Multisensor dense estimation of 3D motion and object tracking based on a particle approach

P. Lanvin **J.-C. Noyer** **M. Benjelloun**
Laboratoire d'Analyse des Systèmes du Littoral (EA2600), Université du Littoral Côte d'Opale
50 rue Ferdinand Buisson, B.P. 699, 62228 Calais Cedex
France
lanvin@lasl.univ-littoral.fr noyer@lasl.univ-littoral.fr benjello@lasl.univ-littoral.fr

Abstract – *We present in this article a method to track and estimate the 3D motion of an object of geometry given, by fusing informations from reflectance and range image sequences. This top-down approach uses a dense modelling of the object to be tracked. The fusion of measurement and the estimation are carried out by a global particle filter, of which interest is to process efficiently the non-linearities of the state equations. To illustrate the subject, the method is applied to synthetic and real image sequences.*

Keywords: Object tracking; 3D motion estimation; Sensor fusion; Particle filtering

1 Introduction

In the field of computer vision, tracking and motion estimation of 3D objects take on a significant place. That it is in robotics, in transport or in the military domain, a lot of applications requires an accurate knowledge of localization and motion of the objects in the scene. Many works were carried out on the subject, from which emerge 3 classes of methods. Feature-based methods [1] aim at extracting characteristics such as points, line segments from image sequences, tracking stage is then ensured by a matching procedure at every time instant. Differential methods are based on the optical flow computation, i.e. on the apparent motion in image sequences, under some regularization assumptions [2, 3]. The third class use the correlation to measure inter-images displacements [4, 5]. These classes has advantages and drawbacks which are related to the applicative context. The method presented in this paper is based on a 3D model of the object to be tracked, and can directly use the images delivered by the sensors. Another interest lies in the motion modelling. Indeed, a feature extracted from an image, a point for example, will have a displacement model often more complex than the object which is part of. In addition the occlusion phenomena are in this case more difficult to deal with. All these points are detailed in [6]. The origin of the model-based approaches can be found in [7] and some similar work in [8, 9].
The solution relies on a state modelling of the problem, which is strongly non-linear. In many works, the extended Kalman filter is used to track and estimate the motion of the features from image sequences [10]. The EKF linearizes the state equations to obtain a model that is locally linear. This linearization stage may give rise to instability problems. For this reason, we use the particle filtering [11] to solve this non-linear estimation problem. The advantage of this filter lies in its ability to deal with non-linear models. Similar approaches exist, known as the Bootstrap filtering [12] and Condensation algorithm [13].
This article details the most significant aspects of the proposed solution. Section 2 gives an overview of the problem and the adopted solution. Section 3 describes the state modelling of the problem. Section 4 introduces the particle filtering and section 5 validates our approach on synthetic and real data.

2 Overview

Previous works [14, 15] that directly use the greyscale levels have been developped for 2D objects in a monocular image sequence to estimate the position of the object. We focused now on the tracking of a 3D object in the scene and its 3D motion estimation using a range camera. This active vision device delivers, at each time, depth and reflectance (intensity) images. The depth measures associate each pixel with a measurement of the camera-object distance and the reflectance images provide a measurement based on the amplitude of the beam reflected by the scene.
To solve this estimation problem, we develop a particle filter, which allows a natural processing of the non linearities of the state model. We also use a centralized fusion structure, since it guarantees the optimality of the processing [16].
This model based is a top-down method : one use an *a priori* shape modelling of the object of his form which can thus be directly compared with the images delivered by the sensor. It also avoids any pre-processing stages that may generate additionnal localisation errors. This process requires a dense representation of the object for the sensor. To this end, we use a CAD model of the object and a 3D rendering engine which project the model in the image plane, for all position and orientation. Figure 1 presents, as an example, the model of the vehicle used for the synthetic sequence and its wireframe representation.

3 3D Modelling

The proposed solution relies on a 3D state modelling. Then the problem is summarized with a set of equations which describe the way the system evolves and the way it is measured by the sensors.

