HAL
open science

Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles

Farid Kendoul, Isabelle Fantoni, Kenzo Nonami

To cite this version:

Farid Kendoul, Isabelle Fantoni, Kenzo Nonami. Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. Robotics and Autonomous Systems, 2009, 57 (6-7), pp.591-602. 10.1016/j.robot.2009.02.001 . hal-00445972

HAL Id: hal-00445972
https://hal.science/hal-00445972v1

Submitted on 11 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optic Flow-Based Vision System for Autonomous 3D Localization and Control of Small Aerial Vehicles

Farid Kendoul [a,*], Isabelle Fantoni [a], Kenzo Nonami [b]

[a] *Heudiasyc Lab., UMR CNRS-UTC 6599, University of Technology of Compiegne, 60200 Compigne, France*

[b] *Robotics and Control Lab., Electronics and Mechanical Engineering Dept., Chiba University, 263-8522, Chiba City, Japan*

Abstract

The problem considered in this paper involves the design of a vision-based autopilot for small and micro Unmanned Aerial Vehicles (UAVs). The proposed autopilot is based on an optic flow-based vision system for autonomous localization and scene mapping, and a nonlinear control system for flight control and guidance. This paper focusses on the development of a real-time 3D vision algorithm for estimating optic flow, aircraft self-motion and depth map, using a low-resolution onboard camera and a low-cost Inertial Measurement Unit (IMU). Our implementation is based on 3 Nested Kalman Filters (3NKF) and results in an efficient and robust estimation process. The vision and control algorithms have been implemented on a quadrotor UAV, and demonstrated in real-time flight tests. Experimental results show that the proposed vision-based autopilot enabled a small rotorcraft to achieve fully autonomous flight using information extracted from optic flow.

Key words: UAV, autonomous localization, optic flow, Structure-From-Motion (SFM), flight guidance and control, visual SLAM.

* Corresponding author.
Email addresses:
`fkendoul@restaff.chiba-u.jp` (Farid Kendoul),
`ifantoni@hds.utc.fr` (Isabelle Fantoni),
`nonami@faculty.chiba-u.jp` (Kenzo Nonami).

1 Introduction

Recent advances in cost effective inertial sensors and accurate navigation systems, such as the GPS, have been key determinants of the feasibility of UAV systems. Milestones in manned and unmanned aircraft have been achieved using conventional navigation sensors such as standard IMUs for orientation, GPS for position, pressure sensors for altitude sensing, radar, ultrasounds and laser range finder for obstacles detection. Our particular interests, however, involve small and micro UAVs flying close to the ground in cluttered environments like urban and indoor environments. Therefore, GPS information may not be available. Furthermore, the substantial weight and energy constraints imposed by small and micro UAVs preclude the use of conventional sensors. On the other hand, visual sensors are passive, lightweight and can provide rich information about the aircraft self-motion and surroundings structure. Therefore, computer vision can be used for autonomous localisation, which is a crucial step for small aerial robots control and guidance. However, the design of a reliable vision system for aerial vehicles has many unsolved problems, ranging from hardware and software development to pure theoretical issues, which are even more complicated when applied to small flying machines operating in unstructured environments. Moreover, the difficulty found when using imaging sensors is the high bandwidth of data, and the resulting heavy computational burden.

Using computer vision for autonomous localization leads to the visual servoing problem [1]. Recently, there is a growing interest in applying visual navigation principles of insects for UAVs control and guidance. Indeed, recent experimental research in biology has discovered a number of different ways in which insects use optic flow in order to deal with the 3D flight control problem [2,3]. We believe that flying insects can provide us with ingenious solutions and efficient flight behaviours which are appropriate for implementation on artificial flying machines. We thus, take inspiration from insects in order to develop a vision-based autopilot that functionally imitates the vision-control system of flying insects. indeed, the proposed autopilot is based on a minimum sensor suite that includes a single onboard camera and a low-cost IMU. This paper focuses on the development of a real-time vision algorithm that jointly performs optic flow computation and interpretation (recovery of 3D motion and structure) using insect-inspired sensory systems. Our motivation for this work is to develop a fully embedded, lightweight, and low-cost solution for autonomous localization in arbitrary and unknown environment using optic flow. The proposed computational framework is based on 3 Nested Kalman Filters (3NKF) which allowed to combine three algorithmic concepts stemming from different areas of research (optic flow computation, data fusion, SFM problem) in a favorable manner (see Fig. 1). The novelty of the approach is based on the fact that each sub-system is connected to other systems, thereby allowing

bidirectional exchange of data between them. The resulted 3NKF-based algorithm is fast, accurate and robust, which make it suitable for aerial robotic applications.

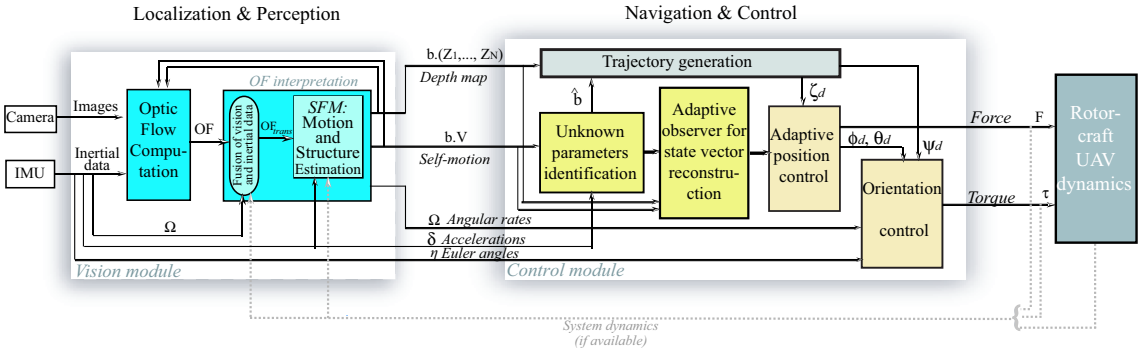

Fig. 1. Optic flow-based autopilot for small rotorcraft UAVs control and guidance

Section 2 summarizes related research in optic flow computation and interpretation, and presents some existing works on bio-inspired vision-based aerial navigation. In Sections 3 and 4, the philosophy behind the vision algorithm as well as its mathematical bases are described. Section 5 provides details about the aerial platform and the real-time implementation of the vision-control system. Experimental results are presented and discussed in Section 6. Conclusions are given in Section 7 with some directions for future work.

2 Related work and the proposed 3NKF framework

In this section, we give a short overview on existing methods for optic flow computation, cite some approaches for camera ego-motion and structure estimation, and present some examples about the application of image optic flow for UAVs navigation control.

2.1 Optic flow computation

The optic flow is defined as *the apparent motion of the image intensities caused by the 2D projection onto a retina of the relative 3D motion of scene points.* The standard optic flow techniques can be classified into four main groups according to the assumptions they make: *differential or gradient methods* [4,28], *correlation and block matching schemes* [5], *energy and phase-based methods* [6], and *sensor-based approaches* [7]. The simplest and the most used technique is the image matching or correlation method. However, this technique suffers

from lack of sub-pixel precision, quadratic computational complexity and inaccuracy in presence of image deformation due to rotation. On the other hand, the well known differential models suffer from sensitivity to noise due to derivative computation, and their inability to handel large optical flow. A general way of circumventing these problems is to apply optical flow techniques in a hierarchical, coarse-to-fine framework [5]. Although the hierarchical techniques improve the optic flow computation, they suffer from a severe trade-off between efficiency and the maximum image displacement that can be computed. A review and comparison of the most popular OF algorithms can be found in [8]. Much progress has been made in optical flow computation and yet, its efficient and accurate estimation in real-time remains difficult, especially when it is used for robotics applications.

2.2 Structure From Motion problem

The problem of Structure From Motion (SFM) concerns the estimation of the camera ego-motion and the reconstruction of the 3D structure of a scene from its projection onto a moving two-dimensional surface (image sequences). SFM has been a central problem in computer vision for many years, and the literature comprises a variety of schemes that differ in the description of the system model (linear, non-linear), the projection model (orthographic, affine, perspective), input measurements (optic flow, feature tracking, image brightness), time-frame (continuous-time or discrete-time models), and data processing techniques (batch optimization, recursive estimation). The paper [9] provided a critical study of existing SFM techniques.

SFM estimation using sparse feature correspondences has been investigated for nearly 30 years [10,27,15]. Because of the high-speed requirement, low image quality, and rapidly changing of camera attitude, feature tracking may not be a trivial task [11]. Optic flow-based algorithms are an interesting alternative for the SFM problem [12]. Our choice for using optic flow is also motivated by the fact that insects rely heavily on optic flow to extract useful information for flight control and navigation. Given a set of measured optic flows, motion and structure parameters can be recovered modulo some scale factor using optimization algorithms like least-squares technique [13]. Fusing-based algorithms like Extended Kalman Filter (EKF) [14,27] are the second traditional approach to SFM after optimization framework.

