2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Three Nested Kalman Filters-Based Algorithm for Real-Time Estimation of Optical Flow, UAV Motion and Obstacles Detection

Farid Kendoul, Isabelle Fantoni and Gérald Dherbomez
University of Technology of Compiègne, 60200 Compiègne, France
fkendoul@hds.utc.fr, ifantoni@hds.utc.fr, gdherbomez@hds.utc.fr

***Abstract*— We aim at developing a vision-based autopilot for autonomous small aerial vehicle applications. This paper presents a new approach for the estimation of optical flow, aircraft motion and scene structure (range map), using monocular vision and inertial data. The proposed algorithm is based on 3 Nested Kalman Filters (3NKF) and results in an efficient and robust estimation process. The 3NKF-based algorithm was tested extensively in simulation using synthetic images, and in real-time experiments.**

***Index Terms*— Small flying robots, optical flow computation, structure from motion, vision-based autopilot.**

I. Introduction

Milestones in aerial robots have been recently achieved using sensor suites that include GPS, Inertial Measurement Units (IMU), laser altimeters, ultrasound sensors to perform missions like terrain-following, autonomous landing, etc. Our particular interests, however, involve small UAVs flying close to the ground in complex environments like urban and indoor environments. Traditional sensors such as GPS, IMU, pressure sensors, radar, sonar, laser are not adapted for this category of UAVs and associated applications. Indeed, the sensors commonly employed in robotics are too heavy and energy-consuming to match the limited payload of small aircraft. Recently, there is a growing interest in applying Optical Flow (OF) for small aircraft control and navigation. However, the difficulty found when using imaging sensors is the high bandwidth of data, and the resulting heavy computational burden. Furthermore, reliably computing optical flow or extracting and tracking features in video is a challenge; accurate and robust estimation of arbitrary camera ego-motion and structure given noisy visual measurements is another challenge. So, the question is how to use vision for the control of small UAVs considering the constraints of real-time applications?

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 behaviors, and they are appropriate for implementation in a biomimetic autopilot for small UAVs and robotics in general [1], [2]. Potential applications of optical flow for small aerial vehicles are altitude control and terrain following [3], autonomous landing [3], [4], [5] and obstacles avoidance [6], [7], [8]. However, most existing OF-based strategies use a simplified model of the 3D vision problem, where the UAV motion is usually restricted to one dimension.

The work presented in this paper represents a major step toward our goal of developing autonomous small-flyers capable of navigating within houses or urban and small indoor environments. Based on the literature review and the restrictions imposed by robotics applications, we developed a new real-time algorithm that jointly performs optical flow estimation and 3D interpretation (recovery of 3D motion and structure) using monocular images and inertial rate data. The proposed computational framework is based on 3 Nested Kalman Filters (3NKF) which allowed to combine three algorithmic concepts stemming from completely different areas of research (OF computation, data fusion, Structure From Motion (SFM) problem) in a favorable manner. The novelty of the approach is based on the fact that each sub-module is connected to other modules, thereby allowing bidirectional exchange of data between them (see Figures 1 and 2). The resulted 3NKF-based algorithm is very fast (30 Hz), accurate and robust. It is designed to operate in a static, structurally unconstrained environment, with no prior knowledge of scene content.

