3D Motion Estimation Using a Token Tracker *

G. TOSCANI, R. DERICHE, O.D. FAUGERAS

INRIA

78153 Le Chesnay - France

Abstract

In this article we describe the implementation of a system for tracking tokens on a sequence of perspective views and use the result to compute the 3D motion of a camera and the 3D structure of the scene (up to a scale factor).

We provide first an overview of the entire process for measuring image flow. We then present a parametric representation for line segments which simplifies the implementation of tracking edge lines. This is followed by the description of prediction, matching and updating of the image flow model. This formalism makes possible a fast matching algorithm.

Finally we consider the 3D motion estimation using the *extremities of the segments* being tracked. The fact that they do not correspond in general to the same physical point, is a major source of error. Another one is the presence of *incorrect matches*. In this case the existing techniques for motion estimation fail while our technique proved particularly robust : the incorrect matches are detected by the motion estimation unit and the information on false correspondences are given to the token tracker.

Experiments have been carried out on real image sequences taken by a moving camera, showing that 3D motion and 3D structure estimation is fast and reliable.

1 THE TOKEN TRACKER

Given a sequence of images, one has to track moving objects in the scene. Our approach uses tokens based on line segments corresponding to the edges extracted from the scene. However, it is worthwhile to note that other tokens as points of interest (corners, triple points..) can be considered without affecting deeply the algorithm. The edges are obtained through the use of an optimal operator previously developed [1]. An edge linking step and a polygonal approximation give the line segments on which the tracking is done. Our tracking approach is based on a combination of a prediction step and a matching process. Kalman filtering is used to help tracking by providing reasonable estimates of the region where the matching process has to seek for a possible match between tokens. Correspondence in the search area is done through the use of a similarity function based on strong features of the line segments. When working with a large sequence of frames, it is possible that some objects may appear or disappear totally or partially ; our Kalman filtering based approach allows to handle this problem of occlusion in a effective way. Experiments have been carried out on noisy synthetic data and on real scenes obtained from a mobile robot. Some experimental results concerning the real scenes are shown.

2 Prediction

Each token (i.e. line segment) is characterized by the following five parameters :

- The orientation θ of the line segment.
- The magnitude of the gradient along the line segment.
- The length L of the line segment.
- The distance of the origin to the line segment denoted by the parameter c.
- The distance denoted d, along the line from the perpendicular intersection to the midpoint of the segment.

A Kalman filter is used to aid tracking by providing reasonable estimates of the region where the matching process has to seek for a possible match between tokens.

Kalman filtering is a statistical approach to estimate a time-varying state vector X_t from noisy measurements Z_t. Consider the estimation of X_{t+k} from the measurements up to the instant t, Kalman filtering is a recursive estimation scheme designed to match the dynamic system model, the statistics of the error between the model and reality, and the uncertainty associated with the measurements.

In our approach, a Kalman filter is used separately on each of the five parameters defined above to estimate a state vector which is simply the time varying motion parameters of interest namely the position, the velocity and the acceleration for the parameters c and d, the angular position and its angular velocity for the parameter θ and only the position for the parameters length L and magnitude of the gradient G assumed to be constant.

The Kalman filter equations used in this paper involve discrete time steps: State vector notation is used such that $X_t^T = (x_t, \dot{x}_t, \ddot{x}_t)$ is the position, velocity, and acceleration of the parameter considered (c,d,θ,L,or G) at the t^{th} time step. The Kalman filter can be viewed as consisting of the following steps:

- The model of the system dynamics is:

$$X_t = \Phi_{t,t-1} X_{t-1} + W_{t-1}$$

The error of the model from reality is given by W_t, a zero-mean white Gaussian process of covariance Q_t :

$$E(W_t) = 0 \ ; \ E(W_t W_t^T) = Q_t$$

$\Phi_{t,t-1}$ is a matrix which evolves the position x, the velocity $\dot{x}$ and the acceleration $\ddot{x}$ from one time sample to another. It is easy to verify that assuming a motion with constant acceleration leads to the following matrix $\Phi_{\Delta t}$:

$$\Phi_{\Delta t} = \begin{pmatrix} 1 & \Delta t & \frac{(\Delta t)^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix}$$

This is the well known equation of a dropped object for a time interval Δt.