There are many critical issues and limitations that are related to the SFM problem. Firstly, extracting useful visual information (features detection and tracking or optic flow computation) in real-time and in natural environments is a challenging task which is not entirely solved yet. Secondly, it is well known that the mathematical system describing the SFM problem is not observable

[11]. Hence, absolute translational velocities and real distances to perceived objects can not be recovered. This is known as the scale factor problem. Thirdly, ambiguities in 3D motion recovery from noisy flow fields have been reported by many researchers [15,16]. One dominant ambiguity arises from the similarity between the flow fields generated by translation parallel to the image plane and associated rotation. An interesting way to eliminate this translation-rotation confusion is to exploit angular rate data obtained from a camera-mounted rate sensor [15].

2.3 Bio-inspired vision-based aerial navigation

Many researchers have been interested by the world of flying insects, and recent experimental research in biology has discovered a number of different ways in which insects use cues derived from optical flow for navigational purposes. Indeed, insects like bees and flies have evolved alternative, simple and ingenious stratagems for dealing with the problem of 3D vision to perform navigational tasks. These behaviors originated in research on insect flight are appropriate for implementation in a biomimetic autopilot for small UAVs and robotics in general [17,18,19]. Potential applications of optical flow for small aerial vehicles include altitude control and terrain following [20,21], autonomous landing [20,22,23] and obstacles avoidance [24,25,26].

In the last decade, very interesting and promising results have been obtained on insect-inspired flying robots. The realized experiments proved the effectiveness and usefulness of insects to provide artificial systems with innovative solutions. The implementation of such biologically-inspired autopilots on real UAVs evolving in unstructured environments, however, remains an unresolved problem. The OF-based strategies, previously presented, consider simple cases, where the UAV motion is usually limited to motion in a horizontal or vertical plane with one or two degrees of freedom. In real-world applications, an UAV may execute an arbitrary 3D motion. Therefore, computing and interpreting robustly and accurately the OF issued from 3D motion is a challenging task. So, to achieve real navigational tasks in natural environments, appropriate and robust 3D vision algorithms are required.

2.4 Brief description of the proposed framework

Applying optic flow for controlling aerial vehicles and robots in general is a multidisciplinary research field that includes image processing (OF computation), computer vision (OF interpretation or SFM problem), robotics and control (vision-based control). The vast majority of classical approaches for vision-based control of robots treat separately the three mentioned problems.

We believe that although there have been good examples of vision-based control of UAVs, there is a gap between the research conducted in computer vision, control and robotics communities. In robotics, vision can not be considered as an isolated component, but it is instead a part of a system resulting in an action. Thus, in our research work, we have considered the entire perception-action loop. In other words, the vision system included consideration of the controller and vice versa (see Figure 2). We have thus, proposed a robotics-

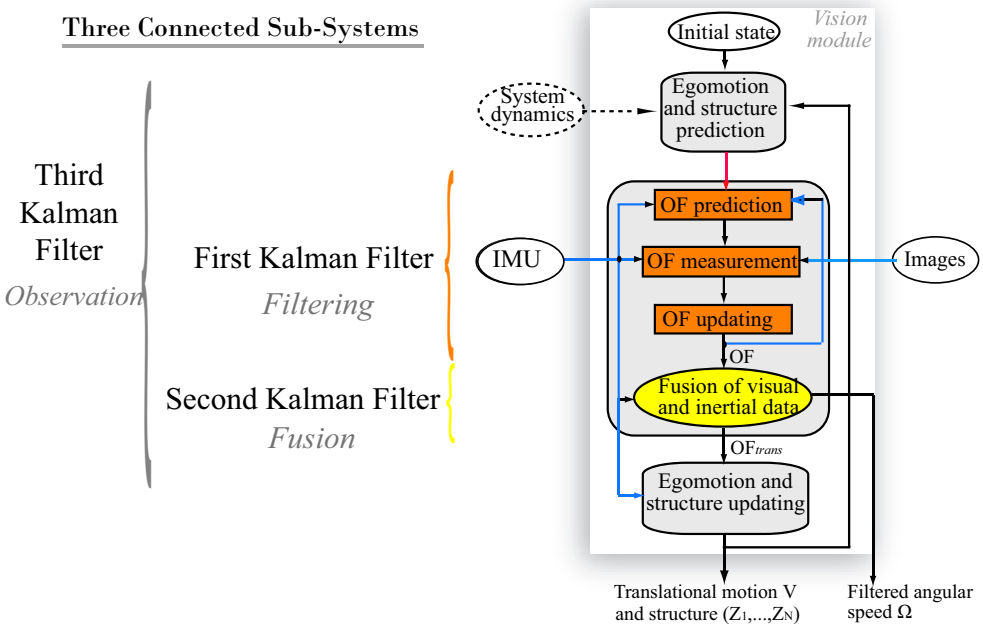

Fig. 2. 3NKF-based vision algorithm for optic flow computation and interpretation

oriented framework which is inspired from insects and takes advantage from robotics systems specifications. Functionally, the proposed vision system is decomposed into three major blocks(OF computation, fusion of visual and angular rate data, motion and structure recovery) as shown in Figs. 1 and 2. In fact, the proposed framework is based on three Nested Kalman Filters (3NKF) allowing effective and useful connection (exchange of data) between the different sub-systems. We have designed an efficient OF algorithm that has as inputs image sequences and angular rate data. SFM sub-system predictions are also exploited in order to improve the efficiency and robustness of optic flow computation. Then, the estimated OF is considered as measurements for the second Kalman Filter (KF) that fuses visual data and inertial measurements in order to cancel the rotational component of OF. Finally, the estimated translational component of OF is exploited by an EKF-based SFM algorithm in order to recover the translational velocity and the scene structure. Therefore, the difference between our algorithm and most other SFM methods like those in [27] and [15] is significant in that the 3NKF-based algorithm jointly performs accurate optic flow estimation and robust 3D interpretation in real-time.

3 Prediction-based algorithm with adaptive patch for accurate and efficient optic flow computation

In this section, we present an OF algorithm that has many advantages for robotics applications. Indeed, we present an efficient algorithm that combines matching and differential techniques for accurate measurement of large and sub-pixel OF. The proposed method takes advantage of UAV dynamics constraining the camera motion to be continuous and smooth. Therefore, based on the 3NKF framework, inertial data and SFM-module outputs (velocity and depth) are exploited to predict the displacement of a given patch (block of pixels) in the subsequent images with adapting the patch shape (deformation), thereby limiting or reducing the search area and avoiding erroneous measures. Thus, a block matching technique computes efficiently the nominal image displacement without suffering from the main issues of standard block matching methods namely, quadratic complexity and sensibility to deformations. Once the nominal displacement $d_n \in \mathbb{Z}^2$ is computed, we translate the new image I_2 by this amount (i.e., d_n) and we obtain a translated image I_2'. Now, the displacement between the reference image I_1 and the new translated image I_2' does not exceed one pixel, and the well-known Lucas-Kanade [28] differential algorithm may compute easily and accurately the remaining sub-pixel displacement $d_s \in \mathbb{R}^2$. Finally, the total image displacement is obtained by summing d_n and d_s, which is then filtered with a Kalman Filter. The main steps of the proposed OF algorithm are shown in Figure 3, and they are described in the following subsections.

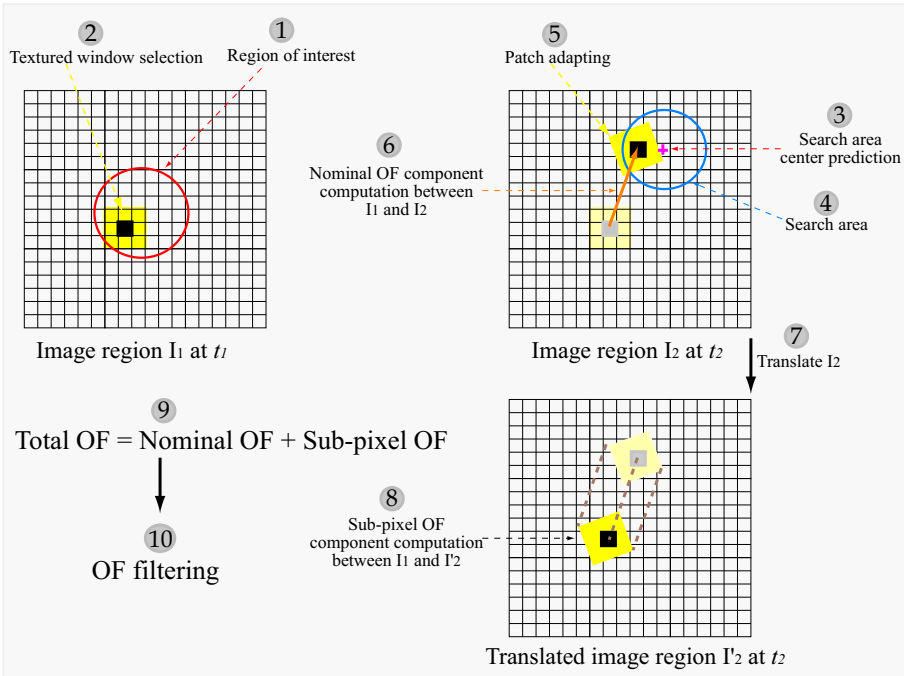

Fig. 3. Prediction-based OF algorithm with adaptive patch

3.1 Search center prediction

In order to overcome the major limitation (computational complexity) of block matching algorithms, we have developed a new block-based OF estimation algorithm that employs motion vector prediction to locate an initial search point, which is called a search center.