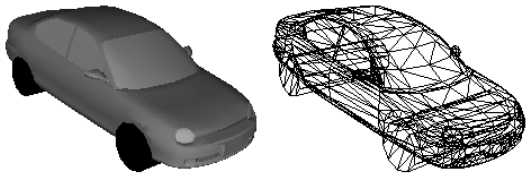

Fig. 1: Solid and wireframe representation of the 3D synthetic model of the car

3.1 Object modelling

Our approach uses a 3D rigid model of the object, defined by its geometry, which can be built using a 3D modeller for instance. This model is defined in a local reference frame, whose origin corresponds to the center of gravity of the object. It has 6 degrees of freedom : three translation parameters and three rotation parameters. The tracking stage leads to determinate the pose of the object [17]. It consists in finding the transformation (translation/rotation) of the object's coordinates between the local reference frame and the reference frame of the 3D sensor.

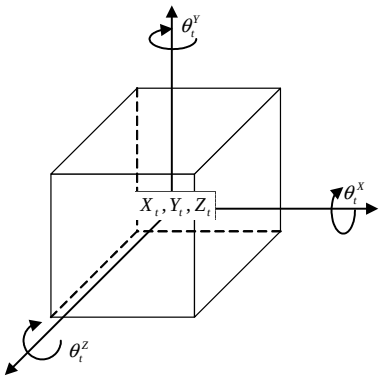

Fig. 2: 3D object model in the local reference frame

The state vector is composed by the pose and motion parameters:

$$\Omega_t = [X_t Y_t Z_t \theta_t^X \theta_t^Y \theta_t^Z T_t^X T_t^Y T_t^Z V_t^{\theta^X} V_t^{\theta^Y} V_t^{\theta^Z}]^T \quad (1)$$

- X_t, Y_t, Z_t are the coordinates of the center of gravity in the sensor reference frame;
- $\theta_t^X, \theta_t^Y, \theta_t^Z$ are the Euler angles which represent the object orientation;
- T_t^X, T_t^Y, T_t^Z are the coordinates of the translation vector;
- $V_t^{\theta^X}, V_t^{\theta^Y}, V_t^{\theta^Z}$ are angular speed of the object.

3.2 Dynamics equations

The dynamics equations, expressed in matrix form can be written as follows:

$$\Omega_{t+1} = F\Omega_t + W_t \quad (2)$$

- Ω_t is the state vector of the object
- F is the system's flow. The object evolves according to a rigid (translation/rotation) motion:

$$F = \begin{bmatrix} I_{6X6} & I_{6X6} \\ 0_{6X6} & I_{6X6} \end{bmatrix} \quad (3)$$

- W_t is an additive white Gaussian noise with zero mean and covariance Q_t.

3.3 Observation equations

The measurement equations for the intensity and range images are:

$$\begin{aligned} Z_{t+1}^I &= h^I(\Omega_{t+1}) + V_{t+1}^I \\ Z_{t+1}^R &= h^R(\Omega_{t+1}) + V_{t+1}^R \end{aligned} \quad (4)$$

- h^I and h^R are non-linear functions which link the object model (appearence and position) to the intensity and range pixels in the images;
- Z_{t+1}^I and Z_{t+1}^R are the intensity and range measures, respectively;
- V_{t+1}^I and V_{t+1}^R are additive white Gaussian noises with means μ^I and μ^R, and covariances R^I and R^R, respectively.

4 Particle estimation

The state equations show clearly the non-linear aspect of the problem. It is known that the usual estimation method (the Extended Kalman Filter) is not optimal in this case since it linearizes these equations. Moreover, it cannot ensure the stability and the convergence of the solution. The proposed method relies on the particle approach.

The basic principle of the particle method is to develop an approximation of the probability density function of the state vector conditionnaly to the available measures, solution of the estimation problem, by random particles. The limitation of this method lies in the finite number N of particles. Convergence results can be found in [18, 19], for instance.

4.1 Dynamic estimation of Markov processes

We shall deal here with discrete-time Markov processes that arise from dynamical models of the type:

$$\begin{aligned} X_{t+1} &= F_{t+1}(X_t, \pi_{t+1}) \\ Y_{t+1} &= H_{t+1}(X_{t+1}) + v_{t+1} \end{aligned} \quad (5)$$

where F_{t+1} and H_{t+1} are non-linear functions and π_{t+1}, v_t are independent noise sequences.

The optimal estimator $\widehat{X}_{t|t}$ (in the minimum mean square error sense) of X_t, from the knowledge of $Y_0^t = \{Y_0, \ldots, Y_t\}$, may be written in compact form:

$$\begin{aligned} \widehat{X}_{t|t} &= E\left[X_t | Y_0^t\right] \\ &= \int_{X_0^t} X_t P(X_0^t | Y_0^t) dX_0^t \\ &= \int_{X_t} X_t P(X_t | Y_0^t) dX_t \end{aligned} \quad (6)$$

The construction of the estimator lies in the knowledge of the probability density function $P(X_t|Y_0^t)$ of the state vector conditional to the observations.

