R}_{\text{in}}(\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3)$ assumed to be Galilean and $\mathcal{R}(\mathsf{X}_c, \mathsf{Y}_c\, \mathsf{Z}_c)$ the frame attached to the UAV (refer to Fig. 2). The position of the center of mass of the robot with respect to the inertial frame $\mathcal{R}_{\text{in}}$ is denoted p. The orientation of the airframe is given by a rotation $\mathbf{R} : \mathcal{R} \to \mathcal{R}_{\text{in}}$. Let v (respectively Ω) be the linear (resp. angular) velocity of the center of mass expressed in the inertial frame $\mathcal{R}_{\text{in}}$ (resp. in $\mathcal{R}$). The control inputs to send to the vehicle are: T, a scalar input termed thrust or heave, applied in direction Z_c and $\Gamma = [\Gamma_1\, \Gamma_2\, \Gamma_3]^T$ (the control torques relative to the Euler angles). The geometry of the robot is supposed to be perfect. Let us denote m the mass of the airframe, g the gravity constant and $\mathbf{I}$ be the 3×3 constant inertia matrix around the centre of mass. Newton's equations of motion yield the following dynamic model for the motion of a rigid object:

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

B. Control Design

1) Translational control: In this Section, the translational dynamic is supposed to be controlled by an on-board controller [13]. Consequently, the desired matrix $\mathbf{R_d}$ is already assigned to have the velocity v converging to the desired velocity $\mathsf{v_d}$ and the control torques $\mathbf{\Gamma}$ are assigned in order to have the rotational matrix $\mathbf{R}$ converging to this desired matrix $\mathbf{R_d}$. This control assures that the tilt angle is limited to small-angle and that the velocity is bounded in order to stay in quasi-stationary flight conditions if $\mathsf{v_d}$ allows this condition. The goal of this

part is to control the position of the UAV in assigning the desired translational velocity and to assure that the system stays in quasi-stationary flight conditions. Let us define the position error $\tilde{\mathsf{p}}$ and the velocity error $\tilde{\mathsf{v}}$:

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

$$\tilde{\mathsf{v}} = \mathsf{v} - \mathsf{v_d} \tag{3}$$

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

Theorem 3.1: The control input defined by:

$$\mathsf{v_d} = -\kappa \text{sat}_\epsilon(\tilde{\mathsf{p}}) \tag{4}$$

with κ small compared to the translational dynamic gains, is stabilizing the system 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.

Proof: Let us consider the storage function:

$$S = \frac{1}{2}||\tilde{\mathsf{p}}||^2 \tag{5}$$

Taking into account Equation (1) and the control input (4), the time derivative of S is $\dot{S} = \tilde{\mathsf{p}}^T \mathsf{v}$ and this Equation may be written:

$$\dot{S} = -\kappa \tilde{\mathsf{p}}^T \text{sat}_\epsilon(\tilde{\mathsf{p}}) + \tilde{\mathsf{p}}^T \tilde{\mathsf{v}} \tag{6}$$

The term $\tilde{\mathsf{p}}^T \tilde{\mathsf{v}}$ acts as a perturbation on the position stabilization. If we assume that this term converges to 0 according to the translational gains, then $\dot{S} = -\kappa \tilde{\mathsf{p}}^T \text{sat}_\epsilon(\tilde{\mathsf{p}})$. This function is definite negative which assures the convergence of p to $\mathsf{p_d}$. In order to limit the effect of the perturbation term, we have to choose κ small compared to the translational dynamic gains. ■

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. Let us note $\mathcal{R}_i = (O_i, \mathsf{X_i}, \mathsf{Y_i}, \mathsf{Z_i})$ and $\mathcal{R}_{i+1} = (O_{i+1}, \mathsf{X_{i+1}}, \mathsf{Y_{i+1}}, \mathsf{Z_{i+1}})$ the frames attached to the vehicle when $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ were stored and $\mathcal{R}_c = (O_c, \mathsf{X_c}, \mathsf{Y_c}, \mathsf{Z_c})$ 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 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 $\mathcal{R}_{i+1}$. If the lateral error $\tilde{\mathsf{p}}$ and yaw error $\tilde{\theta}$ of $\mathcal{R}_c$ with respect to $\mathcal{R}_{i+1}$ are regulated to zero thus the task is achieved. In the next Section, we describe how 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 $\tilde{\mathsf{p}}$ and $\tilde{\theta}$ are derived.

