Reducing "Structure From Motion": a General Framework for Dynamic Vision Part 2: Implementation and Experimental Assessment

Stefano Soatto and *Pietro Perona*

Abstract

A number of methods have been proposed in the literature for estimating scene-structure and ego-motion from a sequence of images using dynamical models. Despite the fact that all methods may be derived from a "natural" dynamical model within a unified framework, from an engineering perspective there are a number of trade-offs that lead to different strategies depending upon the applications and the goals one is targeting. We want to characterize and compare the properties of each model such that the engineer may choose the one best suited to the specific application. We analyze the properties of filters derived from each dynamical model under a variety of experimental conditions, assess the accuracy of the estimates, their robustness to measurement noise, sensitivity to initial conditions and visual angle, effects of the bas-relief ambiguity and occlusions, dependence upon the number of image measurements and their sampling rate.

1 Introduction

"Structure From Motion" (SFM) is concerned with estimating the 3-D motion and structure of a rigid object from a sequence of monocular images. SFM has been a central problem in computer vision over the past decade, and the literature comprises a variety of schemes that differ for the description of structure employed (point-features, lines, curves, surfaces, partial models of the environment), for the projection model (orthographic, affine, perspective), input measurements (optical flow, feature tracking, image brightness, occluding contours), time-frame (continuous-time or discrete-time models) and data processing technique (batch optimization, recursive estimation).

Particular choices may be forced by the specific circumstances. For instance, if one can afford the memory space to store a whole sequence of images and the computational power to process it at once, it is most advisable to employ a *batch* estimation technique. If, on the other hand, the estimates of 3-D motion and/or structure are to be used for performing some control action, such as moving a robot or driving a vehicle, the visual information must be processed in a *causal* fashion and in *real-time*. In such a case a *recursive* estimation technique is most appropriate, since only the current image is processed in order to update the estimates in an incremental fashion. Also, if the sequence has been taken while fixating some feature on the image, *image-feature tracking* is greatly facilitated, for single features are visible over a long interval of time. However, if the sequence is taken, say, from a car, then the most informative area is the periphery of the image, where features move quickly out of the visual field. In such a case it is impossible to track features over an extended period of time, and therefore it may be necessary to use *optical flow*. Moreover, if the scene is viewed under a wide angle (as for instance in autonomous navigation), then perspective projection is the most appropriate model. If, however, the scene consists of a single object that covers a small portion of the visual field, one may consider simpler projection models to approximate the imaging geometry, such as affine or orthographic projection.

In order to make a rational choice of the best algorithm for a given task, it is also vital to assess comparatively the performance of all models that results from different choices. Which choice results in the most accurate scheme? Which one is the most robust to measurement noise? Which one is the least sensitive to the aperture angle? or to the bas-relief ambiguity? Does fixating some particular feature in the scene make the problem better constrained or simpler to solve? Often times, depending upon the task, one is interested only in *part* of the unknown parameters. For instance, in photogrammetry one is more interested in the *structure* of the environment, regardless the motion undergone by the viewer. In navigation or motion control applications, one cares about the *motion* of the viewer, while the structure of the scene is relatively less important. In satellite landing, the main interest is in the direction of heading, since the attitude is controlled independently. In such cases, is it better to estimate *all unknowns* together (for instance structure *and* motion), or is it more appropriate to try to estimate only the parameters of interest *independent* of the other unknowns (for instance the direction of heading alone)?

Even if we restrict our attention to sequences of *perspective* images of *point-features* processed in a *causal* fashion, there are still quite a few schemes available in the literature for estimating structure for known motion [16], motion from known structure [6, 9] or both structure and motion simultaneously [1, 4, 5, 7, 11, 17, 19, 29, 30]. More recently, recursive schemes have been proposed for estimating motion *independent of structure* [25] or structure independent of motion [24].

Soatto and Perona [28] have proposed a framework that unifies all geometric models for estimating structure and/or motion from sequences of images. In order to achieve a fair evaluation of the *geometric properties* of each model it is necessary to employ *the same estimation technique* and the same dynamics for the unknown parameters. Therefore, we will design a *filter* for each one of the models using a technique introduced in [25], which essentially resorts to local implicit Extended Kalman Filtering (EKF [12]) (section 2). In each instance the filter remains the same (and therefore its tuning), and only the space of unknown parameters changes.

Once we have designed a filter for each of the representative classes of models, we need to design experiments which are "sufficiently informative". The experimental conditions depend upon a number of parameters that describe the type of 3-D structure, the aperture angle under which it is viewed, the type of motion the camera (or the scene) is undergoing, the sampling frequency of the measurements, the number of visible features, the noise levels, the initial conditions and the tuning parameters for the estimators.

We will choose and motivate one particular paradigm experiment, and then vary systematically all relevant parameters (section 3).

2 A framework for estimating structure and/or motion

In this section we are going to review a method for obtaining a recursive estimator of motion and/or structure for all possible dynamic models derived from the constraints of rigidity and perspective. First (section 2.1) we summarize the results of a companion paper [28], where we derive all models from the basic constraints following the idea of model reduction for dynamical systems. Then (section 2.2) we show how to transform the parameter identification task into a standard form suitable for using an Extended Kalman Filter [25]. In section 2.4 we describe how to actually realize a filter for each of the models described in section 2.1, and we discuss some issues on the implementation.

2.1 Modeling "Structure From Motion"

In a companion paper [28] we have seen how different models for estimating motion from sequences images can be cast within the same framework. We have started from the model that is "defined" by the rigidity constraint and the perspective projection, either in a continuous-time or in a discrete-time fashion:

$$\begin{cases} \mathbf{X}^i(t+1) = R(t)\mathbf{X}^i(t) + T(t) \\ \mathbf{y}^i(t) = \pi(\mathbf{X}^i(t)) + n^i(t) \end{cases} \qquad \begin{cases} \dot{\mathbf{X}}^i = \Omega \wedge \mathbf{X}^i + V \\ \mathbf{y}^i = \pi(\mathbf{X}^i) + n^i \end{cases} \qquad \forall\, i = 1 \dots N \tag{1}$$

where the *states* $\mathbf{X}^i = [X^i \; Y^i \; Z^i]^T \in \mathbb{R}^3$ are the 3-D coordinates of each of the N feature-points in the scene relative to the viewer's moving frame, $\mathbf{x} \doteq \pi(\mathbf{X}) \doteq [\frac{X}{Z} \; \frac{Y}{Z} \; 1]^T \in \mathbb{RP}^2$ represents an ideal perspective projection (pinhole), and $n^i \in \mathcal{N}(0, R^i)$ is a white, zero-mean and Gaussian measurement noise. The 3×3 rotation matrix $R(t)$ describes the change of coordinates of the viewer's moving frame between time $t+1$ and time t, and is orthonormal with positive determinant. When the rotational velocity Ω is held constant between time samples, R is related to Ω via the exponential map: [1] $R = e^{\Omega\wedge}$. Therefore, a rotation matrix has only 3 degrees of freedom, encoded in the three-dimensional rotation vector Ω. T is a 3-dimensional vector that describes the translation of the origin of the moving frame.

It is possible to integrate the above models from the initial time-instant, and end up with an "integral" model of the form

$$\mathbf{X}^i(t) = {}^tR_{t_0}\mathbf{X}^i(t_0) + {}^tT_{t_0} \qquad \mathbf{X}^i(t_0) = \mathbf{X}^i_0 \tag{2}$$

where the coordinates of each point relative to the initial time-frame are constant and unknown $\mathbf{X}^i_0 = \mathrm{const}$, and the current configuration is described by the unknown translation ${}^tT_{t_0}$ and rotation ${}^tR_{t_0}$, relative to the initial time instant.

[1] The notation $\Omega\wedge$ stands for the operator that performs the vector product on $\mathbb{R}^3$: $(\Omega\wedge)\mathbf{X} \doteq \Omega\wedge\mathbf{X}\; \forall\, \mathbf{X} \in \mathbb{R}^3$. In coordinates $\Omega\wedge \doteq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$. Alternative (local) representations for rotation matrices include various types of "Euler angles"; global (embedded) representations can be obtained through unit quaternions [18].

We have then *dynamically extended* the models above in order to include all unknown parameters T, R or V, Ω in the state-space. In order to do so, one needs to know how such parameters evolve in time. In the absence of any dynamical model, one may assume that they evolve statistically according to a random walk of some order[2]. In the case of a discrete-time first-order random walk, one ends up with the extended model

$$\begin{cases} \mathbf{X}^i(t+1) = R(t)\mathbf{X}^i(t) + T(t) \\ T(t+1) = T(t) + n_T(t) \\ R(t+1) = R(t)e^{n_R\wedge(t)} \\ \mathbf{y}^i(t) = \pi(\mathbf{X}^i(t)) + n^i(t) \end{cases} \qquad \forall\ i\ =\ 1 \ldots N(t) \tag{3}$$

where $n_T \in \mathbb{R}^3, n_R \in \mathbb{R}^3$ are well as $n^i \in \mathbb{RP}^2$ are white, zero-mean Gaussian processes. We have applied the idea of the "reduced-order observer" [13] in order to reduce the dimension of the state by the number of the measurements, and be left with one state for each visible point, which encodes its depth in the moving frame. Depending on whether we use a first-order model or an integral (sencond-order) model, we have

$$\begin{cases} Z^i(t+1) = R_{3.}(t)Z^i(t) + T_3(t) \\ T(t+1) = T(t) + n_T(t) \\ R(t+1) = R(t)e^{n_R\wedge(t)} \\ \mathbf{y}^i(t) = \pi(Z^i(t)\mathbf{y}^i(t)) + n^i(t) \end{cases} \quad \text{or} \quad \begin{cases} Z_0^i(t+1) = Z_0^i(t) \\ \Omega(t+1) = \Omega(t) + n_\Omega(t) \\ V(t+1) = V(t) + n_V(t) \\ R(t+1) = e^{\Omega(t)\wedge}R(t) \\ T(t+1) = e^{\Omega(t)\wedge}T(t) + V(t) \\ \mathbf{y}^i(t) = \pi\left(R(t)\mathbf{y}^i(t_0)Z_0^i(t) + T(t)\right) + n^i(t) \end{cases} \tag{4}$$

where Z_0^i is the depth of each point at the initial time 0, which is obviously constant. Then we have pushed the idea of the reduced-order observer in order to decouple structure from the motion parameters, and we have applied *"output stabilization"* in order to further decouple rotation from translation in the discrete-time case. In all instances we have ended up with implicit dynamical models in the form

$$\begin{cases} h(\mathbf{x}^i, \phi)\dot{\mathbf{x}}^i = 0 \\ \mathbf{y}^i = \mathbf{x}^i + n^i \quad \forall\ i = 1 \ldots N \end{cases} \qquad \phi \in M \tag{5}$$

where ϕ are unknown parameters constrained to belong to the set M. In the discrete-time case we end up with a similar form where $\mathbf{x}(t+1)$ replaces $\dot{\mathbf{x}}$. By simply changing the set M we may obtain all the different reduced models. Some relevant instances are:

[2]The choice of a random walk is made for shear engineering convenience, for it results in a model which is suitable for "recipe-design" of an Extended Kalman Filter.

