International Journal of Computer Vision 22(3), 235–259 (1997)

Recursive 3-D Visual Motion Estimation Using Subspace Constraints

STEFANO SOATTO*
Control and Dynamical Systems, California Institute of Technology 136-93, Pasadena, CA 91125
soatto@caltech.edu

PIETRO PERONA
Electrical Engineering and Computation and Neural Systems, California Institute of Technology 136-93, Pasadena, CA 91125; and Dipartimento di Elettronica ed Informatica, Università di Padova, Padova, Italy

Received August 30, 1994; Revised March 9, 1995; Accepted October 23, 1995

Abstract. The 3-D motion of a camera within a static environment produces a sequence of time-varying images that can be used for reconstructing the relative motion between the scene and the viewer. The problem of reconstructing rigid motion from a sequence of perspective images may be characterized as the estimation of the state of a nonlinear dynamical system, which is defined by the rigidity constraint and the perspective measurement map. The time-derivative of the measured output of such a system, which is called the "2-D motion field" and is approximated by the "optical flow", is bilinear in the motion parameters, and may be used to specify a subspace constraint on the direction of heading independent of rotation and depth, and a pseudo-measurement for the rotational velocity as a function of the estimated heading. The subspace constraint may be viewed as an implicit dynamical model with parameters on a differentiable manifold, and the visual motion estimation problem may be cast in a system-theoretic framework as the identification of such an implicit model. We use techniques which pertain to nonlinear estimation and identification theory to recursively estimate 3-D rigid motion from a sequence of images independent of the structure of the scene. Such independence from scene-structure allows us to deal with a variable number of visible feature-points and occlusions in a principled way. The further decoupling of the direction of heading from the rotational velocity generates a filter with a state that belongs to a two-dimensional and highly constrained state-space. As a result, the filter exhibits robustness properties which are highlighted in a series of experiments on real and noisy synthetic image sequences. While the position of feature-points is not part of the state of the model, the innovation process of the filter describes how each feature is compatible with a rigid motion interpretation, which allows us to test for outliers and makes the filter robust with respect to errors in the feature tracking/optical flow, reflections, T-junctions. Once motion has been estimated, the 3-D structure of the scene follows easily. By releasing the constraint that the visible points lie in front of the viewer, one may explain some psychophysical effects on the nonrigid percept of rigidly moving objects.

Keywords: dynamic vision, recursive rigid motion estimation, nonlinear identification, implicit Extended Kalman Filter

1. Introduction

When a camera moves within a static environment, the stream of images coming out of the sensor contains

*Corresponding address: Stefano Soatto, Division of Applied Sciences, Harvard University, 29 Oxford Street, Cambridge, MA 02138. Email: soatto@hrl.harvard.edu

enough information for reconstructing the relative motion between the camera and the scene. "Visual motion estimation" is one of the oldest (Gibson et al., 1959; Helmholtz, 1910) and at the same time one of the most crucial and challenging problems in computer vision. Even in the simplest cases, when the scene is represented as a *rigid* set of feature-points in 3-D space viewed under *perspective* projection, most of the early algorithms based upon the analysis of two frames at a time are not robust enough to be employed in real-world situations. Multi-frame analysis may be performed either in "batch" or recursively. While batch techniques process the whole sequence at once and therefore are, in principle, more accurate, recursive methods have a number of desirable features: (a) they process information in an incremental and causal fashion, so that they can be employed for real-time closed-loop operations, (b) they allow to easily incorporate model information about motion, (c) require minimal memory storage and computational power, for at each time the past history is summarized by the present estimate, and only the current measurement is being processed.

In this paper we study the recursive estimation of rigid three-dimensional motion of a scene viewed from a sequence of monocular perspective images. Since our main interest is on real-time causal processing, we do not review batch techniques here. Recursive estimation techniques have started being applied to special instances of the visual motion estimation problem only in the last decade (Dickmanns, 1994; Gennery, 1982). A number of schemes exist for recursively estimating structure for known motion (Matthies et al., 1989), motion for known structure (Broida and Chellappa, 1986; Gennery, 1982, 1992) or both structure and motion simultaneously (see for instance (Adiv, 1985; Azarbayejani, 1993; Heel, 1990; Oliensis and Thomas, 1992; Young and Chellappa, 1990) and references therein).

We argue against simultaneous structure and motion estimation for three reasons: (a) complexity—including the structure of the scene into the state of the filter makes it computationally demanding and requires sophisticated heuristics for dealing with a variable number of visible point-features; (b) convergence problems—the schemes proposed so far have poor model-observability (see (Soatto, 1997) for a thorough discussion of this issue); (c) occlusions—having structure in the state allows integrating motion information only to the extent in which all features are visible. While in realistic sequences the life-time of each individual feature is typically very short (2 frames when optical flow is measured instead of feature tracking), it is indeed possible to integrate motion information using a changing set of features, as long as they move according to the same rigid motion.

The recursive estimation of motion alone is a relatively unexplored subject: to our knowledge, the only recursive 3-D motion estimation scheme that is independent of the structure of the scene is the so-called "essential filter" (Soatto et al., 1994; 1996).

We present a recursive motion estimator, which we call the "subspace filter", that is based upon the differential version of the epipolar constraint introduced by Longuet-Higgins (1981) along the lines proposed by Heeger and Jepson (1992). The main advantage consists in the fact that the exponential representation of motion allows us to "decouple" the estimator of the direction of heading from that of the rotational velocity, in the lines of Adiv (1985). We can therefore design two filters, one on a two-dimensional state-space and one on a three-dimensional one, which are significantly more constrained and therefore more robust than algorithms based upon Longuet-Higgins' coplanarity constraint, as we will show in the experimental section.

1.1. Organization of the Paper

We start by showing how the assumptions of rigidity and perspective projection *define* a nonlinear dynamical model that can be used for designing a filter that simultaneously estimates structure and motion (Section 2).

Although the model follows naturally from the definition of the problem, simultaneous structure and motion estimation is both problematic from the theoretical point of view, and impractical (Section 2.3). The discussion in Section 2.4 serves as a motivation for introducing, in Section 3, an alternative implicit constraint on the motion parameters, which is derived from the work of Heeger and Jepson (1992) and called the "subspace constraint".

The core of the paper starts with the observation that the subspace constraint may be viewed as an implicit dynamical system, rather than a nonlinear system of algebraic equations defined for a pair of images. In Section 4, we formulate the problem of estimating the direction of translation as the identification of an implicit dynamical model with parameters on a sphere. The identification task is then carried on using local techniques based upon the Implicit Extended Kalman Filter. The estimates of the rotational

velocity come as a byproduct using a simple linear Kalman filter derived from Section 3.2. Once motion has been estimated, the estimates can be fed, along with the variance of the motion estimation error, in any structure-from-motion module; alternatively, structure may be estimated independent of motion using essentially the same techniques employed for recovering the direction of translation.

The experimental Section 5 comprises a number of tests both on noisy synthetic image sequences and on real indoor and outdoor scenes, which highlight some of the main features of the algorithm, such as its robustness to measurement noise.

Some further issues, such as implementation, tuning, measurement validation and outlier rejection, are discussed in the experimental section. There we also show some experiments on the "rubbery percept" of rigid shapes when the "positive depth constraint" is not enforced.

2. Visual Motion Estimation from a Dynamic Model

Let a scene be described by the position of a set of N feature points in 3-D space. Suppose such points move *rigidly* relative to the viewer, while their *perspective projection* onto an ideal image-plane is measured up to white and zero-mean noise (see Fig. 1). In this section we will see how the rigidity constraint and the perspective measurements *define* a nonlinear dynamical system involving both structure (position of each point in 3-D) and motion (translational and rotational velocity).

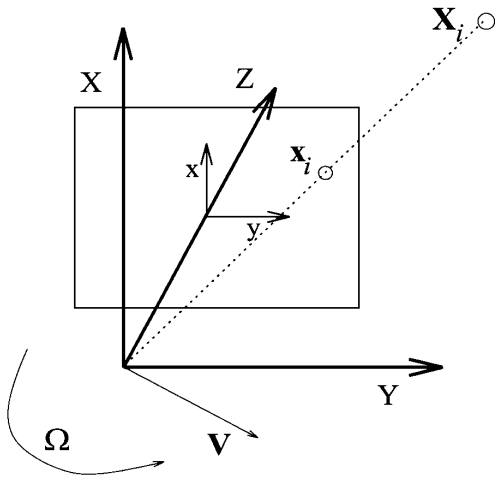

Figure 1. Notation: the viewer-centered reference frame.

2.1. Notation

Let us call $\mathbf{X}_i \doteq [X_i\ Y_i\ Z_i]^T \in \mathbb{R}^3$ the coordinates of the ith point in the viewer's reference frame, which is a right-handed frame with origin in the center of projection. The Z-axis points along the optical axis and the X and Y axes form a plane parallel to the imaging sensor. We call

$$\mathbf{x}_i \doteq [x_i\ y_i]^T = \pi(\mathbf{X}_i) \doteq \left[\frac{X_i}{Z_i}\ \frac{Y_i}{Z_i}\right]^T \in \mathbb{R}^2 \qquad (1)$$

the corresponding projection onto the image-plane (Fig. 1). Under the assumption that the scene moves *rigidly* relative to the viewer, with a translational velocity V and a rotational velocity Ω, the 3-D coordinates of each point evolve according to

$$\begin{cases} \dot{\mathbf{X}}_i = \Omega \wedge \mathbf{X}_i + V & \mathbf{X}_i(0) = \mathbf{X}_{i_0} \\ \mathbf{y}_i = \pi(\mathbf{X}_i) + n_i & \forall\, i = 1:N \end{cases} \qquad (2)$$

where n_i represents an error in measuring the position of the projection of the point i, and π represents an ideal perspective projection. Throughout the paper, $\mathbf{y}_i$ indicates the noisy version of the projection $\mathbf{x}_i = [x_i\ y_i]^T$.

2.2. Simultaneous Structure and Motion Estimation

The Eqs. (2) may be regarded as a nonlinear dynamical model having the 3-D position of each feature-point in the state, and having unknown inputs (or parameters) V, Ω. Solving the visual motion estimation problem consists in reconstructing the ego-motion parameters V, Ω from all the visible points, i.e., estimating the unknown inputs of the above system from its noisy outputs (model inversion).

Since the state of the model (2) is also not known, a first approach consists in enlarging it as to include all the unknown parameters, and then use a state observer (for instance an Extended Kalman Filter), for estimating both 3-D structure and motion simultaneously. The reasons why this approach is problematic are both theoretical and practical, as discussed in (Soatto, 1997); the reader interested in the details can consult that reference along with Isidori (1989) for an introductory treatment on nonlinear observability. In the next Section 2.3, which may be skipped at a first reading, we briefly summarize the conclusions that motivate the introduction of structure-independent models for estimating motion.

2.3. *Against Simultaneous Structure and Motion Estimation*

The model (2) is not observable "as is". Metric constraints must be imposed on the state-space manifold in order to achieve local-weak observability. Even after imposing such metric constraints, the observable manifold is covered with three levels of Lie-differentiation, which causes the dynamics of the observer to be slow[1].

Secondly, having structure in the state causes the dimension of the observer to be very large, as the number of features visible in a typical realistic scene is on the order of few hundreds. Also, features enter/exit the field of view or appear/disappear due to occlusions, so one is forced to deal with a variable number of points[2], and motion information can only be integrated to the extent in which all features are visible. In fact, whenever a new feature is inserted into the state, it needs to be initialized, and the initialization error affects all the other states—including the motion components—causing discontinuities in their estimates.

Moreover, the model (2) is *block-diagonal* with respect to the structure parameters, in the sense that the coordinates of each point $\mathbf{X}_i$ in (2) are directly coupled only to themselves and to the motion parameters, but not to the coordinates of other points $\mathbf{X}_j$ $i \neq j$ (of course points are related to each other *indirectly* through the motion parameters). This implies that the observability of the motion parameters does not depend upon the number N of visible features. On the contrary, it is highly intuitive that, the more points are visible, the better the perception of motion ought to be.

These observations, which are discussed in Soatto (1997), serve to motivate the introduction of structure-independent models for estimating motion.

2.4. *Towards Structure-Independent Motion Estimation*

In this paper we will show that it is possible to recursively invert the system (2) and estimate motion (the input) *independent of structure* (the state) using a technique which has been recently introduced in Soatto et al. (1996) for identifying nonlinear implicit systems with parameters on a manifold.

Our scheme is motivated by the work of Heeger and Jepson, who formulated the task as a *static* optimization problem in Heeger and Jepson (1992), Jepson and Heeger (1991).

The scheme we present may be considered as a recursive solution to the task of Heeger and Jepson using methods which pertain to the field of nonlinear estimation and identification theory. As a result, the minimization task which is the core of the subspace method for recovering rigid motion can be solved in a principled way using an Implicit Extended Kalman Filter (IEKF) Bucy (1965), Jazwinski (1970), Kalman (1960), Soatto et al. (1996) according to nonlinear Prediction-Error criteria (for an introductory treatment of Prediction-Error methods in a linear context, see for example Soderstrom and Stoica (1989)). The method exploits in a pseudo-optimal manner the information coming from a long stream of images, making the scheme robust and computationally efficient.

In the next Section 3, we will re-derive the subspace constraint proposed by Heeger and Jepson (1992), and in Section 4 we will view such a constraint as an implicit dynamical model, and introduce the appropriate tools for identifying it.