The image motion depends on camera motion and the structure of the observed scene. Thus, we use the predicted camera motion (V_{pred}), the predicted structure (Z_{pred}) and the measured angular velocity vector Ω in order to predict the image displacement $d_{pred} \in \mathbb{R}^2$ in the next frame. This prediction process is possible thanks to the 3NKF scheme which connects the different modules. Therefore, the predicted position in the new image I_2 of some pixel located at X_1 in the reference image I_1 is given by

$$X_{pred} = X_1 + d_{pred} \tag{1}$$

X_{pred} is considered as the center of the Search Area (SA) that contains the true position X_2 of the moved pixel. In classical Full Search (FS) matching algorithms, SA is centered at X_1 with a radius chosen equal to the maximum expected image displacement d_{max}. In our case, the chosen radius r is rather equivalent to the variation of image displacement between two subsequent frames. Therefore, r is much lower than d_{max}. In simulations and real-time experiments, r is set to 3 for the computation of image displacements that exceed 25 pixels/frame [29]. The size of the search area is $(r+1) \times (r+1)$ which is independent of the amplitude of image displacement.

3.2 Combined block-matching and differential algorithm

A. Nominal OF computation using a Block-Matching Algorithm (BMA):

The BMA approximates the image motion by a displacement $d = (d_x, d_y)$ that yields the best match between image regions at different times. In other words, to determine the motion of a pixel $X_1 = (x_1, y_1)$ in a reference image $I_1(x, y, t)$, we choose a patch P_ν (block of pixels) centered at (x_1, y_1) and composed of $\nu \times \nu$ pixels. We will then try to find the correspondence of this patch in the successive image $I_2(x, y, t+\delta t)$ by minimizing the following cost function (Sum of Absolute Differences SAD) among the search area.

$$SAD(X_1, d) = \sum_{i=-\nu}^{\nu} \sum_{j=-\nu}^{\nu} |I_1(x_1+i, y_1+j, t) - I_2(x_1+i+d_x, y_1+j+d_y, t+\delta t)| \tag{2}$$

Then, the nominal displacement d_n obtained for the block P_ν located at X_1

can be generally formulated as follows:

$$d_n(X_1) = \arg\min_{d \in SA}(SAD(X_1, d)) \tag{3}$$

Let us define $SAD_n = SAD(X_1, d_n)$. This matching error is used to detect optical flow discontinuities. Indeed, when SAD_n is higher than some user-defined threshold, SA is enlarged progressively until finding the true displacement.

In order to improve the accuracy and robustness of this efficient block-matching algorithm with respect to image deformation, we have integrated the inertial rate data into the matching process. Therefore, the shape of the patch is adapted by modifying equation (2) as follows

$$\begin{aligned} SAD(X_1, d) = & \sum_{i=-\nu}^{\nu} \sum_{j=-\nu}^{\nu} |I_1(x_1 + i, y_1 + j, t) - I_2((x_1, y_1) \\ & + \underbrace{\varphi(x_1{+}i, y_1{+}j)^T - \varphi(x_1, y_1)^T}_{adaptive\ \ term} + (d_x, d_y), t + \delta t)| \end{aligned} \tag{4}$$

with $\varphi(i, j)$ is a transformation given by (see eq. (14))

$$\varphi(i, j) = \begin{bmatrix} \beta ij & -(\frac{1}{\beta} + \beta i^2) & j \\ (\frac{1}{\beta} + \beta j^2) & -\beta ij & -i \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \tag{5}$$

B. Sub-pixel OF computation using a Differential Algorithm (DA):

We know that the BMA is not accurate enough since the measured displacement d_n is a signed integer. Then, in order to improve the accuracy of our algorithm, we use a complementary DA that computes the sub-pixel component d_s (floating part) of image displacement.

Differential methods are based on the assumption that the observed brightness I of any object point is constant over time. This assumption is mathematically stated as

$$I_1(x, y, t) = I_2(x + \delta x, y + \delta y, t + \delta t) \tag{6}$$

with $(\delta x, \delta y)$ is the image displacement during the inter-frame time δt. By applying Taylor's series about (x, y, t), we obtain the standard OF equation: $\nabla I.(\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t}) + I_t = 0$, with $\nabla I = (I_x, I_y)$ are the intensity spatial derivatives and I_t is the temporal derivative. The validity of the later equation requires small image displacements, in general lower than the pixel. This is the main limitation of differential methods since they can not compute large image displacements. To overcome this problem, we have modified (6) by translating

the image I_2 using the previously computed displacement d_n by the BMA. In fact, the vector movement could be decomposed into nominal and small displacements, $\delta x = d_{n_x} + d_{s_x}$ and $\delta y = d_{n_y} + d_{s_y}$. Thus, we write

$$I_1(x, y, t) = I_2(x + d_{n_x} + d_{s_x}, y + d_{n_y} + d_{s_y}, t + \delta t) \tag{7}$$

At this stage, the only unknown variables in equation (7) are (d_{s_x}, d_{s_y}) because d_n has been already computed by the BMA (3). Then, translating I_2 by subtracting d_n, we obtain the new translated image I_2' and we write

$$I_1(x, y, t) = I_2'(x + d_{s_x}, y + d_{s_y}, t + \delta t) \tag{8}$$

Now, the remaining displacement d_s is very small and by expanding the right side term as Taylor's series, we obtain

$$I_x.\frac{d_{s_x}}{\delta t} + I_y.\frac{d_{s_y}}{\delta t} + I_t = 0 \tag{9}$$

For our algorithm, we have used the Lucas-Kanade technique [28] which assumes that in a small image region all the pixels have the same displacement d_s. Then, the two components of d_s are estimated by minimizing the following equation in a small spatial neighborhood S:

$$\sum_{(x,y)\in S} W^2(x, y)[\nabla I(x, y, t).d_s + I_t(x, y, t)]^2 \tag{10}$$

where $W(x, y)$ is a weighting diagonal matrix that gives more influence to constraints at the center of S. The solution of (10) is obtained in closed form using a weighted least-squares.

$$d_s = [A^T W^2 A]^{-1} A^T W^2 b \tag{11}$$

where, for n points $(x, y) \in S$ at a single time, we have:
$A = [\nabla I(x_1, y_1), ..., \nabla I(x_n, y_n)]^T$, $b = -[I_t(x_1, y_1), ..., I_t(x_n, y_n)]$.

The total measured displacement $d_m \in \mathbb{R}^2$ is then obtained by summing the nominal part $d_n \in \mathbb{Z}^2$ and the small sub-pixel $d_s \in \mathbb{R}^2$ (i.e., $d_m = d_n + d_s$).

As showed in Figure 2, a Kalman Filter (KF) is used for optical flow estimation. The benefits of this KF are first its prediction characteristic that has permitted to reduce the search area. Another interesting advantage of the KF is that it filters the measurement d_m from noise.

The state vector of our KF is denoted by $X = (d_x, d_y)^T \in \mathbb{R}^2$ which dynamics can be modelled by a brownian process.

$$X_{k+1} = AX_k + \alpha_k, \;\; Y_k = d_m = CX_k + \beta_k \tag{12}$$

where α_k is the state noise vector, i.e. the prediction error with covariance matrix Q_α. $Y_k \in \mathbb{R}^2$ is the measurement vector and β_k is the measurement noise vector with covariance matrix Q_β. $A \in \mathbb{R}^{2\times 2}$ and $C \in \mathbb{R}^{2\times 2}$ are identity matrices. Based upon this very basic state-space representation for the motion, KF equations can be easily implemented.

4 Optic flow interpretation for UAV 3D motion estimation and obstacles detection (SFM problem)

The computational framework that we use for recursive estimation of UAV motion and structure is the Extended Kalman Filter (EKF), which has been the subject of much work on image sequences. For formalizing the SFM problem, we have used the differential version of the representation given in [27]. However, we have integrated some system dynamics, resulting in a reduction of scale ambiguity. We have also developed an effective procedure for fusing vision data with inertial measurements, thereby overcoming the translation-rotation ambiguity.

