Three-Dimensional Motion Estimation of Objects for Video Coding

Giancarlo Calvagno, *Member, IEEE,* Roberto Rinaldo, *Member, IEEE,* and Luciano Sbaiz

Abstract—**In this work, three-dimensional (3-D) motion estimation is applied to the problem of motion compensation for video coding. We suppose that the video sequence consists of the perspective projections of a collection of rigid bodies which undergo a rototranslational motion. Motion compensation can be performed on the sequence once the shape of the objects and the motion parameters are determined. We show that the motion equations of a rigid body can be formulated as a nonlinear dynamic system whose state is represented by the motion parameters and by the scaled depths of the object feature points. An extended Kalman filter is used to estimate both the motion and the object shape parameters simultaneously. The inclusion of the shape parameters in the estimation procedure adds a set of constraints to the filter equations that appear to be essential for reliable motion estimation. Our experiments show that the proposed approach gives two advantages. First, the filter can give more reliable estimates in the presence of measurement noise in comparison with other motion estimators that separately compute motion and structure. Second, the filter can efficiently track abrupt motion changes. Moreover, the structure imposed by the model implies that the reconstructed motion is very natural as opposed to more common block-based schemes. Also, the parameterization of the model allows for a very efficient coding of motion information.**

Index Terms—**Image coding, Kalman filtering, motion analysis, motion compensation, video signal processing.**

I. Introduction

EFFICIENT video coding techniques usually take advantage of the temporal redundancy between adjacent frames in the image sequence to greatly reduce the data transmission rate. A way to accomplish this is by estimating the displacement between frames of image elements, which can be individual pixels [1], picture blocks of fixed dimension (as in block matching motion compensation [2]), or groups of pixels corresponding to moving objects in the scene [3].

Typical video sequences consist of a few moving rigid objects and a static background. In particular, videoconference scenes have an almost fixed scene content, consisting of the head and shoulders of the speaker and the background. The movement of the speaker mainly consists of the global movement of the shoulders and head, which can be approximated as rigid objects, and of the local motion due to facial expression changes and speech.

Two basic approaches to three-dimensional (3-D) motion estimation have been proposed in the literature [4]. The first approach computes the two-dimensional (2-D) field of instantaneous velocities or optic flow based on local spatial and temporal luminance gradients [5]. The optic flow map is segmented, and different regions are associated with distinct objects in the scene. Constraints and *a priori* information about the scene are used to estimate the actual object motion and structure. It has been observed that this approach is very sensitive to noise since it is based on the use of differential operators. Also, good results are obtained only for smooth and small motion in the scene [5], [6].

The alternative approach, which is usually followed in the computer vision literature, basically consists of three steps [4]: extracting a set of characteristic points on the object called features, establishing a correspondence between points in adjacent frames by matching the features, and finally, computing the structure and motion parameters based on the feature matches.

Our attention will focus on low bit-rate coding of videoconference sequences. Because of the highly constrained content of the considered video material and of the low rate of image acquisition, which allows for large motion between successive frames, feature-based methods are very good candidates for motion estimation.

In this paper, we will consider a model which describes the global motion of the objects in the scene, and we will show that such a model is effective to perform the motion compensation step in a video coder. In particular, it can be used in an object-oriented coder [3] to perform the motion compensation of model compliance objects.

In our work, we exploit some of the results on the problem of "structure and motion" recovery that has been addressed in the field of computer vision [7]–[15]. This problem arises, for example, in applications such as autonomous navigation and robot vision. The most basic formulation of the "structure and motion" recovery problem consists of estimating the translation and rotation parameters and structure of an object in a 3-D space.

In this paper, we show that the motion equations of a rigid body can be formulated as a nonlinear dynamic system whose state is represented by the motion parameters and by the scaled depths of the object feature points. An extended Kalman filter is then used to estimate the object motion and structure, represented in our formulation by the scaled depths of the features, from which successive frames can be predicted. In our approach, the 3-D object structure and the motion parameters are estimated *simultaneously*, rather than computed from inaccurate measurements and estimates, as is often done

Manuscript received September 1, 1996; revised March 1, 1997.

The authors are with the Dipartimento di Elettronica e Informatica, Università di Padova, Padova 35131, Italy.

Publisher Item Identifier S 0733-8716(98)00308-4.

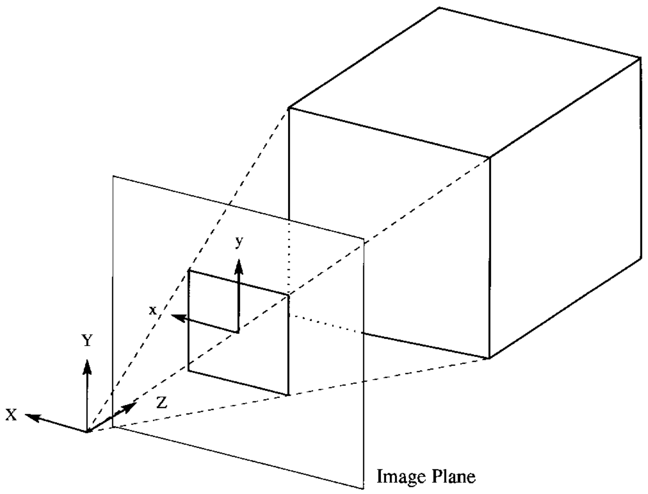


Fig. 1. Reference coordinate system.

in the literature [9], [12], [16]. The inclusion of the shape parameters in the estimation procedure adds a set of constraints to the filter equations that appear to be essential for reliable motion estimation.

Some authors recently proposed the simultaneous estimation of motion and structure [17], [18], and demonstrated its appropriateness for motion estimation. Our recursive formulation permits us to take into account previous measurements in the estimation process. As a matter of fact, the optimization procedure of the Kalman filter can make use of the motion and structure information from the past entire sequence, rather than from a few frames as done in [17] and [18]. The experimental results confirm that the proposed technique can give reliable estimates in the presence of noise and despite abrupt motion variations. In particular, our formulation proved to be essential for motion compensation of real-world video sequences.

In Section II, we review two motion estimation methods already reported in the literature, and we describe the proposed algorithm. In Sections III and IV, we present some experimental results on synthetic and real-world video sequences, confirming the increased robustness of the proposed method compared to the other considered solutions. In Section V, some conclusions are drawn.

II. Problem Formulation

In this section, we formulate the problem of the estimation of the 3-D motion parameters of a rigid object from perspective projections of object points onto the image plane. In particular, we review two estimation methods already proposed in the literature, and present an original formulation that appears to have superior performance in the case of noisy measurements. The new formulation was essential for successful motion estimation of real-world video sequences.

In the following, we suppose that the Cartesian reference system is centered at the pupil of the observer, the Z axis points forward and coincides with the optical axis, while the X and Y axes are parallel to the image plane and form with Z a right-handed reference (see Fig. 1).

Let $\boldsymbol{X}_i(t) = [X_i(t), Y_i(t), Z_i(t)]^T$ denote the coordinates of the generic point i of a rigid body at time t. The velocity of any point i of the rigid body can be represented by the sum of a translation velocity $\dot{\boldsymbol{X}}_O(t)$ and a rotation velocity, namely

$$\dot{\boldsymbol{X}}_i(t) = \boldsymbol{\Omega}(t) \wedge \boldsymbol{X}_i(t) + \dot{\boldsymbol{X}}_O(t) \tag{1}$$

where $\boldsymbol{\Omega}(t) = [\Omega_X(t), \Omega_Y(t), \Omega_Z(t)]^T$ is the vector of the angular velocities. Thus, six parameters are sufficient to characterize the motion. We can rewrite (1) in matrix form as

$$\dot{\boldsymbol{X}}_i(t) = \tilde{\boldsymbol{\Omega}}(t)\boldsymbol{X}_i(t) + \dot{\boldsymbol{X}}_O(t) \tag{2}$$

where

$$\tilde{\boldsymbol{\Omega}}(t) = \begin{bmatrix} 0 & -\Omega_Z(t) & \Omega_Y(t) \\ \Omega_Z(t) & 0 & -\Omega_X(t) \\ -\Omega_Y(t) & \Omega_X(t) & 0 \end{bmatrix} \tag{3}$$

is a skew symmetric matrix. The continuous time equation (2) can be solved to derive a discrete-time equation for $\boldsymbol{X}_i(t)$, namely

$$\boldsymbol{X}_i(t+1) = \boldsymbol{R}(t)\boldsymbol{X}_i(t) + \boldsymbol{T}(t). \tag{4}$$

If $\boldsymbol{\Omega}(t)$ is constant between t and $t+1$, as we will assume in the following, we have, in particular,

$$\begin{aligned} \boldsymbol{R}(t) &= e^{\tilde{\boldsymbol{\Omega}}(t)} \\ \boldsymbol{T}(t) &= [T_X(t), T_Y(t), T_Z(t)]^T \\ &= \int_t^{t+1} e^{\tilde{\boldsymbol{\Omega}}(t)(t+1-\tau)} \dot{\boldsymbol{X}}_O(\tau)\, d\tau. \end{aligned} \tag{5}$$

Let $\boldsymbol{X}_i(t)$ denote the vector of the coordinates of point i on the image plane at time t. The coordinates on the image plane are related to the 3-D coordinates by perspective projection, as shown in Fig. 1. Assuming a focal length equal to one, we obtain

$$\boldsymbol{x}_i(t) = \frac{\boldsymbol{X}_i(t)}{Z_i(t)} = \begin{bmatrix} X_i(t)/Z_i(t) \\ Y_i(t)/Z_i(t) \\ 1 \end{bmatrix}. \tag{6}$$

Using (4), one can easily derive

$$\boldsymbol{x}_i(t+1) = \frac{\boldsymbol{R}(t)Z_i(t)\boldsymbol{x}_i(t) + \boldsymbol{T}(t)}{\boldsymbol{R}_3(t)Z_i(t)\boldsymbol{x}_i(t) + T_Z(t)} \tag{7}$$

where $\boldsymbol{R}_3(t)$ denotes the third row of matrix $\boldsymbol{R}(t)$ and $T_Z(t)$ is the third component of vector $\boldsymbol{T}(t)$, as specified by (5). Equation (7) gives the position on the image plane of the projected point i at time $t+1$ when one knows its position at time t, the rotation matrix $\boldsymbol{R}(t)$, the translation vector $\boldsymbol{T}(t)$, and the depth $Z_i(t)$.