3. Motion Reconstruction via Least-Squares Inversion Constrained on Subspaces

Consider the following expression of the first derivative of the output of the model (2), which is referred to in the literature as the "motion field" and represents the velocity of the projection of the coordinates of each feature-point in the image-plane (Heeger and Jepson, 1992):

$$\dot{\mathbf{x}}_i(t) = \left[\frac{1}{Z_i}\mathcal{A}_i \mid \mathcal{B}_i\right]\begin{bmatrix} V(t) \\ \Omega(t) \end{bmatrix} \tag{3}$$

where

$$\mathcal{A}_i \doteq \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \end{bmatrix}$$
$$\mathcal{B}_i \doteq \begin{bmatrix} -x_i y_i & 1+x_i^2 & -y_i \\ -1-y_i^2 & x_i y_i & x_i \end{bmatrix}. \tag{4}$$

The motion field is not directly measurable. Instead, what we measure are brightness values on the imaging sensor. For practical purposes, the motion field is approximated by the "optical flow", which consists in the velocity of brightness patches on the image-plane. Such an approximation is by and large satisfied in the presence of highly textured Lambertian surfaces and constant illumination. However, outliers are quite common in realistic image sequences, due to the presence of occlusions, specularities, shadows etc. Any motion

estimation algorithm willing to operate in real-time on realistic sequences must be able to deal with such situations in an automatic fashion.

In the next sections we will assume that we can measure directly the motion field, neglecting outliers. Only later, in Section 4.5, will we show how it is possible to spot-out outliers due, for instance, to T-junctions, specularities, matching errors from the feature-tracking algorithm, and reject them before the can affect the estimates of 3-D motion.

3.1. *Recovery of the Direction of Translation from Two Views*

By observing a sufficient number of points $\mathbf{x}_i \ \forall \ i = 1 \cdots N$, one may use Eq. (3) for writing an overdetermined system which can be solved for the inverse depth and the rotational velocity in a least-squares fashion. To this end, rearrange Eq. (3) as

$$\dot{\mathbf{x}}_i(t) = [\mathcal{A}_i V(\theta, \phi) \mid \mathcal{B}_i] \begin{bmatrix} \frac{1}{Z(t)_i} \\ \Omega(t) \end{bmatrix}.$$

Since the translational velocity V multiplies the inverse depth of each point, both can be recovered only up to an arbitrary scale factor. Due to this scale ambiguity, we may only reconstruct the direction of translation; hence V may be restricted to be of unit norm, and represented in local (spherical) coordinates[3] as $V(\theta, \phi) \in \mathbf{S}^2$. For instance, θ may denote the azimuth angle in the viewer's reference, and ϕ the elevation angle. If some scale information becomes available, as for example the size of a visible object, it is possible to rescale the depth and the translational velocity, as we will discuss in the experimental section. When N points are visible, the equations above may be rearranged into a vector equality:

$$\dot{\mathbf{x}} = \tilde{\mathcal{C}}(\mathbf{x}, \theta, \phi) \left[\frac{1}{Z_1}, \ldots, \frac{1}{Z_N}, \Omega \right]^T, \tag{5}$$

where

$$\tilde{\mathcal{C}}(\mathbf{x}, \theta, \phi) \doteq \begin{bmatrix} \mathcal{A}_1 V & & \mathcal{B}_1 \\ & \ddots & \vdots \\ & & \mathcal{A}_N V \quad \mathcal{B}_N \end{bmatrix}$$

and $\mathbf{x}$ is a $2N$ column vector obtained by stacking the $\mathbf{x}_i \ \forall \ i = 1 \cdots N$ on top of each other. At this point one could solve the above Eq. (5) in a least-squares fashion for the inverse depth and rotation:

$$\begin{bmatrix} \frac{1}{\hat{Z}_1} \\ \vdots \\ \frac{1}{\hat{Z}_N} \\ \hat{\Omega} \end{bmatrix} = \tilde{\mathcal{C}}^\dagger \dot{\mathbf{x}} \tag{6}$$

where the symbol † denotes the pseudo-inverse. By substituting this result into Eq. (5),

$$\dot{\mathbf{x}} = \tilde{\mathcal{C}} \tilde{\mathcal{C}}^\dagger \dot{\mathbf{x}},$$

one ends up with an *implicit constraint* on the direction of translation, which is represented by $V(\theta, \phi)$. After rearranging the terms and writing explicitly the pseudo-inverse, one gets the following subspace algebraic constraint (Heeger and Jepson, 1992):

$$[I - \tilde{\mathcal{C}}(\tilde{\mathcal{C}}^T \tilde{\mathcal{C}})^{-1} \tilde{\mathcal{C}}^T] \dot{\mathbf{x}} \doteq \tilde{\mathcal{C}}^\perp \dot{\mathbf{x}} = 0. \tag{7}$$

It is then possible to exploit this constraint for recovering the direction of translation by solving the following nonlinear optimization problem:

$$\hat{V} = \arg \min_{V \in \mathbf{S}^2} \| \tilde{\mathcal{C}}^\perp(\mathbf{x}, V) \dot{\mathbf{x}} \|. \tag{8}$$

In other words one seeks for the best vector in the two-dimensional sphere such that $\dot{\mathbf{x}}$ is the null space of the orthogonal complement of the range of $\tilde{\mathcal{C}}(\mathbf{x}, V)$. If the matrix $\tilde{\mathcal{C}}$ was invertible, the above constraint would be satisfied trivially for all directions of translation. However, when $2N > N + 3$, $\tilde{\mathcal{C}}\tilde{\mathcal{C}}^\dagger$ has rank at most $N + 3$, and therefore $\tilde{\mathcal{C}}^\perp$ is not identically zero.

Note that the solution consists in "adapting" the orthogonal complement of the linear space generated by the columns of $\tilde{\mathcal{C}}$—which is highly structured as a function of $V(\theta, \phi)$—until a given vector $\dot{\mathbf{x}}$ is its null space. Heeger and Jepson (1992), in their early work, first solved this task by minimizing the two-norm of the above constraint (8) using a search over θ, ϕ on a sampling of the sphere.

In Section 4 we rephrase the subspace constraints described in this section as a nonlinear and implicit dynamic model. Estimating motion corresponds to identifying such a model with the parameters living on a sphere: we propose a principled solution for performing the optimization task, which takes into account the temporal coherence of motion and the geometric structure of the residual (8).

3.2. Recovery of Rotation and Depth

Once the direction of translation has been estimated as $\hat{V} = V(\hat{\theta}, \hat{\phi})$, we may use Eq. (6) to compute a least-squares estimate of the rotational velocity and inverse depth from

$$\begin{bmatrix} \frac{1}{Z_1} \\ \vdots \\ \frac{1}{Z_N} \\ \Omega \end{bmatrix} = \tilde{\mathcal{C}}^{\dagger}(\mathbf{x}, \hat{\theta}, \hat{\phi})\dot{\mathbf{x}}. \tag{9}$$

Note that, from the variance/covariance of the estimation error of the direction of translation θ, ϕ, it is possible to characterize the second order statistics of the estimate of the rotational velocity, R_Ω. We may therefore design a simple linear Kalman filter which uses the above estimates as "pseudo-measurements" and is based upon the linear model

$$\begin{cases} \Omega(t+1) = \Omega(t) + n_{rw} \\ \tilde{\mathcal{C}}^{\dagger}_{2N+1:2N+3}(\mathbf{x}, \theta, \phi)\dot{\mathbf{x}} = \Omega(t) + n_\Omega \end{cases} \tag{10}$$

where the notation $\tilde{\mathcal{C}}^{\dagger}_{2N+1:2N+3}$ stands for the rows from $2N+1$ to $2N+3$ of the pseudoinverse of the matrix $\tilde{\mathcal{C}}$; n_{rw} is the noise driving the random walk model, which is to be intended as a tuning parameter, and n_Ω is an error whose variance R_Ω is inferred from the variance of the estimation error for θ, ϕ.

The equations for the Kalman filter corresponding to the above (linear) model are standard, and can be found in textbooks; see for example Jazwinski (1970).

3.3. Recovery of Structure

After the rotational and translational velocities have been recovered, they may be fed, together with the variance of their estimation error, into a recursive structure-from-motion module which processes motion error, such as for example Oliensis and Thomas (1992), Soatto et al. (1993). The main focus of this paper is the estimation of motion, and in the experimental section we have estimated structure using the estimates of motion, as in the scheme presented in Soatto et al. (1993).

However, we just point out in this section an alternative way of estimating structure, that comes from observing that the inverse depth of each point and the direction of translation play interchangeable roles, as it is evident from the motion field Eq. (3). One may therefore "pseudo-invert" the system (3) with respect to the direction of translation and the rotational velocity, and then derive an optimization problem similar to (8) for the scaled inverse depth of each point. This idea is pursued in (Soatto et al., 1995), where the subspace constraint is used to derive a dynamic filter for estimating structure independent of motion.

4. Solving the Subspace Optimization with a Dynamic Filter

In this section we will view the subspace constraint from a different perspective. Instead of considering it an algebraic set of nonlinear equations to be solved for the direction of heading, we view it as a nonlinear and implicit dynamical system, which has parameters constrained onto a two-dimensional sphere. Then we introduce a local identifier based upon an Implicit Extended Kalman Filter in order to recursively estimate the heading direction. Once the heading is estimated, it can be fed into a simple linear Kalman filter that estimates the rotational velocity.

Let us define $\alpha \doteq [\theta, \phi]^T$ as the local coordinate parametrization of the translational velocity V; θ is the azimuth angle, and ϕ the elevation. $\mathbf{x}_i$ are measured up to some error,

$$\mathbf{y}_i \doteq \mathbf{x}_i + n_i, \tag{11}$$

which we model as white, Gaussian and zero-mean: $n_i \in \mathcal{N}(0, R_{n_i})$. In the presence of outliers, this hypothesis is violated, and we will show in Section 4.5 how to detect and reject such outlier measurements before they can affect the estimation process. The error in the location of the features induces an error in the derivative,

$$\mathbf{y}'_i = \dot{\mathbf{x}}_i + n'_i,$$

which is usually approximated by either the optical flow, or by first differences of feature positions between time t and $t+1$. Call $\mathbf{x}$ the column vector obtained by stacking the components of $\mathbf{x}_i$, similarly with $\dot{\mathbf{x}}$. Now define $\tilde{\mathcal{C}}^{\perp}(\mathbf{x}, \alpha)$ as in (5). Then the subspace constraint (7) may be written as $\tilde{\mathcal{C}}^{\perp}(\mathbf{x}, \alpha)\dot{\mathbf{x}} = 0$. Now

$$\begin{cases} \tilde{\mathcal{C}}^{\perp}(\mathbf{x}, \alpha)\dot{\mathbf{x}} = 0 & V(\alpha) \in \mathbf{S}^2 \\ \mathbf{y}_i \doteq \mathbf{x}_i + n_i & \forall\, i = 1 \cdots N \end{cases} \tag{12}$$

represents a nonlinear implicit dynamical system of a particular class, called Exterior Differential Systems (Bryant et al., 1992). *Solving for the translational velocity is equivalent to identifying the above Exterior*

Differential System with parameters α on a differentiable manifold (the sphere in this case) from the noisy data $\mathbf{y}$.

4.1. *Identifying Motion Using Local Implicit Filtering*

The direction of translation, encoded by the two-dimensional vector α, is represented in the above model (12) as an unknown parameter which is subject to three types of constraints. First of all, $V(\alpha)$ is constrained to belong to the unit-sphere in $\mathbb{R}^3$. Secondly, the dynamics of the states $\mathbf{x}$ induces trivially a dynamics on the outputs $\mathbf{y}$:

$$\tilde{\mathcal{C}}(\mathbf{y}, \alpha(t))\,\mathbf{y}' = \tilde{n} \tag{13}$$

where $\tilde{n}$ is a residual noise induced by the measurement noise n. The parameters α must evolve in such a way that the outputs $\mathbf{y}$ satisfy the above dynamics. Since the outputs are directly measured, we could call the above constraint the "a-posteriori" dynamics. However, often times the direction of translation is not free to change arbitrarily, for there is some "a-priori" dynamics it must satisfy. For instance, if the camera is mounted on a vehicle, it must move according to its kinematics and dynamics, which results in a model of the generic form

$$\alpha(t+1) = f(\alpha, n_\alpha) \tag{14}$$

where n_α summarizes all the significant parameters of the vehicle. If the camera is hand-held, or the mechanics of its support is unknown, we know at least that velocity must be a continuous function and the acceleration cannot exceed certain values. In lack of a mechanical model, one may employ statistical models as a mean of describing some inertia. For instance models of the form

$$\alpha(t+1) = f(\alpha) + n_\alpha \qquad n_\alpha \in \mathcal{N}(0, R_\alpha) \tag{15}$$

where f is a polynomial function and n_α is a white, zero-mean Gaussian noise.

By putting these three constraints together, we can write a discrete dynamic model for the parameters

$$\begin{cases} \alpha(t+1) = f(\alpha(t)) + n_\alpha(t) \\ \tilde{\mathcal{C}}(\mathbf{y}, \alpha(t))\,\mathbf{y}' = \tilde{n} \\ \alpha \in [0, \pi) \times \left[-\dfrac{\pi}{2}, \dfrac{\pi}{2}\right) \end{cases} \tag{16}$$

which can be used for designing an Implicit Extended Kalman filter, whose equations we report in the next subsection. Before doing that, however, we would like to stress that the function f in the model equation (16) is a design parameter which is left to the engineer, and depends upon the circumstances in which the algorithm is to be used.

If the algorithm is intended for general purposes, one may choose a *conservative* model, which is a model that fits a larger class than the actual one, neglecting more specific dynamics that may be present, for instance, in vehicle guidance, helicopter flight etc. Should further information about the dynamics of the support of the camera be available, it can easily be exploited by inserting it into the model (15).

A typical case in which no model like (15) can be found is when there is no temporal coherence between subsequent images, which are snapshots of a scene taken from various points of view at different time instants. In such a case, a batch method is most appropriate. Since we are interested in real-time estimation, we always assume that the images are taken sequentially from a camera, so that temporal coherence between subsequent images is guaranteed.

In this paper, we consider the very simplest instance of a statistical model, which is a first-order random walk:

$$f(\alpha) = \alpha. \tag{17}$$

It is not superfluous to point out that the first-order random walk (Brownian motion) does not restrict the motion to having constant velocity. The variance of the noise driving it, R_α, can be considered a tuning parameter that trades off the "speed of convergence" with the "precision" required. One may consider this as a starting point: if the dynamics of the camera in a particular experiment are not captured by this simple model, one can move up the class and consider richer models. It is our experience, however, that a first order random walk works quite well in most cases, in the sense that it allows decent precision while not limiting the range of possible motions to a significant extent. In the experimental section we will show how the simple Brownian motion performs on a variety of situations, ranging from constant-velocity motion, to sinusoidal, to discontinuous velocity, without changing any tuning or modeling parameters.