4.1 Imaging model

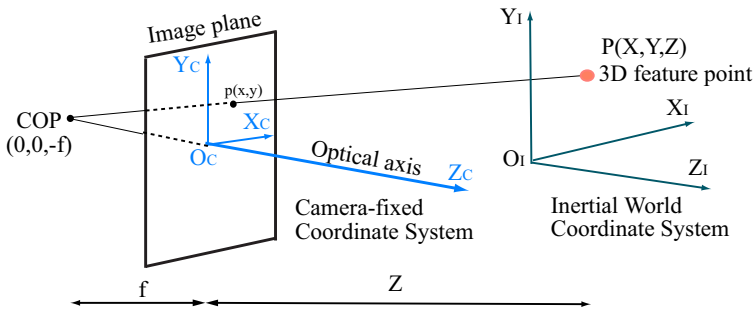

Fig. 4. Imaging model: perspective-central projection

The perspective-central camera model maps the projection of P_i to the focal plane through the following geometrical transformation [27], (see Figure 4):

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{1}{1+\beta Z_i} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \tag{13}$$

with x_i and y_i are the coordinates of p_i which is the projection of P_i on the focal plane as shown in Figure 4. Equation (13) is a model for central projection where $\beta = \frac{1}{f}$ is the inverse focal length. This model is geometrically identical to the usual model, with two representational changes. First, the camera coordinate system origin is fixed at the image plane rather than the

center of projection (COP). Second, inverse focal length β is used as the model parameter.

By differentiating (13) and after geometrical transformations, we find that the optical flow $(\dot{x}_i, \dot{y}_i)$ can be expressed in terms of image coordinates (x_i, y_i), the aircraft body-axis velocities and angular rates $(V_x, V_y, V_z, \Omega_x, \Omega_y, \Omega_z)$, and the depth Z_i [30].

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} \frac{-1}{1+\beta Z_i} & 0 & \frac{\beta x_i}{1+\beta Z_i} \\ 0 & \frac{-1}{1+\beta Z_i} & \frac{\beta y_i}{1+\beta Z_i} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} + \begin{bmatrix} \beta x_i y_i & -(\frac{1}{\beta} + \beta x_i^2) & y_i \\ (\frac{1}{\beta} + \beta y_i^2) & -\beta x_i y_i & -x_i \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \quad (14)$$

4.2 Fusion of OF and angular rate data

Intuitively, by the addition of angular rate data, the performance of SFM estimation should be improved. In the paper [15], the authors have showed that the inertial data can play an important role in improving resistance to tracking noise and reducing inherent ambiguities. They also showed that a smaller number of feature points are sufficient for robust recovery of camera ego-motion when inertial data are available. The approach proposed in [15], treats the inertial data as additional measurements to feature correspondences, and uses an EKF to estimate simultaneously V, Ω and depths Z_i. The fusion strategy that we have proposed aims at simplifying the SFM problem with improvements in accuracy and robustness. Our main idea is to divide the SFM estimation process into two steps. In the first step, we integrate the angular rate data and computed optical flow in a KF in order to estimate the translational component of the optical flow (OF_{trans}) as well as the rotorcraft angular velocity Ω. In fact, at this stage, we aim at subtracting or cancelling the rotational component of the optical flow using a KF that handles measurements noise. The translational OF is the only component that depends on depths Z_i, thereby providing useful information for tasks related to depth perception, such as obstacles detection and collision avoidance. In the second step, the reduced SFM problem is formulated to recover translational motion and structure parameters using the translational OF estimated in the previous step. Unlike the fusion strategy presented in [15] where the structure and the six motion parameters are estimated simultaneously, our fusion procedure separates the estimation of rotation (linear part) and translation (non-linear part). The proposed method results in a noise-resistant SFM algorithm with less ambiguities.

Thus, our fusion strategy is formulated as an estimation problem with the state vector $X^r = (\Omega_x, \Omega_y, \Omega_z, \dot{x}_{1_{trans}}, \dot{y}_{1_{trans}}, ..., \dot{x}_{N_{trans}}, \dot{y}_{N_{trans}})^T$, with N is

the number of the computed OF vectors. The dynamics model in the KF can be chosen trivially as an identity transform plus noise, unless additional prior information on dynamics is available. By recalling (14), the measurement equation can be written as follows

$$Y_k^r = H^r X_k^r + n_k^r, \quad n_k^r \tilde{} \mathcal{N}(0, \Sigma_n^r) \tag{15}$$

with $Y^r = (\dot{x}_1, \dot{y}_1, ..., \dot{x}_N, \dot{y}_N, \Omega_x, \Omega_y, \Omega_z)^T \in \mathbb{R}^{2N+3}$ is the measurement vector and the matrix $H \in \mathbb{R}^{(2N+3)\times(2N+3)}$ can be deduced from (14). The measurement noise n_k^r is assumed to have a Gaussian distribution with zero mean and covariance matrix Σ_n^r.

Then, the KF implementation is straightforward. So, further implementation details will not be repeated here.

4.3 EKF-based algorithm for motion and structure estimation

Using the framework discussed thus far, the initial SFM problem is reduced to estimate translational velocity and structure parameters, considering the previously estimated translational OF as the measurement vector. So, our composite state vector consists of 3 parameters for camera/UAV translational motion, and N variables for structure: $X^c = (V_x, V_y, V_z, Z_1, ..., Z_N)^T$.

Thus, at this stage, the number of unknowns is $3 + N$. Computing OF at N image locations introduces N additional unknowns $(Z_1, ..., Z_N)$ while providing $2N$ equations or measurements. Consequently, the system is completely determined for $2N \geq N + 3 \Rightarrow N \geq 3$. For more accuracy and stability, we have chosen $N = 9$. These nine OF vectors are computed at well-chosen image locations. The selective strategy of these regions is defined to meet some criteria namely: 1) covering a large field of view, 2) increasing the sensibility to vertical velocity V_z in order to obtain a significant divergent OF, and 3) reducing the ambiguity issued from translation and rotation. In order to meet these criteria, we have divided the image into 9 equivalent regions which are symmetrical to the image center. Therefore, the OF, computed in these 9 regions, is rich (translational OF, divergent OF, rotational OF) and exhibits sufficient parallax.

Dynamics model: It is often assumed that the camera motion is smooth in the sense that it can be represented by a dynamic model. This assumption is strongly justified for robotics applications since the camera is mounted on the robot. Therefore, we can write

$$V_{k+1} = V_k + \gamma_k + w_{v_k} \tag{16}$$

with γ_k is the camera/UAV acceleration, available from the IMU.

The scene is assumed to be static and composed of rigid objects. Since accurate 3D reconstruction of the scene is not required for the intended applications, it is then justified to parameterize the scene by depths $Z_i(t),\ i = 1, ..., N$ of the N pixels where OF is computed. In most EKF-based SFM algorithms, the dynamics of the structure parameters Z_i are modeled by an identity matrix. In this paper, we propose the following model:

$$Z_{i_{k+1}} = Z_{i_k} - \delta t\ V_{z_k} + w_{z_{i_k}} \tag{17}$$

The depth map dynamics due to surface variation is included in the noise $w_{z_{i_k}}$.

Thus, the evolution of the state vector X is governed by the following discrete dynamic system (see [29] for more details on the system matrices A and B)

$$X_{k+1} = AX_k + B\gamma_k + w_k, \quad w_k \tilde{\ } \mathcal{N}(0, \Sigma_w) \tag{18}$$

The model noise w_k accounts for modeling errors, and $\mathcal{N}(0, \Sigma_w)$ indicates that the vector w has a Gaussian distribution with zero mean and covariance matrix Σ_w.

Observation/Measurement model: A subset of outputs of the previous KF is considered as measurements for this third EKF. In fact, after fusing inertial data and OF, we obtain a pure translational OF (OF_{trans}) which is related to rotorcraft translational velocity V. From (14), the observation discrete model can be written in the following form

$$Y_k = g(X_k) + \nu_k, \quad \nu_k \tilde{\ } \mathcal{N}(0, \Sigma_\nu) \tag{19}$$

with Y is the measurement vector in $\mathbb{R}^{2N}$ containing the estimated translational OF, and the nonlinear function g is expressed as

$$g(X) = \begin{bmatrix} \frac{-1}{1+\beta Z_1} V_x + \frac{\beta x_1}{1+\beta Z_1} V_z \\ \frac{-1}{1+\beta Z_1} V_y + \frac{\beta y_1}{1+\beta Z_1} V_z \\ . \\ . \\ \frac{-1}{1+\beta Z_N} V_x + \frac{\beta x_N}{1+\beta Z_N} V_z \\ \frac{-1}{1+\beta Z_N} V_y + \frac{\beta y_N}{1+\beta Z_N} V_z \end{bmatrix} \tag{20}$$

The EKF equations: Once the system and measurement/observation models have been specified, then the EKF implementation is straightforward.