It is clear from (7) that the projected point coordinates at time $t+1$ depend on the point depth $Z_i(t)$ as well as on the motion parameters $\boldsymbol{R}(t)$ and $\boldsymbol{T}(t)$. Note that coordinates $Z_i(t)$ describe the 3-D shape of the object. Since we are dealing with rigid bodies, the object shape does not change in time, and therefore it is possible to calculate $Z_i(t+1)$ and $Z_i(t)$ through a simple triangulation process, once the motion parameters $\boldsymbol{R}(t)$, $\boldsymbol{T}(t)$ and the projections $\boldsymbol{x}_i(t)$, $\boldsymbol{x}_i(t+1)$ are known.

In the application at hand, i.e., motion estimation in a video coding system, we can determine the projections onto the image plane of a set of characteristic points, or *features*, of the 3-D object in successive frames. This requires finding a set of point projections which can be easily tracked on the image sequence, typically located along edges or inside highly textured areas. From the measured coordinates $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_i(t+1)$ in successive frames, we will show that it is indeed possible to compute the motion parameters $\boldsymbol{R}(t), \boldsymbol{T}(t)$ and the 3-D structure parameters $Z_i(t)$ up to a scale factor. In a practical environment, projections $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_i(t+1)$ will be affected by observation noise, and therefore the measurement of parameters becomes a typical estimation problem.

As a consequence of the feature-based approach, the structure parameters $Z_i(t)$ will be estimated only for a limited set of points. For the pixels that do not correspond to feature projections, the use of (7) requires that the object structure is extrapolated from $Z_i(t)$. Our approach is to suppose that the objects of the scene have smooth surfaces: as a consequence, the depth of a generic point is approximated by means of a weighted sum of the depths of the neighbor feature points.

In the next subsections, we will describe three algorithms that solve the problem of motion and structure estimation. The first is due to Longuet-Higgins [9], and only uses information from two consecutive frames. The method is not recursive, and does not pose any convergence problem. On the other hand, this procedure proved to be very sensitive to feature position measurement noise.

The second algorithm, which was recently introduced by Soatto *et al.* [12], defines the Longuet-Higgins algorithm in a recursive formulation based on the Kalman filter. In our simulations on synthetic and real-world video sequences, the algorithm performance proved to be significantly better than that of the original approach by Longuet-Higgins, but still not adequate in the case of very noisy measurements and for motion estimation of real-world video sequences.

The last algorithm is original, and is the main contribution of the paper. It is based on the Kalman filter, and includes object shape parameters into the filter state equations. In particular, the object feature scaled depths are *estimated* by the algorithm, together with the motion parameters, rather than *calculated* via a triangulation process from noisy measurements and imprecise estimates of $\boldsymbol{R}(t)$ and $\boldsymbol{T}(t)$. In our experiments, we found that this approach gives reliability to the estimation procedure, at the expense of the necessity to send the scaled depths to the decoder as part of the filter state. Despite the additional cost, the proposed procedure appears to be very promising in comparison to more common block-based motion estimation schemes.

A. Motion and Structure Estimation from Two Frames

Longuet-Higgins developed an algorithm that performs motion and structure estimation using two consecutive frames [9]. Suppose we are given the perspective projections of N feature points of the rigid body in two consecutive frames. From (4), we deduce that the vectors $\boldsymbol{X}_i(t+1)$, $\boldsymbol{R}(t)\boldsymbol{X}_i(t)$, and $\boldsymbol{T}(t)$ are coplanar. As a consequence, their triple product is zero, namely

$$\boldsymbol{X}_i(t+1)^T(\boldsymbol{T}(t) \wedge (\boldsymbol{R}(t)\boldsymbol{X}_i(t))) = 0, \qquad i = 1, \cdots, N. \quad (8)$$

This relation holds for the projected coordinate vectors $\boldsymbol{x}_i(t+1)$ and $\boldsymbol{x}_i(t)$ in place of $\boldsymbol{X}_i(t+1)$ and $\boldsymbol{X}_i(t)$. Hence, we can easily rewrite (8) for $\boldsymbol{x}_i(t+1)$ and $\boldsymbol{x}(t)$ in the form of the "epipolar constraint"

$$\boldsymbol{x}_i(t+1)^T\boldsymbol{Q}\boldsymbol{x}_i(t) = 0, \qquad i = 1, \cdots, N \quad (9)$$

where we defined

$$\boldsymbol{Q}(t) \triangleq \tilde{\boldsymbol{T}}(t)\boldsymbol{R}(t), \quad (10)$$

$$\tilde{\boldsymbol{T}}(t) \triangleq \begin{bmatrix} 0 & -T_Z(t) & T_Y(t) \\ T_Z(t) & 0 & -T_X(t) \\ -T_Y(t) & T_X(t) & 0 \end{bmatrix}. \quad (11)$$

Matrix $\boldsymbol{Q}$ in (10), where we drop the dependence from t for notation simplicity, is referred to in the literature as the "essential matrix."

We note that the epipolar constraint is linear in $\boldsymbol{Q}$. By building a column vector $\boldsymbol{q}$ by stacking one after the other the transposed rows of $\boldsymbol{Q}$, we can arrange the epipolar constraints for the N feature points in the form of a linear system of equations, namely

$$\boldsymbol{\chi}(\boldsymbol{x}(t), \boldsymbol{x}(t+1))\boldsymbol{q} = 0. \quad (12)$$

Here, $\boldsymbol{\chi}$ is an $N \times 9$ matrix whose ith row is built from the feature projected coordinate components, namely,

$$[x_i(t+1)x_i(t), x_i(t+1)y_i(t), x_i(t+1), y_i(t+1)x_i(t), \\ y_i(t+1)y_i(t), y_i(t+1), x_i(t), y_i(t), 1].$$

In summary, from known feature projection coordinates on the image plane, we can build matrix $\boldsymbol{\chi}(\boldsymbol{x}(t), \boldsymbol{x}(t+1))$ and solve (12) to find the essential matrix. If $N \geq 8$ and the feature distribution is not pathologic, the solution of the homogeneous system (12) is a subspace of dimension one. To determine $\boldsymbol{Q}$, up to its sign, we need an additional constraint, and in the following, we will assume $\|\boldsymbol{T}\| = 1$. This ambiguity is not simply a mathematical artifact, but a consequence of the fact that a point at depth Z translating by a vector $\boldsymbol{T}$ has exactly the same projected coordinates on the image plane as a point at depth αZ translating by $\alpha\boldsymbol{T}$. Therefore, matrix $\boldsymbol{Q}$ can be determined only to within a scale factor. It is important to note that the rank of matrix $\boldsymbol{\chi}$ is exactly eight when the measurements are not corrupted by noise, while it becomes nine in the presence of noise. In this case, system (12) does not have any nontrivial solution, and one has to estimate $\boldsymbol{q}$ by means of squared error minimization.

Once $\boldsymbol{Q}$ is known, the necessity of factoring it into $\boldsymbol{R}$ and $\tilde{\boldsymbol{T}}$ arises. This can be done by computing the singular value decomposition of the 3×3 matrix $\boldsymbol{Q}$ [19]. As a matter of fact, it is immediate to verify that we can write $\tilde{\boldsymbol{T}}$ as

$$\tilde{\boldsymbol{T}} = \boldsymbol{V}\boldsymbol{Z}\boldsymbol{\Lambda}\boldsymbol{V}^T \quad (13)$$

where

$$\boldsymbol{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (14)$$

and $\boldsymbol{V}$ is any orthogonal matrix whose third column is $\boldsymbol{T}$. The interpretation of (13) derives from the fact that $\boldsymbol{T} \wedge \boldsymbol{v} = \tilde{\boldsymbol{T}}\boldsymbol{v}$

for all vectors $\boldsymbol{v}$, and that the outer product can indeed be computed by first changing the coordinate system so that the third axis is aligned with $\boldsymbol{T}$, by computing the transformed vector $\boldsymbol{v}$ component perpendicular to $\boldsymbol{T}$ (this is done by setting to zero its third coordinate), by rotating the result by 90°, and finally by changing the coordinates back to the original coordinate system. All of this is done in (13) by multiplications with matrices $\boldsymbol{V}^T$, $\boldsymbol{\Lambda}$, $\boldsymbol{Z}$, and $\boldsymbol{v}$, respectively.

Substituting in (10), we obtain

$$\boldsymbol{Q} = \boldsymbol{VZ\Lambda V}^T\boldsymbol{R} = \boldsymbol{U\Lambda W}^T. \tag{15}$$

Equation (15) represents the singular value decomposition of the essential matrix [20]. Due to the indetermination in the sign of $\boldsymbol{Q}$, we will have to consider two possible solutions for matrix $\boldsymbol{R}$, namely

$$\boldsymbol{R} = \boldsymbol{U}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & s \end{bmatrix}\boldsymbol{W}^T$$

and

$$\boldsymbol{R} = \boldsymbol{U}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & s \end{bmatrix}\boldsymbol{W}^T$$

where $s = \det(\boldsymbol{U})\det(\boldsymbol{W})$, and two possible solutions $\boldsymbol{T} = \pm\boldsymbol{U}_3$, where $\boldsymbol{U}_3$ denotes the last column of $\boldsymbol{U}$. It is easy to verify from (15) that all of the four possibilities, when combined using the product in (10), actually result in $\boldsymbol{Q}$ or $-\boldsymbol{Q}$, and are therefore acceptable in principle. We will see in the following that the positive depth constraint permits us to resolve the ambiguity for $\boldsymbol{R}$ and $\boldsymbol{T}$ (see below).