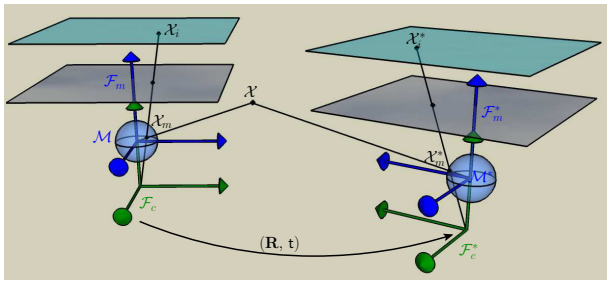

Fig. 3. Geometry of two views.

C. State estimation from the generic camera model

In this work, we propose to use the unified model described in [14], 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 [14]: perspective and catadioptric cameras as well as some fisheye cameras.

1) Camera model: The unified projection model consists of a central projection onto a virtual unitary sphere followed by a perspective projection onto the image plane [14]. This generic model is parametrized by ξ describing the type of sensor and by a matrix $\mathbf{K}$ containing the intrinsic parameters. The coordinates x_i of the point in the image plane corresponding to the 3D point $\mathcal{X}$ are obtained after three steps:

Step 1 : the world points $\mathcal{X}$ of coordinates $\mathsf{X} = [X\ Y\ Z\ ]^T$ in the camera frame $\mathcal{R}_m$ are projected onto the unit sphere on a point $\mathcal{X}_m$ of coordinates X_m in $\mathcal{R}_m$: $\mathsf{X_m} = \mathsf{X}/\|\mathsf{X}\|$

Step 2 : the point coordinates are then changed to a new reference frame $\mathcal{R}_c$ centered in $C = (0,\ 0,\ -\xi)$ and perspectively projected onto the normalized image plane $Z = 1 - \xi$:

$$\begin{aligned} \underline{\mathsf{x}}^\top &= [\mathsf{x}^\top\ 1] = [x\, y\, 1] = f(\mathsf{X}) \\ &= \begin{bmatrix} \dfrac{X}{Z + \xi\|\mathsf{X}\|} & \dfrac{Y}{Z + \xi\|\mathsf{X}\|} & 1 \end{bmatrix} \end{aligned} \tag{7}$$

Step 3 : finally, the coordinates $\underline{\mathsf{x}}_i^\top = [\mathsf{x}_i^\top 1]$ in the image plane are obtained after a plane-to-plane collineation $\mathbf{K}$ of the 2D projective point $\underline{\mathsf{x}}$: $\underline{\mathsf{x}}_i = \mathbf{K}\underline{\mathsf{x}}$.

We highlight that X_m can be computed as a function of the coordinates in the image and the sensor parameter ξ:

$$\mathsf{X}_m = (\eta^{-1} + \xi)\overline{\mathsf{x}} \tag{8}$$

$$\overline{\mathsf{x}} = \begin{bmatrix} \mathsf{x}^T & \dfrac{1}{1 + \xi\eta} \end{bmatrix}^T$$

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

2) Scaled Euclidean reconstruction: Let $\mathcal{X}$ be a 3D point with coordinates $\mathsf{X_c} = [X_c\ Y_c\ Z_c\ ]^T$ in the current frame $\mathcal{F}_c$ and $\mathsf{X}^* = [X_{i+1}\ Y_{i+1}\ Z_{i+1}\ ]^T$ in the frame $\mathcal{F}_{i+1}$. Let $\mathsf{X_m}$ and X^*_m be the coordinates of those points, projected onto the unit sphere (refer to Fig. 3).