- State vector prediction: $X_{pred} = AX_{est} + B\gamma_k$,

- Prediction error: $P_{pred} = AP_{est}A^T + \Sigma_w$
- Compute the Jacobian matrix C: $C = \left[\frac{\partial g}{\partial X}(X_{pred})\right]$
- Compute the Kalman gain: $K = P_{pred}C^T(CP_{pred}C^T + \Sigma_\nu)^{-1}$
- Measurement vector Y computation (KF-based fusion algorithm)
- Update the state vector estimate with the measurement: $X_{est} = X_{pred} + K(Y - g(X_{pred}))$
- Update the error covariance matrix of the motion estimates: $P_{est} = (I-KC)P_{pred}(I-KC)^T + K\Sigma_\nu K^T$ where I is the identity matrix.

---

5 Aerial platform description and real-time implementation

The proposed vision system was implemented on a quadrotor MAV platform. In this section, we describe the air vehicle, give some information about the used sensors and avionics, and present the real-time implementation of the developed vision and control algorithms.

5.1 Quadrotor-based aerial platform

Our platform is based on a miniature four-rotor helicopter, called *X-3D-BL*. The *X-3D-BL* airframe, shown in Figure 5, is designed and commercialized by *Ascending Technologies GmbH*, located in Germany. The vehicle is 53 cm rotor-tip to rotor-tip and weights 400 grams including battery. It has a 300-g payload and an endurance of 22 minutes without payload and 12 minutes with full payload. To demonstrate autonomous flight, we have fitted the helicopter with an embedded autopilot, which was designed and built by the authors at Chiba University. The hardware components that make up the basic flight avionics of our platform include a small micro-controller from *Gumstix Inc.*, and the MNAV100CA sensor from *Crossbow Inc.*.

The *Gumstix* computer is based on a 400-MHz Marvell PXA255 CPU with 16-MB flash memory and 64-MB SDRAM memory. It is the heart of the embedded system and constitutes the Fight Control Computer (FCC). The MNAV100CA includes a digital IMU, a GPS receiver and a pressure sensor in one compact sensor. It is a low-cost (1500 USD) and light-weight (35 grams without GPS antenna) sensor, with a low power consumption, making it ideal for mini and micro UAV applications. The IMU outputs raw data from 3 accelerometers, 3 gyrometers and 3 magnetometers at the rate of 50 Hz to the FCC. The GPS data are updated at 4 Hz and the static pressure sensor

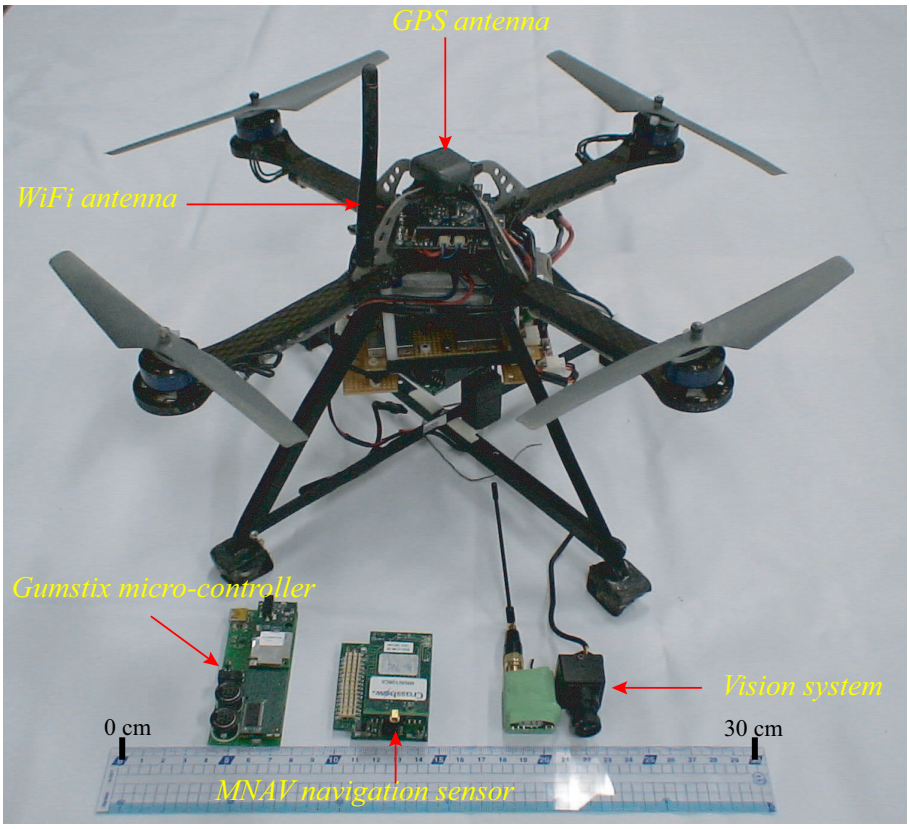

Fig. 5. Our autonomous rotorcraft MAV with navigation sensors and vision system.

measurements are provided at a rate of 50 Hz. All these sensor data are sent to the FCC through an RS-232 serial link. For wireless communication with the Ground Control Station (GCS), we have mounted the *wifistix* expansion card from *Gumstix* on the 92-pin connector of the *Gumstix* motherboard, thereby providing a communication module with high bandwidth (about 50 Mbits/s). The communication range is about 500 m, but it can be increased up to 800 m by reducing the communication bandwidth to 2 Mbits/s.

Our vision system, shown in Figure 5, includes a small analog camera from *RangeVideo* and a 1.3 GHz video transmitter. The camera outputs a NTFS standard composite video signal which is transmitted to the GCS and captured by the frame grabber. The obtained images have a resolution of 320 x 240 pixels, and they are updated at a rate of 25 Hz. The camera is mounted on the bottom of the UAV and it is pointing downwards.

The total weight of the aerial vehicle is about 650 grams, with a total cost of 3500 USD including the air vehicle, battery, FCC, sensors and vision system.

Although this paper focuses on the design of a vision system for self-localisation, the implemented software contains both vision module and control system. Indeed, to demonstrate visual autonomous flight, a 3D flight controller is required. Vision, navigation and control algorithms are implemented on the onboard computer except optic flow computation, which is performed on the GCS. In fact, images are transmitted from the embedded camera to a ground computer where optic flow is computed in real-time at 10 Hz. The calculated optic flow is then, sent to the onboard FCC, through wifi communication using UDP protocol. The GCS software has also other functionalities for displaying flight data and sending some navigation commands such as take-off, landing, hovering, etc.

The real-time embedded software is implemented as a process within Linux OS. This application is composed of different tasks or threads, that are called and scheduled separately for more flexibility and modularity. The Uplink/Downlink thread manages the wireless communication between the FCC and the GCS. The Attitude and Heading Reference System (AHRS) thread is updating the attitude variables at 50 Hz by fusing the IMU raw data using an EKF. The GPS/INS navigation algorithm is implemented in a different thread which is running at 10 Hz and fusing GPS and INS data to provide position and velocity estimates (G*PS estimates are used here for comparison only*). The estimated state vector is then used by a nonlinear flight controller for aircraft stabilization and trajectory tracking tasks. The control thread runs at 50 Hz and provides the four control commands (thrust, pitching torque, rolling torque, yawing torque) for the rotorcraft. Details about the design and implementation of the flight controller can be found in [30,29].

For vision-based navigation, we added another thread that implements the second part of our vision algorithm, namely fusion of optic flow and IMU data, and EKF-based SFM algorithm. This vision thread is running [1] at 10 Hz, and provides useful information (UAV velocity and height modulo some unknown scale factor) for navigation and flight control. These data can be directly used for some reactive navigation without explicit knowledge of MAV velocity and height. Our objective however, is to achieve advanced flight behaviors such as hovering and trajectory tracking based on optic flow and without GPS. As suggested in our previous paper [31], the scale factor can be estimated in real-time using adaptive control tools, Figure 1. In this paper, we have used another effective approach for removing the scale factor ambiguity. Static

[1] The time execution of the vision algorithm onboard the FCC is about 15 ms. The vision algorithm can be run at higher frequencies, but we decided to run it at 10 Hz because it is sufficient for position control.