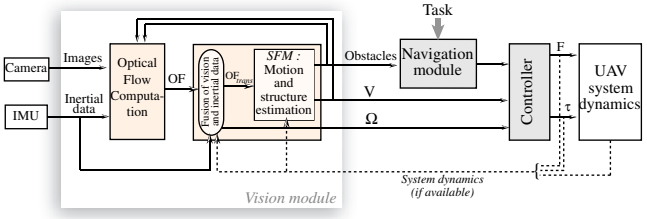


Fig. 1. Optical flow-based vision module for UAV control

II. Prediction-based algorithm with adaptive patch for accurate and efficient OF estimation

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

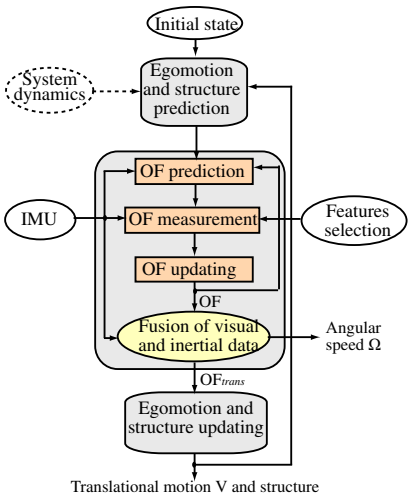


Fig. 2. 3NKF-based algorithm for OF, ego-motion and structure estimation

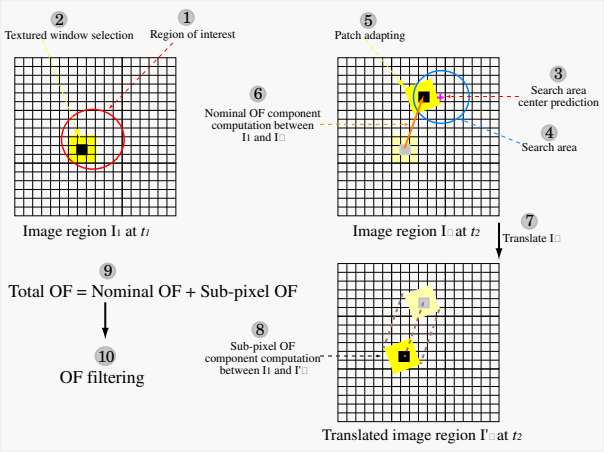


Fig. 3. Prediction-based OF algorithm with adaptive patch

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 [9] 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.

A. 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 (see equation (14)). Thus, we use the predicted camera motion (V_{pred}), the predicted structure (Z_{pred}) and the measured angular velocity (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 (see Figure 2). 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, (see Figure 7).

The size of the search area is $(r+1) \times (r+1)$ which is independent of the amplitude of image displacement.

B. Combined block-matching and differential algorithm

1) Step 1: 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 (i.e., $d \in SA$).

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

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 (rotations) into the matching process. Therefore, the shape of the patch is adapted by modifying equation (2).

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

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

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

2) Step 2: 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

$$I_1(x,y,t) = I_2(x+\delta x, y+\delta y, t+\delta t) \quad (6)$$

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. 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

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

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

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

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

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

For our algorithm, we have used the Lucas-Kanade technique [9] 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:

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

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.

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

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.

$$X_{k+1} = AX_k + \alpha_k, \quad Y_k = d_m = CX_k + \beta_k \quad (12)$$

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.

III. Rotorcraft's 3D motion estimation and obstacles detection using optical flow

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). The paper [10] provided a critical study of existing SFM techniques.

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 [11]. 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.

A. Imaging model

The perspective-central camera model maps the projection of P_i to the focal plane through the following geometrical transformation [11], (see Figure 4):

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

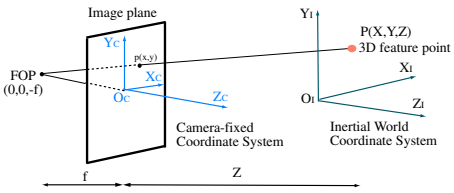


Fig. 4. Imaging model: central projection

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 can be expressed in terms of image coordinates (x_i, y_i), the aircraft body-axis velocities and rates $(V_x, V_y, V_z, \Omega_x, \Omega_y, \Omega_z)$, and the depth Z_i [11].

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

B. Fusion of OF and inertial data

Ambiguities in 3D motion recovery from noisy flow fields have been reported by many researchers [12], [13]. One dominant ambiguity arises from the similarity between the flow fields generated by V_x (V_y) and Ω_y (Ω_x). In robotics applications, the common way to circumvent this problem is the integration of both OF and inertial data in the estimation process. 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 inertial data and estimated OF 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.

