Visual navigation using a single camera

Jean-Yves Bouguet† and Pietro Perona†‡

† California Institute of Technology, 116-81, Pasadena, CA 91125, USA
‡ Università di Padova, Italy
{bouguetj,perona}@systems.caltech.edu

Abstract

We assess the usefulness of monocular recursive motion estimation techniques for vehicle navigation in the absence of a model for the environment. For this purpose we extend a recently proposed recursive motion estimator, the Essential filter, to handle scale estimation. We examine experimentally the accuracy with which the motion and position of the vehicle may be computed on an 8000 frames indoors sequence. The issues of sampling time frequency and number of necessary features in the environment are addressed systematically.

1 Introduction

Estimating motion and trajectory of a vehicle from visual input has been for few years a topic of great interest in the computation vision community. The main applications are assisted and autonomous navigation. The most successful system so far is due to Dickmanns [5, 6, 7] and it is based on strong models of the environment. There are situations (outdoors navigation, navigation in towns) where a model is either outright impossible or impractical to build, therefore techniques for navigation have to be developed that can work well in unstructured environments. In the last few years a number of schemes have come out in the computer vision literature for estimating recursively motion and structure [13, 2, 12]. One could think of using these general structure and motion algorithms for vehicle navigation. However, no large and systematic experiments have been performed so far to assess quantitatively how good motion, position and scale recovery would be in a realistic scenario. This is the purpose of this paper. We present here experiments on a long sequence that represents well typical indoor and city navigation.

1.1 Presentation of the general Scheme

As shown in Figure 1.1, the general scheme applied for full recursive rigid motion recovery can be decomposed into 4 successive stages. The first stage consists of automatically extracting from the images some clearly distinguishable feature points, and then tracking them from frame to frame. This gives the *image flow* information. The second stage computes the motion parameters which include a scale factor ambiguity (from the norm of translation) from this flow. This ambiguity can then be resolved by using scenery information (also called 3-D structure). The third stage is the actual structure reconstruction and the fourth is the scale factor propagation. This paper does not address the issue of image flow computation, since a number of schemes for point feature extraction and tracking already exist. We use a method which is a multi-scale version of the algorithm developed by Lucas and Kanade [11]. We thus consider the image flow as input data (our observations) for the whole system. Section 2 derives the complete estimator, and section 3 describes the experimental results.

2 Dynamical model

2.1 Notations

We take a point feature based approach for the image plane flow as well as for the 3-D structure composing the environment.

The 3-D Structure: We denote by $\mathbf{X}(t)$ the set of the $n(t)$ points $\mathbf{X}^{(i)}(t) = [\ X_i(t)\ \ Y_i(t)\ \ Z_i(t)\ ]^T$ composing the visible 3-D structure at time t. The coordinates are expressed in the camera coordinate system.

The observation: The structure is projected onto the image plane through a perspective projection. If f is the camera focal length, the set of projected points is given by $\mathbf{x}^{(i)}(t) = \prod\left(\mathbf{X}^{(i)}(t)\right) = \frac{f\mathbf{X}^{(i)}(t)}{Z_i(t)}$, letting $\prod$ be the perspective projection operator.

The real observation points are then identical to the $\mathbf{x}^{(i)}(t)$ up to additive measurement noises $\mathbf{n}_x^{(i)}(t)$ that we will assumed white, zero-mean and Gaussian: $\tilde{\mathbf{x}}^{(i)}(t) = \mathbf{x}^{(i)}(t) + \mathbf{n}_x^{(i)}(t)$.

The rigid motion: The rigid motion between times t and $t+1$ is defined by the translation vector $T(t)$ and the rotation matrix $R(t)$ such that $\forall i = 1, \ldots, n(t)$:

$$\mathbf{X}^{(i)}(t+1) = R(t)\mathbf{X}^{(i)}(t) + T(t) \qquad (1)$$

where $R(t)$ is defined from the rotation vector $\Omega(t)$ by $R(t) = e^{\Omega(t)\wedge}$. Finally, define the scale factor $s(t)$ to be the norm of $T(t)$ and the unit length translation $T_u(t) = \frac{T(t)}{s(t)}$.

0-8186-7042-8/95 $4.00

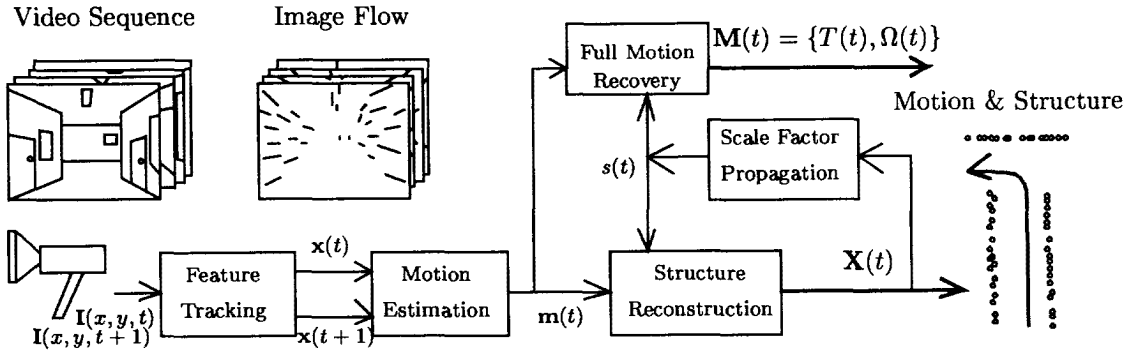

Figure 1: *The Complete system for Full Motion and Structure estimation from a sequence images*