- The measurement model used is:

$$Z_t = H_t . X_t + V_t$$

where Z_t is a vector of measurements with an uncertainty V_t assumed to be a zero-mean Gaussian process of covariance R_t.

$$E(V_t) = 0 \ ; \ E(V_t V_t^T) = R_t$$

*This work was partially supported by Esprit Project P940

When H_t is the identity matrix, the measurements correspond directly to the state vector.

In our application, the measurement model Z_t assumes that the position x (i.e the value of c, d, θ, L or G) is measurable from the matching process while the velocity $\dot{x}$ and the acceleration $\ddot{x}$ are not. Therefore Z_t is the scalar corresponding to the position x and R_t is simply the uncertainty on x. Choosing this uncertainty in a manner reflecting our a priori estimate of the amount of noise to be expected from the previous step (Digitizing effects, edge detection and polygonal approximation) leads to deal with a small uncertainty for the parameters c, θ and G, and a large uncertainty for the unreliable parameters L and d due to the random effects which break line segments.

- State prediction:

$$\hat{X}_{t/t-1} = \Phi_{t,t-1}\hat{X}_{t-1/t-1}$$

After the measurement at time t-1 has been done, this time update equation predicts the system state at time t from the estimated values of the system state at time t-1. In our application, this equation predicts the value for x at time t from the informations available at time t-1.

- Covariance prediction for state vector :

$$P_{t/t-1} = \Phi_{t,t-1}P_{t-1/t-1}\Phi_{t,t-1}^T + Q_{t-1}$$

This equation gives the statistics relating the estimated state vectors to the unmeasurable ideal state vectors.

- Covariance prediction for the measurement vector:

$$U_{t/t-1} = H_t P_{t/t-1} H_t^T + R_t$$

This equation gives the statistics of the estimated model measurement. In our application, it determines the *search area* for the matching process.

- Kalman Gain:

$$K_t = P_{t/t-1}.H_t^T[H_t P_{t/t-1} H_t^T + R_t]^{-1}$$

This equation indicates how much to weight each new measurement. Note that a small uncertainty R_t (precise measurement) causes a large weighting K_t and therefore leads to a corrected state estimate determined mainly by the measurement. A large uncertainty R_t causes a small weighting K_t.

- State correction :

$$\hat{X}_{t/t} = \hat{X}_{t/t-1} + K_t(Z_t - H_t\hat{X}_{t/t-1})$$

This equation is used to update the state model.

- Covariance correction :

$$P_{t/t} = P_{t/t-1} - K_t H_t P_{t/t-1}$$

This equation is used to update the statistics coupling the estimated state vectors to the unmeasurable ideal state vector.

- Covariance correction on the measures :

$$U_{t/t} = R_t(I - K_t^T H_t^T)$$

This equation is used to update the statistics coupling the estimated model measurement after the measurement at time t has been done.

In all the equations the "t_1/t_2" indicates an estimate at time t_1 after a measurement at time t_2 has provided more information. In our application, a new measurement is considered each time a correspondence has been done. All the above equations completely define the Kalman filtering we use in the prediction step.

3 Matching

For each token of the image model, a selection of the region where the matching process has to seek for a possible match between tokens is provided through the use a similarity function. This function uses the uncertainties provided by the Kalman filtering step on each estimate. It allows to keep only the tokens within the search area. A score for each correspondence is then calculated in order to disambiguate some possible multiple matches, using a difference measure. To this end, we use a normalized distance between the tokens, defined as a weighted sum of differences between the respective parameters values. Once a token from the current frame has been matched, we use its parameters as a new measure, first to update its corresponding state vector and second to provide the search area where the matching process has to look for a possible match between the image flow model and the next frame.