4.2. *Equations of the Estimator*

From the model (16), it is immediate to derive the equation for an Extended Kalman Filter (EKF) (Jazwinski,

1970; Kalman, 1960) that estimates the direction of translation α. The only caveat is that the measurement equation is in *implicit* form. The key observation is that the vector

$$\epsilon(t) \doteq \tilde{\mathcal{C}}^{\perp}(\mathbf{y}(t), \quad \hat{\alpha}(t+1\,|\,t))\mathbf{y}' \tag{18}$$

plays the role of the "pseudo-innovation" process, and therefore the standard equations of the EKF can be applied (Jazwinski, 1970). We report here the complete set of equations for the filter that estimates the direction of translation using a first-order random walk model. The reader interested in a detailed derivation of the Implicit Extended Kalman Filter may find it, for instance, in Soatto et al. (1996).

Prediction step

$$\begin{cases} \hat{\alpha}(t+1\,|\,t) = \hat{\alpha}(t\,|\,t) & \hat{\alpha}(0\,|\,0) = \alpha_0 \\ P(t+1\,|\,t) = P(t\,|\,t) + R_{\alpha}(t) & P(0\,|\,0) = P_0 \end{cases}$$

Update step

$$\begin{cases} \hat{\alpha}(t+1\,|\,t+1) = \hat{\alpha}(t+1\,|\,t) + L(t+1) \\ \qquad\qquad \tilde{\mathcal{C}}^{\perp}(\mathbf{y}(t), \hat{\alpha}(t+1\,|\,t))\mathbf{y}' \\ P(t+1\,|\,t+1) = \Gamma(t+1)P(t+1\,|\,t)\Gamma^T(t+1) \\ \qquad\qquad + L(t+1)D(t+1)R_{\tilde{n}}(t+1) \\ \qquad\qquad \times D^T(t+1)L^T(t+1) \end{cases}$$

where

$$\begin{cases} L(t+1) = P(t+1\,|\,t)C^T(t+1)\Lambda^{-1}(t+1) \\ \Lambda(t+1) = C(t+1)P(t+1\,|\,t)C^T(t+1) \\ \qquad\qquad + D(t+1)R_{\tilde{n}}(t+1)D^T(t+1) \\ \Gamma(t+1) = I - L(t+1)C(t+1) \\ D(t+1) \doteq \left(\dfrac{\partial \tilde{\mathcal{C}}^{\perp}\dot{\mathbf{x}}}{\partial[\mathbf{x}(t), \dot{\mathbf{x}}]} \right)_{|[\mathbf{y}(t),\mathbf{y}'],\hat{\alpha}(t)} \\ C(t+1) \doteq \left(\dfrac{\partial \tilde{\mathcal{C}}^{\perp}\dot{\mathbf{x}}}{\partial \alpha(t)} \right)_{|\mathbf{y}(t),\hat{\alpha}(t)} \end{cases}$$

and $R_{\tilde{n}}$ is the variance/covariance matrix of the measurement error $\tilde{n} \doteq [n, n']$, considered as a white noise[4]. R_{α} is a tuning parameter that corresponds to the variance of the noise driving the random walk model.

At each step, the estimates of the direction of translation can be used for *instantaneously* recovering the rotational velocity from (9). Such a pseudo-measurement may also be used for updating the state of a linear Kalman filter based upon the model (10):

Prediction step

$$\begin{cases} \hat{\Omega}(t+1\,|\,t) = \hat{\Omega}(t\,|\,t) & \hat{\Omega}(0\,|\,0) = \Omega_0 \\ P_{\Omega}(t+1\,|\,t) = P_{\Omega}(t\,|\,t) + R_{rw}(t) & P_{\Omega}(0\,|\,0) = P_{\Omega_0} \end{cases}$$

Update step

$$\begin{cases} \hat{\Omega}(t+1\,|\,t+1) = \hat{\Omega}(t+1\,|\,t) + L_{\Omega}(t+1) \\ \qquad\qquad \times \left(\tilde{\mathcal{C}}^{\dagger}_{2N+1:2N+3}(\mathbf{y}, \hat{\alpha})\mathbf{y}' - \hat{\Omega}(t+1\,|\,t) \right) \\ P_{\Omega}(t+1\,|\,t+1) = \Gamma_{\Omega}(t+1)P_{\Omega}(t+1\,|\,t)\Gamma^T_{\Omega}(t+1) \\ \qquad\qquad + L_{\Omega}(t+1)R_{\Omega}(t+1)L^T_{\Omega}(t+1) \end{cases}$$

where the gain matrices $L_{\Omega}, \Gamma_{\Omega}$ are the usual ones of the linear Kalman Filter (Kalman, 1960).

It is easy to verify that both the models (16) and (10) are locally-weakly observable. In fact, the uniqueness results in the analysis of the algorithm of Jepson and Heeger (1991) are equivalent to the assessment of the observability of the model (16), for it is instantaneously observable. The model (10) is observable, for the state and measurement models are the identity and the filter just acts as a smoother. Note that the algorithm just presented produces a measure of the reliability of the estimates in the form of the second order statistics of the estimation error P and P_{Ω}.

4.3. Enforcing Rigid Motion: The Positive Depth Constraint

When estimating motion from visible points, we must enforce the fact that the measured points are *in front of the viewer*. This may be easily done in the prediction step by computing the mean distance of the centroid and checking whether it is positive. If it is negative, the antipodal point of the state-space sphere is chosen as the prediction.

When we do not impose such a constraint, the filter may converge to a rigid motion which corresponds to points moving behind the viewer, and is therefore not physically realizable. However, if we allow such a condition to happen by releasing the positive depth constraint, and then feed the estimate into a structure estimation, such as for example a simple Extended Kalman Filter (Matthies et al., 1989; Oliensis and Thomas, 1992; Soatto et al., 1993) initialized with points at positive depth and a large model-error variance, the result is a *rubbery percept of structure* which has been observed also in psychophysical experiments (Kolb et al., 1994). A pictorial representation of the rubbery percept is illustrated in Fig. 2.

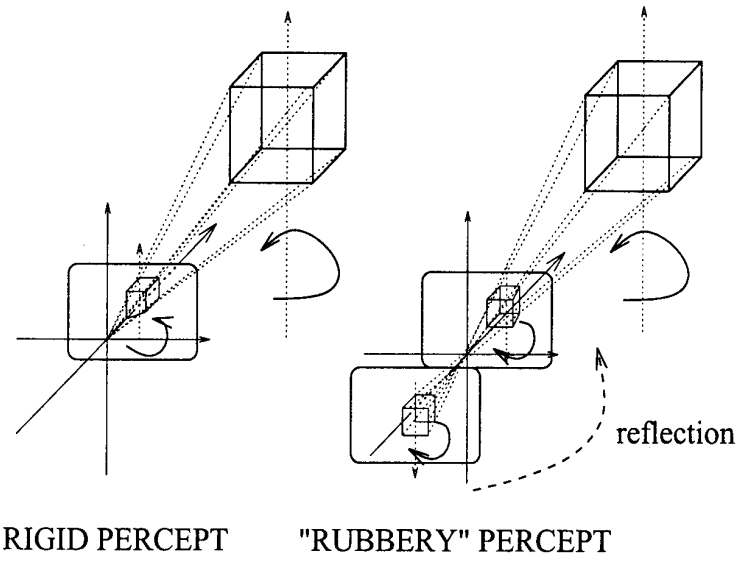

Figure 2. Pictorial illustration of the "rubbery" perception: motion is estimated without imposing the positive depth constraint; this may result in a motion estimate which is compatible with a rigid structure behind the viewer. Once such a structure is interpreted as being in front of the viewer, it gives rise to the perception of a "rubbery" structure rotating in the opposite direction.

4.4. Independence from Structure Estimation

It is worth noting that the state of the filter proposed contains only the motion parameters, and is therefore independent from the structure of the observed scene, provided some general-position conditions. Such conditions are satisfied when the scene cannot embedded in a planar surface, and the motion relative to the viewer generates non-zero parallax. Such conditions describe a zero-measure set in the possible structure and motion configurations, and the noise in the image-plane coordinates is sufficient to set the model in general position. As a consequence, we do not need to track a specific set of features; instead, at each step we can change set of features or locations where we compute the optical flow/feature tracking, without causing discontinuities in the estimates of motion. This is a key property of the filter, since it allows us to deal easily with occlusion and appearance of new features.

Also, note that the filter is able to work properly even when the number of visible features drops down to less than five (for small accelerations), since it integrates over time the information from each incoming frame. This, together with the robustness and noise-rejection properties, is a substantial advantage over two-views schemes.

4.5. Outlier Rejection

One of the crucial features of the subspace filter, as well as the essential filter (Soatto et al., 1994), is its independence from the structure of the scene. However, each feature-point is indirectly represented via the innovation process (18). In particular, for each feature-point with projective coordinates $\mathbf{x}_i$, the components of the innovation ϵ_i, defined in (18), describe how such a feature-point is compatible with the current estimate of motion $\hat{\alpha}$. Since at each step the filter computes the pseudo-innovation vector, it is possible to compare each component against the same at the previous time instant and, using some simple statistics, reject the measurements that give too large a residual before updating the estimates of motion. This technique may be applied both for rejecting outliers, such as mismatches in the optical flow, T-junctions, specularities etc. and for segmenting the scene into a number of independently moving rigid objects, as in Soatto and Perona (1994).

5. Experimental Assessment

In this section we report a series of simulations and experiments on real sequences. Before that, we discuss some of the issues on the implementation, stressing

the fact that the model and the tuning parameters were the same for all the experiments, including the one in Section 5.3.2, which is designed on purpose for challenging the first-order random walk model which we have employed.

5.1. Implementation

We have implemented the filter using `Matlab`. Each update step consists essentially in 15 products of matrices of size varying from 2×2 to $2N\times 2N$, one inversion of the $2N \times 2N$ variance of the pseudo-innovation, 5 sums and the computation of the Singular Value Decomposition (SVD) of $\tilde{\mathcal{C}}$, for a total of circa 1 Mflop for $N = 20$ points. However, the computation can be cut in half by taking into account the sparse structure of the matrices involved in the computation (block-diagonal structure of R_n and $\tilde{\mathcal{C}}$). A time-consuming part of the algorithm is also the linearization of the system with respect to the measurements, $D(t+1)$.

Since the Extended Kalman Filter is based upon the assumption that the linearization error is negligible, which is not often the case, we have added to the variance $DR_{\tilde{n}}D^T$ a small symmetric random matrix in order to account for the linearization error. This practice typically improves the performance of the Extended Kalman Filter for models which are strongly nonlinear.

A crucial part of the design of an EKF consist in "tuning" it, i.e., in assigning a value to the elements of the variance/covariance matrices of the model errors: R_α, R_{rw}. A custom procedure is to assume that these matrices are diagonal, and then play with their values until the prediction error is as white as possible. Standard tests are available for this procedure, such as the "cumulative periodogram" (the integral spectrum of the prediction error). In our experiments we have performed a coarse tuning by changing the variances of the model errors by one order of magnitude at a time. We did not perform any ad-hoc or fine tuning, and the setting was the same throughout the different experiments.

In all experiments, unless stated otherwise, the filter was initialized to zero: $\alpha_0 = 0, \Omega_0 = 0$, and the initial variance of the estimation error P and P_Ω was the identity matrix of dimension 2 and 3 respectively, scaled by 100.

In order to implement the filter, the linearization of the model is needed. In Appendix A we report the detailed computation of the local-linearization of the measurement model.

5.2. Scale Information Recovery

The scheme proposed recovers the direction of translation as a normalized vector of $\mathbb{R}^3$. Such a normalization is necessary because of the presence of a global scale-factor ambiguity that affects the norm of translation and the inverse depth of the visible features, as it can be seen from the Eq. (3). The important fact to realize is that there is only *one* scalar ambiguity for the whole sequence so that, should some scale information become available at any instant, it can be propagated across time and the scale ambiguity resolved.

In fact, at each step the normalized translation and the rotational velocity estimated by the filter may be used for computing some "normalized" structure, which can be re-sized to fit the scale information available, as done in Soatto et al. (1993). If no scale information is available, the initial translation may be used as a unit scale, or the distance between any two features, for instance. The issue of how to propagate scale information is discussed in Bouguet and Perona (1995).

5.3. Simulation Experiments

We have generated at random a set of 20 points in space, distributed uniformly in a cubic volume of side 1 m, with the centroid placed 1.5 m ahead of the image plane. The points are projected onto an image plane of 512×512 pixels with focal length of 750 pixels. The cloud of points rotates about its centroid with a velocity of circa 5°/frame, with the centroid maintained on the optical axis at a fixed distance from the center of projection. White, zero-mean Gaussian noise is added to the projections. The motion is roto-translational in the viewer's reference frame, and is challenging since the effects of rotation and translation superimpose.

Convergence is reached from *zero initial conditions* and noise in the image plane coordinates up to 8 pixel std. The convergence of the main filter with a noise level of 1 pixel std is reported in Fig. 3, while the same experiment is repeated with a noise level of 8 pixels std in Fig. 4. In both cases the positive depth constraint has been enforced. The transient for converging from zero initial conditions ranges from 5 to 40 steps, depending on the noise level, the type of motion and the structure of the scene.