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.

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

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

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.

C. 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: $X^c = (V_x, V_y, V_z, Z_1, ..., Z_N)^T$.

Computing OF in N locations introduces N unknowns and gives $2N$ equations or measurements. Consequently, the system is completely determined for $N \geq 3$. For more accuracy and stability, we have chosen $N = 9$. We have observed in different experiments that 9 OF vectors are sufficient for estimating robustly and efficiently the camera ego-motion and the scene structure. However, these 9 OF vectors should be computed robustly at well-chosen image regions. 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: The equations of motion for an UAV subject to body force $F \in \mathbb{R}$ and torque $\tau \in \mathbb{R}^3$, applied to the center of mass and specified with respect to the body coordinate frame, are given by the following Newton-Euler equations: $\dot{V} = RF$, and $J\dot{\Omega} = \tau - \Omega \times J\Omega$, with R is the rotational matrix and J is the inertial matrix of the rotorcraft body. At this stage, we are only interested by translational motion dynamics. Since the aircraft orientation is measured by an IMU, then we can consider $U = RF$ as the new control force that will be used in the estimation process.

The dynamics of structure parameters can be approximated by the simple following equation: $\dot{Z}_i(t) = -V_z, \ i = 1, ..., N$.

Therefore, the evolution of the state vector X^c is governed by the following discrete dynamical system

$$X_{k+1}^c = A^c X_k^c + B^c U_k + \kappa_k, \quad \kappa_k \tilde{} \mathcal{N}(0, \Sigma_\kappa) \quad (16)$$

$$A^c = \begin{bmatrix}1 & 0 & 0 & 0_1 & 0_2 & ... & 0_N\\ 0 & 1 & 0 & 0_1 & 0_2 & ... & 0_N\\ 0 & 0 & 1 & 0_1 & 0_2 & ... & 0_N\\ 0 & 0 & -\delta t & 1 & 0_2 & ... & 0_N\\ 0 & 0 & -\delta t & 0_1 & 1 & ... & 0_N\\ . & & & & & & \\ . & & & & & & \\ 0 & 0 & -\delta t & 0_1 & 0_2 & ... & 1\end{bmatrix}, \ B^c = \begin{bmatrix}\delta t & 0 & 0\\ 0 & \delta t & 0\\ 0 & 0 & \delta t\\ 0_1 & 0_1 & 0_1\\ . & & \\ . & & \\ 0_N & 0_N & 0_N\end{bmatrix} \quad (17)$$

The model noise κ_k accounts for modeling errors, and $\tilde{} \ \mathcal{N}(0, \Sigma_\kappa)$ indicates that the vector κ has a Gaussian distribution with zero mean and covariance matrix Σ_κ.

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

$$Y_k^c = g^c(X_k^c) + n_k, \quad n_k \tilde{} \mathcal{N}(0, \Sigma_n) \quad (18)$$

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

$$g^c(X^c) = \begin{bmatrix} \frac{-1}{1+\beta Z_1} V_x + \frac{\beta x_1}{1+\beta Z_1} V_z \\ \frac{-1}{1+\beta Z_1} V_y + \frac{\beta y_1}{1+\beta Z_1} V_z \\ \vdots \\ \frac{-1}{1+\beta Z_N} V_x + \frac{\beta x_N}{1+\beta Z_N} V_z \\ \frac{-1}{1+\beta Z_N} V_y + \frac{\beta y_N}{1+\beta Z_N} V_z \end{bmatrix} \quad (19)$$