3.1 Bootstrapping step

First, at $t = 0$ and before the tracking algorithm can operate that is in the bootstraping step and each time that a new token appears, the matching algorithm has no idea where to look for the matching process. Therefore, a set of initial position velocity and acceleration guesses is used to initialize the tracking process. This is done by choosing somewhat arbitrarily values for the position velocity and acceleration but assigning large uncertainty so these initial values will not be weighted heavily as the measurement process continues. In our application, the position is set to be constant while the velocity and the acceleration are set to zero.

3.2 Search area and correspondences

The search area is determined through a simple set of attribute tests using the result of the Kalman filtering. For each token of the image flow model, represented by a feature vector of 5 components, we wish to know which token might correspond to it. This is done through the use of a simple set of attribute tests using the current values of the measure, the expected value of the component and its uncertainty. This leads to calculate the Mahalanobis distance, explained below, for each components and to declare a token of the new frame inside a search area if all the distances are less than a fixed threshold.

The correspondence is controlled by the use of a normalized distance based on the attributes of each line segment. This distance works as a cost function. It is calculated for each possible match and the best score is used to validate the most consistent. In selecting a cost function for correspondences, we wanted to take into account the distance between the respective parameters values and the uncertainty associated with each parameter. A good measure is then the so called Mahalanobis distance. It weights each difference measure between the new token and the estimated token by the uncertainty of the estimated token. It is defined as follows:

Let each new token, issued from the matching process, be represented by a feature vector of N components denoted T_m and the estimated token represented by T_e with an uncertainty Λ. The Mahalanobis distance is then defined to be :

$$M(T_m, T_e) = (T_m - T_e)^T\Lambda^{-1}(T_m - T_e)$$

The difference between T_m and T_e is easy to calculate for the length, norm gradient, c and d components. In order to deal with the problem of the difference between two angles θ_1 and θ_2, the cosine of the difference has been used.

4 Experimental Results of the Token Tracker

Experiments have been carried out with the proposed token tracking approach. The tracking algorithm has been applied to several sequences of real images taken from a mobile robot.

In figure 1 to 3 are shown the extracted line segments on a sequence of 3 images.

Figure 1: Frame number 1.

Figure 2: Frame number 2.

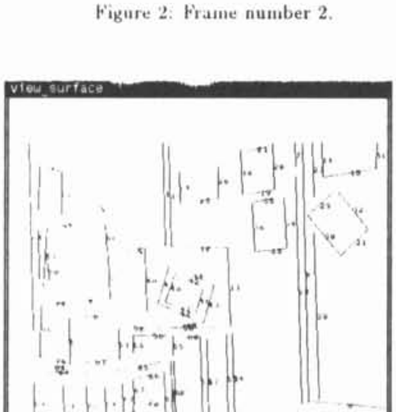

Figure 3: Frame number 3.

The tracking is illustrated through the numbers assigned to each line segment. A close look at the results reveals how some line segments can appear or disappear. A new label is affected as soon as a new segment appears and the process continues without affecting the tracking algorithm. A label corresponding to a good correspondence remains during all the process while false correspondences are removed after three frames generally. It should be pointed out that the algorithm can cope with line segments moving with differents motions. This illustrates the efficiency of the used approach.

5 THE MOTION ESTIMATION

The problem is now to use the tokens matched over the sequence of images to compute the 3D motion and 3D structure of the scene.

In our case the tokens being matched are the *extremities of segments*. The fact that they do not correspond in general to the same physical point is a new major source of error. Another one is the presence of *incorrect matches*.

Many techniques have been proposed to solve this problem with linear [4,7] or iterative [3] algorithms. Unfortunately they reach their limit very quickly as the noise in the data increases [2,5].

The iterative algorithm we used, which prove to be robust on real images, is based on a very simple concept: the idea is to look, among all the possible motions, the one that minimizes the difference between the measured, actual image and the image obtained, synthesized using this motion.

6 Problem Definition

In order to compute the rigid motion of the moving camera we shall consider a number of 3D points $\mathbf{P}$ matched by the token tracker over 2 perspective views : the image coordinates $[x, y, 1]^T$ of the 3D points are known while the 3D coordinates are unknown.