Essential model: it is the well-known coplanarity constraint introduced by Longuet-Higgins [14], interpreted as a discrete-time implicit dynamical system. The unknown motion parameters T and R are encoded into a 3×3 "essential matrix" $\mathbf{Q}$, which belongs to the space of matrices of the form $(T\wedge)R$. Such a space $\mathbf{E}$ is called "essential manifold". The function h is simply $h(\mathbf{x}, \mathbf{Q}) \doteq \mathbf{x}^T\mathbf{Q}$.

Subspace model: it consists of the subspace constraint introduced by Heeger and Jepson [10], interpreted as a *dynamical system*, rather than as an algebraic constraint. $\phi = V$ is the direction of heading, which is a three-dimensional vector with unit norm. The space of unknown parameters is the sphere of all possible directions of translation: $M = \mathbf{S}^2$. The function h is the orthogonal complement of the range space of a matrix $\mathcal{C}(\mathbf{x}, V)$ of coefficients of the 2-D motion field equation, which depends upon the image projection of each feature point $\mathbf{x}^i \doteq \pi(\mathbf{X}^i)$ and the direction of heading V (see [27] and equation (15) of [28]).

Point-fixation model: it arises when the sequence of images is taken while fixating some particular feature-point on the image plane [8]. Such a fixation constraint may be specified simply by considering essential matrices of the form $\mathbf{Q} = RS^T + vSR$ with $v \in \mathbb{R}_+$ the velocity along the fixation axis and $S \doteq [0\ 0\ 1]\wedge$. The model h remains the same as in the essential model.

Point-plus-line fixation model: if, in addition to fixating a point, we impose that another point passes through a given line, we further restrict the parameters to be of the form $\mathbf{Q} = RS^T + vSR, R = e^{[\omega_1\ \omega_2\ 0]\wedge}, v \in \mathbb{R}_+, S = [0\ 0\ 1]\wedge$.

Plane-plus-parallax model: it describes the residual motion after the image has been warped as to compensate for the motion of a plane [3]. We can impose the plane-fixation constraint simply by restricting the parameters of the essential model to unit-norm 3×3 matrices of the form $\phi = T\wedge$, and the parameter space is the two-dimensional unit sphere, as in the subspace model: $T \in M = \mathbf{S}^2$.

For details on the derivation of such models the reader is referred to the companion paper [28]. Here we just notice that all of these models are in the form (5), and obtained from (4-left) through *model reduction*. In each instance, the motion parameters may be estimated by *identifying* the unknown parameters of the corresponding model.

2.2 Formulating the estimation task for the extended models

In the extended models (4) derived from the basic constraints of rigidity and perspective, all unknown parameters are *state variables* of the model. Such states evolve in a space that is *not a linear space*. For instance, rotation matrices do not sum up to produce another rotation matrix, and so for unit-norm vectors. Rotation vectors and spherical coordinates are an instance of a system of *local coordinates* on a curved space (such as the set of rotation matrices or the unit-sphere).

The first step in order to make the model (4) suitable for designing an EKF that estimates the state from the measurements is to transform the model into local coordinates: to this end we substitute to R its local-coordinate correspondent rotation vector $\Omega_R \in \mathbb{R}^3$, such that [3] $R = e^{\Omega_R \wedge}$. The state of the model becomes $\xi \doteq [\dots\ Z^i\ \dots\ T\ ,\ \Omega_R] \in \mathbb{R}^{N+6}$. We have already assumed that the measurement noise n^i is white, zero-mean and Gaussian and that the motion parameters are described by a random walk, so that the model in local coordinates is driven by a white, zero-mean Gaussian process. In order to avoid *saturation* of the filter (see section 2.4), we add a Gaussian noise n_{Z^i} with a small variance also to the first N components of the state model:

$$Z^i(t+1) = R_{3.}(t)Z^i(t) + T_3(t) + n_{Z^i}(t). \tag{6}$$

We can proceed in a similar way for the "integral" model (4-right), whose state-space is transformed into local coordinates using the exponential map: A small residual noise is added to all components of the state model in order to prevent saturation:

$$\begin{cases} Z_0^i(t+1) = Z_0^i(t) + n_{Z^i}(t) \\ \Omega(t+1) = \Omega(t) + n_\Omega(t) \\ V(t+1) = V(t) + n_V(t) \\ \Omega_R(t+1) = Log_{SE(3)}\left(e^{\Omega(t)\wedge} e^{\Omega_R(t)\wedge}\right) + n_{\Omega_R}(t) \\ T(t+1) = e^{\Omega(t)\wedge}T(t) + V(t) + n_T(t) \\ \mathbf{y}^i(t) = \pi\left(R(t)\mathbf{y}^i(t_0)Z_0^i(t) + T(t)\right) + n^i(t) + n_y^i \end{cases} \tag{7}$$

[3]Note that Ω_R is just an alternative way of representing R and is different from Ω, which represents the instantaneous rotational velocity of the viewer moving frame.

where the last error term in the measurement equation takes into account the error in measuring the coordinates of the projections at the initial time instant $\mathbf{y}^i(t_0)$. The function $Log_{SE(3)}$ indicates the (local) inverse function of the exponential map $R = e^{\Omega_R \wedge}$ (see [18] for details [4]). The variance of the measurement error, Σ_n and Σ_{n_y} can be inferred from the properties of the optical flow/feature tracking algorithm [2]. The variance of the noises that drive the random walk model, Σ_*, with $* = n_{Z^i}, n_\Omega, n_V, n_{\Omega_R}, n_T$ are *tuning parameters*, and must be assigned by the engineer according to some criteria which we will discuss in section 2.4.

The models in eq. (4), modified according to (6) and (7) respectively, are of the general form

$$\begin{cases} \xi(t+1) = f(\xi(t)) + n_\xi(t) \\ \mathbf{y}^i(t) = g(\xi(t)) + n_{\mathbf{y}^i}(t) \quad \forall\ i = 1 \dots N \end{cases} \tag{8}$$

where f and g are locally smooth functions and the unknown parameters are encoded into the state ξ that belongs to the linear space $\mathbb{R}^{N+6}$ for (4-left) or $\mathbb{R}^{N+6+6}$ for (7). Such models are in a form suitable for applying an Extended Kalman Filter, whose equations can be derived from any standard textbook on stochastic filtering, for instance [12]. The only caveat is the *scale factor* ambiguity, which we discuss in section 2.6.

2.3 Formulating the estimation task for the reduced models

The reduced models (5), unlike the extended ones just discussed, are not yet in a form like (8) suitable for applying an EKF. In the remainder of this section we are going to outline a method for performing the identification of the class of models (5), which is essentially derived from [25].

The first step consists in transforming the identification task into a state-estimation task; this is done by postulating some dynamics for the unknown parameters ϕ. In the case when the camera is mounted on a vehicle, or on a robotic arm, we have some dynamic constraints that govern its motion, typically in the form $\phi(t+1) = f(\phi(t), n_\phi(t))$, where f is some smooth function and n_ϕ some unknown input. In the most conservative approach, we may assume that there are some bounds on the acceleration, due to the fact that the relative motion between the camera and the scene is somewhat smooth, so we may write $f(\phi(t), n_\phi(t)) = \phi(t) \oplus n_\phi(t)$

[4] A Matlab routine to compute the exponential map and its inverse can be retrieved via anonymous ftp from helper.caltech.edu under `/pub/matlab/vision/rodrigues.m`.

with the constraint that $n_\phi(t)$ is (unknown but) small in some norm. We will explain shortly the meaning of the symbol $\oplus$. If a camera is hand-held, or if there is no information on the device that produced the sequence, then we may want to assume a statistical model for the motion parameters, for instance a random walk. The simplest instance of a random walk is a Brownian motion (first order), where $f(\phi(t), n_\phi(t)) = \phi(t) \oplus n_\phi(t)$ with n_ϕ a white, zero-mean Gaussian process. The choice of the dynamics of the parameters is part of the design process and depends upon the specific application one is targeting. Here we will restrict to first-order random walks just because they are the simplest models and flexible enough to deal with most situations we have encountered:

$$\phi(t+1) = \phi(t) \oplus n_\phi(t) \qquad\qquad \phi(t_0) = \phi_0 \tag{9}$$

where $n_\phi \in \mathcal{N}(0, \Sigma_n)$. The reader may now wonder what we mean with the symbol $\oplus$. Since the parameters ϕ do not lie on a linear vector space, we cannot simply sum two elements and hope to obtain a point on M. If we want to induce a sum operation we have to map each point into its local-coordinate correspondent, perform the sum in the local coordinates, and then map the result back onto the original space. If we call $\xi \doteq \psi(\phi) \in \mathbb{R}^m$ the local-coordinate correspondent of $\phi \in M$, we have $\oplus : M \times M \longrightarrow M \;;\; (\phi_1, \phi_2) \mapsto \phi_1 \oplus \phi_2 \doteq \psi^{-1}(\psi(\phi_1) + \psi(\phi_2))$. The symbol $+$ denotes the usual sum on $\mathbb{R}^m$. For instance, if $\phi = V \in \mathbf{S}^2$ is a unit-norm three-dimensional vector with spherical coordinates θ, λ, such that $V(\theta, \lambda) \doteq \begin{bmatrix} cos(\theta)cos(\lambda) & sin(\theta)cos(\lambda) & sin(\lambda) \end{bmatrix}^T$ then $V_1 \oplus V_2 \doteq V(\theta_1, \lambda_1) \oplus V(\theta_2, \lambda_2) = V(\theta_1 + \theta_2, \lambda_1 + \lambda_2)$, where the last sums are intended modulo 2π.

Equation (9), transformed into local coordinates, will be the state of the filter that estimates the parameters ϕ:

$$\xi(t+1) = \xi(t) + n_\xi(t) \tag{10}$$

where $\xi \doteq \psi(\phi)$ and $n_\xi(t) = \psi(n_\phi(t))$ and $+$ denotes the usual sum in $\mathbb{R}^m$. Now, if we substitute $\mathbf{y}^i - n^i$ for $\mathbf{x}^i$ in the state of the model (5), we get

$$h(\mathbf{y}^i(t-1), \phi(t))\mathbf{y}^i(t) = \tilde{n}^i(t) \qquad \forall\ i = 1 \dots N, \tag{11}$$

where $\tilde{n}^i$ is a noise process induced by n^i. Notice that $\tilde{n}^i$ is *not* a white noise, for it is correlated within one time step. A method for dealing with such a problem is described in [25], while in this paper we will assume that $\tilde{n}^i$ is approximated by a white noise, whose variance is inferred from the variance of n^i and the linearization of h. If we now put together equations (9) and (11), after assuming that $\tilde{n}^i$ is white, we end up with a dynamic model for the unknown parameters, having an implicit measurement constraint:

$$\begin{cases} \phi(t+1) = \phi(t) \oplus n_\phi(t) & \phi(t_0) = \phi_0 \\ h(\mathbf{y}^i(t-1), \phi(t))\mathbf{y}^i(t) = \tilde{n}^i(t) & \forall\ i = 1 \ldots N, \end{cases} \quad \phi \in M \tag{12}$$

which has a local-coordinate correspondent

$$\begin{cases} \xi(t+1) = \xi(t) + n_\xi(t) & \xi(t_0) = \xi_0 \\ h(\mathbf{y}^i(t-1), \psi^{-1}(\xi(t)))\mathbf{y}^i(t) = \tilde{n}^i(t) & \forall\ i = 1 \ldots N. \end{cases} \quad \xi \in \mathbb{R}^M \tag{13}$$

The above model is now in a form suitable for applying an EKF in its version for implicit measurement constraints. This can be easily derived from the standard equations of the EFK, after observing that the variational model about the best estimate of the current trajectory is linear and *explicit*, and the quantity

$$\epsilon^i(t) = h(\mathbf{y}^i(t-1), \psi^{-1}(\hat{\xi}(t+1|t)))\mathbf{y}^i(t) \tag{14}$$

plays the role of the *innovation* (the output prediction error [12]) of the filter. A derivation of the equations of the implicit EKF, which are summarized in the next section, can be found in [25].