The least-squares pseudo-measurements of the rotational velocity, computed as described in Section 3.2, are plotted in Fig. 5 (dashed lines), and compared with

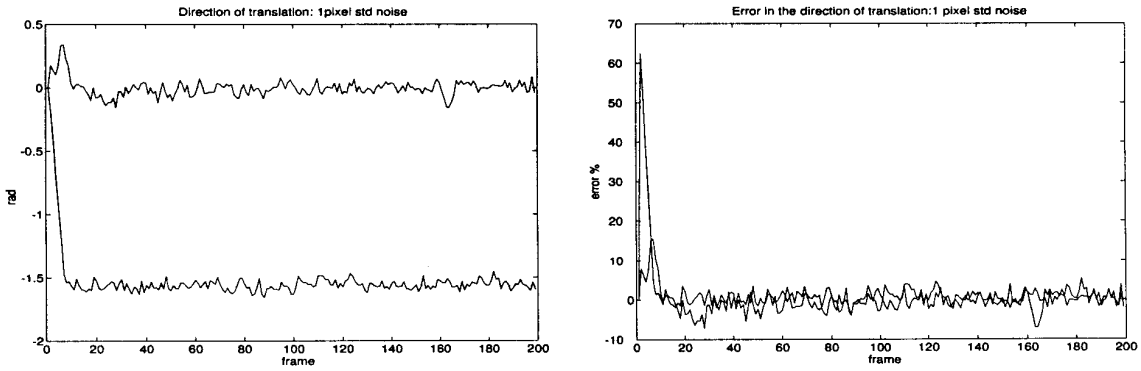

Figure 3. Estimates and errors for the direction of translation when the noise in the image plane has a standard deviation of 1 pixel (according to the performance of common optical flow/feature tracking schemes). Ground truth is displayed in dotted lines. In the left plot the elevation angle ϕ is constant and equal to zero, the azimuth θ is close to $-\frac{\pi}{2}$. Note that convergence is reached from zero initial conditions in about 10 steps.

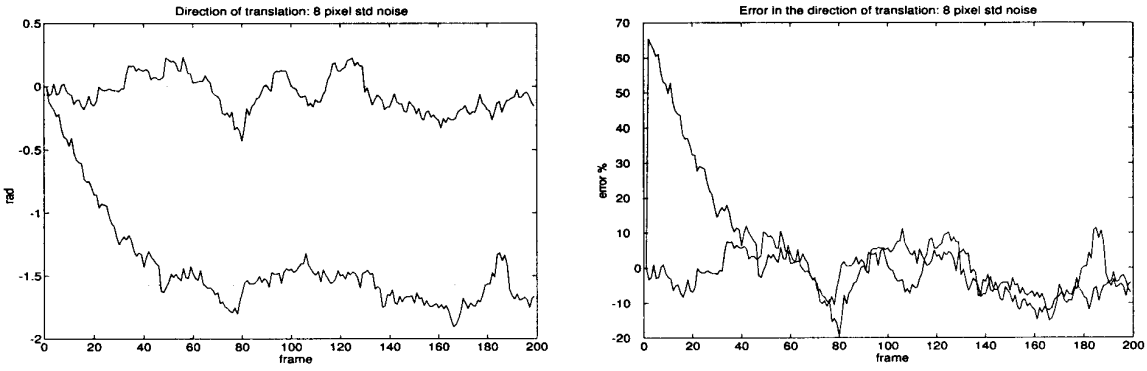

Figure 4. (Left) estimates of the two components of the direction of translation. In the left plot the elevation angle ϕ is constant and equal to zero, the azimuth θ is close to $-\frac{\pi}{2}$. The noise in the image plane measurements had 8 pixel standard deviation. The initial conditions were zero for both components. The ground truth is in dotted lines. (Right) estimation error for the direction of translation. With noise of 8 pixel std in the data, the estimates are still within 20% of the true value.

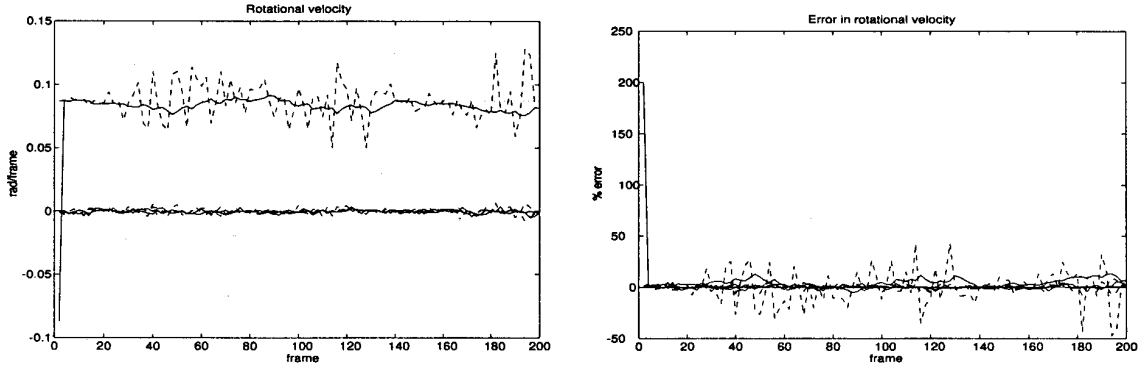

Figure 5. Estimates for the components of rotational velocity (left) and corresponding error (right). Ground truth is displayed in dotted lines; the filtered estimates are in solid lines. The least-squares computation of the rotational velocity is in dashed lines.

the recursive estimates (solid line) using the linear Kalman Filter described in Section 3.2 with a noise level of 1 pixel std.

5.3.1. Altering the Basic Experiment. The basic experiment has been modified in order to test the filter against increasing levels of noise, aspect ratio of the visible scene, size of the visual field and magnitude of motion.

In Figs. 6 and 7 we show the variance of the pseudo-innovation and the norm of the estimation error respectively, as a function of the measurement noise, which ranges from 1 to 4 pixels std. The same plots are reported for the recursive version of Horn's two-frames algorithm (Horn, 1989; Soatto et al., 1996), which breaks down consistently for noise levels higher than 1.1 pixels std.

In Fig. 8 we report the minimum "thickness" of the rotating cloud that can be tolerated before the scheme breaks down. Again, there is a significant advantage over two-frames based algorithms.

In Fig. 9 we report the smallest aperture angle under which the scene can be seen and its motion estimated correctly. The subspace filter has a slight advantage with respect to a two-frames based algorithm. However, all schemes based upon a full perspective model need to have a large visual field.

In Fig. 10, we experiment with dependence upon image-velocity. The model of the subspace filter is based on a differential (exponential) representation of motion, and assumes that the velocity of the brightness patches $\mathbf{y}'$ can be measured. In practice, first differences of feature positions are employed as an approximation to the velocity. Such an approximation

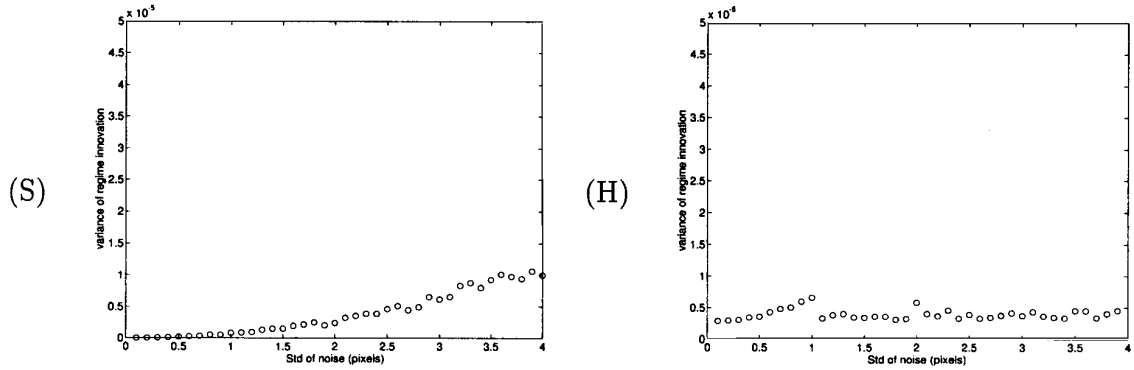

Figure 6. **Statistics of the innovation vs. noise level and initial conditions.** (S = subspace, H = recursive Horn.) The average variance of the innovation over a window of 20 frames is plotted as a function of the noise level. The subspace filter (S) proves robust, and converges for zero initial conditions and noise larger than 4 pixels. The variance of its innovation follows the ideal parabola, while the estimate of (H) breaks down at a noise level of 1.1 pixel std. Notice that, while the variance of the innovation decreases when the filter diverges, the estimation error, as expected, increases (Fig. 7).

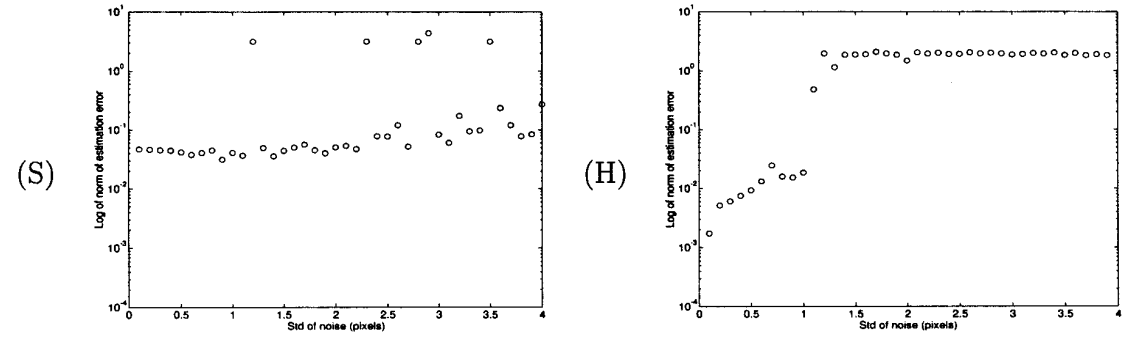

Figure 7. **Estimation error vs. noise level and initial conditions.** (S = subspace, H = Horn. log-scale.) The subspace filter converges in all instances but in 5 cases, where the sample of the estimation error was taken before the filter reached convergence while it was temporarily trapped into a local minimum. The algorithm based on two frames (H) fails consistently for noise levels higher than 1.1 pixel std.

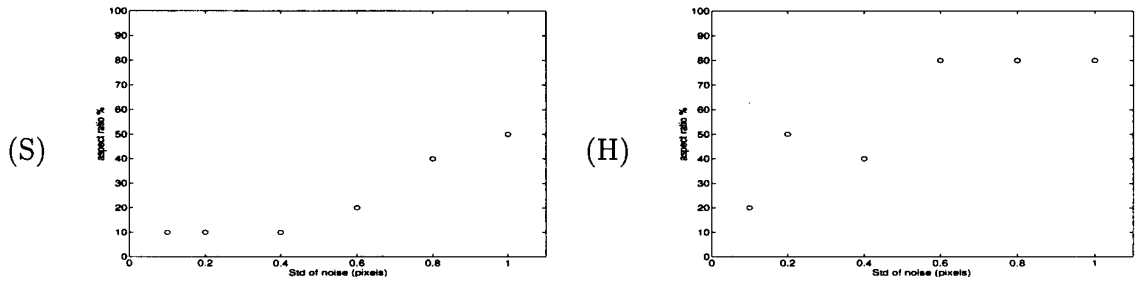

Figure 8. **Critical aspect ratio vs. noise level.** (S = subspace, H = Horn.) In this experiment we "flatten out" the structure by decreasing the ratio between the depth and the width of the cloud of points. For any given noise level, we plot the minimum aspect ratio (maximum "flatness") tolerated by the filters. The aperture angle was 28°. If the noise is small (for example one tenth of a pixel), then the filter can tolerate a very flat structure (for instance 10% aspect-ratio). As the noise increases, the filter is more and more sensitive to the presence of depth in the structure. The reduction of the aspect ratio could be viewed as a reduction of the aperture while the cloud shows its narrower face to the viewer.

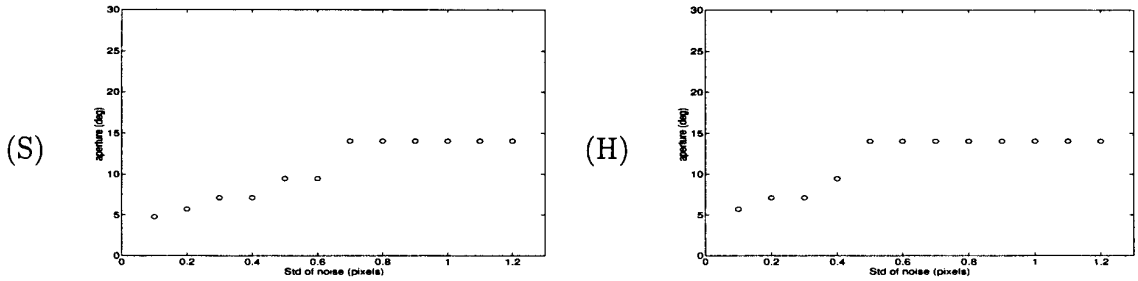

Figure 9. **Critical aperture vs. noise level.** (S = subspace, H = Horn.) The minimum aperture angle tolerated by each filter depends upon the noise level as indicated in the plot above. When the noise is one tenth of a pixel, the filters can estimate the motion of structures viewed up to an angle of 5° circa, while as the noise increases up to 1.2 pixels, the aperture angle has to be larger than 15°.

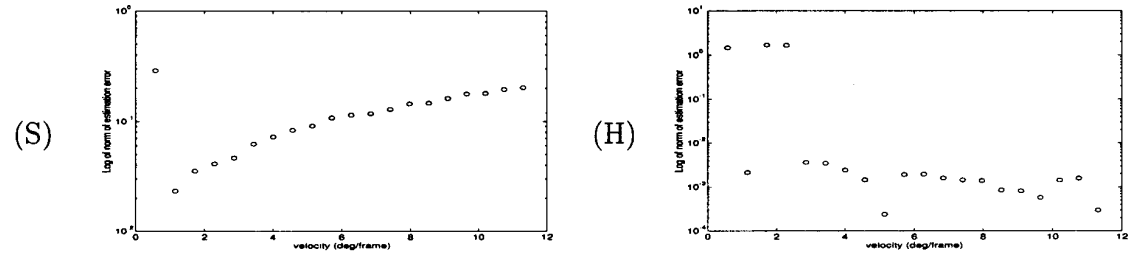

Figure 10. **Norm of the estimation error vs. rotation angle.** (S = subspace, H = Horn. Log-scale.) The schemes based upon the epipolar constraint from two frames (H) do not converge for baselines shorter than a threshold (2.2° in this case). Once the threshold is reached, they improve marginally by increasing the instantaneous baseline. On the contrary, the subspace filter (S), which is based upon an instantaneous constraint, degrades as the baseline increases. Note that the subspace filter is implemented in exponential coordinates, which helps correcting for the finiteness of the sampling interval under the assumption of slow accelerations. The field of view was 28 degrees and the noise half a pixel std.

is honest as long as the image-plane motion is small. As the image-plane motion increases, the performance degrades as shown in Fig. 10.