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

- State vector prediction: $X^c_{pred} = A^c X^c_{est} + B^c U_k$,
- Prediction error: $P^c_{pred} = A^c P^c_{est} A^{c^T} + \Sigma_\kappa$
- Compute the Jacobian matrix C^c: $C^c = \left[\frac{\partial g^c}{\partial X^c}(X^c_{pred})\right]$
- Compute the Kalman gain: $K^c = P^c_{pred} C^{c^T} (C^c P^c_{pred} C^{c^T} + \Sigma_n)^{-1}$
- Measurement vector Y^c computation (KF-based fusion algorithm)
- Update the state vector estimate with the measurement: $X^c_{est} = X^c_{pred} + K^c(Y^c - g^c(X^c_{pred}))$
- Update the error covariance matrix of the motion: $P^c_{est} = (I_{12} - K^c C^c) P^c_{pred} (I_{12} - K^c C^c)^T + K^c \Sigma_n K^{c^T}$ where I_{12} is the identity matrix.

IV. Synthetic and Real-Time Experiments

In this section, we present the results of applying the 3NKF-based algorithm for estimating the flow field and SFM parameters.

A. Synthetic images

The synthetic image sequence we used is a sinusoid which is created by superposing two sinusoidal plane waves with spatial wavelengths of 11 pixels.

The configuration of the simulation is as follows: The virtual camera has a focal length $f = 1230$ pixels and an image size of 640 $\times$ 480 pixels. The considered structure is composed of nine objects, each one has a different depth according to the camera $((Z_1, Z_2, Z_3..., Z_8, Z_9) = (50, 60, 70, ..., 120, 130)[cm])$.

In this experiment, we have conducted comparisons between our 3NKF-based algorithm and two well-known SFM algorithms, including those by *Azarbayejani and Pentland* (A-P algorithm) [11], and *Qian and Chellappa* (Q-C algorithm) [13]. In the A-P algorithm, the EKF uses only the feature correspondences as measurements while the Q-C algorithm uses inertial data as additional measurements to feature correspondences. Zero mean noise with STD of 0.3 pixels/frame (9 pixels/s) is added to each coordinate of the computed OF. Also, inertial rate data are corrupted by white noise with zero mean and STD of 0.01 rad/s.

Figure 5 shows the motion estimation results obtained by the three algorithms. It can be seen that under this noise level, the performance of the A-P algorithm deteriorates much faster (translation-rotation ambiguity) than with the other two algorithms. We can also see that the 3NKF-based algorithm

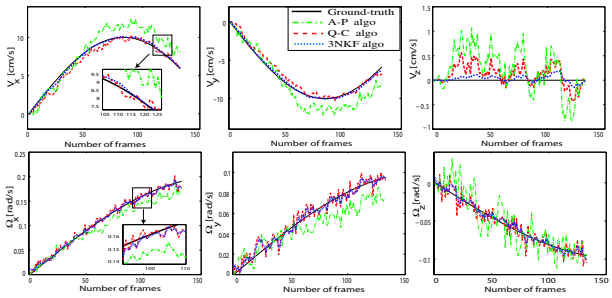
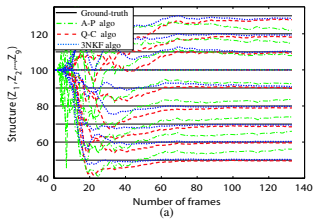


Fig. 5. Camera ego-motion estimation with added noise



Variable / Algorithm	Velocity V [cm/s]		Velocity Ω [rad/s]		Structure Zi [cm]	
	Mean	Std	Mean	Std	Mean	Std
Azarbay-ejani algo	-0.026	1.17	50e-4	0.016	-4.6	0.70
Qian *et al.* algo	-0.037	0.46	-5e-4	0.005	0.43	0.23
3NKF algo	0.001	0.14	-2e-4	0.007	-0.4	0.06

(b)

Fig. 6. Structure estimation with added noise and the estimation errors

performs better than the Q-C algorithm. Figure 6(a) shows the obtained results for structure estimation. It can also be seen that our algorithm results in best performance compared to the two other algorithms. The table in Figure 6(b) confirms these observations and demonstrates quantitatively the good performance of the 3NKF-based algorithm.