We shall consider the system of Figure 4, where the camera has been desplaced from position 1 to position 2 by a rigid motion of rotation $\mathbf{R}$ (a (3x3) rotation matrix) and translation $\mathbf{t}$. $\mathbf{R}$ and $\mathbf{t}$ are the unknowns of our problem.

There are 3 parameters to be computed for the rotation. A translation vector $\mathbf{t} = [t_x, t_y, t_z]^T$ has 3 components, but it is well known that, from a sequence of projections, we can recover the motion up to a scale factor at most. This means that the parameters to be computed for the translation are just 2.

7 Reconstruction & Reprojection

Let $\mathbf{Q}_1$ and $\mathbf{Q}_2$ be the effective projections of a 3D point $\mathbf{P}$ on the two image frames, with $\mathbf{w}_1 = \mathbf{O_1Q_1} = [x_1, y_1, 1]^T$ and $\mathbf{w}_2 = \mathbf{O_2Q_2} = [x_2, y_2, 1]^T$ as shown in Figure 4.

Let us make a simple geometric remark, helping ourselves with Figure 4. In this figure we can see that, given $\mathbf{Q}_1$ in image plane 1, all possible physical points which may have produced $\mathbf{Q}_1$ are on the infinite half-line O_1Q_1.

Given an estimate of rotation and translation, $\mathbf{R}_e$ and $\mathbf{t}_e$, we can then consider the two infinite half-lines O_1Q_1 and O_2Q_2. If $\mathbf{R}_e$ and $\mathbf{t}_e$ are the true rotation and translation and the image coordinates of the two points $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are exactly known, O_1Q_1 and O_2Q_2 intersect in $\mathbf{P}$.

In general, due to incorrect knowledge of rotation and translation and noise on the measured image points $\mathbf{Q}_1$ and $\mathbf{Q}_2$, the two lines O_1Q_1 and O_2Q_2 do not intersect, as shown in Figure 4. In this case we have to decide what would be a good estimate of $\mathbf{P}$. This can be chosen in many ways:

1. the middle point $\mathbf{P}^{(1)}$ of the intersections of O_1Q_1 and O_2Q_2 with their common perpendicular σ; this is the reconstruction used in [5] and is the one shown in figure 4,
2. the point $\mathbf{P}^{(2)}$ that minimize the "error vectors" (see section 8) ; this is the reconstruction implicitly used in [6],
3. the intersection $\mathbf{P}_r^{(3)}$ of line O_1Q_1 with the plane Ω defined as follows : it contains O_2Q_2 and it is perpendicular to the plane Σ passing thru $\mathbf{O}_1$ and $\mathbf{O}_2$ and containing O_1Q_1; in order to keep the symmetry of the problem we shall consider also the point $\mathbf{P}_l^{(3)}$, defined in a similar way.

Our tests show that the reconstruction plays an important role.

Let $\mathbf{P}^{(i)}(i = 1, 2, 3)$ be the reconstructed 3D point using one of the 3 methods. Let $\mathbf{Q}_1'(\mathbf{R}_e, \mathbf{t}_e)$ (resp. $\mathbf{Q}_2'(\mathbf{R}_e, \mathbf{t}_e)$) be the projection of $\mathbf{P}^{(i)}$ on the first (resp. second) image plane.

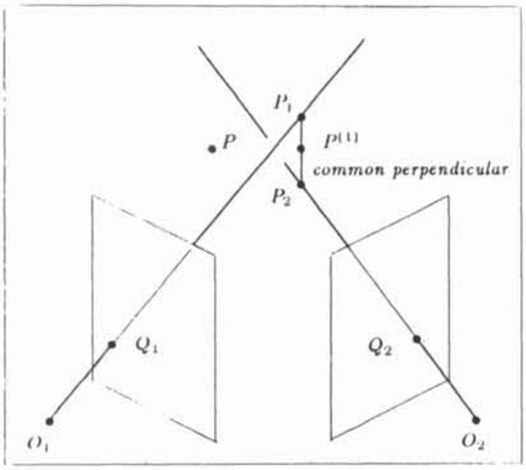

Figure 4: Reconstruction using the middle point of the common perpendicular.