5.3.2. Challenging the Model. In designing the estimator of the parameters α for the model (16), we have wide open choice on the dynamical model for the state f, depending upon the conditions in which the algorithm is applied. For instance, if the camera is mounted on a mobile vehicle, we may use the kinematics and dynamics of the support for describing the evolution of the state. If we know that the camera is moving with considerable inertia, we may employ a smoothness constraint etc. In the lack of any model,

we can employ statistical models, for example fixed order random walks. In the experiments reported here we have chosen the simplest possible, which is the first order, corresponding to a Brownian motion. Whether this model is rich enough to capture the possible motions undergone by the camera is a question of modeling which is left to the engineer, who has to judge the intrinsic tradeoff between flexibility (large model variance) and accuracy or "smoothness" (small model variance).

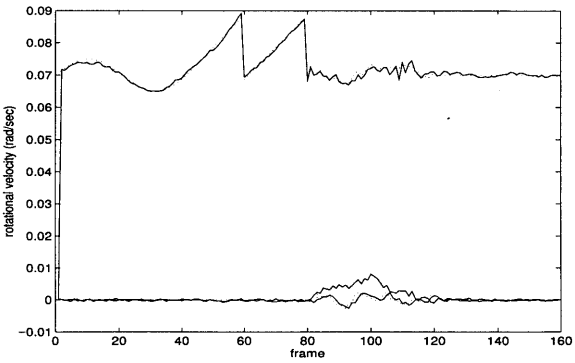

Figure 11. Convergence of the filter with a first-order random walk state model in the presence of non-smooth parameter dynamics. The components of the rotational velocity of the camera are first modulated by a sinusoidal, then by a discontinuous saw-tooth and then they drift with a second order random walk before returning to the initial constant-velocity setting. The estimates (solid lines) follow the ground truth (dotted lines) despite it evolves according to dynamics which are not captured by the state model of the filter.

Just for the sake of illustration, we have considered the same synthetic experiment described in the previous section, and modulated the speed of rotation about the object's axis first with a *sinusoid*, then with a *saw-tooth* discontinuous function, and then with a *second order* random walk (which is one step up the ladder of the class of random walks, and cannot be captured in principle by the Brownian motion). During the latter phase we have also altered the other components of the rotational and translational velocity. Eventually, motion resumed to constant velocity. Note that the parameter which is modulated is the most difficult to estimate, since the effects of rotation and translation are similar (it is one of the manifestations of the so-called "bas-relief ambiguity"). In order to appreciate the precision of the tracking, we have lowered the noise level down to a tenth of a pixel. In Fig. 11 we show the three components of the rotational velocity (solid lines) superimposed to the ground truth (dotted lines). The two spherical coordinates of the direction of translation are plotted in Fig. 12 (solid lines) along with the ground truth (dotted lines). The estimates of the filter follow closely the motion parameters, even at the discontinuities. It is worth pointing out that the tuning was exactly the same in all the experiments in this paper, and no ad-hoc tuning was performed. It is possible to see a small, but not zero-mean, estimation error, which is a clear symptom that the model employed (a first order random walk) does not capture the true dynamics of the parameters (sinusoidal, discontinuous or a second-order random walk). If one wanted

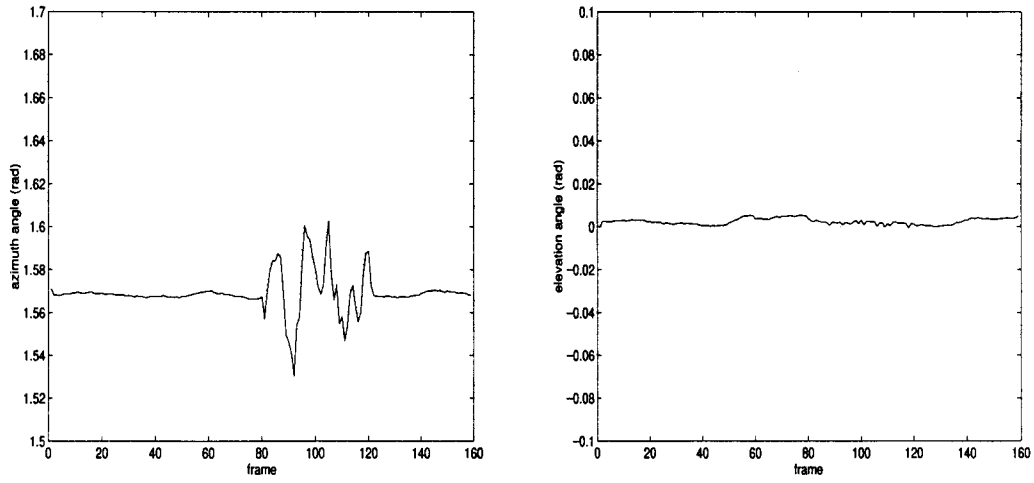

Figure 12. Spherical components of the translational velocity for the experiment with non-constant velocity: azimuth (left) and elevation (right). While the rotational velocity is modulated with sinusoids and saw-tooths, translation is held constant. Between frames 80 and 120 the parameters drift according to a second-order random walk. It can be noticed that the filter follows the estimates with a small but non-zero-mean estimation error. This is due to the fact that the model that generates the data is not captured by the model used for the estimation.

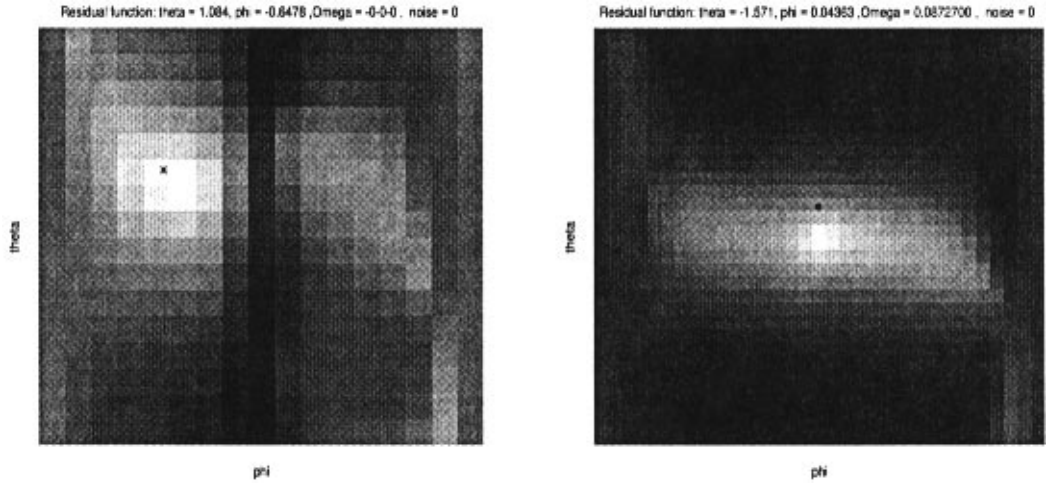

Figure 13. Brightness plots of the residual function. The value of the residual is plotted on the state-space of the filter, which are the local coordinates of the sphere of directions of translation. Bright regions denote small residuals. The black asterisk is the "true" motion which generated the residual. Note that for small rotations (left) the minimum of the residual coincides with the true motion. When the rotational velocity is large (right) the Euler step approximation is no longer valid, and the minimum moves from the true location.

to get rid of these effects, a higher-order random walk should be considered. However, the one just performed is an extreme experiment, and usually real sequences taken from video exhibit a considerable amount of inertia. Therefore we will restrict ourselves to the simplest first-order random walk.

5.3.3. The Residual Plot in the State-Space. A typical plot of the residual function, which is the value of the subspace constraint (18) as a function of the parameters $\theta \in [0, \pi)$, $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2})$, is shown in Fig. 13 for a particular value of the states. The residual depends both on the motion and structure parameters. For an isotropic cloud of dots undergoing constant-velocity motion, the residual is nearly constant. Therefore, it is sufficient to show just one frame of the residual with the filter trajectory superimposed. In the following subsections we restrict our attention to the constant-velocity case just because—the residual function being constant—it is possible to display it. The bright areas indicate a small residual value. The black asterisk indicates the motion (in the local coordinates of the sphere of directions of translation) which generated the residual. It is noted that the minimum of the residual is displaced from the true motion when the norm of the rotational velocity is large. This is due to the fact that we approximate the velocity of the projected points (motion field) with first differences; the approximation is good as long as $R \doteq e^{\Omega\wedge} \cong I + \Omega\wedge$, i.e., as long as the norm of the rotational velocity is small.

5.3.4. Convergence and Local Minima. The reader may have noticed the presence of local minima in the plots of the residual function (Figs. 13–17): if motion is estimated *instantaneously* from two frames, as in Heeger and Jepson (1992), the estimate can be trapped into a local minimum. In our experiments, however, we have rarely witnessed convergence to a local minimum, unless temporary. This is due to the

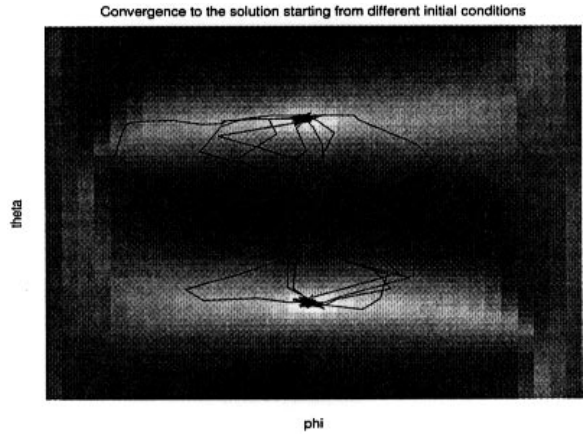

Figure 14. Convergence when the positive depth constraint is not imposed and the initial condition is chosen at random around the origin (which appears in the center of the plot): a number of trajectories is shown in black solid lines superimposed on the brightness plot of the residual function. The filter may converge to either the correct rigid interpretation (bright region on the top half of the plot) or to the local minimum corresponding the "rubbery" interpretation (bright area on the bottom half of the plot).

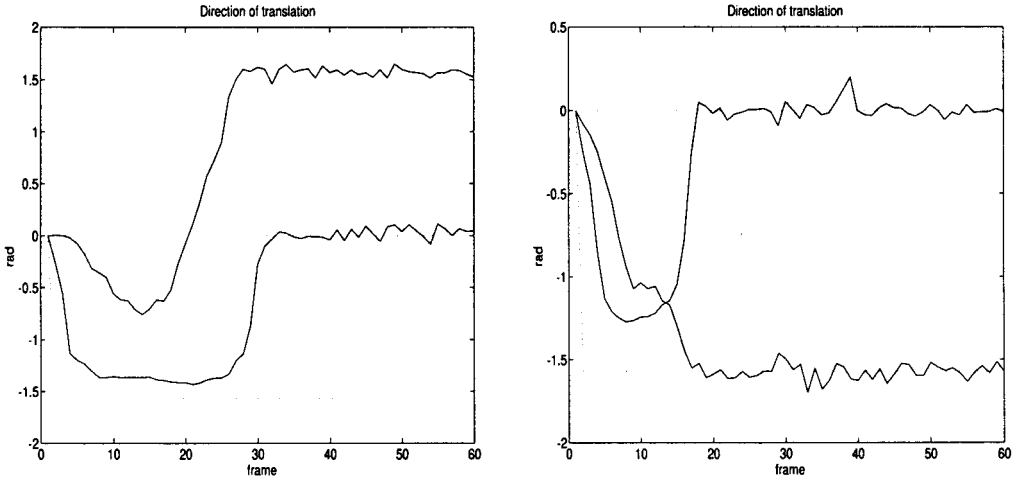

Figure 15. (Left) convergence to a shallow local minimum and then to the local minimum corresponding to the rubbery interpretation when the positive depth constraint is not enforced. (Right) convergence to a shallow local minimum and then to the correct rigid motion (see also Fig. 16).

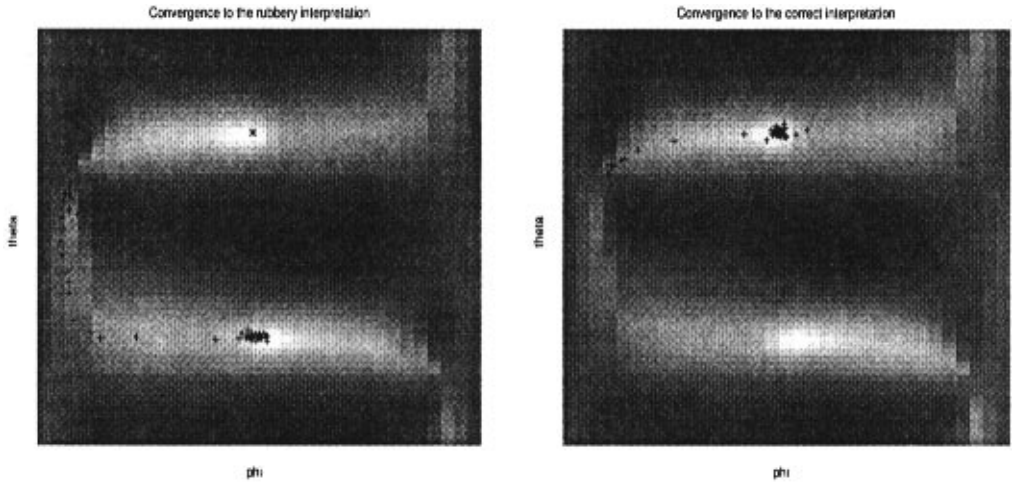

Figure 16. Convergence to the "rubbery interpretation" (left) versus convergence to the rigid motion interpretation (right). The state of the filter at each step is represented as a black '+' and superimposed to the average residual function (darker tones for larger residuals). After the transient, the states accumulate either around the local minimum corresponding to the rubbery interpretation (the one on the bottom half of the plot) or to the one corresponding to the true motion, on the upper half of the plot. The trajectory of the state is also plotted component-wise in Fig. 15.

recursive nature of the scheme, which integrates information over a large baseline. In Figs. 15 and 16 we show a typical example of the temporary convergence of the filter to a local minimum: after few iterations the observations are no longer compatible with the motion interpretation, forcing the filter out of the local minimum.