The epipolar plane contains the projection centers O_c and O_{i+1} and the 3D point X. $\mathsf{X_m}$ and X^*_m clearly belong to this plane. The epipolar relation is obtained from this complanarity condition and is written as in the case of the pinhole model:

$$\mathsf{X_m}^{*T}\mathbf{E}\mathsf{X_m}^{T} = 0 \tag{9}$$

where $\mathbf{R}$ and t represent the rotational matrix and the translational vector between the current and the desired frames expressed in the current vehicle frame and $\mathbf{E} = \mathbf{R}[\mathsf{t}]_\times$ is the essential matrix.

In Equation (9), $\mathsf{X_m}$ (respectively 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 thanks to the relation (8) and to the coordinates of the point matched in the first and second images. The essential matrix $\mathbf{E}$ between two images can be estimated using five couples of matched points as proposed in [15] if the camera calibration (matrix $\mathbf{K}$) is known. Outliers are rejected using a random sample consensus (RANSAC) algorithm. From the essential matrix, the camera motion parameters (that is the rotation $\mathbf{R}$ and the translation t up to a scale) can be determined. Finally, the estimation of the input of the control law (4), *i.e* $\tilde{\mathsf{p}}$ can be computed straightforwardly from t up to a scale factor and the yaw error $\tilde{\theta}$ is extracted from $\mathbf{R}$.

IV. Experimental results

In this section, we discuss the results obtained with our experimental platform. The UAV used for the experimentation is a quadrotor manufactured by the CEA (see Figure 1). It is a Vertical Take Off and Landing (VTOL) vehicle ideally suited for stationary and quasi stationary flight [3].

A. 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. The final board provides a serial wireless communication between the operator's joystick and the vehicle. The embedded camera with a view angle of 120 degrees pointing in front, transmits 640x480 pixels images to a laptop using RTAI-Linux OS with a 2GHz Centrino Duo processor via a wireless analogical link. Vision algorithms are implemented in C^{++} language in this laptop. The images sent by the embedded camera are received at a frequency of 12.5fps and processed. Then, the state required by the control law is sent to the ground station (PC) by an ethernet connection. In parallel, the X4-flyer sends the inertial data to the station on the ground at a frequency of 15Hz. Desired orientation and desired thrust are generated on the ground station PC and sent to the drone. The velocity in X and Y directions is estimated on-board as well as the attitude of the UAV. For orientation dynamics, an embedded high gain controller in the DSP running at 166Hz, independently ensures the exponential stability of the orientation towards the desired one.

Fig. 4. Key image to reach (Exp. 1)

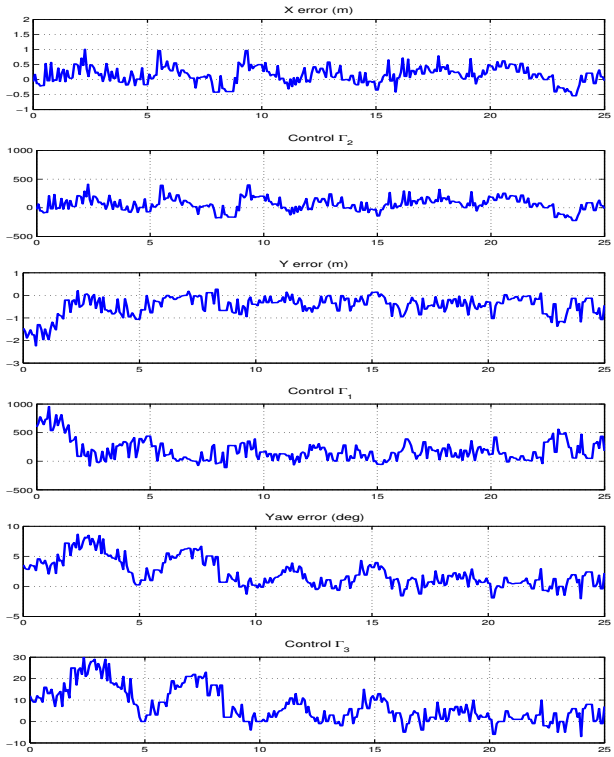

Fig. 5. Errors in translation along the X-axis and Y-axis and in rotation (yaw angle) and associated control input vs time (s) (Exp. 1)