2.2 Kinematic model

If no constraint is imposed on the motion, it can be described by 5 parameters embedded in the motion state variable $\mathbf{m}(t)$ (no scale included). These parameters can be taken as the association of 2 variables for defining the direction of the translation $\theta_{az}(t)$ and $\theta_{el}(t)$ (azimuth and elevation) and 3 variables for the rotation $\Omega(t)$ (its coordinates):

$$\mathbf{m}(t) = [\ \theta_{az}(t) \quad \theta_{el}(t) \quad \Omega^T(t)\ ]^T \qquad (2)$$

For road vehicles, which is one of the most important applications, it makes sense to study reduced models. It is then possible to reduce the state down to 2 variables. This is not studied in this paper.

2.3 A model for full motion $\mathbf{M}(t)$

We denote by $\mathbf{M}(t)$ the full motion state variable ($T(t)$ and $\Omega(t)$) calculated from the motion $\mathbf{m}(t)$ and the scale $s(t)$. We will denote:

$$\mathbf{M}(t) = s(t) \otimes \mathbf{m}_p(t) = [\ T^T(t) \quad \Omega^T(t)\ ]^T \qquad (3)$$

Since we want to estimate full motion, it is necessary to propagate not only the motion state $\mathbf{m}(t)$ but also the scale $s(t)$. However, only the 3-D structure carries the scale information. It is therefore necessary to perform the structure estimation in addition to the motion.

The motion state $\mathbf{m}(t)$: Assuming that we have available some dynamical model for the motion , then we can write :

$$\begin{cases} \mathbf{m}(t+1) &= F(\mathbf{m}(t)) + \mathbf{n}_m(t) \\ 0 &= \mathbf{x}^T(t+1)\mathbf{Q}(\mathbf{m}(t))\mathbf{x}(t) \\ \tilde{\mathbf{x}}(t) &= \mathbf{x}(t) + \mathbf{n}_x(t) \end{cases} \qquad (4)$$

where F is the prediction function describing the dynamics of the system, $\mathbf{n}_m(t)$ the white noise (with covariance matrix Q_m) and the second equation is the well known coplanarity constraint [10] with $\mathbf{Q} \doteq (T_u \wedge) R$ the essential matrix. Note that for clarity, we drop the superscripts $^{(i)}$ for the points $\mathbf{x}^{(i)}(t)$. In cases where the motion is smooth (or simply when no other model is available), one can take $F = id$, and $\mathbf{n}_m(t)$ is then a *random walk.*

Applying the coplanarity constraint on the points $\tilde{\mathbf{x}}(t)$, measured on the image plane, one get the following model:

$$\begin{cases} \mathbf{m}(t+1) &= F(\mathbf{m}(t)) + \mathbf{n}_m(t) \\ \tilde{\mathbf{n}}(t) &= \tilde{\mathbf{x}}^T(t+1)\mathbf{Q}(\mathbf{m}(t))\tilde{\mathbf{x}}(t) \end{cases} \qquad (5)$$

where $\tilde{\mathbf{n}}(t)$ is the induced residual noise whose second order statistic $R_{\tilde{n}}(t)$ can be characterized in terms of the variance R_n of the measurement error $\mathbf{n}_x(t)$. It is also called *pseudo-innovation vector.*

The state of the model (5) is defined on a linear space and can now be estimated using a variation of the Extended Kalman Filter for implicit measurements constraints, which is derived in [13]. The equations of the estimator based on Kalman filtering theory [9, 4, 8] can be derived from the model system (5). This filter provides an estimate for the motion $\hat{\mathbf{m}}(t)$ and the covariance matrix of the error in estimation $P_m(t)$.

For fast convergence reasons, the implemented scheme includes a first motion estimation using Longuet-Higgins 2-frames algorithm [10].

The structure $\mathbf{X}(t)$: In order to estimate motion and position, the scale factor needs to be propagated across time and therefore we must estimate recursively the 3-D structure of the environment as the vehicle moves. This is done through a recursive structure from motion Extended Kalman filter.This a non linear estimator due to the non linear observation function $\prod$ (perspective projection). The equations for this estimator can be directly derived from the structure model equations [12]. This gives after update, some estimates $\hat{\mathbf{X}}^{(i)}(t)$ of the positions

of the points and provides covariance matrices of errors on these estimations $P_i(t)$.