5.3.5. Rubbery Motion. A qualitatively different local minimum is the one corresponding to the "rubbery motion". When the positive depth constraint is not enforced the filter may converge either to the rigid or to the rubbery interpretation (Fig. 14). In Figs. 15 and 16 (left) we show the convergence to the "rubbery motion interpretation" when the positive depth constraint is released.

In Figs. 15 and 16 (right) we show the convergence of the filter to the rigid interpretation. Note that, when the positive depth constraint is enforced, the estimate is reflected onto the correct rigid interpretation (Fig. 17).

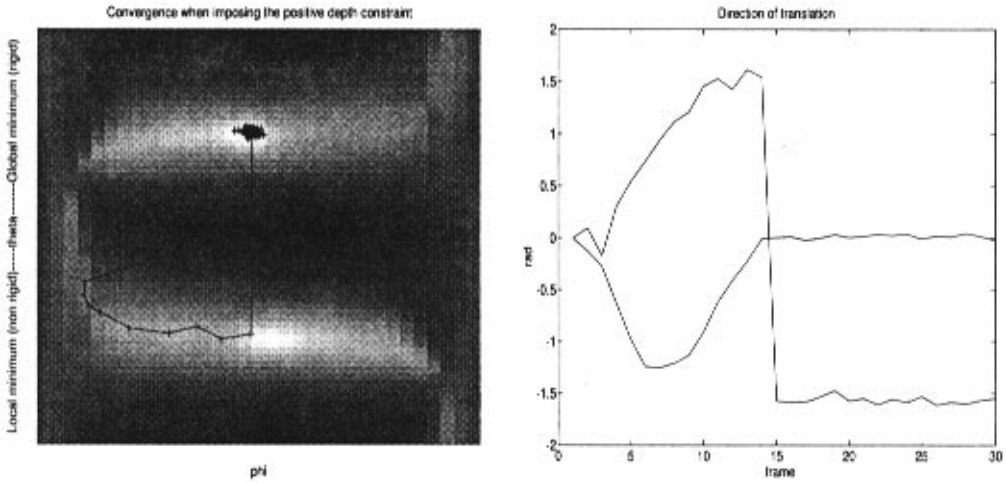

Figure 17. Convergence when the positive depth constraint is enforced: (left) trajectory of the filter on top of the brightness plot of the residual function, (right) corresponding motion components. Initial conditions are zero.

5.3.6. Structure Estimation. When we feed the motion estimates into a structure-from-motion module initialized with points at positive depth and a large model-error variance (Soatto et al., 1993), we may observe either a rigid set of points which move according to the correct motion (a top view of the points is shown in Fig. 18 left) or to a "rubbery" percept (Fig. 18 right). This is in accordance with the experience in psychophysical experiments (Kolb et al., 1994). Note that the rubbery solution disappears as soon as we impose the positive depth constraint.

5.3.7. Comparison with the Essential Filter. The filter proposed in this paper proves significantly less sensitive to noise in the measurements and to the initial conditions than the essential filter (Soatto et al., 1994).

In particular, for 20 observed points and 1 pixel std noise, the essential filter converges for initial conditions within 30% of the correct solution, while the subspace filter converges from any initial condition. Furthermore, the subspace filter is less sensitive to disturbances, and may tolerate up to 5 times more noise on the measured image plane coordinates than the essential filter. This is due to the simple structure of the state-space of the filter as well as its low dimensionality.

Once properly initialized, however, the essential filter proves more accurate, achieving easily less than 1% error in the components of velocity for one pixel

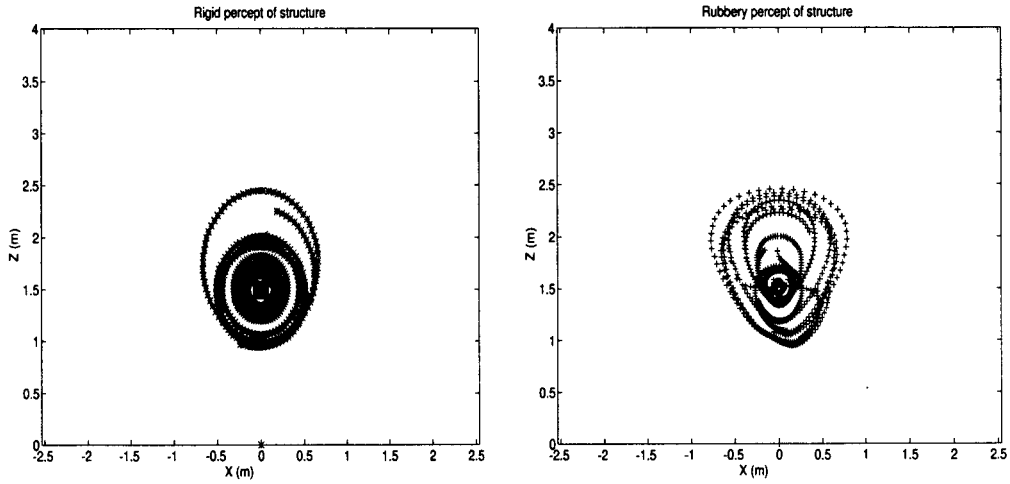

Figure 18. Convergence of a structure-from-motion module to a rigid interpretation of structure (left) or to a rubbery object rotating in the opposite direction (right). The plots show a top view of the points, with the image plane on the lower end.

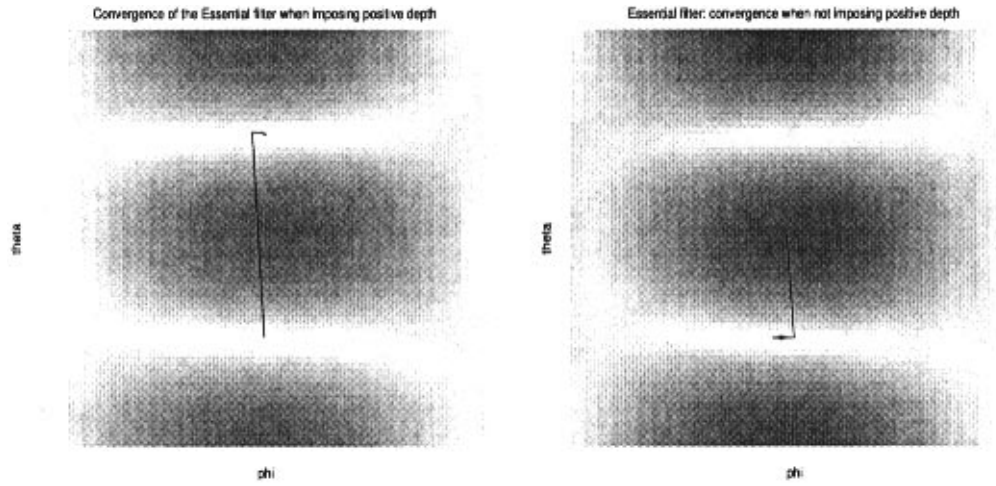

Figure 19. Convergence of the essential filter: the residual function is plotted on a two-dimensional slice of the five-dimensional state space. The remaining states that are not represented (the ones corresponding to the rotational velocity) are set to the ground truth. On the left plot the filter is initialized with a motion close to the minimum corresponding to the rubbery interpretation. The filter, however, imposes automatically the positive depth constraint and the estimate switches fast to the correct motion interpretation. (Right) by releasing the positive depth constraint, it is possible for the filter to converge to the rubbery interpretation. The initial condition is assigned with the rotational velocity corresponding exactly to the rubbery interpretation, and the remaining two states, corresponding to the direction of translation, biased towards the local minimum of the rubbery interpretation.

std error or less, while the subspace filter is more robust but less accurate, achieving accuracies in the order of 2–5% under the same conditions.

The essential filter has, in our current implementation, an advantage in terms of complexity as the number of points increases. In fact the linearization of the measurement equation C in the subspace filter has dimensions $2N \times N+3$, where N is the number of visible feature-points, while in the essential filter it is $2N \times 9$. However, the linearization of the subspace filter has a sparse structure that could in principle be exploited.

In the essential filter the positive depth constraint is encoded directly in the definition of the state-space manifold (the essential manifold). The convergence of the essential filter is illustrated in Fig. 19: on the left the convergence is shown when starting from the rubbery motion interpretation and imposing positive depth. On the right the positive depth constraint has been released (equivalently, reflections are allowed in the essential manifold), and therefore we may observe occasionally convergence to the local minimum corresponding to the rubbery interpretation.

5.4. Experiments with Real Image Sequences

5.4.1. The "Rocket" Scene. As a first example we report here the filter estimates for the rocket scene, for comparison with Soatto et al. (1994). Due to the fact that the filter takes about 10 frames to converge, we have doubled the sequence, which is displayed in Fig. 20. The sequence was provided to us by Oliensis and Thomas, along with approximately 20 point-features tracked through the 11 frames. A qualitative ground truth has also been provided with the original sequence. The results are reported in Fig. 20.

5.4.2. The "Box" Sequence. In a second experiment we consider the motion of a box rotating on top of a chair (see Fig. 21). The box has a side of approximately 25 cm and its centroid is placed at a distance of about 45 cm from the camera. The features are detected and tracked using a multiscale Sum of Square Difference (SSD) method (Lucas and Kanade, 1981). The distance between two features is chosen as reference in order to evaluate the scale factor. In order to get rid of the features belonging to the background, the scene is first segmented using an algorithm described in Soatto and Perona (1994).

The estimates of the direction of translation, with the error-bars corresponding to the variance of the prediction error, are plotted in Fig. 22 (left), and similarly for the rotational velocity, which is estimated using the pseudo-measurements $\tilde{\Omega} = \tilde{C}^{\dagger}_{2N+1:2N+3}\mathbf{y}'$ as input to a linear Kalman filter as described in Section 3.2 (see Fig. 22 right).

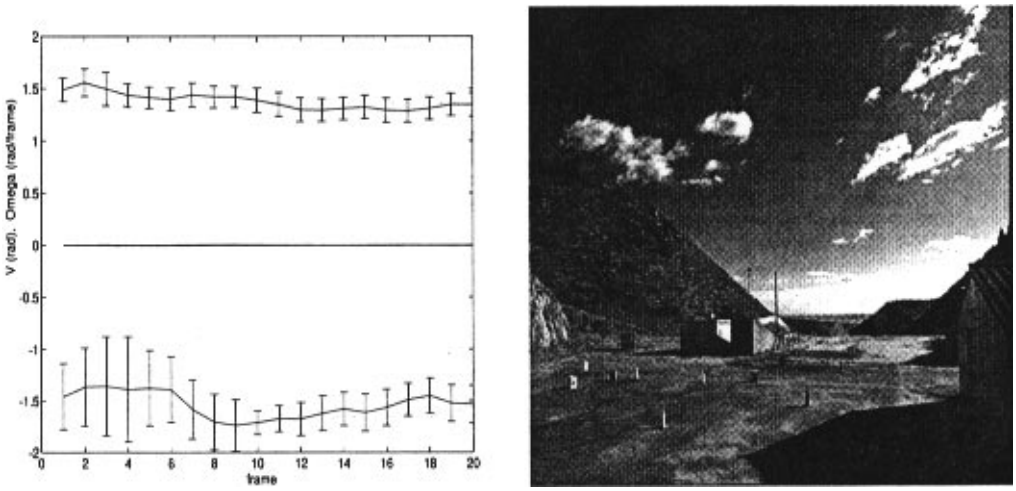

Figure 20. (Left) estimate of the direction of translation for the rocket scene. (Right) one image of the rocket scene. The ground truth is shown in dotted lines, while the filter estimates are in solid lines. The error-bars are three times the variance of the estimation error.

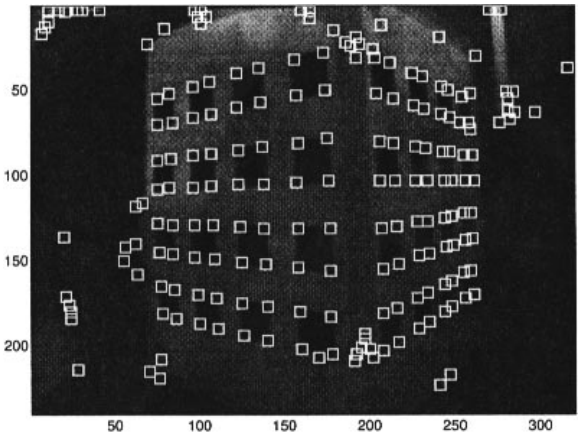

Figure 21. One image of the box sequence. Features (marked as white boxes) are selected using the Sum of Square Difference (SSD) criterion and then clustered according to their rigid motion as estimated between the first two time instants. The distance between two features is chosen as reference in order to update the scale factor.

Once motion is estimated—together with the appropriate variance of the estimation error—it is fed into a "structure-from-motion" module that processes motion error (Soatto et al., 1993) in order to estimate the structure of the scene. A slice of the scene viewed from the top is plotted in Fig. 23 (left), and the corresponding image-plane view is depicted in Fig. 23 (right).