B. Waypoint reaching (Exp. 1)

This section deals with the vision-based control of the UAV in order to reach the key image drawn in Fig.4. The robot is manually guided to an initial position and then automatically controlled in order to reach the key image. Results are reported in Fig. 5. Errors in translation are expressed in meters and yaw angle in degrees, versus time (in seconds).

The experiment lasts 25 seconds. A mean of 73 robust matches for each frame has been found. The mean computational time during the online navigation is of 94 ms / image. X and Y errors are converging to zero as well as the yaw angle. 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

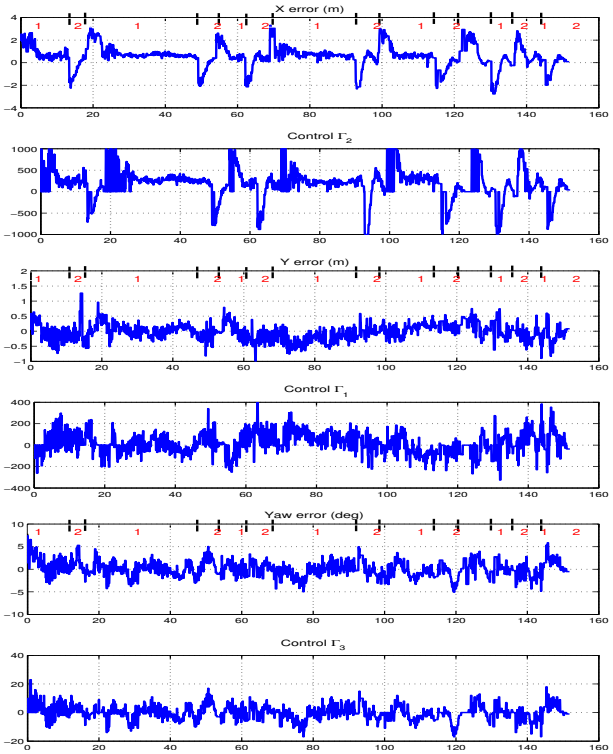

Fig. 7. Errors in translation along the X-axis and Y-axis and in rotation (yaw angle) and associated control input vs time (s) (Exp. 2)

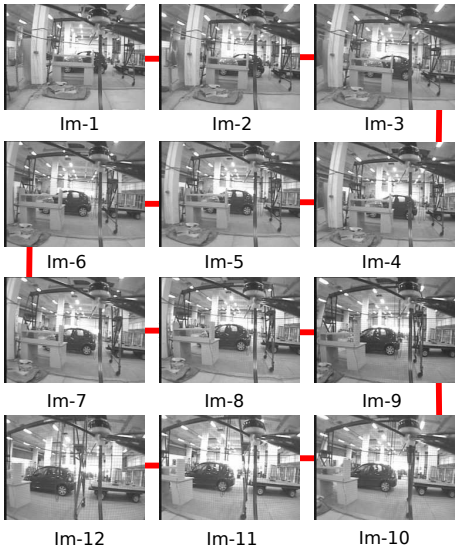

Fig. 8. Visual memory of the UAV consisting on a graph with 12 key images.