2.4 Implementation and tuning

In the previous sections we have seen that both the extended models (4) and the reduced models (5) can be put in a form that is suitable for designing an Extended Kalman Filter in a recipe-like manner, which are (8) and (13) respectively.

If such models were linear and the model and measurement noises were white, zero-mean and Gaussian, the Kalman filter would guarantee that the innovation ϵ be white, zero-mean and have minimum variance. In the case of a nonlinear model, the "whiteness" of the innovation is considered to be a reliable diagnostic of the filter performance, and it may be evaluated using standard statistical tests, for instance Bartlett's Cumulative Periodogram (the integral spectrum of the prediction error).

What are the statistics of the measurement noise in typical vision applications? The feature-correspondence is known up to some uncertainty, summarized in the noise process $\tilde{n}^i$. Such uncertainty comprises both localization noise, which is usually zero-mean and in the order of few pixels standard deviation, and large errors due to mismatches. Such errors are intrinsic in the functioning of feature tracking/optical flow algorithms, which are based upon a local brightness constancy assumption often violated in real-life situations [2]. These errors cannot be eliminated by the optical flow/feature tracking algorithms; indeed, it is responsibility of the methods that use the optical flow/feature tracking in order to estimate 3-D structure and motion to treat properly both sources of errors, by rejecting outlier measurements due to mismatches, and by exploiting the statistics of the localization error and the redundancy in the measurements in order to minimize their effects. When the noise in the measurements is far from white and zero-mean, the statistics of the innovation changes dramatically, which suggests that by doing some simple test on the innovation process we may be able to spot out the outlier measurements due to mismatches in the optical flow/feature tracking. In fact, each component of the innovation measures how consistent each visible feature point is with the current estimate of motion. A test for rejecting outliers based upon such a principle has been proposed in [26]. Therefore, we are going to assume that the measurement noise is white and zero-mean, and we will reject as outliers those feature-points that produce an innovation residual which is not consistent with our statistical model.

We report here, for the sake of completeness, the equations for the Implicit EKF, which can then be applied to the reduced model (13), and the to extended model (8)

$$\textbf{Prediction step} \begin{cases} \hat{\xi}(t+1|t) = f(\hat{\xi}(t|t)) & \hat{\xi}(0|0) = \hat{\xi}_0 \\ P(t+1|t) = F(t)P(t|t)F^T(t) + \Sigma_\xi(t) & P(0|0) = P_0 \end{cases}$$

$$\textbf{Update step} \begin{cases} \hat{\xi}(t+1|t+1) = \hat{\xi}(t+1|t) + L(t+1)h\left(\mathbf{y}(t-1), \hat{\xi}(t+1|t)\right)\mathbf{y}(t) \\ P(t+1|t+1) = \Gamma(t+1)P(t+1|t)\Gamma^T(t+1) + L(t+1)\Sigma_{\tilde{n}}(t+1)L^T(t+1) \end{cases}$$

$$\textbf{Gain:} \begin{cases} L(t+1) = P(t+1|t)C^T(t+1)\Lambda^{-1}(t+1) \\ \Lambda(t+1) = C(t+1)P(t+1|t)C^T(t+1) + \Sigma_{\tilde{n}}(t+1) \\ \Gamma(t+1) = I - L(t+1)C(t+1) \end{cases}$$

$$\textbf{Residual variance:}\ \Sigma_{\tilde{n}}(t+1) = D(t+1)\Sigma_{\bar{n}}D^T(t+1)$$

where $F \doteq \left(\frac{\partial f}{\partial \xi}\right)$, $C \doteq \left(\frac{\partial h}{\partial \xi}\right)$ and $D \doteq \left(\frac{\partial h}{[\partial \mathbf{x}(t)\ \mathbf{x}(t-1)]}\right)$, Σ_* indicates that variance of the process $*$, and P is the variance of the estimation error. In the extended (explicit) models of the form (8), we have $h^i(\mathbf{y}(t-1), \xi(t))\mathbf{y}(t) \doteq \mathbf{y}^i(t) - g^i(\xi(t))$; in the reduced models (13) we have $f(\xi) = \xi$.

The only ingredients that are needed in order to complete the implementation of the filters are the measurement and model variances $\Sigma_{\bar{n}}$ and Σ_ξ. For the measurements, we have assumed that the error in the location of each feature-point is independent, with a standard deviation of 1 pixel (0.002 units of focal length in the simulation experiments described in section 3), according to the average performance of optical flow/feature tracking techniques [2]. $\Sigma_{\bar{n}}$ is therefore a $4N \times 4N$ matrix [5] with diagonal elements $4 * 10^{-6}$.

We assume that the model errors n_ξ are uncorrelated, and therefore their variance Σ_ξ is a diagonal matrix. In principle the elements of Σ_ξ corresponding to the structure parameters (in the extended models), and the ones corresponding to Ω_R and T in the integral models should be zero, for the model is *exact*. In order to prevent *saturation* [6] of the filter, we add a noise term whose variance is small relative to the variance of the measurement error (10^{-16}).

The variance of the random walk models for V and Ω is the most crucial to set, for it trades off the "smoothness" of the estimates with the "inertia" of the filter. We have experimented with various types of motion, and finally set the variance of the random walk parameters to 10^{-6}. This number has nothing magic, and has to be regarded as a reference. In order to be consistent, however, we have maintained the same tuning parameters throughout all the experiments we describe in section 3.

2.5 Recovering the reduced parameters

The "reduced models" (5) are obtained from the extended ones (4) via model reduction, as discussed in the companion paper [28]. In essence some of the states are eliminated by solving the measurement equation for such states, and substituted into the model equation. For instance, the subspace model is obtained by eliminating the depth and rotation parameters from the time-derivative of the measurement equation of the model (8).

As a result, the filters based upon the reduced models will only provide an estimate of *some* of the unknown parameters. How can we estimate the remaining ones?

The parameters that are not represented in the state of the reduced models are in a sense

[5] Note that in the reduced filters we need to keep in memory the measurements at time $t-1$, and the measurement vector is effectively $4N$-dimensional (image-plane coordinates at time t and $t-1$), rather than $2N$-dimensional as in the case of the extended models.

[6] Saturation of the filter can be described as follows: if the variance of the model error is zero, the model is perceived by the filter to be exact, the relative weight of the measurements decreases until the gain becomes zero along some direction, and the filter drifts away without paying attention to the measurements [12].

"hidden" and can be recovered easily. In fact, we can use the same equation that we solved for *eliminating* them in order to provide an estimate *from the current estimate of the states of the reduced model*. Equation (21) in section 3.2.1 of the companion paper [28] provides an instance of such an "indirect" estimate for the rotation and structure parameters from the estimated direction of translation. Such indirect estimates can be used as pseudo-measurements by a Kalman Filter that acts as a smoother, as described in [27].

As for the structure parameters, once motion has been estimated, it can be fed, together with the variance of the estimates, to an algorithm for estimating structure that processes motion error, such as [19, 29].

2.6 Dealing with scale factors

As we have anticipated in section 2.1, the structure parameters and the translational velocity are only measurable up to a scale factor which affects the depth of each point and the norm of the relative translation. In fact, it is very well known that an object moving in front of a camera produces the same images as an object which is "twice as far, twice as big and moving twice as fast" [14].

In order to get rid of such an ambiguity we can choose essentially two ways. The first consists in isolating the state variable that corresponds to the scale factor ambiguity and eliminating it. This is done in all reduced filters, where the translational velocity is expressed in spherical coordinates θ, λ (azimuth and elevation). Only the direction of heading, therefore, is estimated while the radius is constant and therefore removed from the state-space.

Alternatively, we may leave the state-space untouched, and saturate the filter along any direction affected by the ambiguity. Note that, by doing so, we are dealing with a model which is globally unobservable, and we just "freeze" our filter onto a slice of the unobservable space. The variance of the model error of any one of the states affected by the ambiguity (for instance the distance of one point in the models (4)), is set to zero, and so is the variance of the initial estimate. Each initial condition determines a slice of the state-space which is an observable subset of the state-space. Of course we can observe the trajectory of the model along such slices, but we cannot infer from the measurement in which slice we are. This strategy has been used, for instance, by Azarbayejani and Pentland [1].

2.7 Integral reduced models

Reduced filters may be implemented in their integral form, simply by referring the structure to the initial time instant and integrating the motion parameters. For instance, in the case of the essential constraint, the corresponding integral filter is based upon the model

$$\begin{cases} \Omega(t+1) = \Omega(t) + n_\Omega(t) \\ V(t+1) = V(t) + n_V(t) \\ R(t+1) = e^{\Omega(t)\wedge} R(t) \\ T(t+1) = e^{\Omega(t)\wedge} T(t) + V(t) \\ \mathbf{y}^i(t)^T \mathbf{Q}(T(t), R(t)) \mathbf{y}^i_0 = \tilde{n}^i(t). \end{cases} \tag{15}$$

Here the scale factor may be set by imposing that the initial translation has norm one, by giving it as an initial condition and saturating the initial variance of the estimation error for the norm of translation. This solution, unlike when the scale factor is associated to structure parameters, is very sensitive to drifts since the translational velocity changes in time and therefore the initial guess cannot be updated.

2.8 Dealing with occlusions

It must be noticed that, unlike incremental model, all filters based upon an "integral" model (defined relative to the initial time instant) need all the features to be visible throughout the experiment. In the presence of occlusions and appearance of new features one has to use some ad-hoc heuristics [7]. While all other schemes based upon a first-order random walk estimate *velocity* (or rather relative attitude between successive time instants), the integral filters estimate the attitude of the viewer relative to the initial time instant.

However, we remark that one of the major strengths of the reduced models is that they can integrate motion information over time in absence of continuative tracking of the same point-features, or even using optical flow at a fixed number of locations on the image. In fact, since structure is not represented in the state, we can add and remove features by adding or deleting rows of the measurement equation of the model (13), without affecting the continuity

[7] A technique for dealing with a variable number of features is outlined in [17].

of the state. Structure, however, is represented *indirectly* through the innovation process (14), whose components are a measure of how consistent each feature is with the current motion interpretation.

3 Experiments

We have chosen to use a simulation framework in order to make careful comparisons, since a rigorous *ground truth* is available while the relevant parameters are varied systematically. Such a ground truth is difficult to obtain and impossible to validate for real image sequences.