These results show the effectiveness of inertial data for achieving robust and accurate SFM estimation. It can also be observed that the 3NKF-based algorithm outperforms the Q-C algorithm which indicates that the strategy or manner of fusing visual and inertial data plays also an important role in improving the performance of SFM estimation.

B. Real images obtained from a vehicle-mounted camera

In this section, we present results from applying our 3NKF-based algorithm in real-time experiments. The used camera is a *Basler 601f* with an image size of 640 $\times$ 480 pixels, rate frame of 60 fps and a focal length of 1230 pixels. The camera is connected to a PC via the adapter card *IEEE1394 FireWire Notebook Adapter*. All the parts of the 3NKF-based algorithm, presented in this paper, are implemented in C^{++}. The experiments were performed with the platform STRADA, which is equipped with an onboard computer, a GPS, two laser telemeters, an Inertial Measurement Unit, ABS (4 wheels) and many other sensors. The four ABS sensors are used to provide ground-truth on translational velocity, due to their increased accuracy compared to GPS.

To demonstrate the robustness of the algorithm with respect to natural environments with poor texture, we have performed the first test by a vehicle moving on a straight road parallel to a flat terrain. The *Basler* camera was fixed on the roof of the vehicle, and oriented laterally towards ground with an angle $\alpha = 0.1$ rad.

Figure 7(a) shows the two components of the computed optical flow at 9 image locations, the estimated translational and rotational velocities, and the recovered 9 depths. Firstly, we can notice that the algorithm computes robustly large

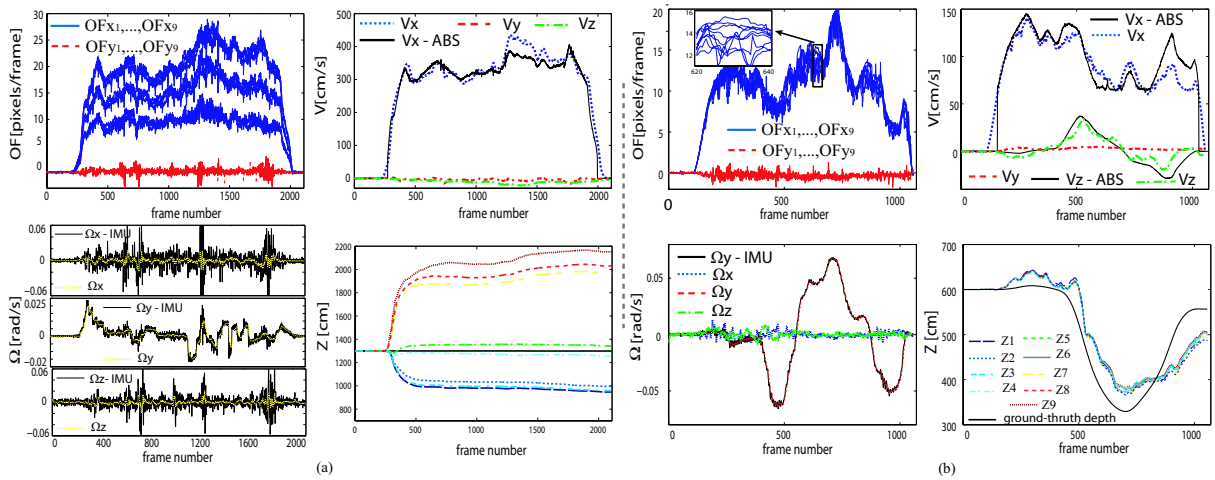


Fig. 7. Vehicle-mounted camera: (a) translational motion and lateral-downward-looking camera. (b) translational + rotational motions

optical flows (until 28 pixels/frame), and the estimated angular velocities are less noisy than the IMU measurements.

Secondly, the estimated velocity V_x is almost identical to the velocity computed by wheels-mounted sensors. Concerning the structure recovery, the results are also satisfactory. The estimated depths, shown in Figure 7(a), correspond to the ground-truth values (9.8,9.8,9.8,13,13,13, 20, 20, 20 m).