The initial predictions of the filter is done by triangulation using the projections of the points on two first appearance frames. This technique has been largely addressed in previous publications [2]. Define the operator Δ to be the triangulation function returning the 3-D position of one or several points $\mathbf{X}(t)$, from their perspective projections $\mathbf{x}(t)$ and $\mathbf{x}(t+1)$ onto 2 successive image planes: $\mathbf{X}(t) = \Delta(\mathbf{x}(t), \mathbf{x}(t+1), \mathbf{M}(t))$, where $\mathbf{M}(t)$ is the full rigid motion of the camera from time t to time $t+1$.

Such a method for estimating the 3-D structure is known to be very ill-conditioned with respect to observation noise (on the image plane) and the errors on the motion parameters. 3D positions are especially poorly estimated for points far away from the observer (with a large relative depth, or a large depth over focal length ratio), or more generally with small motion parallax. Then, at the initialization of each point, the covariance matrix of the error of estimation P_i can be either set to an arbitrary large value (taking into consideration the large uncertainty on the initialization of $\mathbf{X}^{(i)}$), or computed by propagation of the variance errors through Δ from the covariance error matrices on the observations $\mathbf{x}^{(i)}$.

The scale propagation $s(t)$ **:** Going back to the structure estimator, the scale $s(t)$ is only required to perform the prediction step. It is therefore possible (and required) to estimate scale immediately after the structure update.

At time t, given some estimates of the structure $\hat{\mathbf{X}}(t)$ and the motion $\hat{\mathbf{m}}(t)$, how can we estimate the scale factor $s(t)$?

Let $\mathbf{X}_u(t)$ be the 3-D structure obtained by triangulation from the projective points $(\mathbf{x}(t), \mathbf{x}(t+1))$ and the full rigid motion $\mathbf{M}_u(t) = 1 \otimes \mathbf{m}(t)$ computed from $\mathbf{m}(t)$ with *unit norm translation*: $\mathbf{X}_u(t) = \Delta(\mathbf{x}(t), \mathbf{x}(t+1), \mathbf{M}_u(t))$. Then:

$$\mathbf{X}(t) = s(t)\mathbf{X}_u(t) \tag{6}$$

which intuitively means that the whole structure gets scaled by $s(t)$. Now, in the real data case, we can only get some noisy estimate of $\mathbf{X}_u(t)$, $\hat{\mathbf{X}}_u(t)$:

$$\hat{\mathbf{X}}_u(t) = \Delta\left(\tilde{\mathbf{x}}(t), \tilde{\mathbf{x}}(t+1), \hat{\mathbf{M}}_u(t)\right) \tag{7}$$

Where $\hat{\mathbf{M}}_u(t) = 1 \otimes \hat{\mathbf{m}}(t)$. This can be written:

$$\left\{ \begin{array}{rcl} \hat{\mathbf{X}}_u(t) & = & \mathbf{X}_u(t) + \tilde{\mathbf{n}}_{Xu}(t) \\ \hat{\mathbf{X}}(t) & = & \mathbf{X}(t) + \mathbf{n}_X(t) \end{array} \right. \tag{8}$$

where $\tilde{\mathbf{n}}_{Xu}(t)$ is some model noise process assumed to be white, with zero-mean, a second order statistics which can be derived from the error covariance matrices attached to the observations $\tilde{\mathbf{x}}(t)$ and the motion $\mathbf{m}(t)$. The covariance matrix $P_u(t)$ attached to $\tilde{\mathbf{n}}_{Xu}(t)$, is a good estimator of the reliability of the 3-D positions of the points by triangulation. Considering each point individually, we have:

$$P_u(t) = diag\left(P_u^{(1)}(t), P_u^{(1)}(t), \ldots, P_u^{(N)}(t)\right) \tag{9}$$

where $P_u^{(i)}(t)$ are the 3×3 covariance matrices attached to the error in estimating the 3-D positions of the points $\mathbf{X}_u^{(i)}$ (for $i = 1, \ldots, n(t)$).

It follows that the "larger" the matrix $P_u^{(i)}(t)$ the more inaccurate the point (to be precise, one should consider the maximum eigenvalues of these matrices). Triangulation fails when relative depth is large, or when the motion flow is small compared to the observation noise. Points with small motion parallax will have very large associated covariance matrix $P_u^{(i)}(t)$ (especially on the depth component). We can then make use of this in the computation of the scale. From (6), one can derive the following estimator of scale:

$$s_m(t) = \arg\min_s \sum_{i=1}^{N} \omega_i(t) \|\hat{\mathbf{X}}^{(i)}(t) - s\hat{\mathbf{X}}_u^{(i)}(t)\|^2 \tag{10}$$

with $s_m(t)$ the computed value of $s(t)$, and $\{\omega_i(t)\}_{i=1}^N$ some weighting coefficients in order to take into consideration the *relative reliability* of the points in the computation. Of course, these coefficients not only depend on the covariance matrices $P_u^{(i)}(t)$ but also on the matrices $P_i(t)$, the covariance matrices of the error of estimation coming out of the recursive structure estimator. From (10), it is possible to derive an explicit expression for $s_m(t)$:

$$s_m(t) = \frac{\sum_{i=1}^{N} \omega_i(t) \left(\hat{\mathbf{X}}_u^{(i)}(t)\right)^T \hat{\mathbf{X}}^{(i)}(t)}{\sum_{i=1}^{N} \omega_i(t) \|\hat{\mathbf{X}}_u^{(i)}(t)\|^2} \tag{11}$$