The idea is to look, among all the possible rotations and translations, the ones that produce the "best" reconstructed image.

We define as the "best reconstructed image" the one that minimize the "error vectors" $| \mathbf{Q}'_1(\mathbf{R}_e, \mathbf{t}_e)\mathbf{Q}_1 |$ and $| \mathbf{Q}'_2(\mathbf{R}_e, \mathbf{t}_e)\mathbf{Q}_2 |$ for all matched points (figure 5)

The criterion to minimize can be written as follows :

$$\mathcal{C}^{(i)}(\mathbf{R},\mathbf{t}) = \sum_{\substack{\text{all matched}\\ \text{points}}} \left(| \mathbf{Q}'_1(\mathbf{R},\mathbf{t})\mathbf{Q}_1 |^2 + | \mathbf{Q}'_2(\mathbf{R},\mathbf{t})\mathbf{Q}_2 |^2 \right)$$

$$\stackrel{\text{def}}{=} \mathcal{C}_l^{(i)}(\mathbf{R},\mathbf{t}) + \mathcal{C}_r^{(i)}(\mathbf{R},\mathbf{t}) \tag{1}$$

where $\mathcal{C}_l^{(i)}$ (resp. $\mathcal{C}_r^{(i)}$) refers to the error vectors of the left (resp. right) image using the $i^{th} (i = 1, 2, 3)$ reconstruction method.
We have considered also the Longuet-Higgins [4] criterion :

$$\mathcal{C}^{(4)}(\mathbf{R},\mathbf{t}) = \sum_{\substack{\text{all matched}\\ \text{points}}} (\mathbf{w_2} \cdot (\mathbf{t} \wedge \mathbf{R}\mathbf{w_1})) \tag{2}$$

Criteria $\mathcal{C}^{(1)}$, $\mathcal{C}^{(3)}$ and $\mathcal{C}^{(4)}$ have been derived and minimized using a steepest descent method and Kalman filtering, while criterion $\mathcal{C}^{(2)}$ has been minimized using a finite difference technique.

8 Minimization of the Criterion $\mathcal{C}^{(2)}$

In order to minimize criterion $\mathcal{C}^{(2)}$ we have to reconstruct the point $\mathbf{P}^{(2)}$ for every matched point $\mathbf{P}$. This is the solution of a minimization problem, which can be written in the following equivalent ways :

Minimization of 4 variables subject to nonlinear constraint

The point $\mathbf{P}^{(2)}$ is the one that minimize the magnitude of the error vectors :

$$\mathbf{w_1} + \Delta\mathbf{w_1} \stackrel{\text{def}}{=} \mathbf{Q}'_1\mathbf{Q}_1 \quad \text{and} \quad \mathbf{w_2} + \Delta\mathbf{w_2} \stackrel{\text{def}}{=} \mathbf{Q}'_2\mathbf{Q}_2$$

where :

$\Delta\mathbf{w_1} = [\Delta x_1, \Delta y_1, 1]^T$ and $\Delta\mathbf{w_2} = [\Delta x_2, \Delta y_2, 1]^T$

The criterion to be minimized is then the following :

$$\mathcal{C}_{(a)}(\Delta x_1, \Delta y_1, \Delta x_2, \Delta y_2) = \Delta\mathbf{w_1}^2 + \Delta\mathbf{w_2}^2$$

under the constraint :

$$(\mathbf{w_2} + \Delta\mathbf{w_2}) \cdot (\mathbf{t} \wedge \mathbf{R}(\mathbf{w_1} + \Delta\mathbf{w_1})) = 0$$

Minimization of 3 variables without constraint

Let $\mathbf{O_1}\mathbf{P}^{(2)} = \mathbf{X} = [x, y, z]^T$. We have :

$$\Delta\mathbf{w_1} = [\tfrac{x}{z}, \tfrac{y}{z}, 1]^T \text{ and } \Delta\mathbf{w_2} = \left[\frac{(\mathbf{R}\mathbf{X})_x}{(\mathbf{R}\mathbf{X})_z}, \frac{(\mathbf{R}\mathbf{X})_y}{(\mathbf{R}\mathbf{X})_z}, 1\right]^T$$