The depths of the feature points $Z_i(t)$, $Z_i(t+1)$ can be computed by triangulation. The corresponding equations can be derived by computing the vector product of (4) with $\boldsymbol{x}_i(t+1)$ which, of course, gives zero

$$\begin{aligned} &\boldsymbol{X}_i(t+1) \wedge \boldsymbol{x}_i(t+1) \\ &\quad = \boldsymbol{RX}_i(t) \wedge \boldsymbol{x}_i(t+1) + \boldsymbol{T} \wedge \boldsymbol{x}_i(t+1) \\ &\quad = Z_i(t)(\boldsymbol{Rx}_i(t) \wedge \boldsymbol{x}_i(t+1)) + \boldsymbol{T} \wedge \boldsymbol{x}_i(t+1) = 0. \end{aligned}$$

Then, we compute the scalar product with $\boldsymbol{Rx}_i(t) \wedge \boldsymbol{x}_i(t+1)$ and obtain

$$Z_i(t) = -\frac{\langle \boldsymbol{T} \wedge \boldsymbol{x}_i(t+1), \boldsymbol{Rx}_i(t) \wedge \boldsymbol{x}_i(t+1)\rangle}{\|\boldsymbol{Rx}_i(t) \wedge \boldsymbol{x}_i(t+1)\|^2}. \tag{16}$$

In a similar way, one can show that

$$Z_i(t+1) = -\frac{\langle \boldsymbol{T} \wedge \boldsymbol{Rx}_i(t), \boldsymbol{x}_i(t+1) \wedge \boldsymbol{Rx}_i(t)\rangle}{\|\boldsymbol{x}_i(t+1) \wedge \boldsymbol{Rx}_i(t)\|^2}. \tag{17}$$

Equations (16) and (17) allow us to resolve the ambiguity on $\boldsymbol{T}$ and $\boldsymbol{R}$. Actually, the visibility of the features implies that $Z_i(t) > 0$ and $Z_i(t+1) > 0$. This condition is satisfied if, and only if, both the correct sign of $\boldsymbol{T}$ and the right solution for $\boldsymbol{R}$ are chosen. It should be noted again that $Z_i(t)$ and $Z_i(t+1)$ are determined to within a positive scale factor given by the norm of $\boldsymbol{T}$. Such an unknown scaling factor does not prevent us from using (7) in a motion compensation scheme, as soon as the depth and the translation are scaled by the same quantity.

B. Motion Estimation in Local Coordinates

Soatto *et al.* [12] impose the epipolar constraint to compute motion and structure from measurements on a sequence of more frames. They formulate the problem in terms of the identification of a nonlinear implicit dynamic system. As a matter of fact, one can suppose that abrupt changes of motion are not very likely in typical video sequences. This makes it convenient to use the information of more than two frames to estimate motion and structure with greater accuracy. Using (4) and (5), one can derive an expression for $\boldsymbol{R}(t)$ as a function of $\boldsymbol{\Omega}(t)$ (Rodrigues' formula [11]). The essential matrix $\boldsymbol{Q}(t)$ can therefore be represented as a function of the translation $\boldsymbol{T}(t)$ and of the angular velocity $\boldsymbol{\Omega}(t)$.

As noted before, the norm of the translation is not observable from measurements on the image plane. We will estimate only its direction, which we will represent in spherical coordinates by specifying two angles $T_\theta(t)$ and $T_\phi(t)$ on the unit sphere. In summary, $\boldsymbol{Q}(t)$ can be computed from a five-dimensional (5-D) vector, namely

$$\boldsymbol{\xi}(t) = \left[T_\theta(t), T_\phi(t), \boldsymbol{\Omega}^T\right]^T. \tag{18}$$

In the lack of assumptions about motion, a simple random walk model is assumed for the dynamics of $\xi(t)$

$$\boldsymbol{\xi}(t+1) = \boldsymbol{\xi}(t) + \boldsymbol{n}_\xi(t) \tag{19}$$

where $\boldsymbol{n}_\xi(t)$ is the model noise. The epipolar equation (12) represents a constraint between feature coordinate measurements and the motion vector $\boldsymbol{\xi}(t)$. We therefore obtain a nonlinear implicit model for motion dynamics, namely

$$\begin{cases} \boldsymbol{\xi}(t+1) = \boldsymbol{\xi}(t) + \boldsymbol{n}_\xi(t), \\ \boldsymbol{\chi}(\boldsymbol{y}(t) - \boldsymbol{w}(t), \boldsymbol{y}(t+1) - \boldsymbol{w}(t+1))\boldsymbol{q}(\boldsymbol{\xi}(t)) = 0. \end{cases} \tag{20}$$

In (20), $\boldsymbol{y}(t) = \boldsymbol{x}(t) + \boldsymbol{w}(t)$ represents the noisy measurement of the feature position $\boldsymbol{x}(t)$ and $\boldsymbol{w}(t)$ is the observation noise. The estimation scheme for the system state $\xi(t)$ is based on an implicit extended Kalman filter (IEKF) [21]. The equations for the IEKF update and prediction steps are given in the Appendix.

As noted above, the correspondence between $\boldsymbol{\xi}(t)$ and $\boldsymbol{Q}(t)$ is not injective. To avoid that the IEKF may converge to the wrong solution, one has to check that the positive depth condition is satisfied at each step of the Kalman filter. Once the estimate $\hat{\boldsymbol{\xi}}(t \mid t)$ of $\xi(t)$ is computed, it is possible to calculate $\boldsymbol{R}$ and $\boldsymbol{T}$, and apply the triangulation equations (16) and (17) to find the depths.

C. The Proposed Algorithm

In this section, we propose an algorithm that simultaneously estimates the object motion and structure, without using triangulation. This is obtained by including depths in the state vector of the dynamic system describing the motion of the object.

We define by $\bar{Z}(t) = \Sigma_{i=1}^N Z_i(t)/N$ the average depth, by $s_i(t) = Z_i(t)/\bar{Z}(t)$ the scaled depth, and by $\tilde{\boldsymbol{T}}(t) = \boldsymbol{T}(t)/\bar{Z}(t)$ the scaled translation. Using this position, (7) becomes

$$\boldsymbol{x}_i(t+1) = \frac{\boldsymbol{R}(t)s_i(t)\boldsymbol{x}_i(t) + \tilde{\boldsymbol{T}}(t)}{\boldsymbol{R}_3(t)s_i(t)\boldsymbol{x}_i(t) + \tilde{T}_Z(t)}. \tag{21}$$

We can interpret (21) as an implicit relation between the coordinates $\boldsymbol{x}_i(t)$ and the state $\boldsymbol{R}(t), \boldsymbol{T}(t), s_i(t), i = 1, \cdots, N$, of a nonlinear system governing the motion of the rigid body. Our objective is to estimate the system state, i.e., the object motion parameters and the scaled depths of the features, from the feature projections $\boldsymbol{x}_i(t)$. In the following, we derive the state update equations for the system.

As in the previous subsection, we will use $\boldsymbol{\Omega}(t)$ instead of $\boldsymbol{R}(t)$ in the state equations, and assume for its dynamics a random walk model

$$\boldsymbol{\Omega}(t+1) = \boldsymbol{\Omega}(t) + \boldsymbol{n}_{\boldsymbol{\Omega}}(t) \tag{22}$$

where $\boldsymbol{n}_{\boldsymbol{\Omega}}(t)$ is a zero-mean white noise. The update equation for the scaled translation $\tilde{\boldsymbol{T}}(t)$ can be derived by rewriting the third equation of (7)

$$\begin{aligned} Z_i(t+1) &= \boldsymbol{R}_3(t)\boldsymbol{X}_i(t) + T_Z(t) \\ &= \Big(\boldsymbol{R}_3(t)s_i(t)\boldsymbol{x}_i(t) + \tilde{T}_Z(t)\Big)\bar{Z}(t) \end{aligned} \tag{23}$$

and by assuming a random walk model also for $\boldsymbol{T}(t)$. We obtain

$$\tilde{\boldsymbol{T}}(t+1) = \frac{\boldsymbol{T}(t+1)}{\bar{Z}(t+1)} = \frac{\tilde{\boldsymbol{T}}(t)}{\boldsymbol{R}_{3\cdot}(t)\bar{\boldsymbol{x}}(t) + \tilde{T}_Z(t)} + \boldsymbol{n}_{\tilde{\boldsymbol{T}}}(t) \tag{24}$$

where $\bar{\boldsymbol{x}}(t) = (1/N)\,\Sigma_{i=1}^N s_i(t)\boldsymbol{x}_i(t)$ and $\boldsymbol{n}_{\tilde{\boldsymbol{T}}}(t)$ is assumed to be a zero-mean white noise.

From (23), we derive the update equation for $s_i(t)$

$$s_i(t+1) = \frac{Z_i(t+1)}{\bar{Z}(t+1)} = \frac{\boldsymbol{R}_{3\cdot}(t)s_i(t)\boldsymbol{x}_i(t) + \tilde{T}_Z(t)}{\boldsymbol{R}_{3\cdot}(t)\bar{\boldsymbol{x}}(t) + \tilde{T}_Z(t)}. \tag{25}$$

Moreover, we have the constraint

$$\frac{1}{N}\sum_{i=1}^N s_i(t) = \frac{1}{N}\sum_{i=1}^N \frac{Z_i(t)}{\bar{Z}(t)} = 1. \tag{26}$$

In summary, the system equations are

$$\begin{cases} \boldsymbol{\Omega}(t+1) = \boldsymbol{\Omega}(t) + \boldsymbol{n}_{\Omega}(t) \\ \tilde{\boldsymbol{T}}(t+1) = \frac{\tilde{\boldsymbol{T}}(t)}{\boldsymbol{R}_{3\cdot}(t)\bar{\boldsymbol{x}}(t)+\tilde{T}_3(t)} + \boldsymbol{n}_{\tilde{\boldsymbol{T}}}(t) \\ s_i(t+1) = \frac{\boldsymbol{R}_{3\cdot}(t)s_i(t)\boldsymbol{x}_i(t)+\tilde{T}_3(t)}{\boldsymbol{R}_{3\cdot}(t)\bar{\boldsymbol{x}}(t)+\tilde{T}_3(t)} + n_{s_i}(t) \\ \displaystyle\sum_{i=1}^N s_i(t) = N \\ \boldsymbol{x}_i(t+1) = \frac{\boldsymbol{R}(t)s_i(t)\boldsymbol{x}_i(t)+\tilde{\boldsymbol{T}}(t)}{\boldsymbol{R}_{3\cdot}(t)s_i(t)\boldsymbol{x}_i(t)+\tilde{T}_3(t)} + \boldsymbol{n}_x(t) \end{cases} \tag{27}$$

where $n_{s_i}(t)$ and $\boldsymbol{n}_x(t)$ are model noises that may take into account slow deformations of the object.