Using Bayes rule and the Chapman-Kolmogorov equation, one gets a decomposition of the probability density function [20]:

$$P(X_t|Y_0^t) = \frac{\int_{X_0^{t-1}} \prod_{\tau=1}^{t} P(X_\tau|X_{\tau-1})P(X_0) \prod_{\tau=1}^{t} P(Y_\tau|X_\tau) dX_0^{\tau-1}}{\int_{X_0^t} \prod_{\tau=1}^{t} P(X_\tau|X_{\tau-1})P(X_0) \prod_{\tau=1}^{t} P(Y_\tau|X_\tau) dX_0^{\tau}} \quad (7)$$

So the solution of Markov processes filtering $P(X_t|Y_0^t)$ is based on two probability density functions:

- the transition law $P(X_\tau|X_{\tau-1})$ from the state $X_{\tau-1}$ at time $\tau - 1$ to the state X_τ at time τ;
- the observation law $P(Y_\tau|X_\tau)$ which is linked to the measurement equation.

4.2 Basic particle procedure [18]

The basic particle method uses a probability space representation with N Dirac measures (particles) whose supports X_t^i and weights p_t^i are conditioned by the measurements.

The basic particle approach to the filtering problem can be summarized into four points:

1. Filter initialization: Each particle X_0^i is initialized according to a random sampling of the *a priori* law $P(X_0)$. The initial weights p_0^i are set to $\frac{1}{N}$ (where N is the number of particles);

2. Evolution: The particles $(X_t^i)_{i=\{1,\dots,N\}}$ evolve in the state space according to the system's stochastic flow, through the generation of N random sequences of independent π_{t+1}^i with law P(π_{t+1});

$$X_{t+1}^i = F_{t+1}(X_t^i, \pi_{t+1}^i) \quad (8)$$

3. Weighting: The above evolution is followed by a weighting stage, in which the normalized weights p_{t+1}^i must be corrected by the measures available at time $t+1$, according to Bayes' rule:

$$p_{t+1}^i = \frac{P(Y_{t+1}|X_{t+1}^i)}{\sum_{j=1}^{N} P(Y_{t+1}|X_{t+1}^j)} p_t^i \quad (9)$$

Under the common assumption of noise vector independence, the weights in the multisensor framework can be computed as follows:

$$p_{t+1}^i = \frac{\prod_{m=1}^{M} P(Y_{t+1}^m|X_{t+1}^i)}{\sum_{j=1}^{N} \prod_{m=1}^{M} P(Y_{t+1}^m|X_{t+1}^j)} p_t^i \quad (10)$$

4. Estimation: The particle estimation is the weighted sum of the particles X^i:

$$\hat{X}_{t+1|t+1} = \sum_{i=1}^{N} p_{t+1}^i X_{t+1}^i \quad (11)$$

For a dynamical processing, parts 2,3 and 4 must be time-iterated.

One can represent (fig. 3) the discretization using the particle method of the probability density function $P(X_t|Y_0^t)$ of the state vector conditional to the observations. This figure depicts the basic evolution/weighting procedure where the particles are represented as Dirac measures. The evolution of each particle in the state space (shown in dot line) is done according to eq. 8. At each time, the weights are computed using equation 9. The magnitude of the Dirac measures (fig. 3) is related to the weight of the particle. Finally, the weighted Dirac comb represents a "particle discretization" of the probability density function $P(X_t|Y_0^t)$.

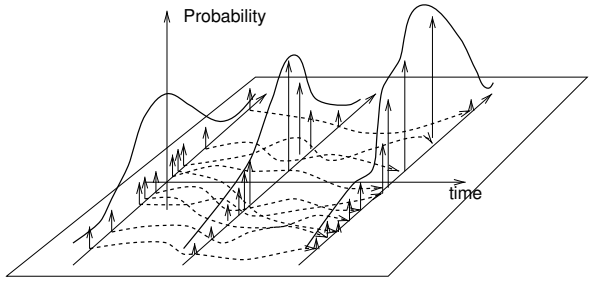

Fig. 3: Evolution of the particle network

4.2.1 Redistribution procedure

It is known that the basic particle procedure does not prevent some of the weights to be low compared to those of other particles and therefore to poorly contribute to the performance of the estimator (eq. 11). Indeed, the probability will concentrate on an unique particle and the optimality of the solution cannot be ensured.