The previous criterion can be written in the following manner :

$$\mathcal{C}_{(b)}(x, y, z) = \left(x_1 - \frac{x}{z}\right)^2 + \left(y_1 - \frac{y}{z}\right)^2 + \left(x_2 - \frac{(\mathbf{R}\mathbf{X})_x}{(\mathbf{R}\mathbf{X})_z}\right)^2 + \left(y_2 - \frac{(\mathbf{R}\mathbf{X})_y}{(\mathbf{R}\mathbf{X})_z}\right)^2$$

Minimization of 2 variables subject to nonlinear constraint

It can be shown that the previous criterion is equivalent to the following :

$$\mathcal{C}_{(c)}(\Delta x_1, \Delta y_1) = \Delta\mathbf{w_1}^2$$

under the constraint :

$$(\Delta x_1)^2 + (\Delta y_1)^2 = \left(x_2 - \frac{(\mathbf{R}\Delta\mathbf{w_1})_x}{(\mathbf{R}\Delta\mathbf{w_1})_z}\right)^2 + \left(y_2 - \frac{(\mathbf{R}\Delta\mathbf{w_1})_y}{(\mathbf{R}\Delta\mathbf{w_1})_z}\right)^2$$

9 Choice of the segments

There are two major sources of errors in using points as tokens to compute the motion:

- noise in the detection of the point,
- incorrect matches.

In using segments we have to add another source of error:

- the extremities of matched segments do not correspond, in general, to the same physical 3D points.

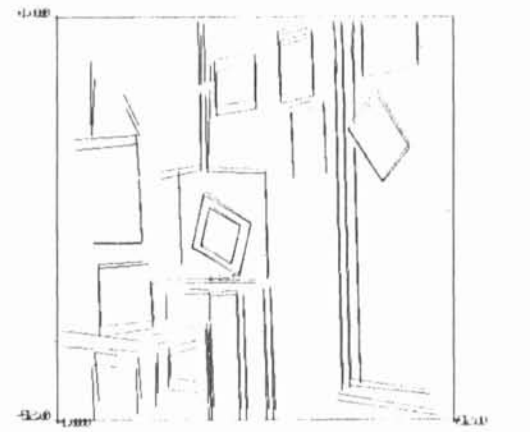

Figure 5: Reprojection of frame 2 of the reconstructed scene using the initial estimate for the motion (translation=(0,0,0), axis of rotation=(0,0.1,0))

In order to reduce the effect of the latter we proceed as follows : we reconstruct the scene using the initial estimate of the motion (in our case translation = $[0, 0, 0]^T$ and axis of rotation = $[0, 0.1, 0]^T$) ; we then reproject the scene on the second image plane and we select only the matched segments that verify the following conditions :

1. they differ, in length, by not more than 5%,
2. they differ, in orientation, by not more than 5 degrees.

The minimum of the criterion is then computed for the selected segments, giving a new estimate for the rotation and the translation. This estimate is used on all the original segments to make a new selection with the additional conditions :

3. the distance of the extremities of the segments must be less than 3 pixels,

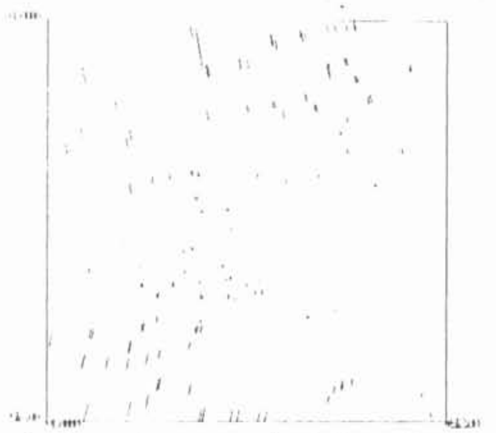

Figure 6: Error vectors corresponding to the previous reprojection.