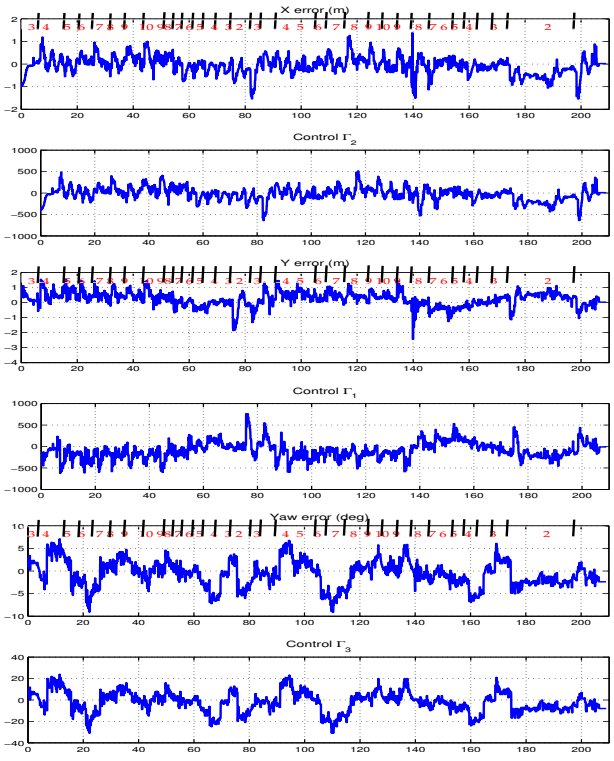

Fig. 9. Errors in translation along the X-axis and Y-axis and in rotation (yaw angle) and associated control input vs time (s) (Exp. 3)

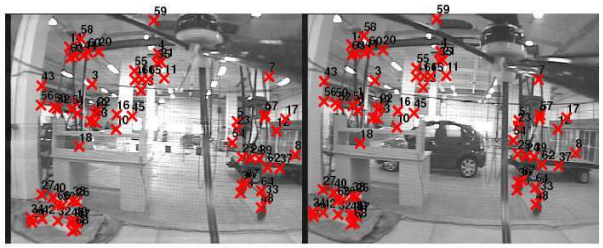

Fig. 10. Robustly matched features between the current image (left) and the image to reach (left; Im-6)

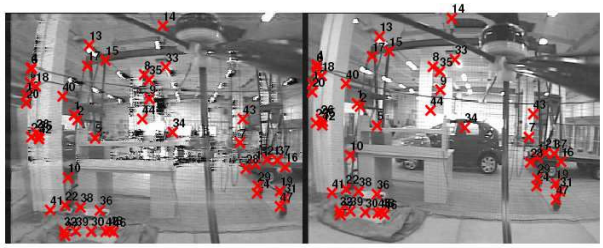

Fig. 11. Robustly matched features between the current image (left) and the image to reach (left; Im-5)

(a) Key image 1 (b) Key image 2

Fig. 6. Key images to successively reach (Exp. 2)

by the asynchronous sensors' data. Nevertheless, the results show the very good behaviour of the visual servoing.

C. Succession of two images (Exp. 2)

In this second experiment, we use two key images as targets (refer to Fig. 6). 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). 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 XY plane is under a threshold. Note that the two key images are approximately situated in the direction of the vehicle. Translations thus occur more in X-axis direction. Results are reported in Fig. 7. In the figures vertical dotted lines denote that a key image is reached. After each change of desired key image, error in Y direction and yaw angles are converging to zero. Error in X direction is also converging. A static error in X direction remains due to errors in velocity estimation. Future works will deal with this point.

D. Waypoints following (Exp. 3)

The UAV is now manually controlled along an approximately linear path situated in the XY plane and at 45° from the direction of the UAV (X_c) and images are acquired by the embedded camera pointing forward. After a selection process, a single edge containing 12 key images is extracted and stored in the visual memory (refer to Fig. 8). 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. Results are drawn in Fig. 9. Even if errors in X and Y directions and in yaw angles 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. 10 (68 matched points) and Fig. 11 (48 matched points). In Fig. 11, the current image has a low quality. Despite this fact, many points have been matched.