Defining the system state by $\boldsymbol{\xi}(t) = [\boldsymbol{\Omega}(t)^T, \tilde{\boldsymbol{T}}(t)^T, s_1(t), \cdots, s_N(t)]^T$ and observations by $\boldsymbol{y}(t) = [\boldsymbol{x}_1(t)^T, \cdots, \boldsymbol{x}_N(t)^T, \boldsymbol{x}_1(t+1)^T, \cdots, \boldsymbol{x}_N(t+1)^T]^T + \boldsymbol{w}(t)$, where $\boldsymbol{w}(t)$ is the observation noise, we may rewrite (27) as

$$\begin{cases} \boldsymbol{\xi}(t+1) = f(\boldsymbol{\xi}(t), \boldsymbol{y}(t)) + \tilde{\boldsymbol{n}}(t) \\ h(\boldsymbol{\xi}(t), \boldsymbol{y}(t) - \boldsymbol{w}(t)) = 0 \end{cases} \tag{28}$$

where $\tilde{\boldsymbol{n}}(t)$ is a function of the noises in (27) and of $\boldsymbol{w}(t)$.

System (28) is nonlinear and implicit; therefore, we can estimate and predict its state by means of the IEKF [21].

The state estimate $\hat{\boldsymbol{\xi}}(t \mid t)$ can be used to predict the feature positions at time $t+1$ from their positions at time t by means of (21). We remark that the inclusion of the scaled depths $s_i(t)$ in the filter state is essential to obtain reliable motion estimates in the case of very noisy observations and of simulations with real video sequences, as will be confirmed by the experimental results of the next sections.

III. Application to Synthetic Sequences

In this section, we will present some simulation results relative to the three methods described in the previous section and relative to synthetic image sequences.

In the first experiment, we considered a set of 30 points randomly selected inside a cube of side 1 m with centroid positioned 2.5 m ahead of the viewer. This cloud of points undergoes a rotational motion around a vertical axis passing through its center of mass. In the camera coordinate system, this corresponds to a rotation around the vertical axis with angular velocity $\Omega_Y = 3°$/frame, and a translation with velocity $2.5(1 - \cos\Omega_Y)$ m/frame along the Z axis and $-2.5\sin\Omega_Y$ m/frame along the X axis. In our simulations, we considered a 60-frame-long sequence, and we observed the scene using a camera with a visual field of 52° and CIF resolution (288 × 352 pixels).

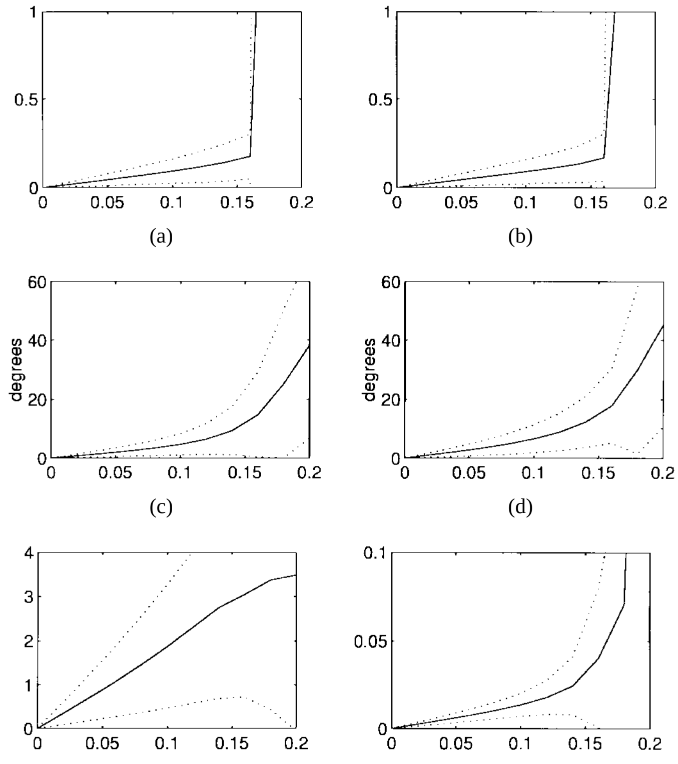
The cloud of points was projected onto the image plane, and a zero-mean Gaussian white noise was added to the point coordinates (x_i, y_i), $i = 1, \cdots, 30$, to simulate the measure error. For each of the three algorithms described in the previous section, we report the results obtained by averaging over 50 experiments. We used an initial null estimate for the translation $\tilde{\boldsymbol{T}}$ and the angular velocity $\boldsymbol{\Omega}$ while the initial estimates of the scaled depths s_i were set to one.

Figs. 2–4 show the estimation errors of the motion parameters, the point positions, and the scaled depths for the last frame of the synthetic sequence versus the standard deviation of the added noise in pixels. The solid lines represent the mean values of the errors, while the dotted lines are obtained by adding and subtracting the error standard deviations. For each figure, the first plot (a) shows the magnitude of the angular velocity relative error, and the second plot (b) shows the magnitude of the translation relative error. The third (c) and fourth (d) plots are relative to the angular velocity direction error and to the translation direction error (in degrees), respectively. The fifth plot (e) shows the point position (on the image plane) prediction error (in pixels), and the last plot (f) shows the scaled point depths error.

It is interesting to note that the Longuet-Higgins algorithm and the local coordinate algorithm do not converge when the standard deviation of the added noise is greater than about 0.15 pixels, while the proposed method maintains convergence and exhibits error values that increase almost linearly with the noise variance.

In the second experiment, we show that the proposed algorithm is capable of tracking fast motion changes in the objects in the scene.

A synthetic sequence of a rotating cloud of points with the same characteristics of the first experiment is generated. In

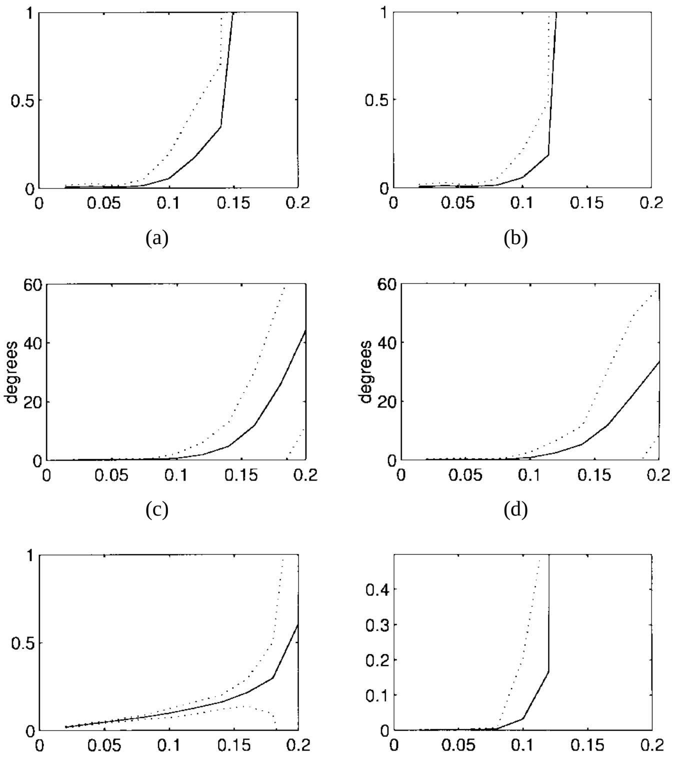


(a) (b)

(c) (d)

(e) (f)

Fig. 2. Estimation errors using Longuet-Higgins' algorithm. The solid lines are the mean values of the errors versus the standard deviation of the additive Gaussian noise. The dotted lines are obtained by adding and subtracting the error standard deviations to the mean values. (a) Angular velocity magnitude relative error. (b) Translation magnitude relative error. (c) Angular velocity direction error (degrees). (d) Translation direction error (degrees). (e) Feature position prediction error (pixels). (f) Scaled depths error.



(a) (b)

(c) (d)

(e) (f)

Fig. 3. Estimation errors using Soatto's algorithm. The solid lines are the mean values of the errors versus the standard deviation of the additive Gaussian noise. The dotted lines are obtained by adding and subtracting the error standard deviations to the mean values. (a) Angular velocity magnitude relative error. (b) Translation magnitude relative error. (c) Angular velocity direction error (degrees). (d) Translation direction error (degrees). (e) Feature position prediction error (pixels). (f) Scaled depths error.

this case, the sequence is 100 frames long. Initially, the object undergoes the same motion as in the previous experiment, and after 50 frames, it inverts its direction of rotation. We used an initial null estimate for the translation $\hat{\boldsymbol{T}}$ and the angular velocity $\boldsymbol{\Omega}$, while the initial estimates of the scaled depths s_i were set to one. Fig. 5 shows the estimates of the translational and rotational velocities as a function of the frame number. In the same figure, the estimated scaled depths s_1 and s_2 of two features are shown. The ground truth is plotted as a dotted line. As can be seen from the plots, the Kalman filter takes about 20 frames to converge. After that, it tracks the object motion even after the abrupt inversion at frame 50.

IV. Application to Video Sequences

In the motion estimation framework we consider, one can think to use (7) to predict at time $t+1$ the luminance of the pixel at position $\boldsymbol{x}_i(t+1)$ by using the luminance of the pixel $\boldsymbol{x}_i(t)$ in the frame at time t. Note that this luminance prediction procedure will not be exact in general, but only in the particular case of a Lambertian object surface, of uniform illumination, and of a pure object translation [22]. Moreover, we should take into account occlusions between objects. In practice, one can think that it is possible to obtain a good approximation for the pixel luminance at time $t+1$, and that it can be improved by coding the residual error between the actual luminance and its prediction. Moreover, the luminance regions that cannot be successfully predicted, such as shadows, specular surfaces, or those corresponding to new objects appearing in the scene, can be detected and coded as "model failure" regions. Similarly, image areas where local movement occurs, like lips or eyes in typical videoconference sequences, will cause large residual errors that must be properly detected and coded.