Then, an analytical expression for the weights $\omega_i(t)$ can be derived, for which the variance of the scale measurement error is minimized (in terms of $P_i(t)$ and $P_u^{(i)}(t)$). For simplicity, we retain the depths (Z components) in equations (10) and (11). This leads to the following simplified measurement expression for scale:

$$\left\{ \begin{array}{rcl} s_m(t) & = & \frac{\sum_{i=1}^{N} \omega_i(t) \hat{Z}_u^{(i)}(t) \hat{Z}^{(i)}(t)}{\sum_{i=1}^{N} \omega_i(t) \left(\hat{Z}_u^{(i)}(t)\right)^2} \\ \omega_i(t) & = & \left(\sigma_{iu}^2(t) + \sigma_i^2(t)\right)^{-1} \end{array} \right. \tag{12}$$

where $\sigma_{iu}(t)$ and $\sigma_i(t)$ are the standard deviations of the estimation error of $Z_{iu}(t)$ and $Z_i(t)$ respectively. Now, consider (12) as a *measurement*

equation for the scale. We then have available at time t some measure of the scale $s_m(t)$ whose variance $\sigma_s^2(t)$ is induced from the covariance matrices $P_i(t)$ and $P_u^{(i)}(t)$. Assuming then smoothness on the velocity of the vehicle:

$$\begin{cases} s(t+1) &= s(t) + n_s(t) \\ s_m(t) &= \frac{\sum_{i=1}^{N} \omega_i(t) \hat{Z}_u^{(i)}(t) \hat{Z}^{(i)}(t)}{\sum_{i=1}^{N} \omega_i(t) \left(\hat{Z}_u^{(i)}(t)\right)^2} \end{cases} \quad (13)$$

Where the first equation in (13) describes the dynamics of the scale (state equation) and the second one the observation (which is, in this case the measurement of the scale itself). We can then write the Kalman filter for that model [9, 4, 8]. Notice that, with such a simple dynamics, the filter will be equivalent to a *smoother*.

Regrouping both motion and scale dynamical models (5) and (13), one can derive a model *including dynamics on the full motion* $\mathbf{M}(t)$:

$$\begin{cases} \mathbf{M}(t+1) &= F(\mathbf{M}(t)) + \mathbf{n}_M(t) \\ \mathbf{M}(t) &= s(t) \otimes \mathbf{m}(t) \\ \tilde{\mathbf{n}}(t) &= \tilde{\mathbf{x}}^T(t+1) \mathbf{Q}(\mathbf{m}(t)) \tilde{\mathbf{x}}(t) \\ s_m(t) &= \frac{\sum_{i=1}^{N} \omega_i(t) \hat{Z}_u^{(i)}(t) \hat{Z}^{(i)}(t)}{\sum_{i=1}^{N} \omega_i(t) \left(\hat{Z}_u^{(i)}(t)\right)^2} \end{cases} \quad (14)$$

where the 2 last equations correspond to the *implicit measurement for motion* and *explicit measurement for scale factor*.

Note that F describes here the dynamical model on the *full motion state*. This constitutes an improvement since it may happen that the dynamic of the scale factor $s(t)$ (or velocity, since it is the norm of translation) and the other motion parameters $\mathbf{m}(t)$ (such as rotation and direction of translation), are coupled. If no dynamic is known, then F can be assumed to be the identity, which is equivalent to assuming smoothness in the motion by taking first order additive random walks on $\mathbf{m}(t)$ and $s(t)$ in the state equations. It is actually this model that we have applied in our experiments. Since after initialization, there are no measurements of the scale available, we expect a drift in the estimate of the scale.

One could also think of propagating scale information without explicitly reconstructing the 3D structure by writing a constraint on the scale directly using observation on 3 consecutive frames. This would give an direct estimate for scale as a function of the projective points coordinates on these 3 frames. This alternative method is still under investigation.

3 Experimental results

This experiment consists of a sequence taken with a CCD video camera mounted onto a cart moving along a closed corridor. The cart was simply pulled by two operators while another operator was sitting on it. The camera was attached to a small tripod on the cart. The camera was such that it was pointing approximately in the direction of the motion. The objective of the experiment was to reproduce city driving as well as indoor navigation. Since the velocity of the motion was about 4 km/h, and the corridor was 2 meters wide, we would have similar size of observation from a vehicle driving at 60 km/h on a 30 meters wide road (from building to building).