To this end, a redistribution technique may be applied which is an application of elementary resampling principle. It consists of redistributing all particles from the explored positions of the state space, by sampling in accordance to the acquired probability weights. The procedure allows "heavy" particles to give birth to more particles at the expense of "light" particles which die.

4.3 Rendering process

The computing of the weights p^i_{t+1} is done according to the measurements at time $t+1$ and their particle-based reconstruction. This reconstruction stage uses the non-linear transfer functions (h^I, h^R) to project the 3D model of the object on each sensor.

In this section, one details the rendering process of the object, which can be mainly split in two stages:

- Model transformation: the characteristics parameters of the object must be converted from the local reference frame to the reference frame of the sensor using a rigid transformation:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{3X3} & T_{3X1} \\ 0_{1X3} & 1_{1X1} \end{pmatrix} \begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix} \quad (12)$$

  - (X_m, Y_m, Z_m) and (X_c, Y_c, Z_c) are the coordinates of a 3D point respectively in the local and the sensor-based reference frames;
  - $R = R_{\theta^X}.R_{\theta^Y}.R_{\theta^Z}$ is the rotation matrix (3x3) defined by the angular parameters of the state vector. $(R_{\theta^i})_{i=\{X,Y,Z\}}$ are the rotation matrices for the axis $\theta^X, \theta^Y, \theta^Z$;
  - $T = (X, Y, Z)^T$ is the vector composed by the coordinates of the center of gravity in the sensor-based reference frame.

- Projection: the transfer function of the sensor is used to project the 3D object in the image planes (range and intensity). Two sensor models are more particularly studied:

  1. The following model have been used for the synthetic sequence:

$$\begin{array}{lcl} u^{R,I} & = & \alpha_u \dfrac{Y_c}{Z_c} + u_0 \\ v^{R,I} & = & \alpha_v \dfrac{X_c}{Z_c} + v_0 \\ n^I(u,v) & = & f_1\left(X_c, Y_c, Z_c\right) \\ n^R(u,v) & = & Z_c \end{array} \quad (13)$$

  2. In the real data case, one uses the *Odetics* sensor which can be modelled by [21]:

$$\begin{array}{lcl} u^{R,I} & = & \alpha'_u tan^{-1}\left(\dfrac{Y_c}{Z_c}\right) + u'_0 \\ v^{R,I} & = & \alpha'_v tan^{-1}\left(\dfrac{X_c}{Z_c}\right) + v'_0 \\ & & +\beta'_v \alpha'_u tan^{-1}\left(\dfrac{Y_c}{Z_c}\right) - \beta'_v u'_0 \\ n^I(u,v) & = & f_2\left(X_c, Y_c, Z_c\right) \\ n^R(u,v) & = & \sqrt{X_c^2 + Y_c^2 + Z_c^2} - a_0 \end{array} \quad (14)$$

     Where:

     - $u^{R,I}$, $v^{R,I}$ are the pixel coordinates of the 3D point respectively in the range (R) and intensity (I) images;
     - $n^R(u,v)$ and $n^I(u,v)$ are the grey levels of the pixel with coordinates u, v in the range and intensity images;
     - $(\alpha_u,\ u_0,\ \alpha_v,\ v_0,\ \beta_v)$ and $(\alpha'_u,\ u'_0,\ \alpha'_v,\ v'_0,\ \beta_v,\ a_0)$ are the intrinsic parameters of the synthetic and *Odetics* sensors, respectively;
     - f_1 et f_2 detail the illumination models for the synthetic and real sequences.

5 Results

The results detailed in the sequel use 10000 particles to provide an accurate estimation, although the method performs well with 1000 particles. Concerning the timing for running this method, it must be noticed that it can be much improved using hybrid methods such as Kalman particle filtering, for instance. Moreover, it is linked to the polygonal complexity of the object since the rendering process is done by the graphic card and to the speed rate between the computer's memory and the graphic memory. Typically, without any optimization, the algorithm performs in approximately 7s/frame for the real sequence and 40s/frame for the synthetic sequence using 10000 particles.