First, we test the scheme on a real image sequence obtained by rotating a box on top of a chair (the "box sequence", section 3.2). Then we build a simulation that mimics the box sequence, and allows us to change the number of visible features, the distance from the viewer, the noise level, the initial conditions for the filters and other structural parameters in a systematic way. The basic setup is described in section 3.3, and the following sections outline the results of the experiments. The particular choice of experiment is then validated by testing the algorithms on other motion and structure configurations (section 3.11).

3.1 Nomenclature

We have implemented a recursive filter for each of the geometric models described in the companion paper [28] and summarized in section 2 of this paper. The filter based upon the extended model (4-left), which we call the **"structure filter"**, needed very accurate initial conditions for the motion parameters, and therefore it did not converge in most of the situations described in this section. Therefore, the filter for simultaneously estimating structure and motion has been implemented only in its "integral" version, based upon the model (4-right). This filter, which we call the **"integral structure filter"**, is the same proposed by Azarbayejani and Pentland [1], except for minor modifications.

We have then implemented the filter derived from the subspace constraint, called the **"subspace filter"** in [27], which corresponds to the model (12) with the parameter space $M = \mathbf{S}^2$. The velocity of image features is approximated by first differences, and exponential coordinates are used to model the discrete motion between successive time instants. The filter based upon the epipolar constraint of Longuet-Higgins [14] is called the **"essential filter in local coordinates"** in [25]. These filters are implemented in their incremental version, which can use both feature tracking or optical flow (velocity vectors at fixed locations on the image-plane) as input. For the sake of comparison with the integral-structure filter, we have also implemented an integral version of the essential filter, which refers motion to the initial time instant; we call this filter the **"integral essential filter"**.

We have then implemented one filter for each of the fixation constraints described in the companion paper [28]. The filter derived from fixating a feature-point is called the **"point-fixation filter"**. Similarly, when we fixate a point and a line, we have the **"point-plus-line fixation filter"**, and when we compensate for the motion of a plane we have the **"plane-plus-parallax filter"**, or **"plane-fixation filter"**. All of these filters are obtained from the model (12) where, in each case, only the parameter space M changes.

It must be noticed that "integral filters" need all features to be visible throughout the sequence, as opposed to "reduced filters" that can integrate motion information over time even in the presence of features with a very short life-span. Therefore, reduced filters have an advantage in real-life situations, since it is extremely difficult to track single features over long sequences; typically feature-tracking algorithms can trace features over the order of ten frames, and then refresh by selecting a new set of features [2]. In the following sections, however, we are mainly interested in comparing the geometric essence of each scheme, and we have therefore selected all features that survived from the beginning to the end of the experiments, in order to compare integral models against reduced ones.

3.2 The basic experiment: the "box sequence"

We report here a test on a sequence of real images that we will later replicate in our simulation environment. This is done mainly for the purpose of motivating the experimental conditions used in the simulations. A box of side approximately $30cm$ is placed on a chair $50cm$ ahead of the camera and rotated by 5 deg/frame circa. The direction of rotation is inverted after 25 frames, and the overall sequence is 40 frames-long.

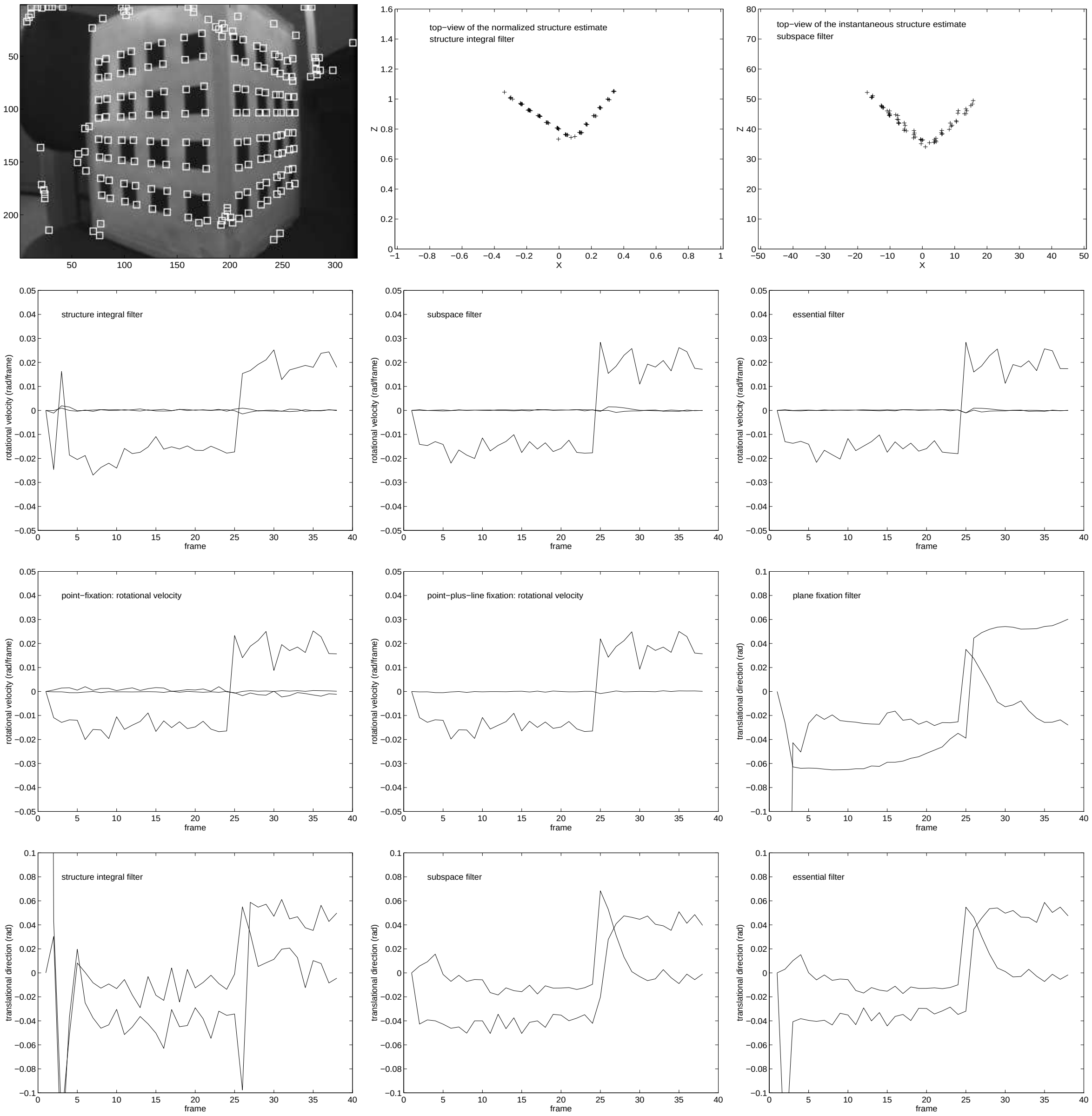

Figure 1: *(Row, Column): (1,1) one image of the* **"box sequence"**. *(1,2) normalized structure estimated by the integral structure filter. (1,3) instantaneous estimate of structure by the subspace filter. Rotational velocity estimated by the integral structure filter (2,1), the subspace filter (2,2), the essential filter (2,3), the point-fixation filter (3,1) and the point-plus-line filter (3,2). The last scheme produces estimates only for two out of the three rotation parameters, since it exploits the fact that the third (cyclorotation) is zero. Direction of translation estimated by the integral structure filter (4,1), the subspace filter (4,2), the essential filter (4,3) and the plane-fixation filter (3,3). We plot the two spherical coordinates (azimuth and elevation) as a function of the frame number.*

We have used a multi-scale version of the classical SSD algorithm [15] for tracking a number of features. In order to test the integral filters we have selected only the features that survived from the first to the last frame.

The setting used for each filter is exactly the same used for the simulation experiments which is described in the next sections, and no ad-hoc tuning was performed. Initial conditions were zero for all schemes, and a noise level of one pixel std was hypothesized for the feature tracking.

In figure 1 (top row) we show one image of the test sequence (left), with the feature points highlighted, and the estimates of structure performed by the integral structure filter (middle), normalized so as to place the center of mass at unit distance from the viewer. The figure shows a top view of the scene at the initial time instant, and it can be seen that the qualitative structure of the box is estimated correctly. In the right plot we show the instantaneous estimate of structure that comes as a byproduct from the subspace filter, as discussed in section 2.5. Note that such estimate only uses the instantaneous measurements and the current estimate of motion, and is therefore less precise. All other schemes do not provide an estimate of structure *directly*. However, their estimates of motion may be fed to any structure-from-motion module that processes motion error, as done for instance in [19, 29].

In figure 1 (rows 2-4) we show the estimates of the rotational velocity and the direction of translation (azimuth and elevation). The plane fixation constraint does not provide an estimate of the rotational velocity *directly*. Similarly, the point-fixation and the point-plus-line fixation constraints do not provide a direct estimate of the direction of translation, but only the translational velocity along the fixation axis.

Of course, in the absence of a ground truth it is only possible to appreciate the qualitative behavior of each estimator. In order to perform a rigorous quantitative evaluation of the properties of each model, it is necessary to employ a simulation platform, which we describe in the next section.

3.3 Simulation setup

We have generated a simulation that mimics the box experiment described in the previous section. A cloud of $N = 20$ dots is distributed at random within a cubic volume of side $1m$ at a distance

$d = 2m$ from the viewer. These dots are projected onto an ideal image plane with unit focal length and 500×500 pixels, corresponding to a visual angle of approximately 30^o and therefore approximately $3.5'$ of visual angle per pixel. White, zero-mean Gaussian noise has been added to the projections with a standard deviation n_0 varying between 0.1 and 12 pixels. The cloud is then rotated about an axis parallel to the image-plane and passing through its center with a constant velocity [8] of 4 deg/frame. The basic experiment is then altered by varying systematically the parameters of the simulation. All tuning parameters remain the same throughout the experiments.

3.4 Accuracy

Each scheme is tested on a sequence containing 20 point-features, with initial conditions distributed normally at random around the true motion parameters, with a standard deviation of 4% of the norm of the true parameters. The noise level is increased from 0.1 to 5.1 pixels std, and the normalized estimation error is evaluated over a window of 10 frames, after the filters have settled (between frame 50 and 60). In figure 2 we plot the norm of the estimation error against the noise level for a window between 0.1 and 1.1 pixels, according to the average performance of feature-tracking/optical-flow techniques [2]. In order to evaluate the *accuracy*, we have plotted only the instances when the filters have convergence in all 50 trials. We display the mean error, and visualize the standard deviation using error-bars.

It may be noticed that the subspace filter does not converge to zero error in the absence of noise and is in general less precise, since it has to cope with the approximation of the derivative of the position of the features on the image-plane using first-differences (upper-left plot). The schemes that impose fixation constraints, either for a point (middle-left), a line (middle-center) or a plane (middle-right) cease converging consistently for noise levels around 0.6 pixels std. This is due to the propagation of the errors in fixating noisy features.

Integral filters (figure 2 bottom) can count on an increasingly large baseline, for structure is referred to the initial time-instant and motion is modeled as a second-order random walk, and exhibit therefore a better performance.

In figure 3, we plot the norm of the estimation error against the noise level that increases

[8] If the reader is not comfortable with this assumption, we suggest a quick look at section 3.12.