Duration and Image characteristics: Figures 2(a) and 2(b) show the first image of the sequence and the path followed by the cart throughout the experiment (the ground truth). The complete path is approximately 145 meters long, and composed of four straight segments, four 90 degrees turns and four little "S turns". The output image rate of the camera is 60 interlaced frames a second. The whole 2 minutes and 11 seconds long sequence consists of 7871 images (even and odd fields).

Each image is 640×240 pixels. A pre-calibration of the camera gave us values for its focal length $f = 527 \pm 6$ pixels, the position of the center of projection on the image plane $c = [348 \pm 2, 211 \pm 6]$ pixels (in the initial format 640×480), and the radial distortion factor $k = -0.121 \pm 0.003$. The images were digitized using a JPEG compression board. A limited number of artifacts due to JPEG are present in the digitized images.

Ground truth recovery: Some 60 makers were put on the floor throughout the path to recover the ground truth position of the cart at any time instant (Fig. 2a). We recovered ground truth knowing the camera field of view, the approximate height of the camera with respect to the floor and the numbers of the frames where the markers were visible. This estimation has been done with an accuracy of about 10 frames (error of about 15 centimeters). The camera position at each frame has been then deduced by interpolation of the trajectory between the markers.

We can then take as an obvious criterion of goodness, the distance between the reconstructed path and the "real path" obtained from ground truth.

From the ground truth trajectory, one can get some estimation of the norm of translation composing the motion between each time instant (or scale factor). Since the ground truth path has been obtained using interpolation, this computation can only give an estimation of the real scale. We refer to this as the "ground truth for scale".

We placed on the walls some white sheets with black contours. This was to make available more feature points to be selected and tracked across time. The actual average number of detected features per frame was approximately 200 (only true for straight segments, at the turns, it went down to about 100). This allowed us to test the robustness of the scheme upon the number of features used for motion estimation. Results concerning these tests

are reported here. Note that during the turns, the features were visible in average for 50 frames.

Computational parameters: The whole sequence has been segmented into four 2000 frames long sub-sequences (constituting the four sub-experiments), each of them corresponding to a different turn (Fig. 2(b)). We ran the motion estimator over these sequences.

We used for feature tracking, a scheme which is a multi-scale version of the algorithm developed by Lucas and Kanade [11]. This scheme selects and tracks point features on the images with an accuracy of about 0.2 to 0.5 pixel. This is the performance of most common existing feature tracking methods [3]. We added to the present scheme a segmentation stage to reject outliers and false features [13]. New selections are done on the images whenever the number of available point features drops below some threshold.

We applied the first order random walk model for the motion and the scale estimations. We tried the algorithm over very different number of feature points for motion estimation. We found that $N = 40$ is a good trade off. We did not tune very accurately the filter parameters for motion. We used one set of tuning parameters that seemed to fit well to the experiment: for the model, we took a diagonal covariance matrix Q_m, with identical variance of $k^2 * 10^{-4} rad^2$ on each component (standard deviation of 5 degrees for a $k = 10$ frames baseline). We set the measurement noise to 1 pixel. The vehicle was moving with an average speed of $4km/h$, which means $18cm$ every 10 frames. Assuming that it could not accelerate by more than $1km/h/s$, one can deduce a value for the variance on the scale $Q_s \approx 7 * 10^{-6} m^2$ (with k=10). In our experiments we found very important to include the smooth model to the scale, otherwise the drifts were too large. We encountered using this model drifts of scale of few percents over 2000 frames. By suppressing the dynamics on the scale, or decreasing its model variance, we experimented much larger drifts: For example, by setting $Q_s = 0.002 m^2$, the scale drifts is "roughly" 50% every 1000 frames.

An important characteristic of the implemented version of the scheme is the time baseline to perform motion and structure estimation. This idea is based on the fact that motion estimation is very ill-conditioned when the image flow is small compared to the image noise [1]. Therefore, we chose to perform it not based on the flow computed from frame t to frame $t + 1$, but on flow from frame t to $t + k$, where k is some integer defining the time baseline ($k > 1$). Actually, we recursively estimated motion at *each time instant t* using a *sliding baseline window*. One may think of taking any arbitrary large value for the baseline duration k, however, as k increases, the smooth model for motion is less and less valid. We found $k = 10$ to be a good trade off for that experiment (which means a baseline rate of 6 frames a second). For general type of motions, and environments one should check if there are means of finding the best time baseline.

4 Results

To "measure" the performance of the algorithm, we extract from the reconstructions two quantities: the computed angle of turn (ideally 90 degrees), and the final computed vertical deviation (ideally 0 meter, since the motion was planar).

The **"4 turns sub-experiments"**: Fig. 3 shows the results in estimating the trajectory and the structure on each each individual turn. Notice the quality of the reconstructions.