5.1 Synthetic sequence

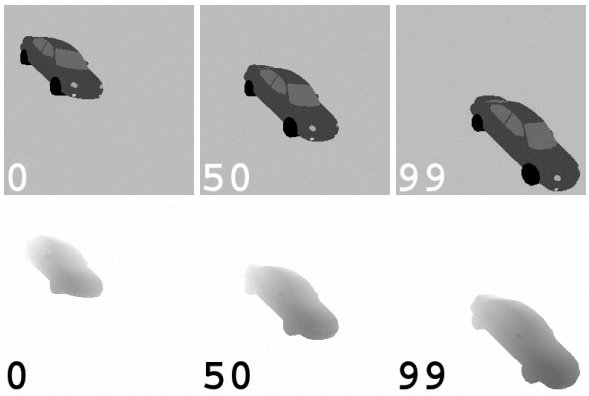

Fig. 4: Intensity and range images of the synthetic sequence at t={0, 50, 99}

The method is first applied to a sequence of 100 synthetic range and intensity images, whose resolution is equal to 256x256 pixels. The tracked object is a vehicle that evolves according to a rigid motion. Figs. 4 show the images at the beginning, the middle and the end of the sequence (the first line shows the intensity images and the bottom line the range images).

	Position	Translation	Orientation (rad)	Speed (rad/frame)
Errors	0.008	0.006	0.002	0.002

Table 1: Estimation errors for the synthetic sequence

Figs. 5 shows the car tracking results. The wireframe estimate is displayed on the range and intensity measurements. As it can be seen on Figs. 6, after a short period of convergence, the algorithm tracks efficiently the vehicle