To test the proposed estimator with a real video sequence, we need to choose the feature points of the object in the first frame and track them in the following frames. For this purpose, we used a multiresolution version of Lucas–Kanade's algorithm [10], [23]. This procedure consists of approximating the luminance at time t around the point at position $\boldsymbol{x}$ with a differentiable function $I(\boldsymbol{x},t)$. In addition, one supposes that the luminance variations are due only to translations. Therefore, denoting by $\boldsymbol{d}$ the displacement of $\boldsymbol{x}$ from time t to $t+1$, one can write

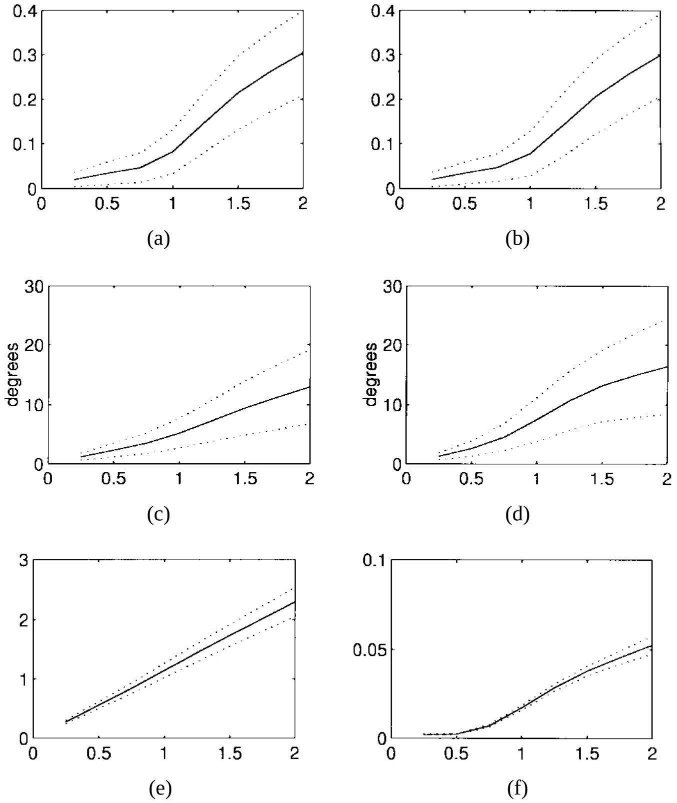
$$I(\boldsymbol{x},t) = I(\boldsymbol{x}-\boldsymbol{d},t+1) \simeq I(\boldsymbol{x},t+1) - \boldsymbol{g}^T\boldsymbol{d} \qquad (29)$$

where $\boldsymbol{g} = \operatorname{grad} I$. Using (29), we determine the minimum squared error solution for $\boldsymbol{d}$ over a region $\mathcal{W}$ around point $\boldsymbol{x}$. We find

$$\boldsymbol{G}\boldsymbol{d} = \boldsymbol{e} \qquad (30)$$

where

$$\boldsymbol{G} = \sum_{\boldsymbol{x}\in\mathcal{W}} \boldsymbol{g}\boldsymbol{g}^T \quad \boldsymbol{e} = \sum_{\boldsymbol{x}\in\mathcal{W}} (I(\boldsymbol{x},t) - I(\boldsymbol{x},t+1))\boldsymbol{g}. \qquad (31)$$



(a) (b) (c) (d) (e) (f)

Fig. 4. Estimation errors using the proposed algorithm. The solid lines are the mean values of the errors versus the standard deviation of the additive Gaussian noise. The dotted lines are obtained by adding and subtracting the error standard deviations to the mean values. (a) Angular velocity magnitude relative error. (b) Translation magnitude relative error. (c) Angular velocity direction error (degrees). (d) Translation direction error (degrees). (e) Feature position prediction error (pixels). (f) Scaled depths error.

System (30) allows one to find the displacement of a feature and, if the eigenvalues of the matrix $\boldsymbol{G}$ are nonzero, it has a unique solution. In practice, due to the presence of noise, to avoid ill conditioning of matrix $\boldsymbol{G}$ and to obtain a reliable solution, the eigenvalues must be greater than a threshold. This suggests considering only those feature points that correspond to local maxima of the minimum eigenvalue of $\boldsymbol{G}$.

In the first group of experiments, we tested the quality of motion compensation obtained with the proposed technique. The feature extraction and tracking procedure was applied to the CIF test video sequence "Miss America," temporally sampled at 15 frames/s. We suppose that the sequence is already segmented into three regions, corresponding to the head, the shoulders, and the background (see Fig. 6) [24]. In our tests, the sequence is manually segmented.

The features obtained using Lucas–Kanade's algorithm were classified into two groups corresponding to the regions of the head and the shoulders (see Fig. 6). The number of features selected and tracked were 27 and 14 for the regions corresponding to the head and the shoulders of "Miss America," respectively. As explained above, the estimator takes as input at each step the positions of the features on two consecutive frames, and yields the estimated state. In order to cope with possible feature occlusions, we always use in the algorithm the features extracted with the best matches. Thus, it is possible that some of the state variables in the Kalman filter are reinitialized from time to time to take into account the new entries in the feature set.

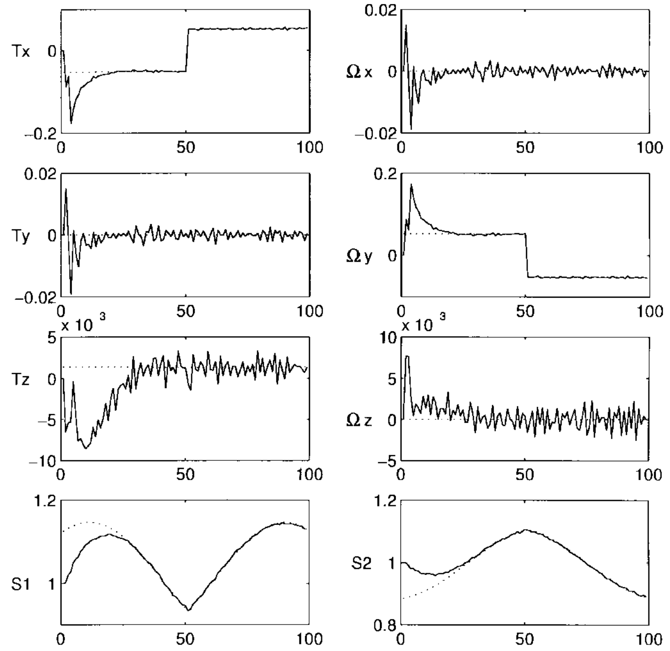


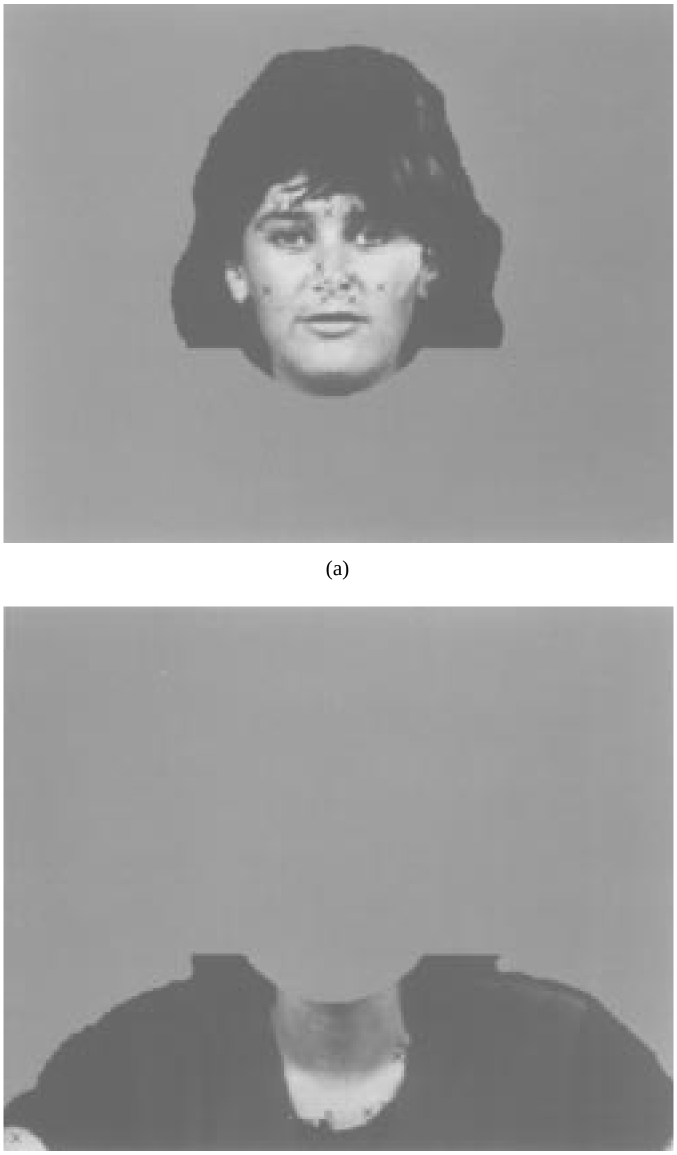
Fig. 5. Estimates of the motion parameter for the synthetic sequence.