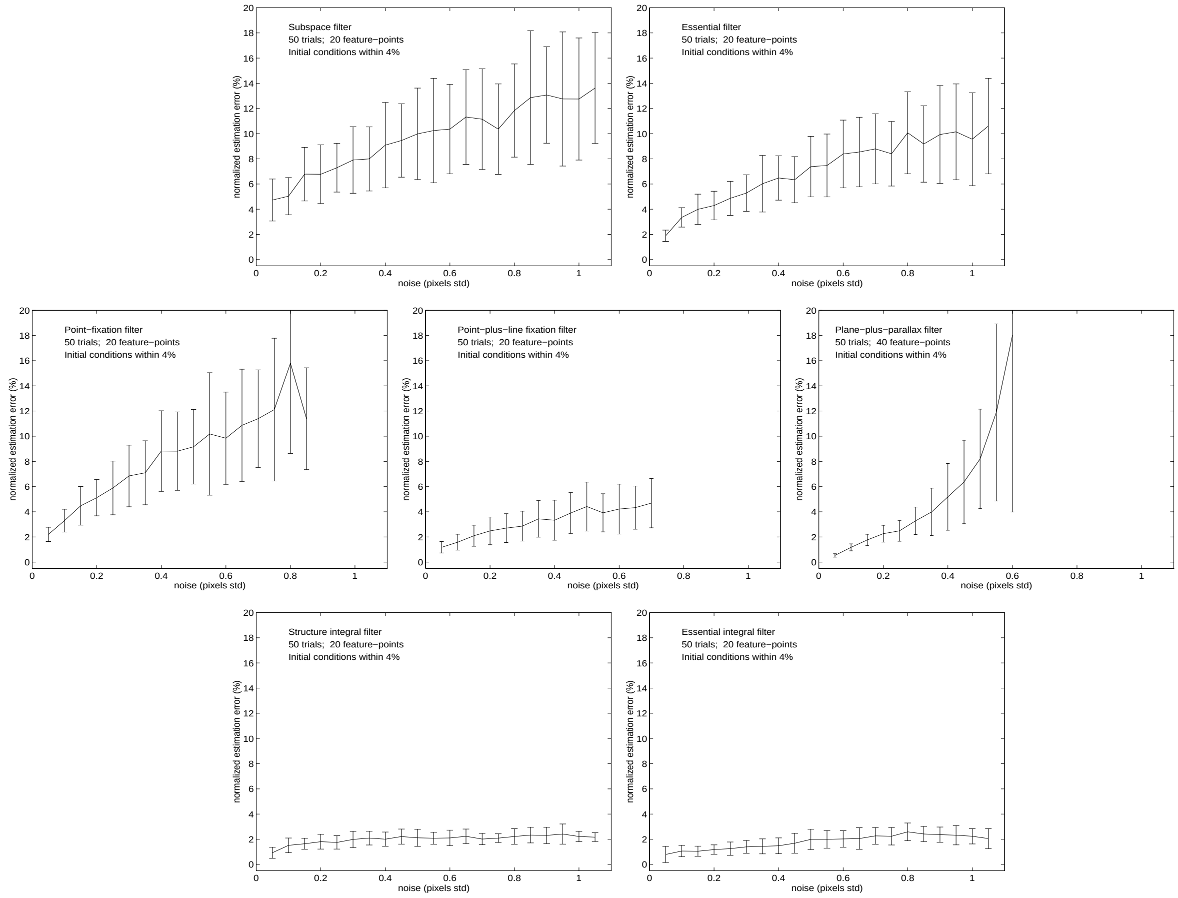

Figure 2: **Accuracy experiment.** *50 trials, with 20 feature-points (except for the plane-fixation filter, see also figure 6), starting at initial conditions distributed at random within 4% of the true parameters while the noise level increases from* 0.1 *to* 1.1 *pixels std, according to the standard performance of feature tracking algorithms. The scaled norm of the estimation error is plotted against the noise level. The filters enforcing a fixation constraint (middle row), cease converging consistently for less than one pixel noise. Note that integral filters (bottom row) have an advantage in performance, since they can count on an increasingly large baseline. For the integral structure filter we display only the error in the estimates of motion parameters.*

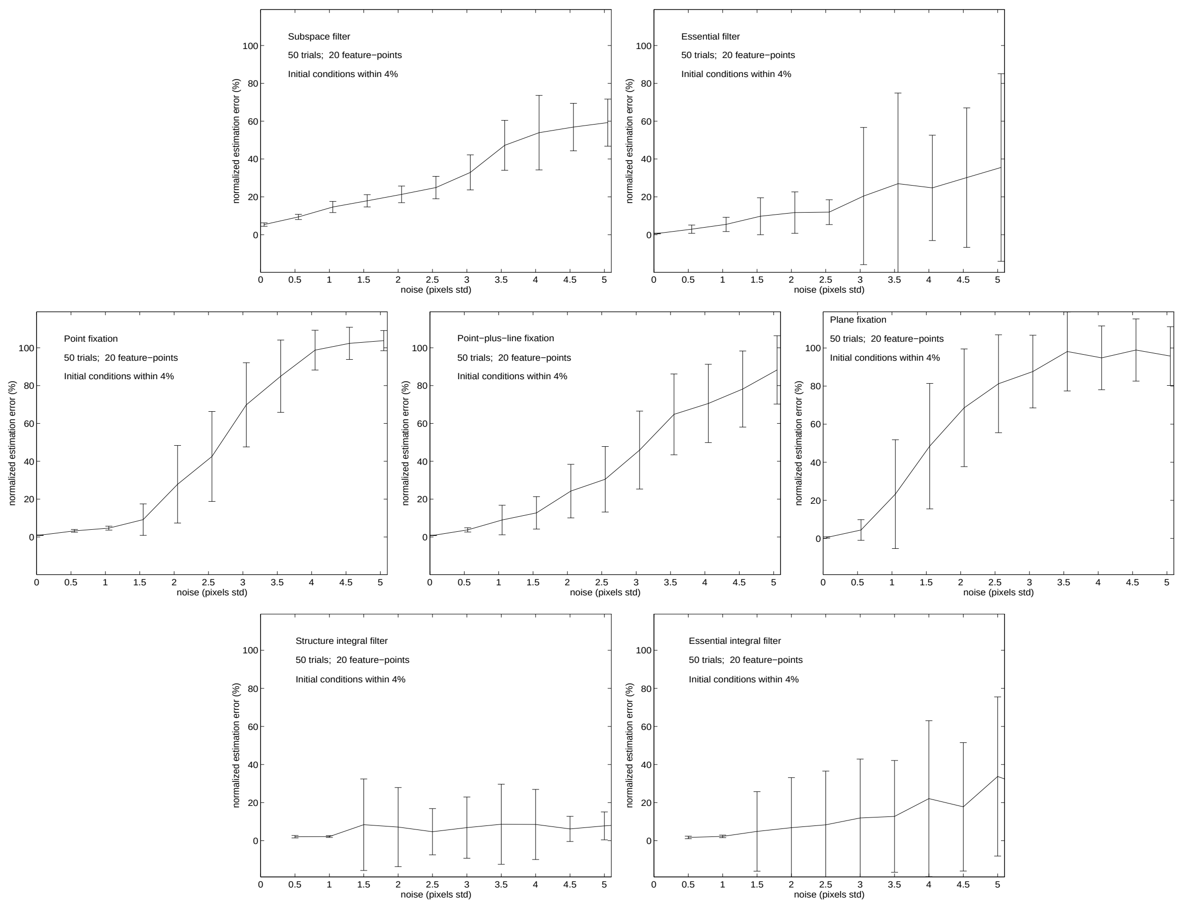

Figure 3: **Accuracy/robustness experiment.** *The conditions were the same described in figure 2, except that the noise level goes from* 0.1 *to* 5.1 *pixels std and we did not remove the instances when the filters did not converge. The scaled norm of the estimation error is plotted against the noise level after the filters have settled. The size of the error-bars can be considered a measure of robustness, for it indicates the consistency of each filter across trials.*

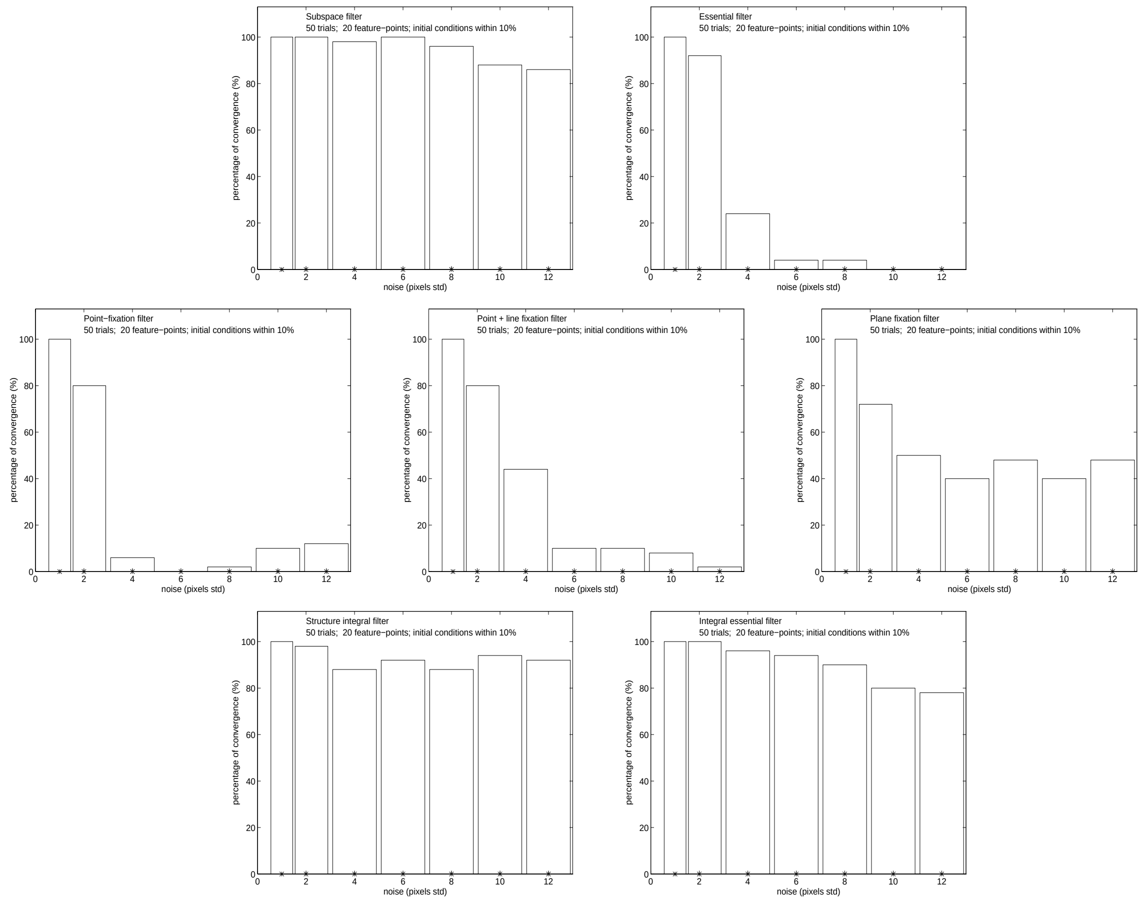

Figure 4: **Robustness experiment.** *50 trials with the initial conditions distributed at random within 10% of the true value, and the noise level increased from 1 to 12 pixels std. The histograms represents the percentage of the experiments in which the filters reached convergence. Integral filters (bottom row) exhibit better robustness properties than reduced filters, with the exception of the subspace filter (top-left).*

from 0.1 to 5.1 pixels without removing the instances when the filters did not converge. We have performed 50 trials of the experiment, and we display the mean error, and visualize the standard deviation using error-bars. This experiment evaluates a mixture of accuracy and robustness, since the size of the error-bars gives an idea of the consistency of the performance across trials.