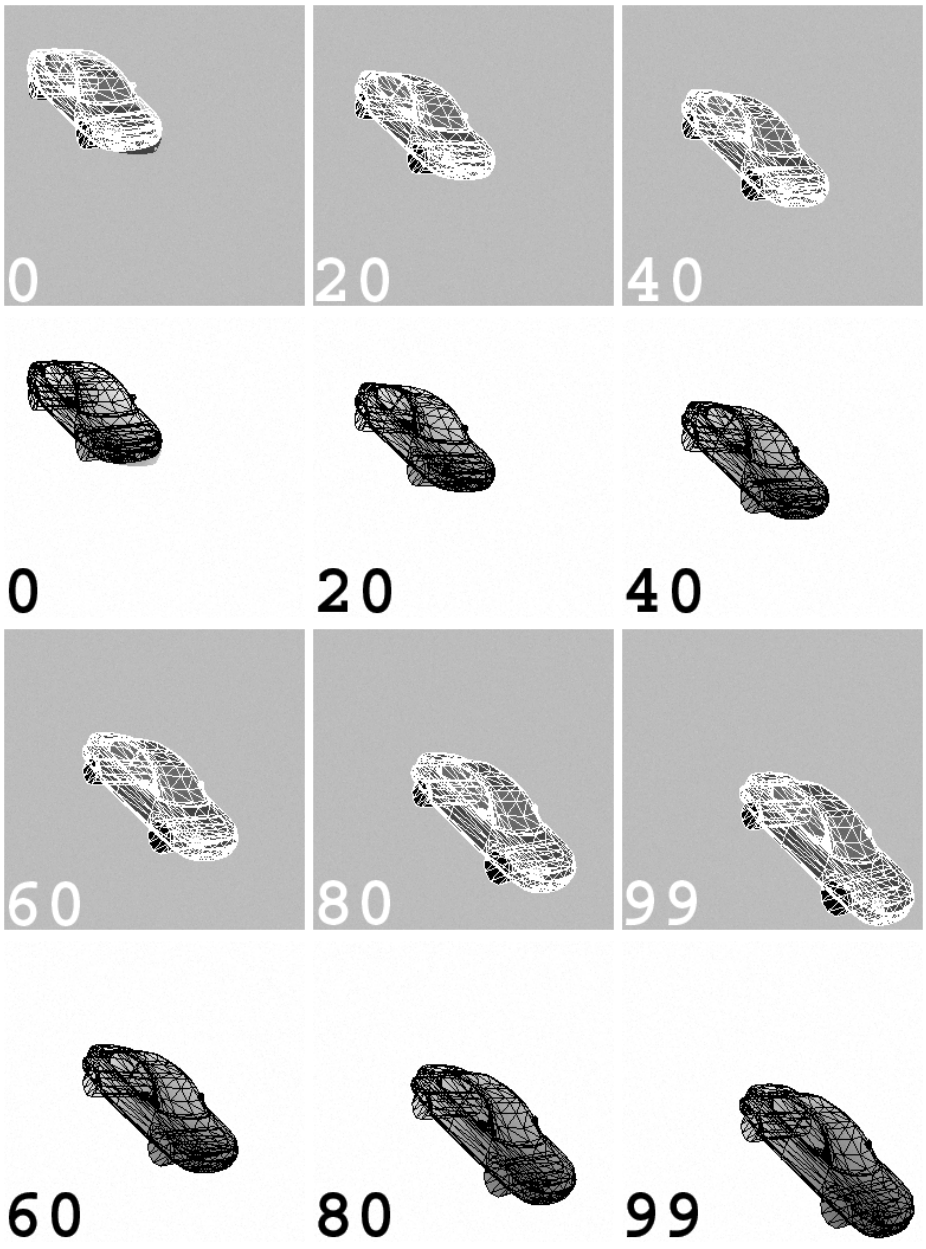

Fig. 5: Car tracking results at t={0, 20, 40, 60, 80, 99}

and provides an accurate estimation of the 3D motion and position parameters. Table 1 summarizes the estimation errors in this synthetic case.

5.2 Real sequence

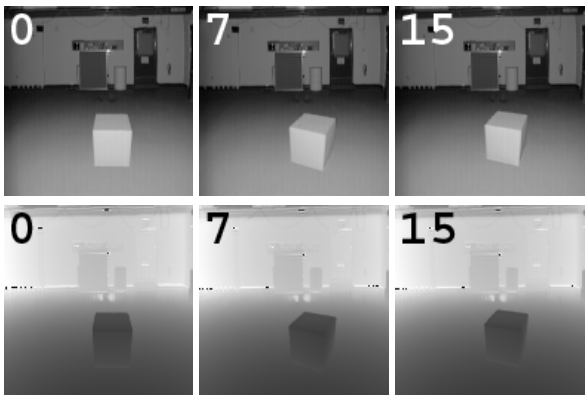

Fig. 7: Intensity and range images of the real sequence at t=$\{0, 7, 15\}$

The previous sequence allows an evaluation of the accuracies of the proposed method on synthetic data. In order to study the robustness of the algorithm, it is now applied on a real image sequence from the range image database of the University of South Florida (available at http://marathon.csee.usf.edu/range/DataBase.html).

This sequence is delivered by an *Odetics* sensor whose resolution is 128x128 and composed by 16 images of a polyhedric object that evolves according to a rigid motion. The sensor delivers at each time instant, range and reflectance images. The proposed particle solution fuses both measurements to track the object in the sequence and jointly estimate the 3D positions and motion.

Figure 8 shows the tracking results in this case. Despite the weak image resolution, a motion with high magnitude ($\approx 20^\circ$/frame) and the lack of information concerning the lighting of the scene, the method allows an efficient tracking of the shape (see Figs. 8).

6 Conclusion

In this paper, we proposed a method for object tracking that jointly estimates its 3D position and motion parameters by fusing intensity and range images. In this model-based approach, the *a priori* information about the shape to be tracked (as, for instance, part of an image database) avoids the usual preprocessing stage (token extraction, ...) which leads to an increasing accuracy in the parameters estimation.

This non-linear estimation problem is solved by the particle filtering due to its ability to deal non-linear models (dynamics/measurement, statistics). The proposed solution is then applied to synthetic and real sequences of range and intensity/reflectance images. More particularly, this method allows an efficient tracking with weak resolution images and high magnitude motion together with accurate estimation of 3D position and motion parameters.

Acknowledgements

The authors would like to thank the Ministère de l'Enseignement Supérieur et de la Recherche of the French Government and the Région Nord-Pas-de-Calais for their financial support.

The authors are also grateful to Dr A. Hoover for his technical support on the USF Range Image Database.

References

[1] C. Boucher, J.-C. Noyer, and M. Benjelloun. 3D structure and motion recovery by fusing range and intensity image sequences. In *Proceedings of the 3rd International Conference on Information Fusion*, Paris, France, July 2000.

[2] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

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

[4] E. de Castro and C. Morandi. Registration of translated and rotated images using finite Fourier transforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5):700–703, september 1987.