The **"whole round-trip experiment"**: Fig. 5(a) is a top view of the complete reconstructed trajectory and corridor for the whole round-trip sequence. Fig. 5(b) are the estimated 5 DOF motion parameters, and Fig. 5(c) displays one image of the sequence with its attached point features and flow, and the current motion.

For these two figures, we took the same experimental conditions: a time baseline $k = 10$ and the number points used for motion estimation $N = 40$.

In addition to the quality of the trajectory, one can remark how accurately the different walls of the corridor are reconstructed (approximate reconstruction thickness of 5 cm).

We tested the algorithm on the first turn with different values for the time baseline k and for the number of features used for motion estimation N.

The tables 1 and 2 present the performances for of the algorithm on the first turn under different conditions of time baseline k and number of features N used for estimating the motion. Note that only $N = 8$ point features are sufficient to provide some relatively acceptable motion estimates (see Fig. 4(a)). We can also notice that the baselines $k = 5, 10$ and 20 give very good and similar performances. Fig. 4(b) shows the degradation of the estimate as we suppress the time baseline.

k	angle of turn (deg)	vert. deviation (m)
1	52.2	1.2
5	92.4	0.78
10	90.9	0.89
20	93.5	0.93
30	91.3	0.98
50	96.6	0.75

Table 1: *Influence of the time baselines k (with $N = 40$, first turn).*

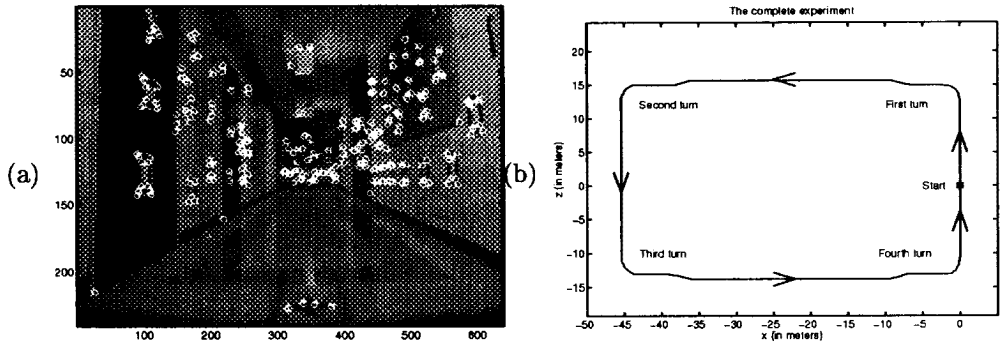

Figure 2: *The **"cart experiment"**: (left) the first frame of the sequence with the point features highlighted. (right) The whole path followed by the cart in the complete experiment. Note the positions of the 4 turns constituting the 4 sub-experiments.*

N	angle of turn (deg)	vert. deviation (m)
< 8	breaks down	breaks down
8	79.3	0.25
10	100.3	4.7
20	91.2	1.79
40	90.9	0.89
100	91.3	0.72
200	91.7	0.4

Table 2: *Influence of the number of points N (with $k = 10$, first turn). The computed angles of turns are in degrees, and the final vertical deviations in meters. The retained points are the N ones with the smallest the innovations (5).*

5 Conclusion and Future work

We showed that it is possible to get a sufficiently accurate estimation of the position in real experimental conditions with minimal equipment required (a simple video camera mounted on the vehicle without any visual control). For road navigation applications, we would like to be able to estimate accurately and robustly any particular type of motion. We developed a scheme which showed some remarkable results for estimating motion in straight segments as well as in the 90 degrees turns. We believe that including a complete dynamical model for the vehicle with a kinematic exploiting the planarity of the motion, would improve the scheme by making it more robust.

Acknowledgments

This work is supported in part by the California Institute of Technology; the Office of Naval Research grant ONR NOOO14-93-1-0990; an NSF National Young Investigator Award; the Center for Neuromorphic Systems Engineering as a part of the National Science Foundation Engineering Research Center Program; and by the California Trade and Commerce Agency, Office of Strategic Technology. We wish to thank all the persons that helped us throughout this work, in especially Stefano Soatto for the very useful discussions, and Luis Goncalves, Mario E. Munich and Enrico Ursella for their great help in setting up the experiment.

References

[1] G. Adiv. Inherent ambiguities in recovering 3-d motion and structure from noisy flow field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:477–489, 1989.

[2] A. Azarbayejani, B. Horowitz, and A. Pentland. Recursive estimation of structure and motion using relative orientation constraints. *Proc. CVPR*, New York, 1993.

[3] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. RPL-TR 9107, Queen's University Kingston, Ontario, Robotics and perception laboratory, 1992. Also in Proc. CVPR 1992, pp 236-242.

[4] R.S. Bucy. Non-linear filtering theory. *IEEE Trans. A.C. AC-10, 198*, 1965.

[5] E. D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[6] E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:199–213, 1992.

[7] B. D. Mysliwetz E. D. Dickmanns and T. Christians. An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. *IEEE Trans. on Systems, Man, and Cybernetics*, 20:1273–1284, 1990.