3.5 Robustness

In this experiment we assess the robustness of each filter, intended as the capability to retain a correct estimate in the presence of increasing noise. We have performed 50 trials, with initial conditions distributed at random within 10% of the true parameters, and we have tested whether the filter has reached convergence after 50 time steps. In order to formulate a convergence verdict we test both the estimation error and the periodogram of the innovation. In fact, the criterion for the filter to be operating correctly is that the innovation be "as white as possible". The periodogram, which is the integral of the prediction error spectrum, is a measure of how "white" the innovation is. However, occasionally filters may get stuck in "local minima" where the innovation is small, but the estimation error is large.

In figure 4 we report a histogram of the percentage of trials that have reached convergence as a function of the noise level that ranges between 1 and 12 pixels std. It can be seen that the filters that enforce fixation constraints (middle row) are significantly less robust than the ones based upon explicit reduction. Integral filters (bottom row) are in general more robust than reduced filters, with the exception of the subspace filter (top-left), which proves remarkably robust.

3.6 Convergence

In this experiment we test the convergence properties of each model, by changing the initial conditions at random within a region that grows from 1% to 100% of the true values of the parameters. In figure 5 we plot an histogram that counts the percentage of successful convergences as a function of the size of the perturbation of the initial conditions. Noise is half a pixel std.

The filters based upon the fixation assumptions (middle row) have convergence problems, most probably due to the effects of noise propagated through the fixation constraint.

Integral filters (bottom row) prove more sensitive to initial conditions than reduced ones. For the structure integral filter this is due to the observability properties of the model, discussed in [23], while for the essential integral filter this is most probably due to the mechanism of propagation of scale, which consists in saturating the norm of the initial translational velocity. Such a filter is subject to a drift that increases with perturbations in the initial conditions.

3.7 Dependence upon the number of visible points

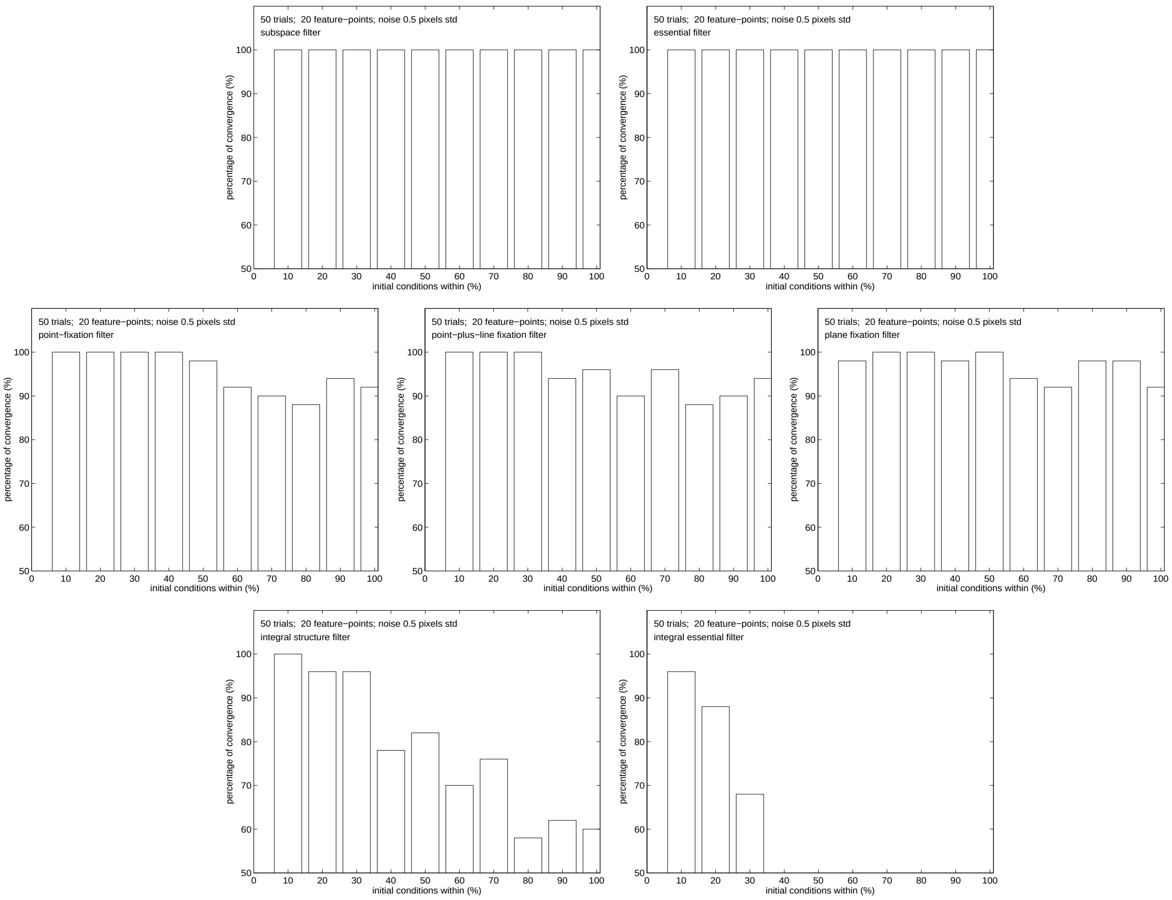

Figure 5: **Convergence experiment.** *50 trials with* 0.5 *pixel std error, while the initial conditions are chosen at random with Gaussian distribution with* σ *ranging from* 10% *to* 100% *of the true parameters. Integral filters (bottom row) exhibit decreased robustness relative to reduced filters. For the structure integral filter (bottom-left) this is mainly due to the observability properties of the model having structure in the state, while for the integral essential filter (bottom-right) this behavior is due to the mechanism of propagation of scale over time.*

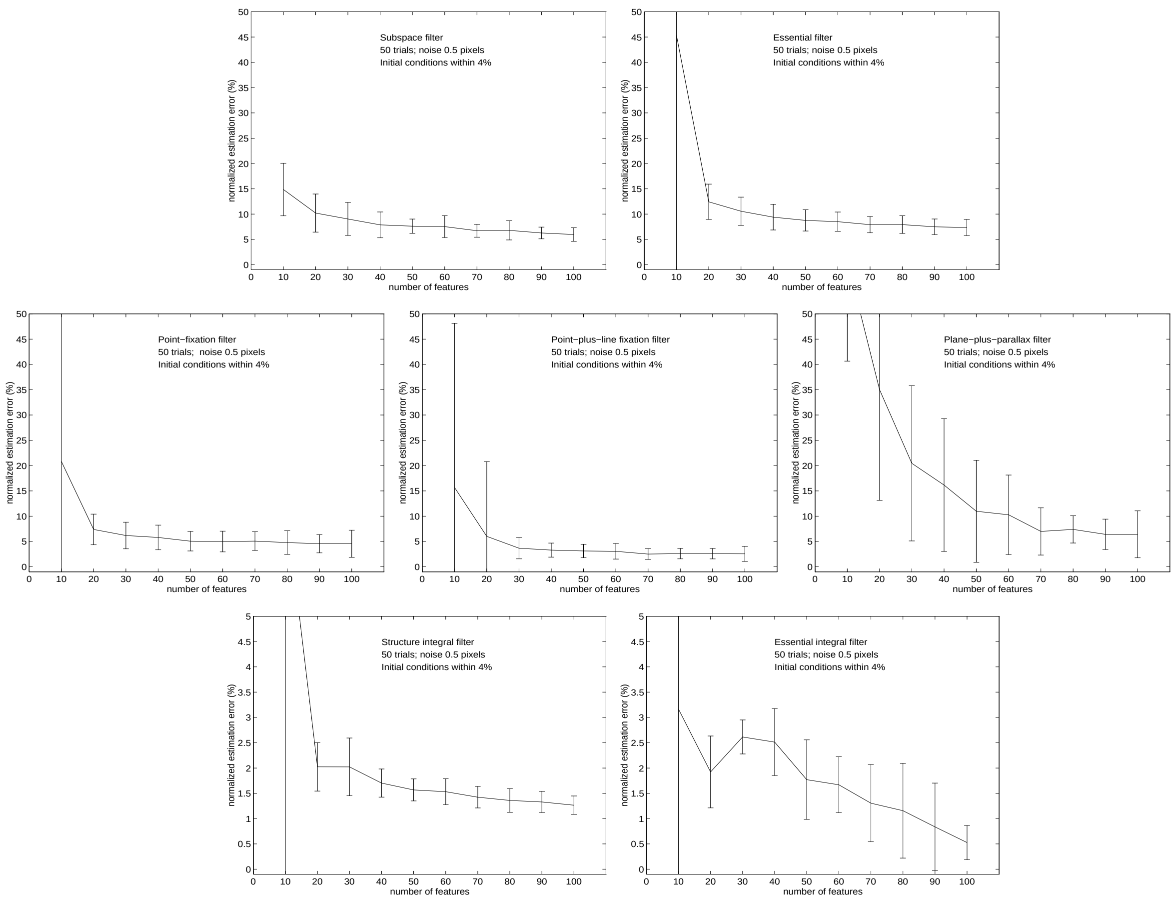

Figure 6: **Dependence upon the number of features.** *The norm of the estimation error is plotted against the number of visible features, for a noise level of half a pixel and initial conditions within 4%. The subspace filter (top-left) has an advantage over other schemes in that it needs fewer features for reaching convergence. However, the computational cost of such a filter is quadratic in the number of features, unlike all other schemes whose complexity is linear. Note that all filters can actually reach convergence in the presence of less than 5 feature-points (for small noise and small acceleration) since motion information is integrated over time. This is an advantage over two-views algorithms that need at least 5 (or 8) features to be visible at all times. Note that the plane-fixation filter needs more features in order to achieve performance similar to other reduced filters. For this reason the accuracy experiment in figure 2 has been performed with 20 feature-points for all filters, except for the plane-fixation filter which had 40. Note that the performance improves marginally beyond 50 features.*