For each region, the proposed Kalman filter was used to estimate the motion parameters and the scaled depths. Frame at time $t+1$ was predicted from frame at time t using the estimated parameters. To all of the image pixels that are not features, we assigned a scaled depth obtained as a weighted average of the estimates of the scaled depths $\hat{s}_i(t \mid t)$ of the feature points, namely

$$s(t) = \frac{\sum_{i=1}^{N} w_i \hat{s}_i(t \mid t)}{\sum_{i=1}^{N} w_i}. \tag{32}$$

The weights were empirically set to $w_i = (|x-x_i|+|y-y_i|)^{-3}$ to take into account the distance between the generic pixel coordinates (x, y) and the coordinates (x_i, y_i) of feature i. To each pixel $\boldsymbol{X}(t)$ of the three regions, we applied (21), using the corresponding motion parameters and estimated scaled depth to predict the pixel coordinates $\hat{\boldsymbol{x}}(t+1)$ at time $t+1$. The luminance value of pixel $\boldsymbol{x}(t+1)$ at time $t+1$ is set to the same luminance value of $\hat{\boldsymbol{x}}(t)$ at time t. We assume no motion in the background, and the corresponding pixels are simply replicated from time t to $t+1$.

We report the results relative to the prediction of frame 58 from the original frame 54 using the proposed algorithm. For comparison purposes, the results are compared with the prediction obtained using a block matching procedure. Block matching was performed using 16 × 16 blocks, motion vectors in the range −15 to +15, and half-pixel refinement. We also consider the case of using frame 54 as an estimate of frame 58, with no motion compensation.



(a)

(b)

Fig. 6. The two regions used for the test on the video sequence “Miss America.” The features computed by Lucas–Kanade’s algorithm are marked with crosses.

TABLE I
RESULTS FOR DIFFERENT PREDICTION METHODS APPLIED TO THE PREDICTION OF FRAME 58 OF THE VIDEO SEQUENCE “MISS AMERICA”

	Proposed method	Block matching	No compensation
MSE	17.3	10.4	73.6

In Table I, the mean-squared error (MSE) between frame 58 and its prediction is given for the three cases. We can see that the MSE of the proposed solution is slightly greater than the MSE obtained with block matching, although the visual quality of the image predicted using the 3-D model is superior and, as will be shown later in more detail, its coding can be more efficient. Fig. 7 shows the original frame 58 of “Miss America,” while Fig. 8 shows the frame predicted using the proposed method. In the presence of relevant local motion around the lips and the eyes, the prediction procedure can give less satisfactory results, as expected. As an example, the prediction of frame 50 from frame 46 gives an MSE $= 32.2$ with our method and an MSE $= 22.1$ with block matching.

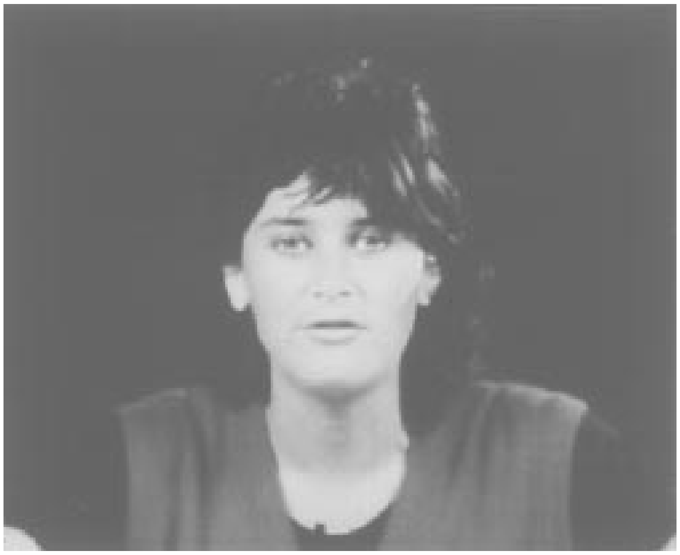


Fig. 7. Original frame 58 of “Miss America.”

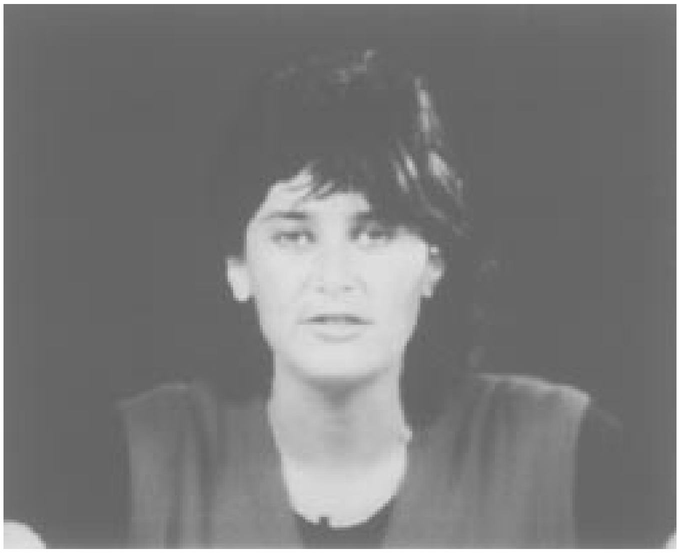


Fig. 8. Predicted frame 58 of “Miss America.”

The procedure was also tested on the video sequence “Flower Garden.” In this case, the scene contains only one rigid body (neglecting the flowers and the people movements), and segmentation is not required. We considered 150 features selected as in the previous test. For the feature tracking step, we obtained better results using a correlation method in place of Lucas–Kanade’s algorithm. In particular, we used blocks of 9 × 9 pixels around each feature position, and searched in the new frame for the blocks that minimize the squared error using half-pixel resolution. Figs. 9 and 10 show the original and the predicted frame 11 of “Flower Garden,” respectively, while Fig. 11 shows the frame prediction error. Table II summarizes the results for this experiment.

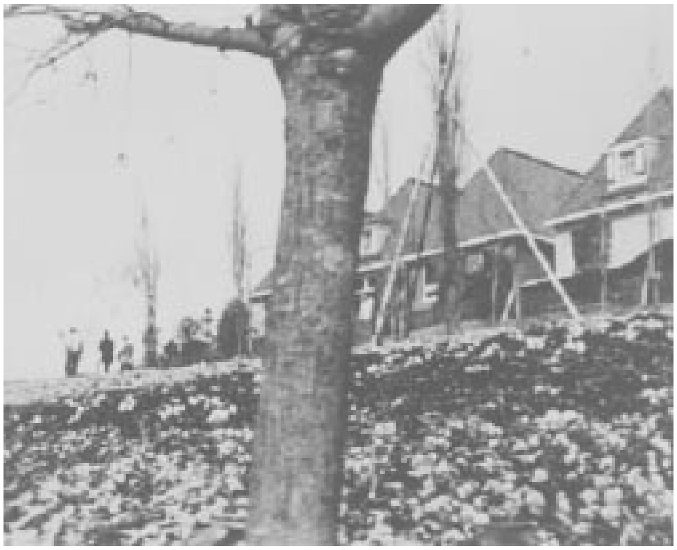


Fig. 9. Original frame 11 of “Flower Garden.”

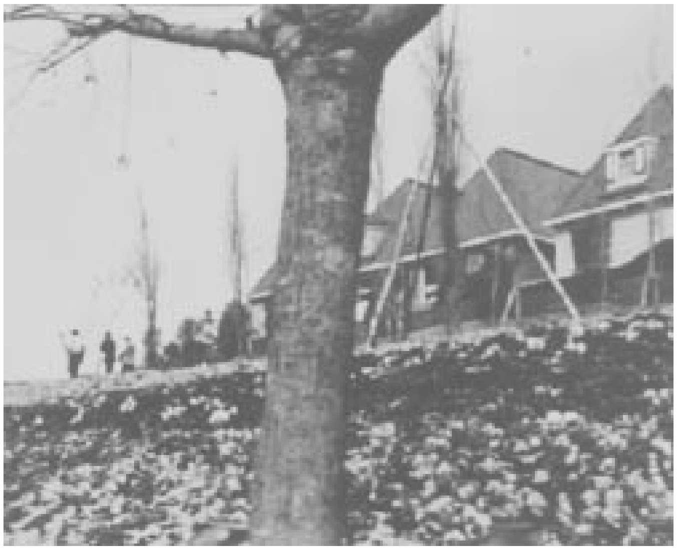


Fig. 10. Predicted frame 11 of “Flower Garden.”

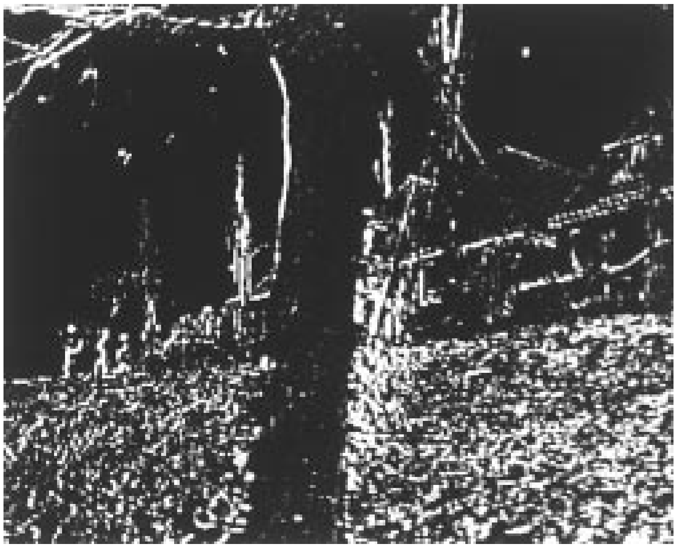


Fig. 11. Prediction error for frame 11 of “Flower Garden.”