5.4.3. The "Beckman Corridor" Sequence. The complete "Beckman corridor" sequence consists of a sequence of approximately 8000 frames taken by Bouguet et al. inside the corridor of the Beckman Institute at the California Institute of Technology. On the walls sheets of paper with high contrast provide sufficient texture for point-feature tracking. The sequence is taken while the camera moves along the corridor on top of a cart which is hand-pushed following a prescribed path on the floor of the corridor, so that qualitative ground-truth can be reconstructed. The sequence, with the tracking of about 400 feature-points, the same employed in Bouguet and Perona (1995), has been kindly provided to us by J.Y. Bouguet. The features come with a condition number that indicates the presence of sufficient contrast along both spatial directions.

We show here only the first 1800 frames, during which the cart was turning of 90 degrees at a corridor angle, and then following a shallow s-turn. The algorithm makes no assumption about the fact that motion occurs on a plane, so that we can check whether the rotation about the fronto-parallel axis and the cyclo-rotation are estimated as zero, and the elevation angle is constant. Rotation about the vertical axis should integrate at about 90 degrees at the end of the experiment.

We have run our algorithm by using only part of the feature-set. We have fixed the maximum number of features to 20, so that the average number that pass the innovation test described in Section 4.5 is about 15, with a minimum of 3 features at frame 400. The number of features used by the algorithm as a function of the current frame is plotted in Fig. 27. It must be noticed that no particular attention is paid to the location in the

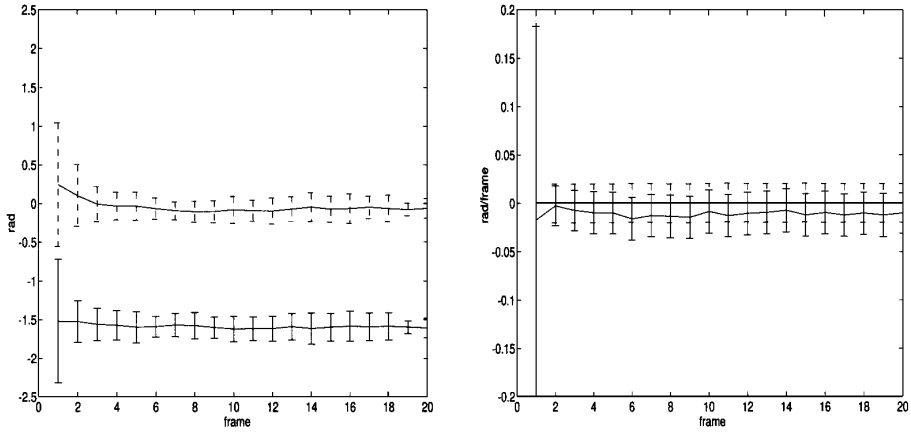

Figure 22. (Left) estimate of the direction of translation for the rotating box. The error-bars are three times the variance of the estimation error (diagonal of the P matrix of the filter). (Right) estimates of the components of rotational velocity, estimated using a linear Kalman filter that processes the pseudo-measurements derived from the direction of translation, as described in Section 3.2.

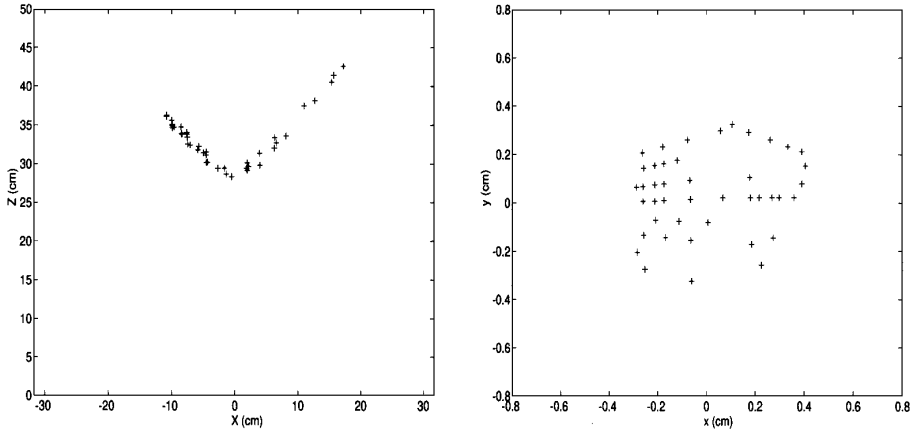

Figure 23. (Left) top view of the estimated scene. Note that some features have been lost during the tracking procedure. The structure was estimated using a simple Extended Kalman Filter having as input the feature points and the motion estimates together with their variance/covariance matrices. (Right) image-plane view of the scene.

image-plane of the features used by the algorithm, so it can happen that at some step the scheme uses few features that cover only a small portion of the visual field.

In Fig. 25 we show the estimated direction of translation, consisting of the azimuth angle (direction of heading) and elevation angle. The latter is constant to about 5 degrees, which corresponds to the angle between the camera and the horizontal axis on the cart. The direction of heading points left during the first turn, then slightly right and then left again during the s-turn. This is consistent with the cart having front steering wheels and the camera being mounted on the front. The rotation angle about the Y-axis (horizontal) and Z-axis (cyclo-rotation) are zero, as reported in Fig. 26. The rotational velocity about the vertical axis-X, reported in Fig. 27, shows first the full left turn, then the s-turn left-right. The integral of the velocity along the whole sequence is 101°, with an overall error of about 10° over 1800 frames. This is the mean integral of the error along the whole sequence. In order to appreciate the convergence of the filter, which was initialized to zero, we show the components of the main filter for the direction of heading, along with the variance of the estimation error—plotted as errorbars—during the first 100 frames (Fig. 28).

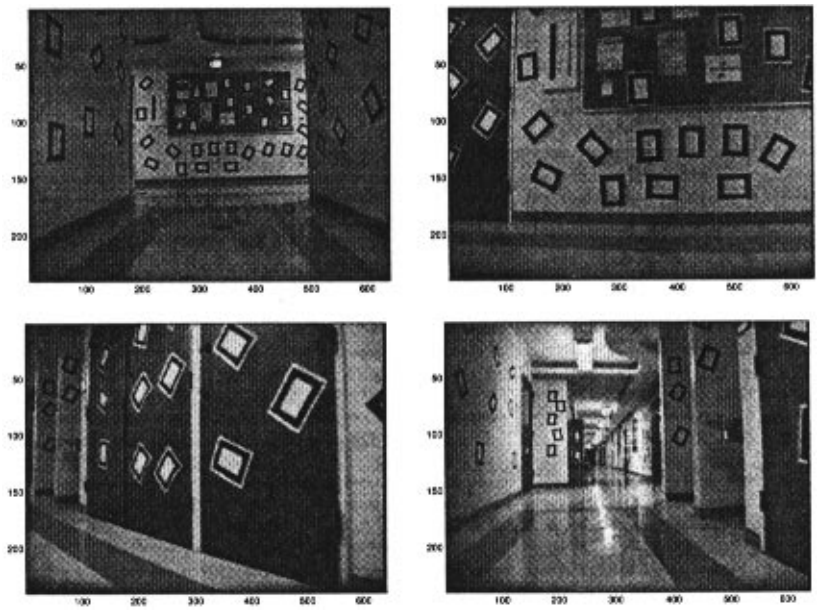

Figure 24. Few images from the "Beckman sequence". The camera is mounted on a cart which is pushed around a corridor. First the cart turns left by 90°, then right and left again on a s-turn. The sequence consists of approximately 8000 frames. We have processed here only the first turn of the corridor, which corresponds to the first 1800 frames. The sequence was taken by Bouguet et al., who also performed the feature tracking using Sum of Square Differences criteria on a multi-scale framework.

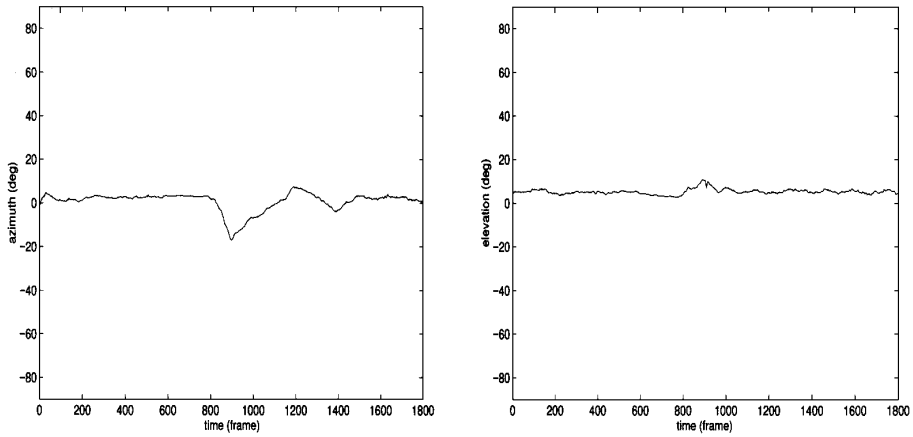

Figure 25. (Left) azimuth angle for the corridor sequence. Zero corresponds to forward translation along the Z-axis. The first peak is due to the left turn, while the subsequent wiggle corresponds to a right-left s-turn. (Right) elevation angle. The camera was pointing downwards at an angle of approximately 5°; therefore the heading direction was approximately constant with an elevation of $+5°$. Since the camera was hand-held, there is quite a bit of wobbling.

6. Conclusions

We have formulated a new recursive scheme for estimating rigid motion under perspective by identifying a nonlinear implicit dynamic model with parameters on a manifold.

The motivation comes from the work of Heeger and Jepson (1992), who first proposed to view motion estimation as an optimization problem constrained on a subspace. Using standard results from nonlinear estimation and identification theory, we formulate a motion estimator which is efficient, accurate and remarkably robust to measurement noise.

One of the crucial features of the scheme is the independence of the motion estimates from the structure of the scene. This allows us to deal with occlusions and

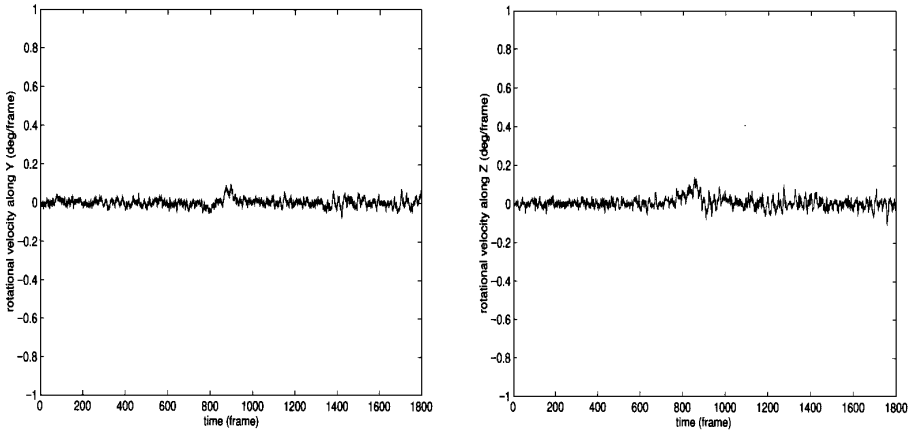

Figure 26. Rotational velocity about the Y-axis (left) and about the Z-axis (right). Since the camera was not pitching nor cyclo-rotating, both estimates are close to zero as expected. Since the camera was hand-held and no accurate ground-truth is available, it is not easy to sort out the effects of noise and the ones of small motions or vibrations of the camera.

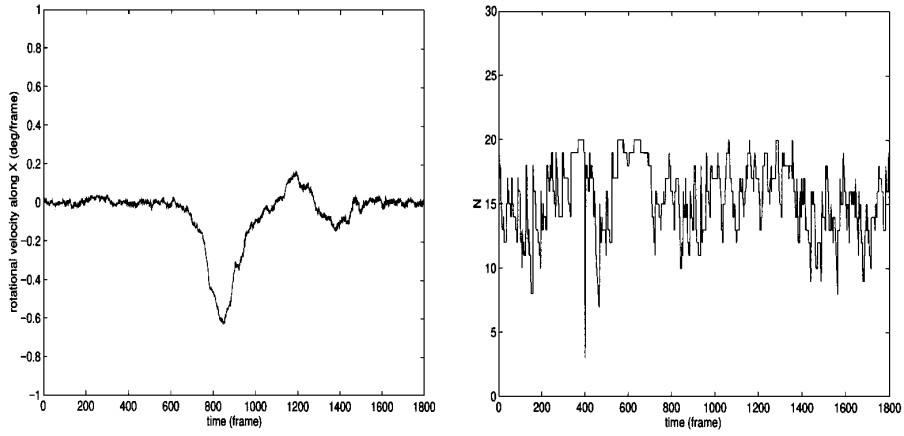

Figure 27. (Left) rotational velocity about the vertical axis. First the camera turns left at the corner of the corridor (frames 700 to 1000), then right and then left again around the s-turn (frames 1000 to 1600). The integral of the rotational velocity should add up to approximately 90°, for this is the change of orientation of the camera from beginning to end. The sum of the estimates is 101°, corresponding to an error of 10% circa on a sequence of 1800 frames. (Right) number of features employed by the algorithm at each time step. On average the algorithm uses 15 feature-points, without particular attention to how they are distributed on the image plane. The maximum number of features used is 20, and the minimum is 3. Note that two-frames algorithms would not perform in such a case, since at least 5 features need to be visible at all times. The temporal integration involved in the filter, on the contrary, allows us to retain the estimates even in presence of less than 5 features.

appearance of new features in a principled way, and results in a filter with a small, constant-dimensional and highly-constrained state-space. While structure is not represented explicitly in the state, the innovation process of the filter describes how each single feature-point is compatible with the current motion interpretation, and may therefore be used for detecting outlier measurements, making the filter robust to error in feature tracking/optical flow. The filter has proven robust to tuning parameters, and needs no ad-hoc adjustments depending upon the experiment. Convergence is reached in fractions of a second of video-rate from arbitrary initial conditions. This, together with the light computational load required, makes our approach suitable for real-time processing on the current generation of PC microprocessors, once optical flow or feature tracking is provided. Extensive experiments have been performed that highlight such features.