[5] H. Foroosh, J.B. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Trans. on Image Processing*, 11(3):188–200, march 2002.

[6] Soon Ki Jung and Kwang Yun Wohn. A model-based 3d tracking of rigid objects from a sequence of multiple perspective views. *Pattern Recognition Letters*, 19, 1998.

[7] D. Koller, K. Daniilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3):257–281, 1993.

[8] A. Gagalowicz P. Gérard. Three dimensional model-based tracking using texture learning and matching. *Pattern Recognition Letters*, 21:1095–1103, 2000.

[9] E. Marchand, P. Bouthemy, and F. Chaumette. A 2D/3D model-based approach to real-time visual tracking. *Image and Vision Computing*, 19(13):941–955, 2001.

[10] C. Boucher, J.-C. Noyer, and M. Benjelloun. 3D structure and motion recovery in a multisensor framework. *International Journal of Information Fusion*, 2(4):271–285, December 2001.

[11] P. Del Moral, J.-C. Noyer, G. Rigal, and G. Salut. Traitement non-linéaire du signal par réseau particulaire: Application radar. In *Actes Du 14ème Colloque Gretsi sur Le Traitement Du Signal et Des Images*, pages 399–402, Juan-les-Pins, France, 13-16 Septembre 1993.

[12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to Non-linear/Non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2), April 1993.

[13] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998.

[14] P. Lanvin, J-C. Noyer, and M. Benjelloun. Non-linear estimation of image motion and tracking. In *IEEE International Conference on Multimedia and Expo*, pages II–781/II–784, Baltimore, USA, July 2003.