[8] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.

[9] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. of the ASME-Journal of basic engineering.*, 35-45, 1960.

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

[11] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th Int. Joinnt Conf. on Art. Intell.*, 1981.

[12] J. Oliensis and J. Inigo-Thomas. Recursive multi-frame structure from motion incorporating motion error. *Proc. DARPA Image Understanding Workshop*, 1992.

[13] S. Soatto, R. Frezza, and P. Perona. Motion estimation on the essential manifold. *In "Computer Vision ECCV 94, Lecture Notes in Computer Sciences vol. 801", Springer Verlag*, May 1994.

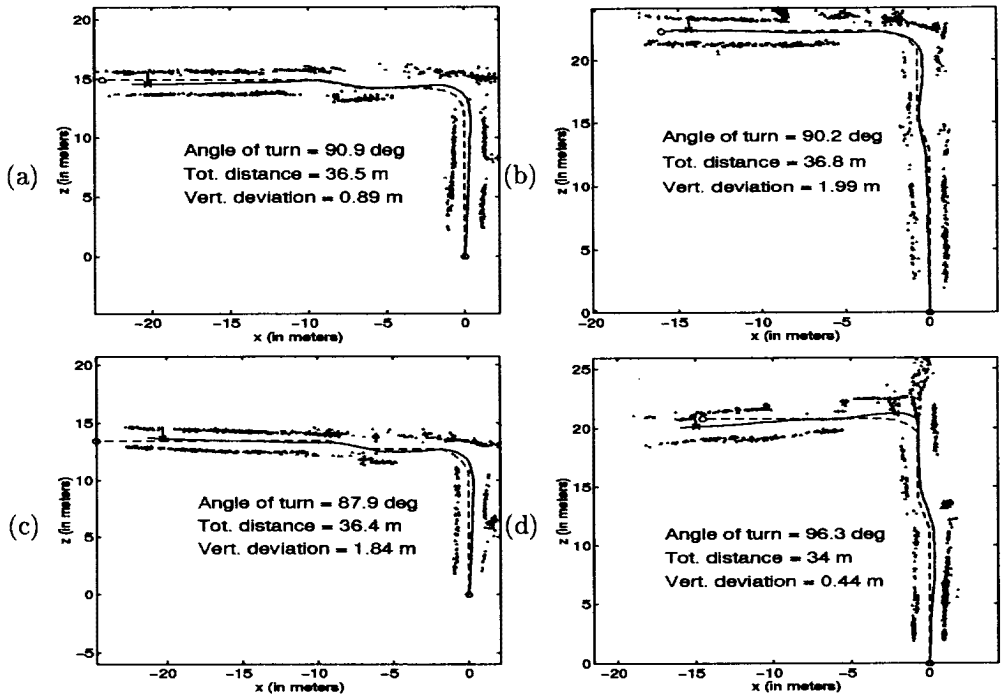

Figure 3: *The* ***"4 turns sub-experiments"****: These four figures are top views of the estimates (in trajectory and structure) on the four successive turns. The real and estimated trajectories are respectively showed in dotted and solid lines, the reconstructed structure (or corridor) is represented in dots. The errors on the turn angles are below 2 degrees for the three first turns and only 6 degrees for the fourth one, the paths have maximal vertical deviation of about 3cm/m, and almost all straight segments are very nicely reproduced. Notice that the last turn presents larger errors in reconstruction of the structure (especially right at the turn). The time baseline is* $k = 10$ *frames (a 6Hz frame rate), with* $N = 40$ *points for estimating motion.*

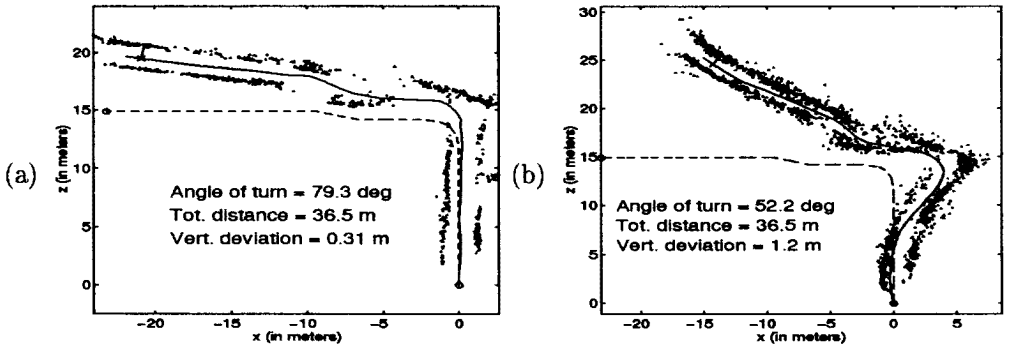

Figure 4: ***Reconstruction of the first turn with different N and k:*** *(a) A small number of features* $N = 8$ *(and a time baseline k=10). (b) Suppression of the time baseline (N=40,k=1).*