V. Conclusion

We have presented a framework for autonomous navigation of an unmanned aerial vehicle using a single camera and natural landmarks and without any recovery of the pose of the vehicle in a reference frame. The robot environment is modeled as a graph of images, called visual memory from which a visual route connecting the initial and goal images can be extracted. The UAV is driven along the images thanks to a vision based control law which takes into account its dynamic model. Note that only a single camera and natural landmarks have been used for control purpose.
Future works will deal with the improvements of the velocity estimation and of the robustness of the visual algorithms. Moreover, the fusion of the camera with other sensors may improve the quality of the path following as well as the robustness to missing or temporarily unavailable sensor data.

Acknowledgment

This work is supported by the EU-Project FP6 IST μDrones, FP6-2005-IST-6-045248. The authors gratefully acknowledge Laurent Eck for his help with problems regarding the UAV control and experiments.

References

[1] Z. Sarris, "Survey of UAV applications in civil markets," in *9th Mediterranean Conference on Control And Automation*, Dubrovnik, Croatia, June 27–29 2001.

[2] T. Hamel and R. Mahony, "Attitude estimation on SO(3) based on direct inertial measurements," in *IEEE International Conference on Robotics and Automation, ICRA'06*, Orlando, Florida, may 2006, pp. 2170–2175.

[3] N. Guénard, T. Hamel, and L. Eck, "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*, Beijing, China, oct 2006, pp. 3249–3254.

[4] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "2D simultaneous localization and mapping for micro aerial vehicles," in *European Micro Aerial Vehicles (EMAV 2006)*, 2006.

[5] E. Frew, J. Langelaan, and M. Stachura, "Adaptive planning horizon based on information velocity for vision-based navigation," in *AIAA Guidance, Navigation and Controls Conference*, Hilton Head, South Carolina, USA, August 2007.

[6] G. Hu, W. Dixon, S. Gupta, and N. Fitz-Coy, "A quaternion formulation for homography-based visual servo control," in *IEEE International Conference on Robotics and Automation, ICRA'06*, Orlando, Florida, may 2006.

[7] N. Guénard, T. Hamel, and R. Mahony, "A practical visual servo control for a unmanned aerial vehicle," in *IEEE International Conference on Robotics and Automation, ICRA'07*, Rome, Italy, apr 2007, pp. 1342–1348.

[8] O. Bourquardez and F. Chaumette, "Visual servoing of an airplane for alignment with respect to a runway," in *IEEE International Conference on Robotics and Automation, ICRA'07*, Rome, Italy, apr 2007, pp. 1330–1335.

[9] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 237–267, february 2002.

[10] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, 2008.

[11] J. Courbon, Y. Mezouar, L. Eck, and P. Martinet, "Efficient hierarchical localization method in an omnidirectional images memory," in *IEEE International Conference on Robotics and Automation, ICRA'08*, Pasadena, CA, USA, may 19-23 2008.

[12] T. Hamel, R. Mahony, R. Lozano, and J.-N. Ostrowski, "Dynamic modelling and configuration stabilization for an x4-flyer." in *15th International Federation of Automatic Control symposium, IFAC'2002*, Barcelona, Spain, July, 21–26 2002.

[13] N. Guénard, V. Moreau, T. Hamel, and R. Mahony, "Synthèse d'un contrôleur permettant la stabilisation de vitesse d'un drone de type x4-flyer *via* la correction d'assiette," *Journal européen des systèmes automatisés (RS-JESA)*, vol. 42, no. 1, pp. 117–138, 2008.

[14] C. Geyer and K. Daniilidis, "Mirrors in motion: Epipolar geometry and motion estimation," in *International Conference on Computer Vision, ICCV03*, Nice, France, 2003, pp. 766–773.

[15] D. Nistér, "An efficient solution to the five-point relative pose problem," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.