In the second experiment, the vehicle moves on a road with performing steering manoeuvres (yaw rotations). In this experiment, the camera is looking sideways (i.e., the camera optical axis Z_c is parallel to the vehicle Y_b axis) and the observed scene is a line of trees (which form a kind of green flat surface) located at about 6 meters from the camera (i.e., $Z_1(0) = Z_2(0) = ... = Z_9(0) = 6$ m). During the experiment, the translational velocity, the angular velocity and steering angle (yaw angle) of the vehicle were recorded. By performing transformations on this data and using a kinematic model of the vehicle, the 3D motion of the camera and the depths of observed points are reconstructed to give ground-truth.

Figure 7(b) shows the computed OF and the recovered motion and structure. Colored lines correspond to the vision estimates, while solid black lines represent the ground-truth trajectories, which are reconstructed from the vehicle's sensors data. We observe that the obtained results are satisfactory and very promising. However, small errors remain, which are mainly due to the errors in the optical flow computation. The observed errors can also be related to uncertainties on sensors used to provide ground-truth.

The obtained results in outdoor natural environments with poor texture are very promising. These results confirm the robustness of the 3NKF-based algorithm and its effectiveness and practicability in real world applications. Furthermore, the program runs in real-time at a frequency of 30 Hz.

V. Conclusion and Future Work

The 3NKF-based algorithm appears to perform well, both in simulation and in real-time experiments. Efficiency (30 Hz), accuracy and robustness are the benefits of the proposed algorithm, and make it suitable for small aerial vehicles navigation.

Future efforts will focus on applying the proposed algorithm for flight control (autonomous vertical landing) in complex and cluttered environments.

References

[1] M. V. Srinivasan, J. S. Chahl, K. Weber, S. Venkatesh, M. Negle, and S. Zhang, "Robot navigation inspired by principles of insect vision," *Robotics and Autonomous Systems*, vol. 26, pp. 203–216, 1999.

[2] M. V. Srinivasan, S. Zhang, M. Lehrer, and T. Collett, "Honeybee navigation en route to the goal: Visual flight control and odometry," *The Journal of Experimental Biology*, pp. 237–244, 1996.

[3] F. Ruffier and N. Franceschini, "Optic flow regulation: The key to aircraft automatic guidance," *Robotics and Autonomous Systems*, no. 50, pp. 177–194, 2005.

[4] J. Chahl, M. Srinivasan, and S. Zhang, "Landing strategies in honeybees and applications to uninhabited airborne vehicles," *Int. J. of Robotics Research*, vol. 23, no. 2, pp. 101–110, February 2004.

[5] W. E. Green, P. Y. Oh, and 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, April 2004, pp. 2347–2352.

[6] J.-C. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Transactions On Robotics*, vol. 22, no. 1, pp. 137–146, 2006.

[7] L. Muratet, S. Doncieux, Y. Briere, and J.-A. Meyer, "A contribution to vision-based autonomous helicopter flight in urban environments," *Robotics and Autonomous Systems*, no. 50, pp. 195–209, 2005.

[8] S. Hrabar and 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/RSJ International Conference on Intelligent Robots and Systems*, Japan, 2004, pp. 2673–2680.

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

[10] J. Oliensis, "A critique of structure from motion algorithms," *Technical Report, NEC Research Institute, Princeton,N.J.*, 2000.

[11] A. Azarbayejani and A. Pentland, "Recursive estimation of motion, structure, and focal length," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 562–575, 1995.

[12] G. Young and R. Chellappa, "Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy flow field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 995–1013, 1992.

[13] G. Qian, R. Chellappa, and Q. Zheng, "Robust structure from motion estimation using inertial data," *Journal of Optical Society of America*, vol. 18, no. 12, pp. 2982–2997, December 2001.