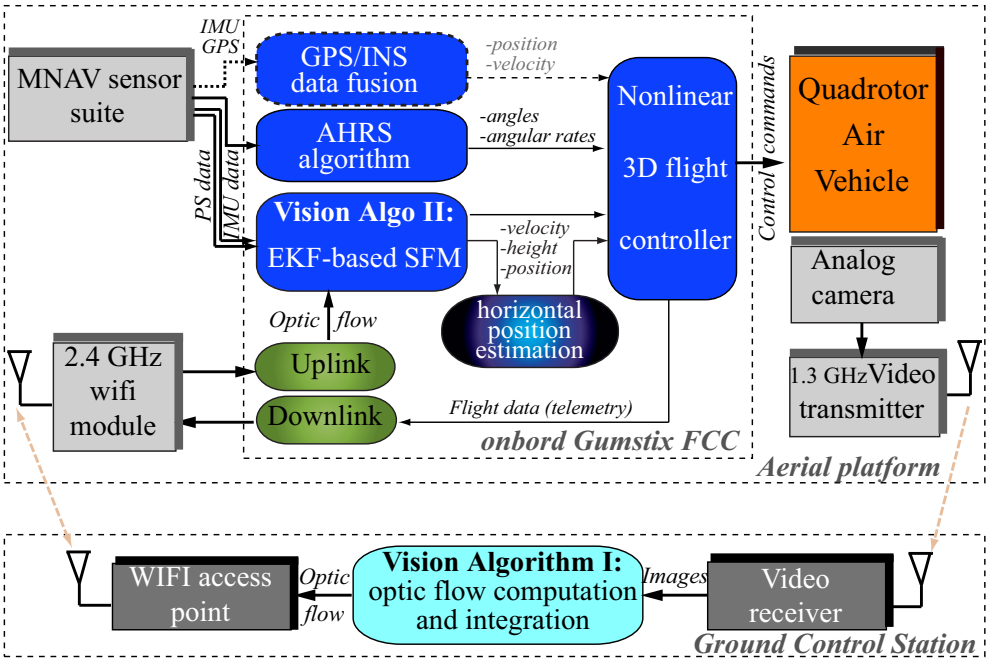

Fig. 6. Real-time architecture of the optic flow-based autopilot.

pressure sensor data is considered as an additional measurement for the SFM EKF measurements, thereby estimating the scale factor and recovering the MAV real velocity and height.

We would like to emphasise the fact that static pressure sensors are light-weight (few grams) and low-cost (few dollars) devices that provide useful information for vertical motion estimation in outdoor and indoor environments. Therefore, they can be easily integrated into micro air vehicles to enhance the vision system.

Furthermore, the MAV horizontal position is estimated by integrating the optic flow computed at the image center. This optic flow integration mechanism acts as a visual odometer that gauges the travelled flight distance. This integrated visual displacement, expressed in pixels, is then converted to real displacement in meters, and used by the flight controller to achieve accurate hovering.

6 3D flight tests and experimental results

In order to demonstrate the robustness and reliability of the designed vision system when used in closed-loop control, we have performed several outdoor and indoor flight tests. In this research and experiments, the flight controller relies on the vision estimates to achieve autonomous flight. GPS data are used here for comparison purposes only.

6.1 Experimental methodology and safety procedures

The developed GCS and embedded software offers the possibility to deploy and operate the MAV easily and safely by one person. Each flight trial is preceded by the check of airframe status as well as sensors and communication reliability. Then, the operator launches the *take-off* procedure by just clicking on the *take-off* button of the GCS interface. When the MAV reaches the desired altitude, it hovers at that point and waits for new commands. For trajectory tracking control, reference velocity and position trajectories are sent in real-time from the GCS. The embedded software offers also a possibility to track some pre-programmed trajectories. Finally, the *landing* button is pushed and the MAV performs an automatic safe landing. All these tasks are performed autonomously by relying on IMU measurements for orientation control and vision data for position and velocity control.

For safety reasons, a number of procedures are implemented on the embedded software. For example, if the communication link is lost for more that 5 seconds, then, the emergency landing procedure is automatically activated. Furthermore, a mechanism was implemented to allow the pilot to take over control of the helicopter in the event of a control or sensor failure.

6.2 Optic flow-based velocity control

This flight test involved a velocity control scheme. It aims at evaluating and demonstrating the ability of the MAV to achieve hovering flight and velocity trajectory tracking by relying on the SFM-algorithm estimates (velocities and height). After automatic take-off, the MAV is tasked to hover and then to achieve autonomous translational flight by tracking some reference trajectories, sent in real-time from the GCS.

Figure 7 shows an extract of the validation flights, demonstrating an autonomous stable flight and a clear match between the visual and GPS velocity measurements. Indeed, the rotorcraft can be seen to clearly respond to commands and to track reference velocity trajectories. Although the closed-loop control of horizontal position is not used in this test, the MAV achieved an accurate stationary flight with very small position drift. During the first 200-s velocity-based hovering flight, shown in Figure 7, the aircraft maintains its position with $\pm 3\ m$ maximum error (ground track measured by GPS). We can also see on Figure 7 that the reference velocity trajectories are well tracked, thereby allowing to guide the MAV by just sending high level commands (fly forward, fly backward, fly sideward, etc.);

This experiment demonstrated the feasibility of using optic flow for vertical

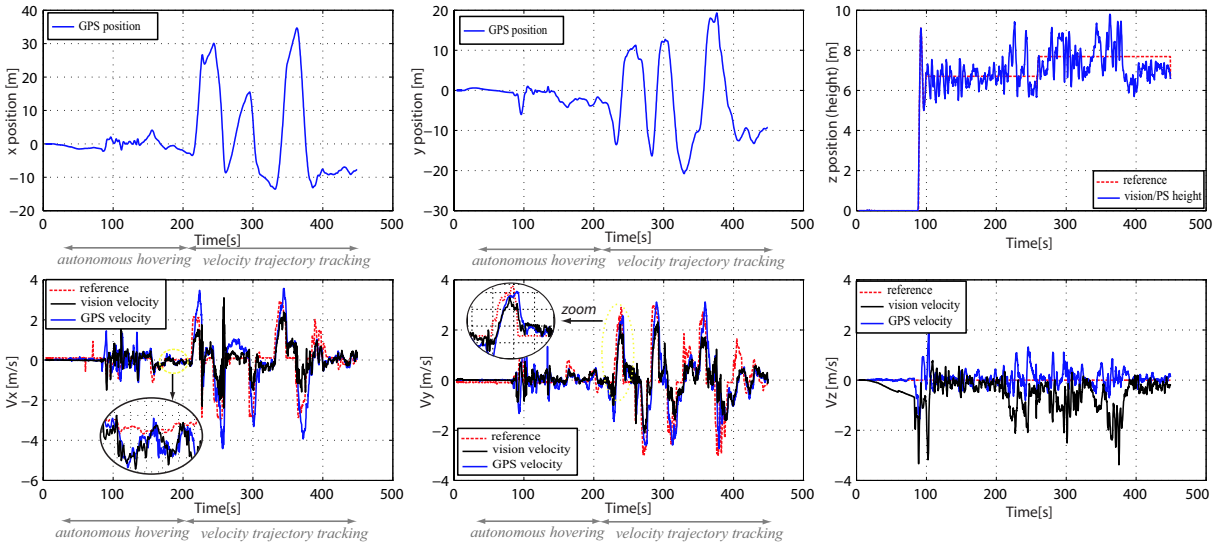

Fig. 7. MAV trajectories of autonomous optic flow-based velocity control.

and horizontal motion control for a rotorcraft in forward flight. This flight behavior is very useful and needed for many real-world applications where GPS signal is not available. Indeed, autonomous velocity control is sufficient to achieve many realistic tasks.

6.3 Optic flow-based position control

As explained in Section 5, the vision algorithm has been augmented by an *integration* mechanism that allows to compute the travelled flight distances. This experimental test consists thus, in achieving an accurate hovering flight by relying on the SFM-algorithm estimates for velocity and height feedback, and integrated image displacement for position feedback.

Figure 8 shows the MAV trajectories (position, height, velocity, orientation) obtained from a fully autonomous vision-based flight including automatic take-off, hovering and auto-landing. The experimental results are very satisfactory and the MAV executed successfully this task despite the outdoor environment (play-ground) which contains poor texture (see Figure 10). GPS [2] data are recorded during flight tests and plotted in Figure 8 for comparison purpose. The obtained experimental results reveal that:

- It is possible to achieve fully autonomous flight in natural environment using information extracted from optic flow.

[2] We are using a low-cost GPS with $\pm 2\ m$ accuracy in the horizontal position and $\pm 5\ m$ in the vertical position.