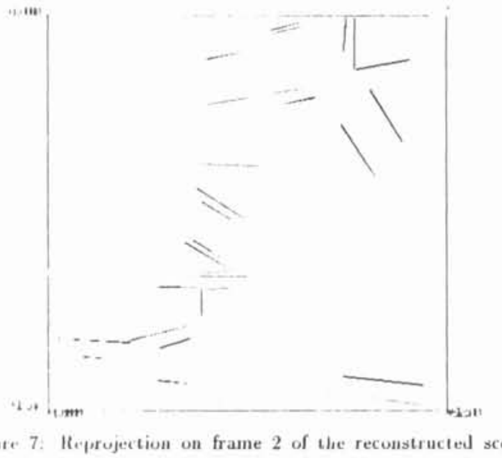

Figure 7: Reprojection on frame 2 of the reconstructed scene using the computed motion on the selected segments after the first iteration.

4. they differ, in orientation, by not more than 2 degrees.

The minimum of the criterion is computed on these segments and gives the final estimate of the motion. This estimate is then used to remove the segments that do not verify condition 3, which are considered as false correspondences.

10 Experimental Results of the Motion Estimation

From the test that we have performed on a number of sequences of real images we can make the following two considerations :

- criteria $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(4)}$ are unreliable,
- criterion $\mathcal{C}^{(3)}$ gives better results than $\mathcal{C}^{(2)}$ but its implementation is slower : every reconstructed point has to be computed as solution of a minimization problem and the minimum of $\mathcal{C}^{(3)}$ is found using finite difference methods, since we could not derive the criterion.

The results show on figure 5 to 8 were computed using criterion [illegible], while the reconstruction implemented for the reprojection was [illegible].

An indication of the quality of the results is shown in figure 8 : we cannot see the difference between the reconstructed and reprojected scene and the original segments.

Another indication of quality of the results is the computed angle of rotation : from frame 1 to 3 the computed angle was 1.91 and 2.14 while the robot moved of about 2 degrees.

Figure 8: Reprojection on frame 2 of the reconstructed scene using the final computed motion (2 iterations) on all the segments of the scene.

Similar results were obtained on all the other sequences used for test.

The cpu time on a Sun workstation was of about 3 seconds for 100 selected segments.

11 CONCLUSION

We have proposed a technique for recovering the 3D motion and 3D structure (up to a scale factor) of a scene which presents the following caracteristics :

- **reliable** : the reconstructed and reprojected scene cannot be differentiated from the original images,
- **fast** : the overall process of tracking, motion and structure computation takes a few seconds on a Sun workstation,
- **of simple material implementation** : only a single camera and hence one calibration is required and no epipolar geometry is needed.

These caracteristics render the technique very attractive and have been illustrated by several experiments that have been carried out on noisy synthetic data and real scenes obtained from a mobile robot.

References

[1] R. DERICHE. Optimal edge detection using recursive filtering. In *Proc. First International Conference on Computer Vision*, London, June 8-12 1987.

[2] J.-Q. FANG and T.S. HUANG. Some experiments on estimating the 3-d motion parameters of a rigid body from two consecutive image frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):547-554, 1984.

[3] D.B. GENNERY. Stereo-camera calibration. In *Proc. Image Understanding Workshop*, pages 101-108, November 1979.

[4] H.C. LONGUET HIGGINS. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133-138, September 1981.

[5] G. TOSCANI and O.D. FAUGERAS. Structure from motion using the reconstruction & reprojection technique. In *Proc. IEEE Computer Society Workshop on Computer Vision*, November 30-December 2 1987. Miami, Florida, USA.

[6] H.P. TRIVEDI. Estimation of stereo and motion parameters using a variational principle. *Image and Vision Computing*, 5(2), May 1987.

[7] R. TSAI, T.S. HUANG, and W.L. ZHU. Estimating three-dimensional motion parameters of a rigid planar patch, II: singular value decomposition. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 30(4):525-534, August 1982.