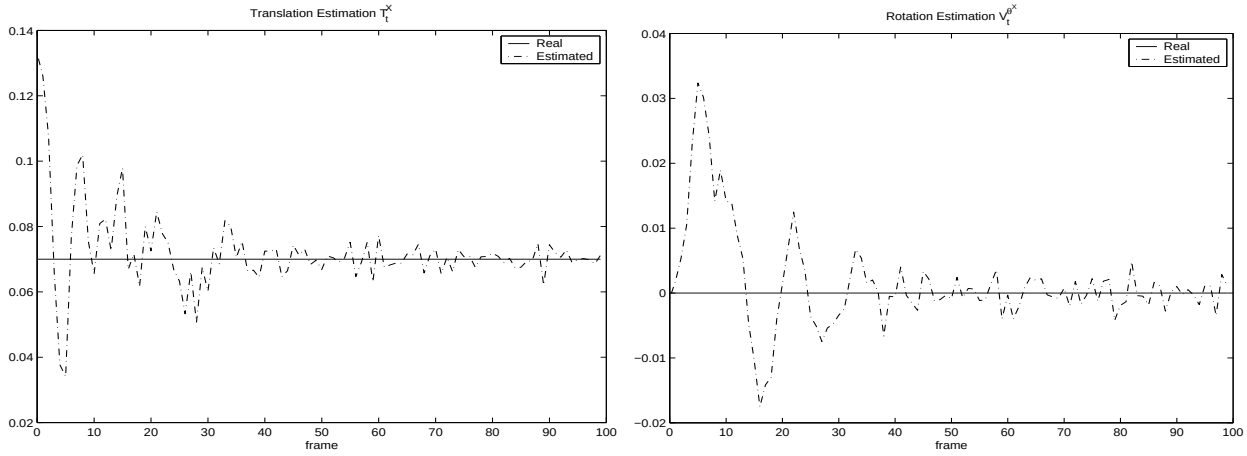

Fig. 6: Estimation errors on the translation and rotation parameters along X

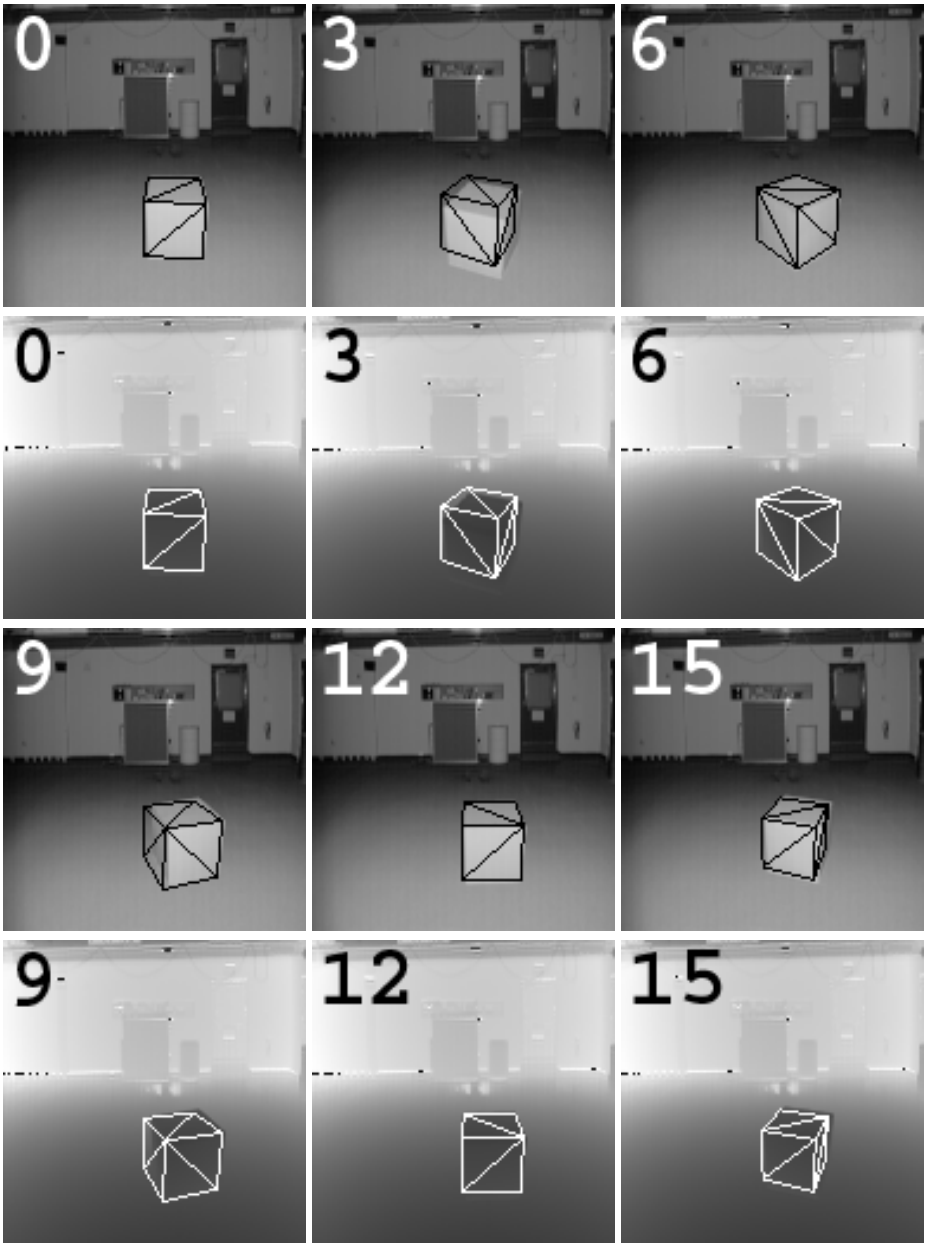

Fig. 8: Tracking results of the cube at t={0, 3, 6, 9, 12 and 15}

[15] J-C. Noyer, P. Lanvin, and M. Benjelloun. Non-linear matched filtering for object detection and tracking. *Pattern Recognition Letters*, 25(6):655–668, 2004.

[16] Y. Bar-Shalom and X. R. Li. *Multitarget Multisensor Tracking*. YBS Publication, 1995.

[17] E. Polat, M. Yeasin, and R. Sharma. A 2D/3D model based object tracking framework. *Pattern Recognition*, 36:2127–2141, 2003.

[18] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte-Carlo Methods in Practice*. Springer Verlag, 2002.

[19] P. Del Moral. *Résolution Particulaire Des Problèmes d'Estimation et d'Optimisation Non-Linéaires*. PhD thesis, Université Paul Sabatier, Toulouse, 1994.

[20] H. Carvalho, P. Del Moral, A. Monin, and G. Salut. Optimal non-linear filtering in GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 33(3), November 1997.

[21] A. Hoover. Descriptions of the odetics LRF and the range images. Technical report, Deparment of Computer Science and Engineering, University of South Florida, Tampa, Florida 33620 USA, April 1994.