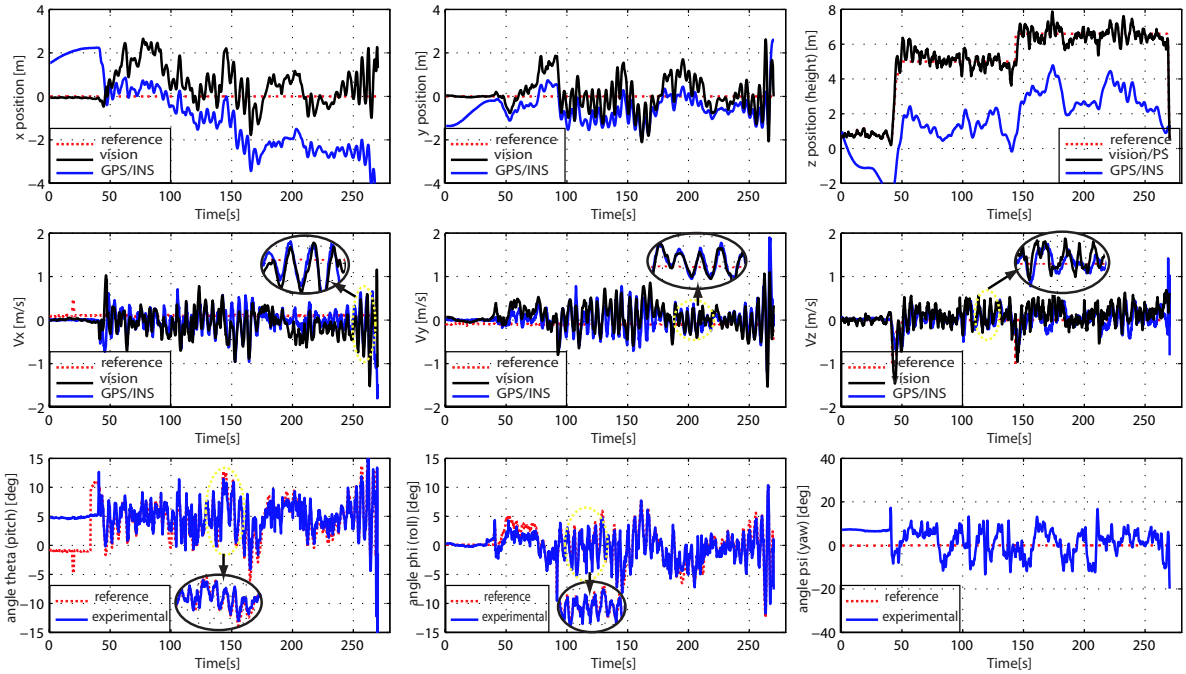

Fig. 8. MAV trajectories of autonomous optic flow-based position control.

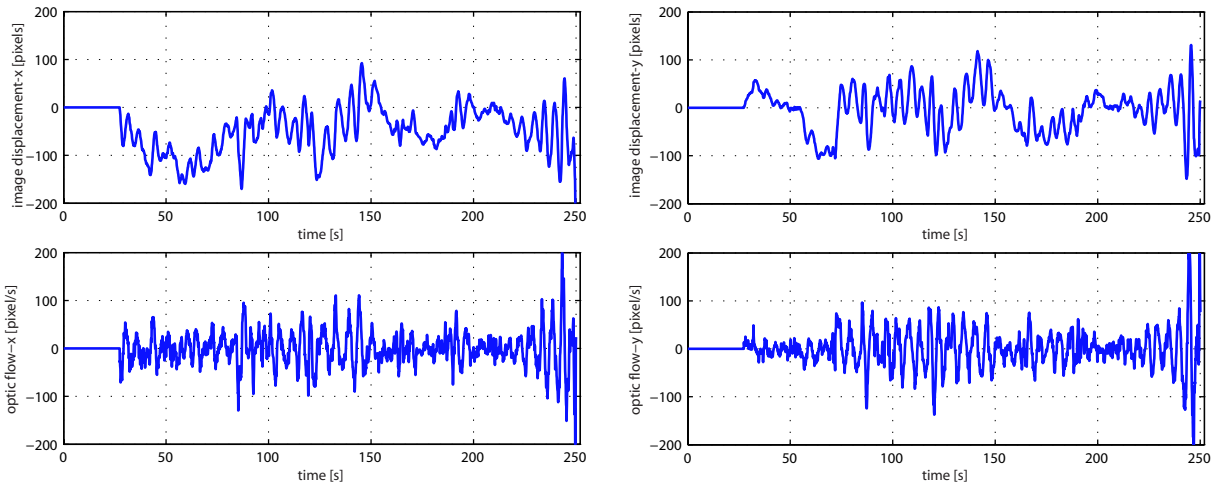

Fig. 9. Visual measurements (optic flow and image displacement) computed at the image center.

- The OF-based velocity estimates are as accurate as the GPS data which proves the effectiveness of the proposed vision system.
- The OF/PS-based height estimate is more accurate and reliable than the GPS measurement.
- The vision-based horizontal position is more accurate than the GPS data, but GPS position measurements are more robust and reliable during long distance translational flights.

The image data (optic flow in [pixels/s] and image displacement in [pixels]),

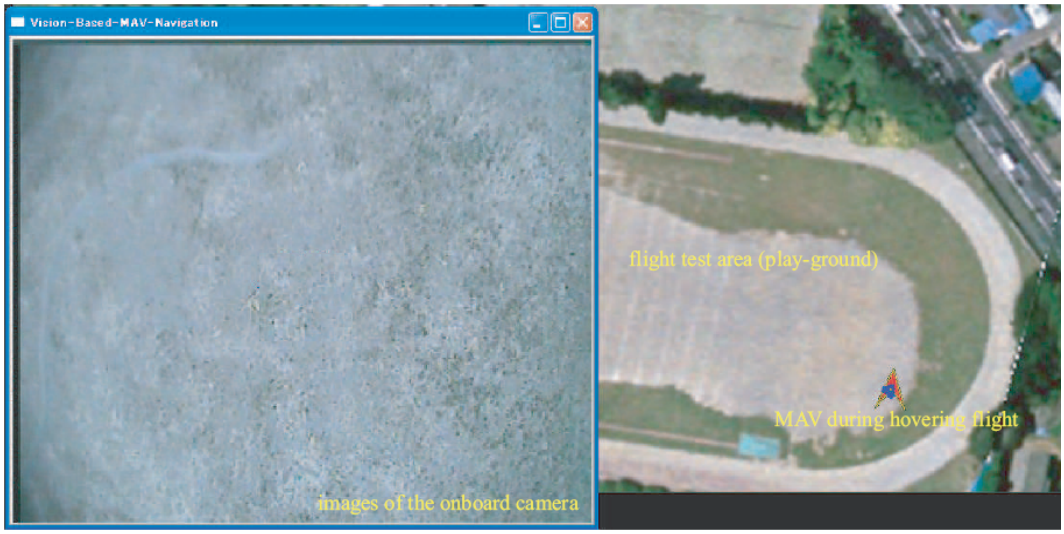

Fig. 10. Part of the GCS interface showing images captured by the onboard camera and the MAV evolution in real-time.

computed at one image location by the first part of the vision algorithm (see Figure 6), is also plotted in Figure 9. We can clearly see that the optic flow is computed and integrated robustly despite the poor image quality and the textureless environment.

6.4 Fully autonomous indoor flight using optic flow

The autonomous flight experiments described above were conducted outdoor. As discussed in the Introduction, most future applications of MAV will include autonomous flight in cluttered and indoor environments. Therefore, this indoor flight test demonstrates the ability of our MAV, equipped with the developed vision system, to achieve fully autonomous indoor flight using information extracted from optic flow. This test has been conducted at *Tokyo Big Sight* for an international exhibition.

Since the floor is homogeneous without any texture, we have put some objects on the ground to provide some texture for optic flow computation (see Figure 11). The task was similar to the one described in Subsection 6.3, that is take-off, autonomous hovering and automated landing. The exception or difference here is that during autonomous hovering, we have switched to manual flight for several seconds and then switched back to autonomous flight. The objective was to show to the audience (visitors) that the rotorcraft is naturally unstable and the vision-control system plays the main role in stabilizing the vehicle.

As it can be seen on Figure 11 and Figure 12, the rotorcraft achieved autonomously the required task with good performance, using optic flow-based visual estimates.

Fig. 11. Indoor fully autonomous flight using optic flow. Demonstration at *Tokyo Big Sight* for an international exhibition.