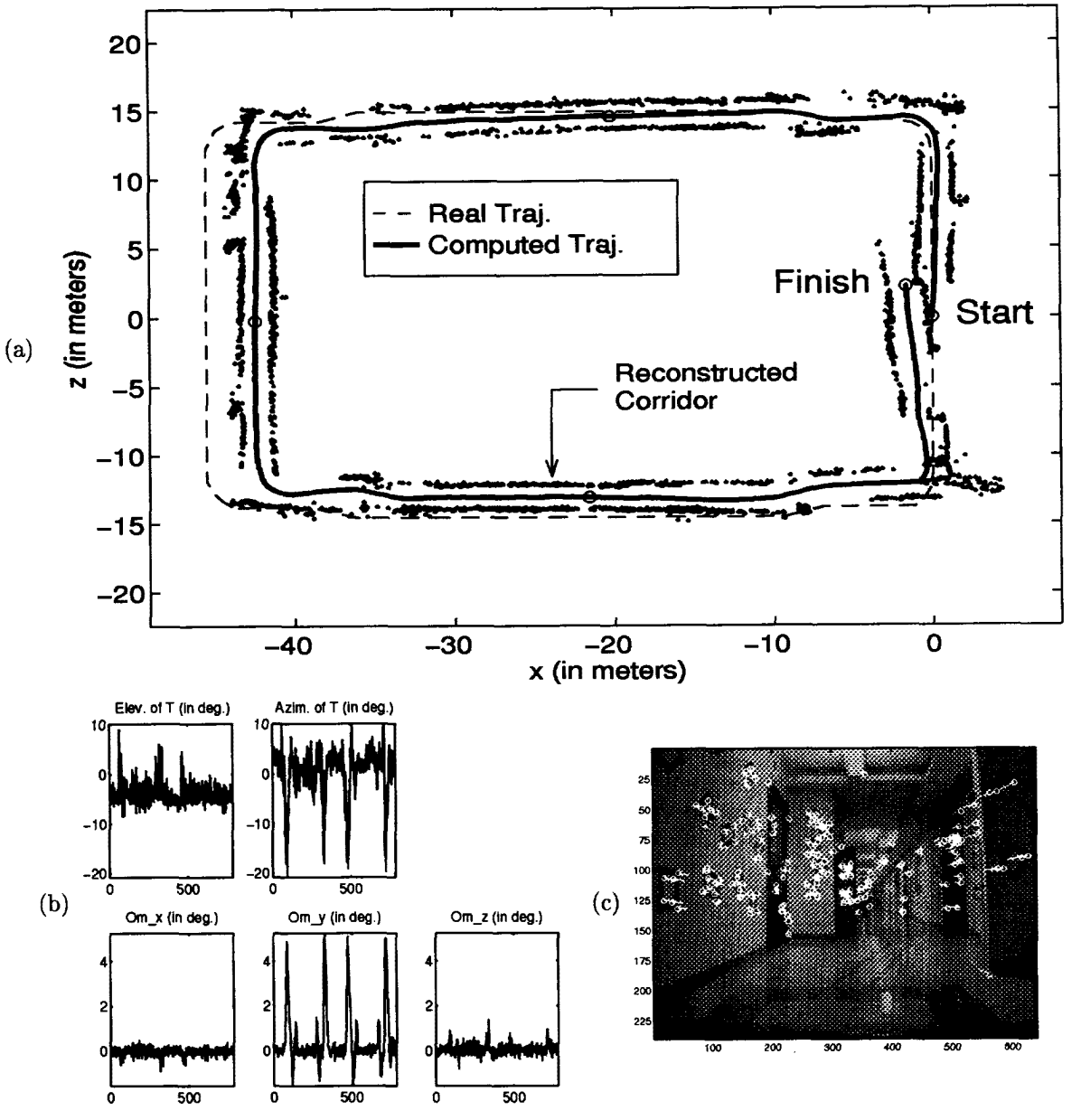

Figure 5: *The* ***"whole round-trip experiment"****: Figure (a) is a top view of the reconstructed trajectory and corridor. The dotted lines represent the real trajectory and the solid lines the estimated one. The dots are the reconstructed positions of the features on the walls of estimated thickness of 5 cm. Note that most of the errors are concentrated at the turns. These errors are at most 10 degrees over each turn. All straight segments are well estimated (with the "S turns"), with an error of at most 10 cm/m. The overall vertical deviation is* 5.16 *m, which represents* 3.6% *of the total traveled distance (145 meters). Figure (b) shows the 5 estimated state motion variables throughout the complete round-trip. Note that the elevation component is roughly constant and equal to* -5 *degrees, and at the turns the rotation vector is mostly along the vertical direction (y axis), as expected from the planar nature of the motion. Figure (c) shows one image with the features and the flow. The time baseline is* $k = 10$ *frames (a* 6*Hz rate), with* $N = 40$ *points for estimating motion.*