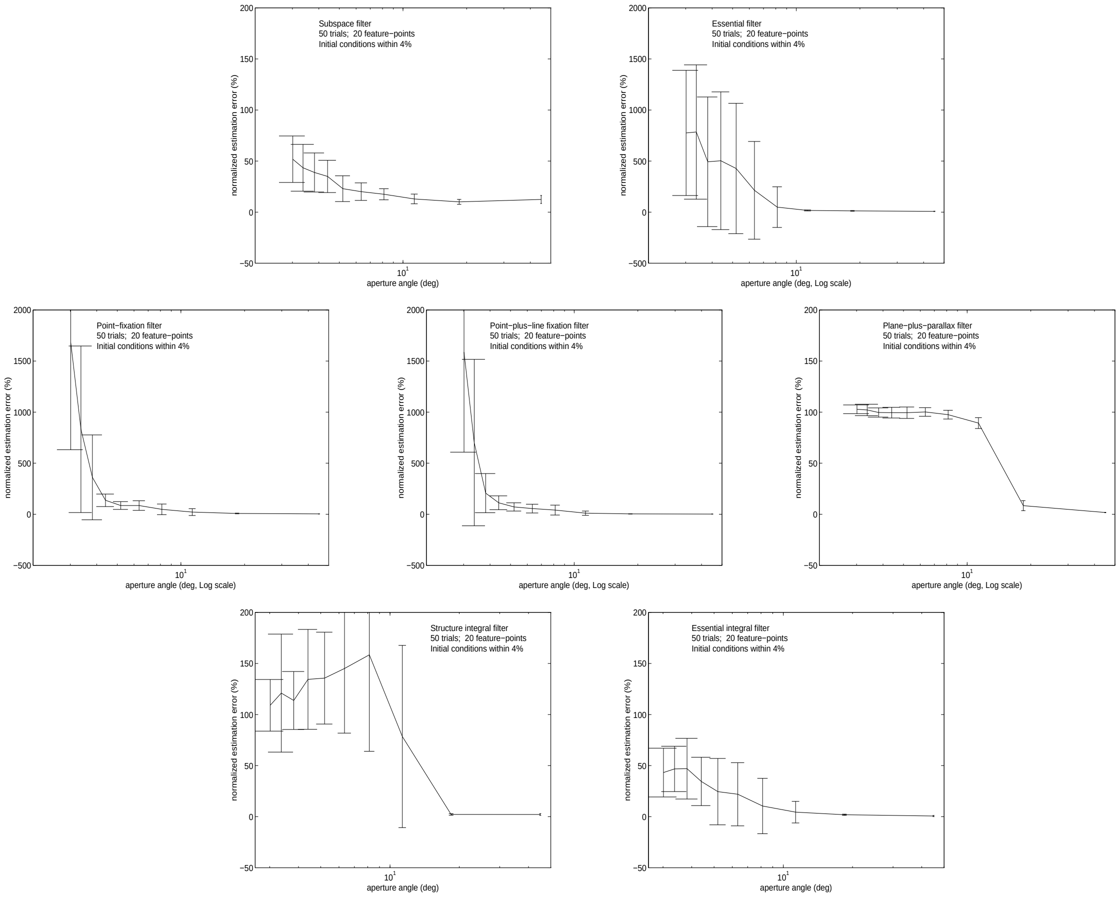

Figure 7: **Dependence upon the aperture angle.** *Norm of the estimation error as a function of the aperture angle that ranges from* 2^o *to* 40^o*.*

In figure 6 we display the norm of the estimation error as a function of the number of features, which range from 10 to 100. In general performance levels at 50 points, for the noise levels and initial conditions considered. An exception is the plane-fixation filter, which needs more points in order to accurately warp the images, and estimate the residual direction of translation. The subspace filter seems to have an advantage in that it needs fewer points. However, such a filter has a quadratic complexity, and therefore it becomes computationally intensive for more than 70 feature-points.

3.8 Dependence upon the aperture angle

All models based upon full perspective projection need a wide field of view in order for the higherorder perspective effects to be appreciable. We have decreased the aperture angle from 40 down to 2 degrees: most filters seem to prefer aperture angles larger than 10 degrees, while the plane-fixation filter and the integral structure filter need at least 20 degrees of visual angle to achieve satisfactory performance (figure 7).

3.9 Sensitivity to the "bas-relief" ambiguity

We have taken the original cubic cloud of points, and reduced one of the dimensions to a fraction of the original side, ranging from 100% (cubic cloud) down to 10% (flat cloud). The norm of the estimation error as a function of the "flatness" of the cloud is plotted in figure 8. Most filters do not seem to be bothered by such a deformation, for the aperture angle considered (30^o). Notice that one can view such a deformation of the cloud as a reduction of the effective field of view, which is however limited to the times when the cloud shows the thinner face.

An exceptional behavior is exhibited by the plane-fixation filter (middle-right). In fact, the estimation error seems to increase dramatically as the cloud approaches a plane. This, however, does not mean that the filter is not operating correctly. In fact, as the cloud approaches a plane, the warping operation stabilizes such a plane up to the point in which the residual parallax is zero (in the limit of a flat plane). Therefore the norm of the residual translation is zero, and its direction is undetermined.

3.10 Dependence upon the parallax (sampling rate)

In the basic experiment the cloud of dots rotates about an axis parallel to the image-plane by 4 degrees per frame. In figure 9 we show how performance changes as the rotational velocity varies between 1 and 12 degrees/frame. The subspace filter is based upon a differential model, and therefore it prefers small rotations. There is, however, a tradeoff between the first-difference approximation of the image-velocity and the amount of parallax in the data. As the velocity increases, the data are better conditioned, but the first-order approximation of the image velocity degrades. The exponential coordinatization of motion helps improving the filter for large image-motions.

The behavior of the essential integral filter (bottom-right) is almost inverse to the other filters.

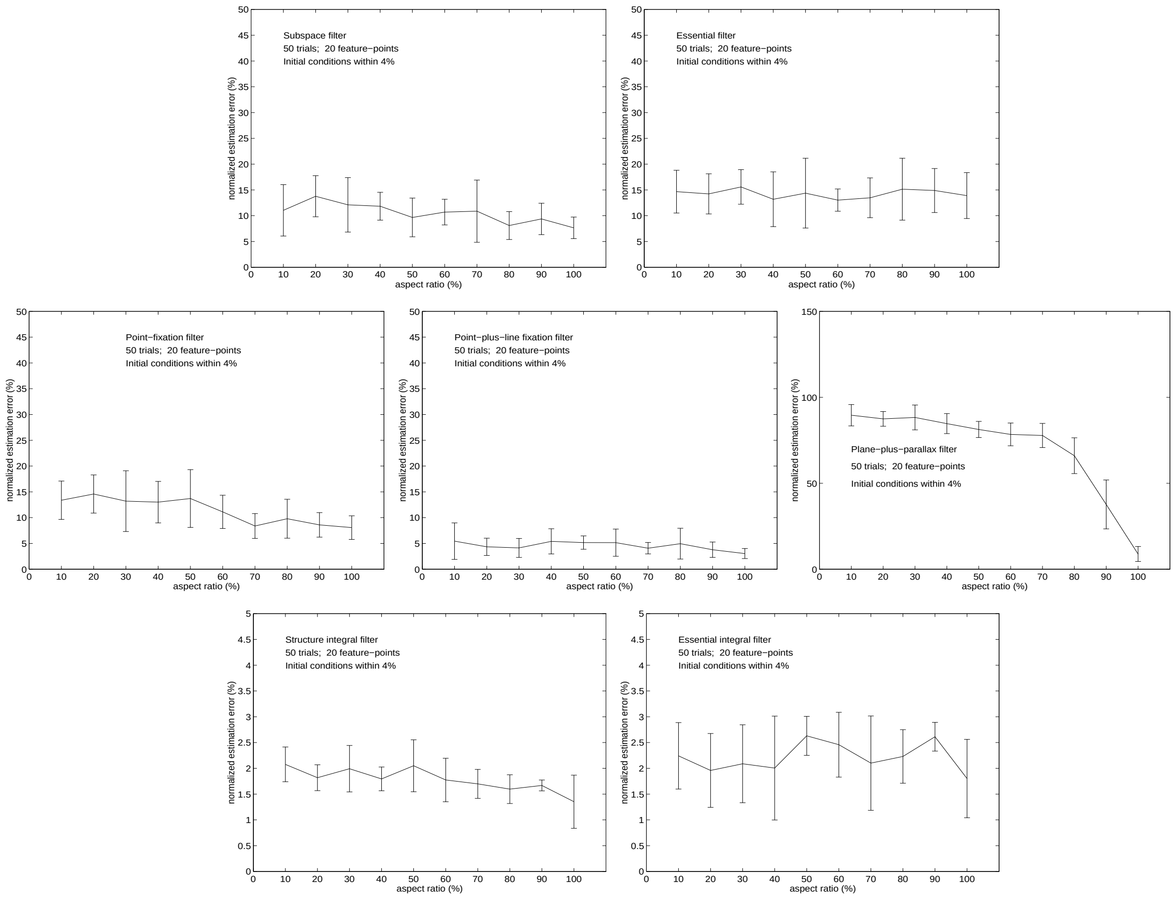

Figure 8: **Dependence upon the bas-relief ambiguity.** *The norm of the estimation error is plotted against the "thickness ratio" of the cloud of points being viewed (ratio between width and depth), which ranges between 10% and 100%. The error curve is almost flat for all schemes, except for the plane-fixation filter (middle-right), whose error increases as the scene approaches a plane. When the scene approaches a plane, the warped images have no parallax, and therefore the residual translation has norm zero, and the direction of translation (which is the state of the filter) can be arbitrary without violating the constraints.*