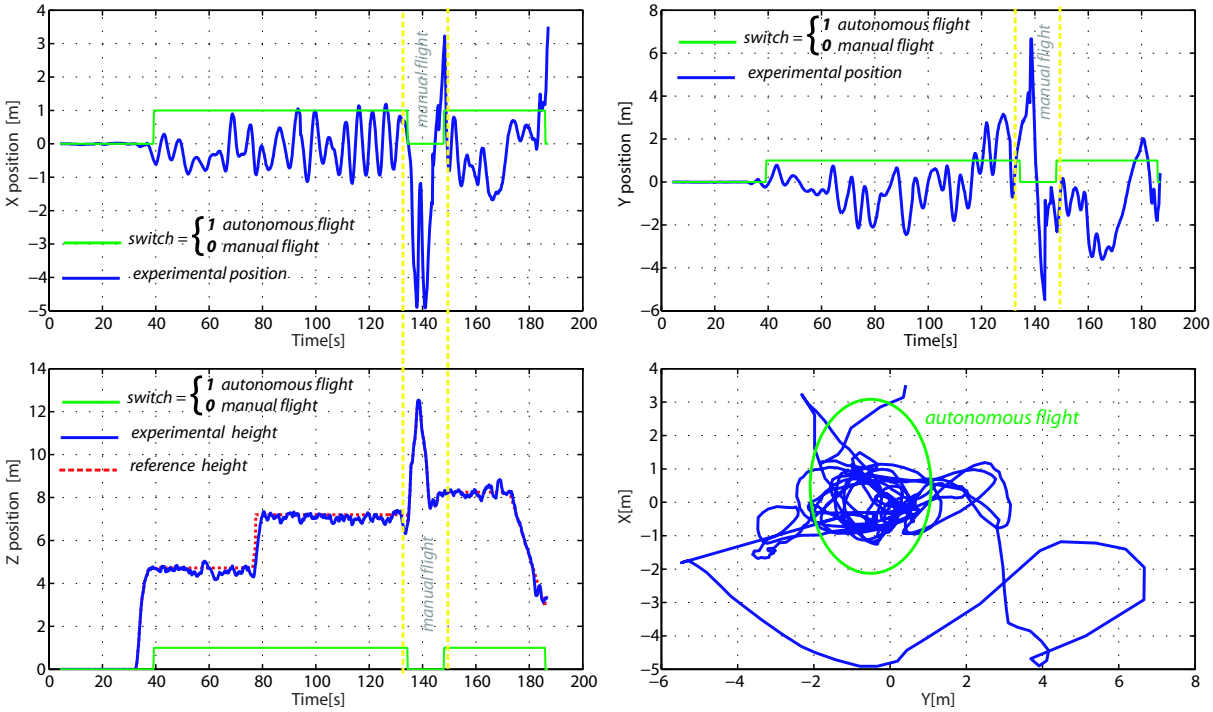

Fig. 12. MAV position and height trajectories of indoor autonomous flight using optic flow.

Video clips of these flight tests and optic flow-based autonomous flights can be found at:
http://www.youtube.com/watch?v=6U0IhPlYXKw

7 Conclusion and future work

A real time optic flow-based vision system for rotorcraft MAVs localisation and control has been designed, demonstrated and benchmarked against GPS. The vision system consists of single downward-looking camera and a vision algorithm for optic flow computation, IMU data and optic flow fusion, and MAV motion and structure estimation. The aircraft self-motion and height are recovered modulo some unknown scale factor. However, when combined with relative height measurements from static pressure sensor, the vision system was able to remove the scale factor ambiguity and to estimate the real MAV position, velocities and height above the ground. Moreover, the vision algorithm has been augmented by an effective mechanism to estimate the travelled flight distance by integrating the optic flow over time. These vision-based estimates are then, used by a nonlinear controller to achieve advanced 3D flight behaviors.

Through a variety of real-time experiments, the efficiency and robustness of the proposed vision system were demonstrated for indoor and outdoor flights. In spite of poor image quality and texture resulting in noisy optic flow estimates, the overall behaviour of the vision-control system is robust and almost independent of the environment and the camera motion pattern. A four-rotor helicopter, equipped with the developed vision system, achieved successfully automatic take-off and landing, hovering and trajectory tracking. The designed vision-based autopilot takes inspiration from insects and represents a major step toward our goal of developing autonomous small-flyers capable of navigating in urban and cluttered environments.

In this work, optic flow, obtained from a downward-looking camera, was mainly used to control the 3D motion of a rotorcraft MAV which is flying over a flat area. In a more general situation, the camera could be pointed in any direction and the proposed vision system could be used to recover the structure of the environment. This would allow to detect a safe landing area and/or to provide the range, the azimuth and the elevation of obstacles. In fact, we have already implemented this vision system on a ground vehicle and demonstrated its ability to estimate the structure parameters Z_i when the camera is sideward-looking [29]. Future work should thus extend these findings and investigate the use of the proposed vision system for environment mapping and obstacles avoidance. We are currently implementing an adaptive control system to identify the unknown scale factor and to estimate the range to lateral and frontal obstacles.

References

[1] T. Kanade, O. Amidi, Q. Ke, Real-time and 3d vision for autonomous small and micro air vehicles, in: Proceedings of the 43th IEEE Conference on Decision and Control (CDC), 2004.

[2] M. Egelhaaf, R. Kern, Vision in flying insects, Current Opinion in Neurobiology 12 (6) (2002) 699–706.

[3] M. V. Srinivasan, S. Zhang, M. Lehrer, T. Collett, Honeybee navigation en route to the gaoal: Visual flight control and odometry, The Journal of Experimental Biology 199 (1) (1996) 237–244.

[4] B. Horn, B. Schunck, Determinig optical flow, Artificial Intelligence 17 (1981) 185–204.

[5] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, International Journal of Computer Vision (1989) 283–310.

[6] D. Fleet, A. Jepson, Computation of component image velocity from local phase information, International Journal of Computer Vision 5 (1) (1990) 77–104.

[7] G. L. Barrows, Mixed-mode VLSI optic flow sensors for micro air vehicles, Ph.D Dissertation, Department of Electrical Engineering, University of Maryland, 1999.

[8] J. Barron, D. Fleet, S. Beauchemin, Performance of optical flow techniques, International Journal of Computer Vision 12 (1) (1994) 43–77.

[9] J. Weng, T. S. Hwang, N. Ahuja, Motion and Structure from Image Sequences, Springer-Verlag, Berlin, 1991.

[10] H. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, Nature 293 (1981) 133–135.

[11] A. Chiuso, P. Favaro, H. Jin, S. Soatto, Structure from motion causally integrated over time, IEEE Tran. On Pattern Analysis and Machine Intelligence 24 (4).

[12] M. Irani, P. Anandan, About direct methods, Vision Algorithms: Theory and Practice (1999) 267–277.

[13] R. I. Hartley, Estimation of relative camera positions for uncalibrated cameras, in: Proc. of the European Conference on Computer Vision, 1992, pp. 579–587.

[14] T. J. Broida, R. Chellappa, Estimation of object motion parameters from noisy images, IEEE Trans. Pattern Analysis and Machine Intelligence 8 (1) (1986) 90–99.

[15] G. Qian, R. Chellappa, Q. Zheng, Robust structure from motion estimation using inertial data, Journal of Optical Society of America 18 (12) (2001) 2982–2997.

[16] G. Adiv, Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field, IEEE Trans. Pattern Analysis and Machine Intelligence 11 (1989) 477–489.

[17] M. V. Srinivasan, J. S. Chahl, K. Weber, S. Venkatesh, M. Negle, S. Zhang, Robot navigation inspired by principles of insect vision, Robotics and Autonomous Systems 26 (2) (1999) 203–216.

[18] M. V. Srinivasan, M. Poteser, K. Kral, Motion detection in insect orientation and navigation, Vision Research 39 (1999) 2749–2766.

[19] A. Si, M. V. Srinivasan, S. Zhang, Honeybee navigation: Properties of the visually driven odometer, The Journal of Experimental Biology (2003) 1265–1273.

[20] F. Ruffier, N. Franceschini, Optic flow regulation: the key to aircraft automatic guidance, Robotics and Autonomous Systems 50 (4) (2005) 177–194.

[21] J. Chahl, A. Mizutani, An algorithm for terrain avoidance using optical flow, in: Proc. of the IEEE American Control Conference, Minnesota, USA, 2006, pp. 2742–2747.

[22] J. Chahl, M. Srinivasan, S. Zhang, Landing strategies in honeybees and applications to uninhabited airborne vehicles, International Journal of Robotics Research 23 (2) (2004) 101–110.

[23] W. E. Green, P. Y. Oh, G. Barrows, Flying insects inspired vision for autonomous aerial robot maneuvers in near-earth environments, in: Proc. of the IEEE International Conference on Robotics and Automation, New Orleans, 2004, pp. 2347–2352.

[24] J.-C. Zufferey, D. Floreano, Fly-inspired visual steering of an ultralight indoor aircraft, IEEE Trans. Robotics 22 (1) (2006) 137–146.

[25] L. Muratet, S. Doncieux, Y. Briere, J.-A. Meyer, A contribution to vision-based autonomous helicopter flight in urban environments, Robotics and Autonomous Systems (Elsevier) 50 (4) (2005) 195–209.

[26] S. Hrabar, G. S.Sukhatme, A comparison of two camera configurations for optic-flow based navigation of a UAV through urban canyons, in: Proc. of the IEEE International Conference on Intelligent Robots and Systems, Japan, 2004, pp. 2673–2680.

[27] A. Azarbayejani, A. Pentland, Recursive estimation of motion, structure, and focal length, IEEE Trans. Pattern Analysis and Machine Intelligence 17 (6) (1995) 562–575.

[28] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proc. DARPA IU Workshop, 1981, pp. 121–130.

[29] F. Kendoul, I. Fantoni, G. Dherbomez, Three nested kalman filters-based algorithm for real-time estimation of optical flow, uav motion and obstacles detection, in: Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 4746–4751.

[30] F. Kendoul, Modelling and control of unmanned aerial vehicles, and development of a vision-based autopilot for small rotorcraft navigation, PhD Thesis Report, University of Technology of Compiegne.

[31] F. Kendoul, I. Fantoni, R. Lozano, Adaptive vision-based controller for small rotorcraft uavs control and guidance, in: Proceedings of the 17th IFAC World Congress, Seoul, Korea, 2008, pp. 797–802.