TABLE II
RESULTS FOR DIFFERENT PREDICTION METHODS APPLIED TO THE PREDICTION OF FRAME 11 OF THE VIDEO SEQUENCE “FLOWER GARDEN”

	Proposed method	Block matching	No compensation
MSE	539	274	1310

TABLE III
AVERAGE NUMBER OF BITS REQUIRED TO CODE EACH OBJECT OF A PREDICTED FRAME

object	average # of bits
eyes and mouth	754
head	168.3
shoulders	82.4
borders	300
total	1304.7

The second group of experiments is relative to a coding scheme that is complete enough to give realistic indications on the performance of the proposed technique, but does not consider all of the possible difficulties that may arise in a practical implementation of a video coder. The test is relative to the image sequence “Miss America” at 15 frames/s. As for the first group of experiments, the sequence is manually segmented and the region borders are coded using the algorithm described in [25]. Inside the face, the region of the eyes, the mouth, and the chin are automatically detected using the algorithm presented in [26]. Two small elliptical regions around the eyes and a square region for the mouth and chin are labeled as model failure regions. The difference between these regions and their motion-compensated prediction is coded using an adaptation of the algorithm of [27] for the case of arbitrarily shaped regions. To avoid error propagation in the motion estimation procedure, we introduce a feedback scheme, and use the feature positions relative to the *decoded* previous image. The estimated state is an efficient way to code model compliance objects. We use arithmetic coding to code the innovation of the Kalman filter that is sufficient to reconstruct the motion parameters at the decoder. As mentioned above, feature occlusions are taken into account by reinitializing some of the filter state variables. A negligible amount of side information is used to notify such an event to the decoder.

The first frame of the sequence was coded in intraframe mode using a target bit rate of 12 kbits. A larger number of bits is used for the region of the head (0.4 bpp), while the region of the shoulders is coded with 0.2 bpp, in order to obtain a better quality for the region of greater interest. In this test, we used an intraframe every 15 coded frames (which correspond to 1 s) with an overall bit rate of 30 kbits/s. Table III shows the average number of bits used to code each region and the region borders for the 14 predicted images between two successive intraframes. In particular, the first entry in the table specifies the average number of bits needed to code the eyes and mouth region with the algorithm of [27]. The other entries

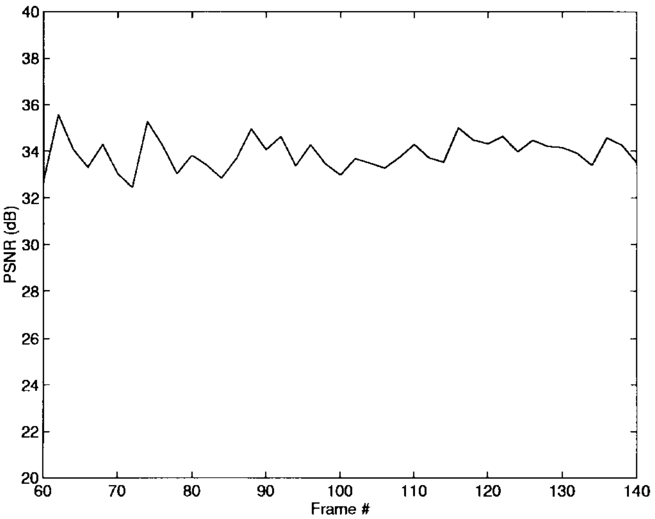


Fig. 12. PSNR for frames 60–140 of the reconstructed sequence "Miss America."

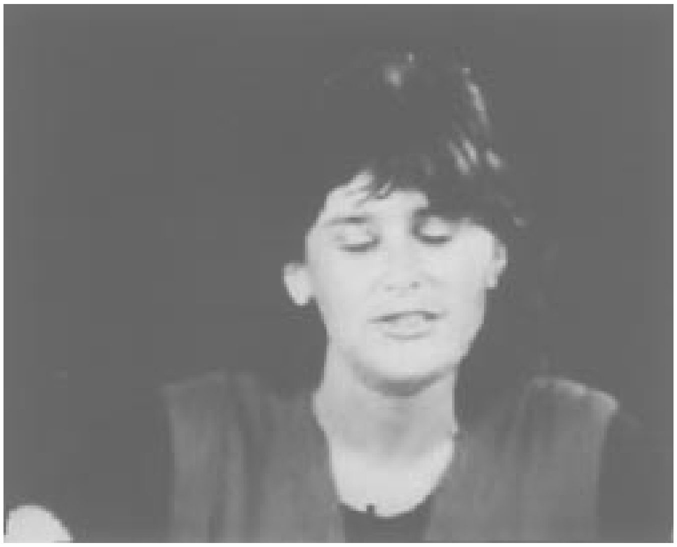


Fig. 13. Reconstructed frame 96 of "Miss America."

specify the number of bits used to code motion information and borders for the model compliance regions. We note that model compliance objects, which need only motion information, are coded with a small amount of bits in comparison with model failure objects. Fig. 12 shows the PSNR for frames 60–140 (where relatively large motion is present in the scene) of the reconstructed sequence. Although the PSNR figures may be inferior to those obtainable with other conventional schemes like H263, the visual quality is quite good and superior at low bit rates. Fig. 13 shows the reconstructed frame 96 of "Miss America." Besides the low bit rate, the visual quality of the decoded sequence is quite good, especially when compared to block-based coding schemes.

V. Conclusion

Statistical-based motion estimation has been widely used in computer vision [12], and more recently in video coding [28]. In this work, a modification of the scheme of [12] is applied to the problem of motion estimation for video coding. The proposed method compares favorably with respect to other algorithms that solve the problem of "structure and motion" estimation, and is very reliable in spite of noise.

The estimated motion parameters for each object in the scene, modeled as the projections of a 3-D rigid body, are used for frame prediction in a motion-compensated video coder. The constraints imposed by the model guarantee that the motion in the reconstructed video sequence is very natural compared to simpler and more common block-based schemes. Moreover, the simple parametrization of the model allows for a very efficient coding of motion information.

The results reported in this paper confirm that the proposed method could be conveniently used in a complete video coding scheme, with appropriate coding of the residual between the original and predicted frame. Of course, future work is required to solve the segmentation problem and to take into account occlusions and scene changes.

Appendix
Extended Kalman Filtering for Implicit Measurement Constraints

We are interested in building an estimator for the state process $\boldsymbol{\xi}(t)$ described by the difference equation

$$\boldsymbol{\xi}(t+1) = f(\boldsymbol{\xi}(t)) + \boldsymbol{v}(t), \qquad \boldsymbol{\xi}(0) = \boldsymbol{\xi}_0 \tag{33}$$

where $\boldsymbol{v}(t)$ is the model noise. We will assume that $\boldsymbol{v}(t)$ is zero-mean Gaussian white noise with autocorrelation matrix $\boldsymbol{R}_v$, i.e., $\boldsymbol{v}(t) \in \mathcal{N}(0, \boldsymbol{R}_v)$. We suppose there is a measurable quantity $\boldsymbol{x}(t)$ related to $\boldsymbol{\xi}(t)$ by means of the implicit constraint

$$h(\boldsymbol{\xi}(t), \boldsymbol{x}(t)) = 0. \tag{34}$$

We will assume that the functions f and h are differentiable, and that the quantity $\boldsymbol{x}(t)$ is known via a noisy measurement

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) + \boldsymbol{w}(t) \tag{35}$$

with $\boldsymbol{w}(t) \in \mathcal{N}(0, \boldsymbol{R}_w)$. In summary, we consider the model

$$\begin{cases} \boldsymbol{\xi}(t+1) = f(\boldsymbol{\xi}(t)) + \boldsymbol{v}(t), & \boldsymbol{\xi}(0) = \boldsymbol{\xi}_0 \\ h(\boldsymbol{\xi}(t), \boldsymbol{y}(t) - \boldsymbol{w}(t)) = 0. \end{cases} \tag{36}$$

Moreover, we will assume $\boldsymbol{v}(t)$, $\boldsymbol{w}(t)$, $\boldsymbol{\xi}_0$ incorrelated.

We will denote by $\hat{\boldsymbol{\xi}}(t \mid t)$ the estimate of the state $\xi(t)$ at time t based on measurements $\{\boldsymbol{y}(\tau): \tau \leq t\}$ and with $\hat{\boldsymbol{\xi}}(t+1 \mid t)$ the prediction of the state at time $t+1$ from measurements $\{\boldsymbol{y}(\tau): \tau \leq t\}$.

As will be seen, the equations for the IEKF are derived by linearizing functions f and h and using a standard Kalman filter estimator for the linearized model.

A. Prediction Step

We linearize the state dynamics (33) around the estimate $\hat{\boldsymbol{\xi}}(t \mid t)$. Defining the Jacobian matrix

$$\boldsymbol{F}(t) \triangleq \left. \frac{\partial f}{\partial \boldsymbol{\xi}} \right|_{\hat{\boldsymbol{\xi}}(t|t)} \tag{37}$$

and neglecting higher order terms in the Taylor expansion, we obtain from (33)

$$\boldsymbol{\xi}(t+1) = f(\hat{\boldsymbol{\xi}}(t \mid t)) + \boldsymbol{F}(t)(\boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t \mid t)) + \boldsymbol{v}(t). \tag{38}$$

The best estimate of $\boldsymbol{\xi}(t+1)$ based on measurements $\{\boldsymbol{y}(\tau) : \tau \leq t\}$ is, from (38)

$$\text{a)} \quad \hat{\boldsymbol{\xi}}(t+1 \mid t) = f(\hat{\boldsymbol{\xi}}(t \mid t)). \tag{39}$$

The prediction error $\delta\boldsymbol{\xi}(t) = \boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t \mid t)$ obeys the dynamics

$$\begin{aligned}\delta\boldsymbol{\xi}(t+1) &= \boldsymbol{\xi}(t+1) - \hat{\boldsymbol{\xi}}(t+1 \mid t)\\ &= f(\boldsymbol{\xi}(t)) + \boldsymbol{v}(t) - f(\hat{\boldsymbol{\xi}}(t \mid t)).\end{aligned} \tag{40}$$

By linearizing $f(\boldsymbol{\xi}(t))$ around $\hat{\boldsymbol{\xi}}(t \mid t)$, we rewrite (40) as

$$\begin{aligned}\delta\boldsymbol{\xi}(t+1) &= f(\hat{\boldsymbol{\xi}}(t \mid t)) + \boldsymbol{F}(t)(\boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t \mid t)) + \boldsymbol{v}(t)\\ &\quad - f(\hat{\boldsymbol{\xi}}(t \mid t)) = \boldsymbol{F}(t)\delta\boldsymbol{\xi}(t) + \boldsymbol{v}(t).\end{aligned} \tag{41}$$

From (41), we can derive an approximation of the prediction error variance (the so-called *pseudovariance*) $\boldsymbol{P}(t+1 \mid t)$, namely

$$\text{b)} \quad \boldsymbol{P}(t+1 \mid t) = \boldsymbol{F}(t)\boldsymbol{P}(t \mid t)\boldsymbol{F}^T + \boldsymbol{R}_v. \tag{42}$$

We will need (42) in the update step of the Kalman filter.