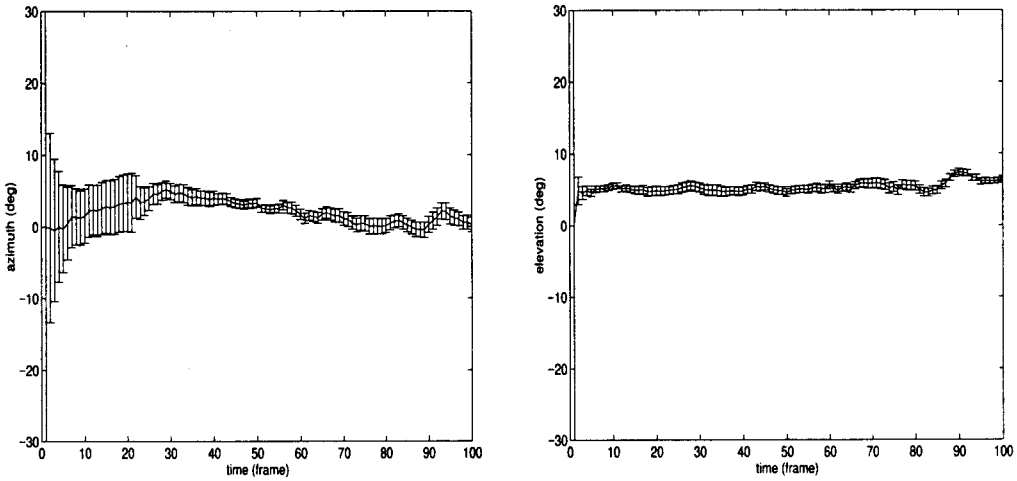

Figure 28. Close-up view of the transient in the estimates of the direction of translation (azimuth on the left, elevation on the right). The variance of the estimation error, represented using the error-bars, decreases during the first 20–30 frames, after which it remains bounded around the current estimate of the parameter.

Appendix A: Computation of the Local Linearization of the Model

In this appendix we give the detailed equations for the linearization of the model of the subspace filter. We compute the derivative of the implicit measurement equation

$$\tilde{\mathcal{C}}^{\perp}(\mathbf{x}, V(\theta,\phi))\dot{\mathbf{x}} \tag{19}$$

as a function of the derivative of $\tilde{\mathcal{C}}$ with respect to the states θ, ϕ and the measurements $\mathbf{x}$. From the definition of $\tilde{\mathcal{C}}^{\perp}$ we have

$$\tilde{\mathcal{C}}^{\perp} \doteq (I - \tilde{\mathcal{C}}(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\tilde{\mathcal{C}}^T) \tag{20}$$

If we call α a scalar parameter (α will be either $\phi(t), \theta(t)$ or one component of the measurements $x^i(t), y^i(t)$) and

$$\tilde{\mathcal{C}}_\alpha \doteq \frac{\partial\tilde{\mathcal{C}}}{\partial\alpha} \tag{21}$$

then we have

$$\begin{aligned}\tilde{\mathcal{C}}_\alpha^{\perp} = &-\tilde{\mathcal{C}}_\alpha(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\tilde{\mathcal{C}}^T - \tilde{\mathcal{C}}(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\tilde{\mathcal{C}}_\alpha^T \\ &- \tilde{\mathcal{C}}\frac{\partial(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}}{\partial\alpha}\tilde{\mathcal{C}}^T.\end{aligned} \tag{22}$$

Since, for a square and invertible matrix A, $A_\alpha^{-1} = -A^{-1}A_\alpha A^{-1}$, we have

$$\begin{aligned}\tilde{\mathcal{C}}_\alpha^{\perp} = &-\tilde{\mathcal{C}}_\alpha(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\tilde{\mathcal{C}}^T - \tilde{\mathcal{C}}(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\tilde{\mathcal{C}}_\alpha^T \\ &- \tilde{\mathcal{C}}(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\big(\tilde{\mathcal{C}}_\alpha^T\tilde{\mathcal{C}} + \tilde{\mathcal{C}}^T\tilde{\mathcal{C}}_\alpha\big)(\tilde{\mathcal{C}}^T\tilde{\mathcal{C}})^{-1}\tilde{\mathcal{C}}^T\end{aligned} \tag{23}$$

we can write, after collecting the common terms,

$$\tilde{\mathcal{C}}_\alpha^{\perp} = -\tilde{\mathcal{C}}^{\perp}\tilde{\mathcal{C}}_\alpha\tilde{\mathcal{C}}^{\dagger} - \tilde{\mathcal{C}}^{\dagger^T}\tilde{\mathcal{C}}_\alpha^T\tilde{\mathcal{C}}^{\perp}. \tag{24}$$

If we call

$$\mathcal{K}_\alpha \doteq \tilde{\mathcal{C}}^{\perp}\tilde{\mathcal{C}}_\alpha\tilde{\mathcal{C}}^{\dagger} \tag{25}$$

and we notice that $\tilde{\mathcal{C}}^{\perp}$ is a symmetric matrix, we end up finally with

$$\tilde{\mathcal{C}}_\alpha^{\perp} = -\mathcal{K}_\alpha - \mathcal{K}_\alpha^T. \tag{26}$$

We now seek for a cheaper and better-conditioned way of computing the matrix $\mathcal{K}$. Consider the Singular Value Decomposition of the matrix $\tilde{\mathcal{C}}$:

$$\tilde{\mathcal{C}} = U_c\Sigma_c V_c^T \tag{27}$$

then it is immediate to notice that

$$\tilde{\mathcal{C}}^{\perp} = I - U_c U_c^T. \tag{28}$$

After substituting for the SVD of $\tilde{\mathcal{C}}$ and exploiting the orthogonality of U and V, we have

$$\mathcal{K}_\alpha = \big(I - U_c U_c^T\big)\tilde{\mathcal{C}}_\alpha V_c\Sigma_c^{-1}U_c^T. \tag{29}$$

In order to compute the full linearization of the implicit measurement equation with respect to the states θ, ϕ and the measurements $\mathbf{x}$, we are only left with computing the derivatives of the matrix $\tilde{\mathcal{C}}$ with respect to these parameters:

$$\tilde{\mathcal{C}}_\theta = \begin{bmatrix} \mathcal{A}_1 \frac{\partial V}{\partial \theta} & & & 0 \\ & \ddots & & 0 \\ & & \mathcal{A}_N \frac{\partial V}{\partial \theta} & 0 \end{bmatrix} \tag{30}$$

$$\tilde{\mathcal{C}}_\phi = \begin{bmatrix} \mathcal{A}_1 \frac{\partial V}{\partial \phi} & & & 0 \\ & \ddots & & 0 \\ & & \mathcal{A}_N \frac{\partial V}{\partial \phi} & 0 \end{bmatrix} \tag{31}$$

$$\tilde{\mathcal{C}}_{x^i} = \begin{bmatrix} \ddots & & & 0 \\ & \begin{bmatrix} V_3 \\ 0 \end{bmatrix} & & \frac{\partial \mathcal{B}_i}{\partial x^i} \\ & & \ddots & 0 \end{bmatrix} \tag{32}$$

$$\tilde{\mathcal{C}}_{y^i} = \begin{bmatrix} \ddots & & & 0 \\ & \begin{bmatrix} 0 \\ V_3 \end{bmatrix} & & \frac{\partial \mathcal{B}_i}{\partial y^i} \\ & & \ddots & 0 \end{bmatrix} \tag{33}$$

where

$$\frac{\partial V}{\partial \theta} = \begin{bmatrix} -\cos(\phi)\sin(\theta) \\ \cos(\phi)\cos(\theta) \\ 0 \end{bmatrix} \tag{34}$$

$$\frac{\partial V}{\partial \phi} = \begin{bmatrix} -\sin(\phi)\cos(\theta) \\ -\sin(\phi)\sin(\theta) \\ \cos(\phi) \end{bmatrix}. \tag{35}$$

The spherical coordinates are defined such that

$$V(\theta, \phi) \doteq \begin{bmatrix} \cos(\theta)\cos(\phi) \\ \sin(\theta)\cos(\phi) \\ \sin(\phi) \end{bmatrix}. \tag{36}$$

We now have all the ingredients necessary for computing the linearization of the model:

$$C \doteq \left(\frac{\partial \tilde{\mathcal{C}}^\perp \dot{\mathbf{x}}}{\partial [\theta \ \phi]} \right) = \left[\tilde{\mathcal{C}}_\theta^\perp \dot{\mathbf{x}} \quad \tilde{\mathcal{C}}_\phi^\perp \dot{\mathbf{x}} \right] \tag{37}$$

$$D \doteq \left(\frac{\partial \tilde{\mathcal{C}}^\perp \dot{\mathbf{x}}}{\partial [\mathbf{x}\ \dot{\mathbf{x}}]} \right) = \left[\tilde{\mathcal{C}}_{x^1}^\perp \dot{\mathbf{x}} \quad \tilde{\mathcal{C}}_{y^1}^\perp \dot{\mathbf{x}} \quad \cdots \quad \tilde{\mathcal{C}}_{y^N}^\perp \dot{\mathbf{x}} \,|\, \tilde{\mathcal{C}}^\perp \right]. \tag{38}$$

Acknowledgments

This research has been funded by the California Institute of Technology, a scholarship from the University of Padova, a fellowship from the "A. Gini" Foundation, the Center for Neuromorphic Engineering as a part of the NSF ERC program and the NSF NYI Award (P. P.). We wish to thank J.-Y. Bouguet for providing us with the Beckman sequence, and J. Oliensis and I. Thomas for the Rocket sequence.

Notes

1. For in introductory treatment on nonlinear observability and its effects on state observation, see Isidori (1989).
2. See McLauchlan (1994) for a way of dealing with a variable state-dimension model.
3. An instance of a spherical coordinate chart is reported in Appendix A.
4. It should be noted that $\bar{n}$ is *not* a white noise, for n and n' are effectively correlated. A technique for fixing this inconvenient is described in Soatto et al. (1996). However, we find that the performance achieved by approximating $\bar{n}$ with a white noise is satisfactory in most cases.

References

Adiv, G. 1985. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Anal. Mach. Intell*.

Azarbayejani, A., Horowitz, B., and Pentland, A. 1993. Recursive estimation of structure and motion using relative orientation constraints. *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, New York.

Bouguet, J.-Y. and Perona, P. 1995. A visual odometer and gyroscope. *Proc. of the 5 IEEE Int. Conf. Comp. Vision*.

Broida, T. and Chellappa, R. 1986. Estimation of object motion parameters from noisy images. *IEEE Trans. PAMI*.

Bryant, R.L., Chern, S.S., Gardner, R.B., Goldshmidt, H.L., and Griffith, P.A. 1991. *Exterior Differential Systems*. Mathematical Research Institute. Springer Verlag.

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

Dickmanns, E.D. 1994. Historical development of use of dynamical models for the representation of knowledge about real-world processes in machine vision. *Signal Processing*, 35(3):305–306.

Gennery, D.B. 1982. Tracking known 3-dimensional object. In *Proc. AAAI 2nd Natl. Conf. Artif. Intell.*, Pittsburg, PA, pp. 13–17.

Gennery, D.B. 1992. Visual tracking of known 3-dimensional object. *Int. J. of Computer Vision*, 7(3):243–270.

Gibson, E.J., Gibson, J.J., Smith, O.W., and Flock, H. 1959. Motion parallax as a determinant of perceived depth. *J. Exp. Psych.*, Vol. 45.

Heeger, D. and Jepson, A. 1992. Subspace methods for recovering rigid motion i: Algorithm and implementation. *Int. J. Computer Vision*, 7(2).

Heel, J. 1990. Direct estimation of structure and motion from multiple frames. AI Memo 1190, MIT AI Lab.

Horn, B.K.P. 1990. Relative orientation. *Int. J. of Computer Vision*, 4:59–78.

Isidori, A. 1989. *Nonlinear Control Systems*. Springer Verlag.

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

Jepson, A. and Heeger, D. 1991. Subspace methods for recovering rigid motion ii: Theory. RBCV TR-90-35, University of Toronto.

Kalman, R.E. 1960. A new approach to linear filtering and prediction problems. *Trans. of the ASME-Journal of Basic Engineering*, pp. 35–45.

Kolb, C., Braun, J., and Perona, P. 1994. Object segmentation and 3d structure from motion. In *Invest. Ophtalmol. Vis. Sci.* (Supplement), p. 1275.

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

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

Matthies, L., Szeliski, R., and Kanade, T. 1989. Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. of Computer Vision*.

McLauchlan, P., Reid, I., and Murray, D. 1994. Recursive affine structure and motion from image sequences. *Proc. of the 3 ECCV*.

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

Soatto, S. 1997. 3-D Structure from visual motion: modeling, representation and observability. *Automatica* (in press).

Soatto, S., Perona, P., Frezza, R., and Picci, G. 1993. Recursive motion and structure estimation with complete error characterization. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR*, New York, pp. 428–433.

Soatto, S. and Perona, P. 1994. Three dimensional transparent structure segmentation and multiple 3d motion estimation from monocular perspective image sequences. In *IEEE Workshop on Motion of Nonrigid and Articulated Objects*, Austin, IEEE Computer Society, pp. 228–235.

Soatto, S., Frezza, R., and Perona, P. 1994. Motion estimation on the essential manifold. In *Proc. 3rd Europ. Conf. Comput. Vision*, J.-O. Eklundh (Ed.), *LNCS-Series*, Springer-Verlag: Stockholm, 800–801:II-61–II-72.

Soatto, S., Frezza, R., and Perona, P. 1995. Structure from visual motion as a nonlinear observation problem. In *Proceedings of the IFAC Symposium on Nonlinear Control Systems NOLCOS*, Tahoe City.

Soatto, S., Frezza, R., and Perona, P. 1996. Motion estimation via dynamic vision. *IEEE Trans. on Automatic Control*, 41(3):393–414.

Soderstorm, T. and Stoica, P. 1989. *System Identification*. Prentice Hall.

Von Helmholtz, H. 1910. *Treatise on Physiological Optics*.

Young, G.S. and Chellappa, R. 1990. 3-d motion estimation using a sequence of noisy stereo images-models, estimation and uniqueness results. *IEEE Trans. Pattern Anal. Mach. Intell.*