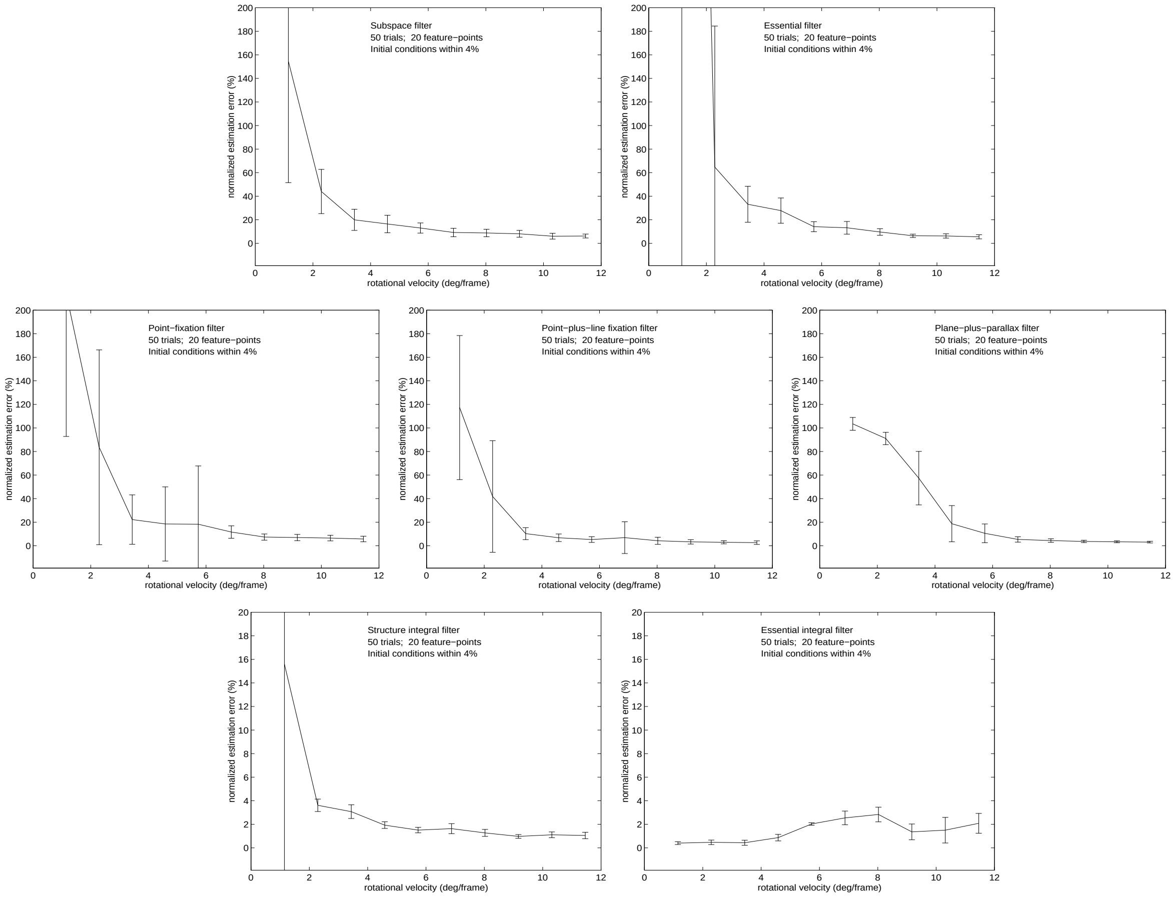

Figure 9: **Dependence upon the sampling rate.** *The subspace filter (top-left), which is based upon a differential model, converges for smaller velocities. In principle its performance should degrade as such velocity increases, since image velocities are approximated by first differences. However, the exponential coordinatization helps maintaining good performance even in the presence of large image-motions. The performance of the integral essential filter is somewhat odd. Since the filter is based upon a second-order model, and therefore it can count on an increasingly large baseline, it can handle small motions quite well. However, when the instantaneous baseline increases, the bias in the estimate of scale increases, which causes a degradation of the performance.*

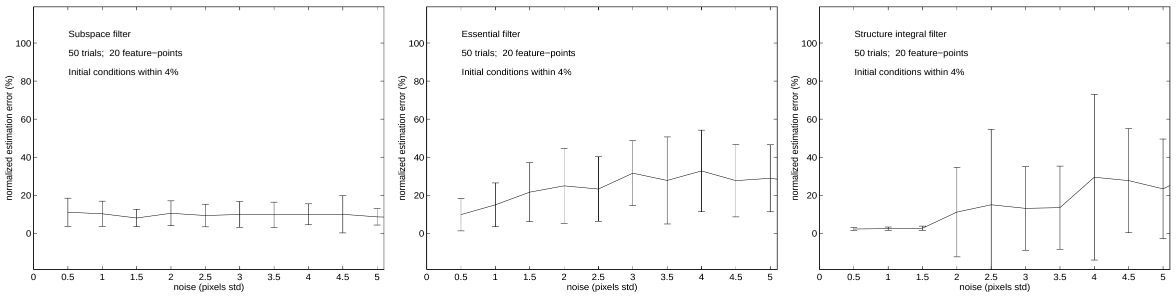

Figure 10: **Alternative motions.** *The accuracy/robustness experiment of figure 3 is repeated for some alternative motions. In the left plot we display the performance of the subspace filter for a forward translation of 30 cm/frame. Although the average norm of image-motion vectors is similar to that of the box experiment, the data are less ambiguous, for the effects of rotation and translation do not superimpose. The same motion has been estimated by the essential filter, and the results are shown in the middle plot. We have also considered translation along a direction parallel to the image-plane by 20 cm/frame. The estimation error for the integral structure filter is reported in the right plot. Compare with figure 3 top-left, top-right and bottom-left respectively.*

In fact, it degrades as the image-motion increases. This is most probably due to the mechanism of propagation of scale, which is subject to biases that increase with the size of the image-motion.

3.11 Other types of motion

Throughout this section we have considered the "box experiment" as a paradigm. Here we consider other types of motion. In a first experiment we consider forward translation within an infinite cloud of points, where only the ones that fall within a visual angle of 30 degrees are seen. Translation is 30 cm/frame in order to produce an image-motion of size comparable to that of the box sequence. Note that we cannot test integral filters on this sequence, for points move out of the visual field as the viewer translates forward. Results are qualitatively similar to those obtained for the "box experiment". As an example, in figure 10 we display the results of the accuracy/robustness experiment for the essential filter and the subspace filter. In general this motion is "simpler" than the roto-translational motion of the box experiment, and performance is better.

We have also considered translation along a direction parallel to the image-plane by 20 cm/frame. The scene is the usual cloud of 20 points of side $1m$ at $2m$ from the viewer. As time

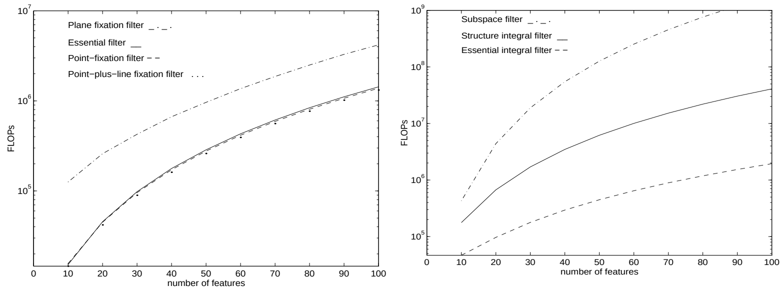

Figure 11: **Complexity**: *number of floating point operations as a function of the number of visible features. This count includes the overhead of our Matlab implementation. The subspace filter has been implemented using a tensor package that does not exploit the sparse structure of the matrices involved in the computation.*

goes by, the cloud moves farther away, and the effective aperture angle decreases. Nevertheless the performance is comparable with that obtained in the box experiment. In figure 10 (right) we show the performance of the structure integral filter.

3.12 A remark on "constant velocity" and first-order random walks

In the incremental models we have chosen a first-order random walk to describe the dynamics of the unknown parameters. Integral models can be interpreted as a second-order random walk. The only reason for choosing such random-walk models is that they are a good compromise between simplicity and flexibility. As we have pointed out already, *any other dynamical or statistical model* can be used in place of the first-order walk in any one of the filters described in this paper, as long as it preserves the observability properties of the overall system. The reader who is uncomfortable with modeling motion as a first-order random walk may consider looking at an experiment presented in [27], where the velocity of the cloud of the same synthetic experiment just described is modulated first by a *sinusoid*, then by a *saw-tooth* discontinuous function, and then by a *second order* random walk.

4 Discussion and interpretation of the results

We have compared the various models under controlled conditions, in order to evaluate the geometric properties of each constraint. It emerges that the models obtained by reduction using *fixation*, i.e. using output-dependent changes of coordinates, are in general less effective in all respects: precision, robustness and convergence properties. This is surprising, for one expects that the fewer the degrees of freedom, the better-conditioned the optimization task should be. Our finding can be explained by the fact that, when reduction is performed using changes of coordinates that depend on the noisy measurements, the effects are propagated in a non-linear fashion across the states of the filter, and even keeping track of the second-order statistics of the errors does not help. "Explicit reduction", on the other hand, does not require use of the measured output, and helps achieving desirable properties such as global observability of the dynamic model [23]. Note that we could reach this conclusion only because the unifying framework allowed us to compare the models that exploit the fixation constraints versus the *same* models based on general motions, simply by changing the geometry of the parameter space while using the same dynamic model and the same estimation technique.

Integral filters are in general more accurate and robust than reduced ones, with the exception of the subspace filter that proves remarkably insensitive to measurement noise. On the other hand, integral models are more sensitive to perturbations in the initial conditions, due either to the observability properties of the model or to the mechanism of scale propagation.

Other practical aspects, such as the presence of occlusions, need also to be taken into consideration. In fact, in the presence of occlusions, the integral structure filter has a disadvantage over the reduced models that do not include structure parameters in the state, for it has discontinuities in the estimates each time a new feature enters the field of view, or each time a feature disappears. Furthermore, the integral structure filter needs full-fledge feature tracking, and cannot use the optical flow at a fixed number of locations on the image.

The computational load of the schemes proposed are comparable, and range approximately between $40KFlops$ per frame and $10MFlops$ per frame depending upon the scheme, the number of features and the implementation. In figure 11 we report the number of floating point-operations as a function of the number of points for our `Matlab` implementation. Such implementation is not optimized and the count includes the overhead from the Matlab server. We feel that each one of the schemes we have tested could be implemented in real-time on standard processors *once the feature tracking/optical flow* is available. Motion and structure estimation are not the crucial bottleneck for real-time systems; feature-tracking/optical flow, on the contrary, is quite demanding and needs to be further optimized in order to run in real-time on low-cost hardware platforms [2].

References

[1] A. Azarbayejani and A. Pentland. Recursive estimation of motion, structure and focal length. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1995.

[2] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. of Computer Vision*, 12(1):43–78, 1994.

[3] J. Bergen, R. Kumar, P. Anandan, and M. Irani. Representation of scenes from collections of images. *Internal Report, Sarnoff Research Center*, 1995.

[4] T. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE trans. AES*, 1990.

[5] T. Broida and R. Chellappa. Estimating the kinematics and structure of a rigid object from a sequence of monocular frames. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1991.

[6] T. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE trans. PAMI*, Jan. 1986.

[7] N Cui, J. Weng, and P. Cohen. Recursive-batch estimation of motion and structure from monocular image sequences. *IEEE trans. AES*, 1990.

[8] C. Fermüller and Y. Aloimonos. Tracking facilitates 3-d motion estimation. *Biological Cybernetics (67), 259-268*, 1992.

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

[10] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion i: algorithm and implementation. *Int. J. Comp. Vision vol. 7 (2)*, 1992.

[11] J. Heel. Direct estimation of structure and motion from multiple frames. *AI Memo 1190, MIT AI Lab*, March 1990.

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

[13] T. Kailath. *Linear Systems.* Prentice Hall, 1980.

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

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

[16] L. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *Int. J. of computer vision*, 1989.

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

[18] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, 1994.

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

[20] P. Anandan R. Kumar and K. Hanna. Shape recovery from multiple views: a parallax based approach. *Proc. of the Image Understanding Workshop*, 1994.

[21] D. Raviv and M. Herman. A unified approach to camera fixation and vision-based road following. *IEEE Trans. on Systems, Man and Cybernetics vol. 24, n. 8*, 1994.

[22] H. S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. *Proc. of the Int. Conf. on Pattern Recognition*, 1994.

[23] S. Soatto. Observability/identifiability of rigid motion under perspective projection. In *Proc. of the 33rd IEEE Conf. on Decision and Control*, pages 3235–3240, Dec. 1994.

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

[25] S. Soatto, R. Frezza, and P. Perona. Motion estimation via dynamic vision. *IEEE Trans. on Automatic Control*, in press, March 1996.

[26] S. Soatto and P. Perona. 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*, pages 228–235. IEEE Computer Society, November 1994.

[27] S. Soatto and P. Perona. Recursive 3-d visual motion estimation using subspace constraints. *Int. J. of Computer Vision*, in press, 1996.

[28] S. Soatto and P. Perona. Reducing "structure from motion" 1: modeling. *submitted to the IEEE trans. PAMI*, Nov. 1995.

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

[30] M. Spetsakis and J. Aloimonos. A multi-frame approach to visual motion perception. *Int. J. Computer Vision 6 (3)*, 1991.

[31] M. A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1992.