B. Update Step

We linearize (34) at time $t+1$ around the prediction $\hat{\boldsymbol{\xi}}(t+1 \mid t)$ and the new measurement $\boldsymbol{y}(t+1)$, namely

$$\begin{aligned}0 = h(\boldsymbol{\xi}(t+1), \boldsymbol{x}(t+1)) &\simeq h(\hat{\boldsymbol{\xi}}(t+1 \mid t), \boldsymbol{y}(t+1))\\ &\quad + \boldsymbol{C}(t+1)\delta\boldsymbol{\xi}(t+1) - \boldsymbol{D}(t+1)\boldsymbol{w}(t+1)\end{aligned} \tag{43}$$

with

$$\boldsymbol{C}(t+1) \triangleq \left.\frac{\partial h}{\partial \boldsymbol{\xi}}\right|_{\hat{\boldsymbol{\xi}}(t+1|t), \boldsymbol{y}(t+1)} \tag{44}$$

$$\boldsymbol{D}(t+1) \triangleq \left.\frac{\partial h}{\partial \boldsymbol{y}}\right|_{\hat{\boldsymbol{\xi}}(t+1|t), \boldsymbol{y}(t+1)}. \tag{45}$$

Therefore, we have

$$\begin{aligned}&h(\hat{\boldsymbol{\xi}}(t+1 \mid t), \boldsymbol{y}(t+1))\\ &\quad \simeq -\boldsymbol{C}(t+1)\delta\boldsymbol{\xi}(t+1) + \boldsymbol{n}(t+1)\end{aligned} \tag{46}$$

where $\boldsymbol{n}(t)$ is the process defined by

$$\boldsymbol{n}(t) = \boldsymbol{D}(t)\boldsymbol{w}(t). \tag{47}$$

Equation (46) together with (41) define a linear system. Thus, we can write the update equation of the traditional linear Kalman filter as

$$\begin{aligned}\text{c)} \quad \hat{\boldsymbol{\xi}}(t+1 \mid t+1) &= \hat{\boldsymbol{\xi}}(t+1 \mid t) + \boldsymbol{L}(t+1)\\ &\quad \cdot h\Big(\hat{\boldsymbol{\xi}}(t+1 \mid t), \boldsymbol{y}(t+1)\Big)\end{aligned} \tag{48}$$

where

$$\text{d)} \quad \boldsymbol{L}(t+1) = -\boldsymbol{P}(t+1 \mid t)\boldsymbol{C}(t+1)^T\boldsymbol{\Lambda}(t+1)^{-1} \tag{49}$$

$$\begin{aligned}\text{e)} \quad \boldsymbol{\Lambda}(t+1) &= \boldsymbol{C}(t+1)\boldsymbol{P}(t+1 \mid t)\boldsymbol{C}(t+1)^T\\ &\quad + \boldsymbol{R}_n(t)\end{aligned} \tag{50}$$

$$\begin{aligned}\text{f)} \quad \boldsymbol{P}(t+1 \mid t+1) &= \boldsymbol{\Gamma}(t+1)\boldsymbol{P}(t+1 \mid t)\boldsymbol{\Gamma}(t+1)^T\\ &\quad + \boldsymbol{L}(t+1)\boldsymbol{R}_n(t+1)\boldsymbol{L}(t+1)^T\end{aligned} \tag{51}$$

$$\text{g)} \quad \boldsymbol{\Gamma}(t+1) = \boldsymbol{I} + \boldsymbol{L}(t+1)\boldsymbol{C}(t+1) \tag{52}$$

with $\boldsymbol{R}_n(t+1)$ the covariance matrix of the noise $\boldsymbol{n}(t)$, namely

$$\text{h)} \quad \boldsymbol{R}_n(t+1) = \boldsymbol{D}(t+1)\boldsymbol{R}_w(t+1)\boldsymbol{D}(t+1)^T. \tag{53}$$

Equations a)–h), together with initial conditions

$$\hat{\boldsymbol{\xi}}(0 \mid 0) = \mathrm{E}[\boldsymbol{\xi}_0], \quad \boldsymbol{P}(0 \mid 0) = \mathrm{E}[\boldsymbol{\xi}_0\boldsymbol{\xi}_0^T]$$

constitute the Kalman filter. Here, E[·] denotes statistical expectation.

The quantity $h(\hat{\boldsymbol{\xi}}(t+1 \mid t), \boldsymbol{y}(t+1))$ takes the place of the innovation in the traditional linear Kalman filter, so it is called the *pseudoinnovation*.

It is important to note that the linear Kalman filter gives the minimum prediction error variance among all estimators, when the model and the measurement noises are Gaussian. It is optimal among all linear estimators when the noises are not Gaussian. On the other hand, when the model is not linear, as in the case considered above, the extended Kalman filter does not guarantee optimality.

Acknowledgment

The authors would like to thank S. Soatto for his useful suggestions and for kindly supplying the feature selection and tracking program. They also thank the anonymous reviewers for their comments.

References

[1] E. Dubois and J. Konrad, "Estimation of 2-D motion fields from image sequences with application to motion-compensated processing," in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Boston, MA: Kluwer, 1993.

[2] *Coded Representation of Picture and Audio Information*, ISO/IEC JCT1/SC29/WG11 MPEG, Test Model 5, Draft Rev. 2, pp. 21–23, Apr. 7, 1993.

[3] H. G. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing: Image Commun.*, vol. 1, pp. 117–138, Oct. 1989.

[4] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.

[5] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, pp. 185–203, 1981.

[6] D. H. Ballard and O. A. Kimball, "Rigid body motion from depth and optical flow," *Comput. Vis., Graph. Image Processing*, vol. 22, pp. 95–115, 1983.

[7] H. Shariat and K. E. Price, "Motion estimation with more than two frames," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 5, pp. 417–433, 1990.

[8] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 90–99, 1986.

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

[10] J. Bouquet and P. Perona, "Visual navigation using a single camera," in *Proc. 5th Int. Conf. Comput. Vision*, 1995.

[11] Z. Zhang and O. D. Faugeras, “Three-dimensional motion computation and object segmentation in a long sequence of stereo frames,” Rapports de Recherche, INRIA, Juillet, France, 1991.

[12] S. Soatto, R. Frezza, and P. Perona, “Motion estimation via dynamic vision,” *IEEE Trans. Automat. Contr.*, vol. 41, pp. 393–413, Mar. 1996.

[13] J. W. Lee, M. S. Kim, and I. S. Kweon, “A Kalman filter based visual tracking algorithm for an object moving in 3D,” in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, vol. 1, pp. 342–347, Aug. 1995.

[14] K. Aizawa and T. S. Huang, “Model-based image coding: Advanced video coding techniques for very low bit-rate applications,” *Proc. IEEE,* vol. 83, pp. 259–271, Feb. 1995.

[15] K. Aizawa, H. Harashima, and T. Saito, “Model-based analysis synthesis image coding (MBASIC) system for a person’s face,” *Signal Processing: Image Commun.*, vol. 1, no. 2, pp. 139–152, 1989.

[16] A. Zakhor and F. Lari, “Edge based 3-D camera motion estimation with application to video coding,” in *Motion Analysis and Image Sequence Processing*, M. I. Sezan and R. L. Lagendijk, Eds. Boston, MA: Kluwer, 1993.

[17] E. Stein and B. Girod, “Estimation of rigid body motion and scene structure from image sequence using a novel epipolar transform,” in *Proc. ICASSP’96*, Atlanta, GA, 1996, pp. 1911–1914.

[18] T. S. Huang and A. N. Netravali, “Motion and structure from feature correspondences: A review,” *Proc. IEEE*, vol. 82, pp. 252–268, Feb. 1994.

[19] R. Y. Tsai and T. S. Huang, “Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 13–26, Jan. 1984.

[20] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

[21] P. S. Maybeck, *Stochastic Models, Estimation and Control, Volume 1 and 2*. New York: Academic, 1979–1982.

[22] A. Verri and T. Poggio, “Motion field and optical flow: Qualitative properties,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 490–498, May 1989.

[23] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proc. 7th Int. Joint Conf. Artif. Intell.*, 1981.

[24] Ad hoc group on MPEG-4 video VM editing, *MPEG-4 Video Verification Model Version 1.22.0*, ISO/IEC JTC1/SC29/WG11, no. 1260, Firenze, Italy, Mar. 1996.

[25] P. Gerken, “Object-based analysis–synthesis coding of image sequences at very low bit rates,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 228–235, June 1994.

[26] A. Eleftheriadis and A. Jacquin, “Automatic face location detection for model-assisted rate control in H.261-compatible coding of video,” *Signal Processing: Image Commun.*, vol. 7, pp. 435–445, 1995.

[27] R. Rinaldo and G. Calvagno, “Image coding by block prediction of multiresolution subimages,” *IEEE Trans. Image Processing*, vol. 4, pp. 909–920, July 1995.

[28] R. J. Crinon and W. J. Kolodziej “Adaptive model-based motion estimation,” *IEEE Trans. Image Processing*, pp. 469–481, Sept. 1994.

Giancarlo Calvagno (M’92) was born in Venezia, Italy, in 1962. He received the “Laurea in Ingegneria Elettronica” degree from the University of Padova, Padova, Italy, in 1986, and the doctorate degree in electrical engineering from the University of Padova in 1990.

From 1988 to 1990, he was at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, as a Visiting Scholar. Since 1990, he has been with the Dipartimento di Elettronica e Informatica, University of Padova, as a Ricercatore. His main research interests are in the areas of digital signal processing, signal reconstruction, image processing, and coding.

Roberto Rinaldo (S’92–M’92) received the “Laurea in Ingegneria Elettronica” degree in 1987 from the University of Padova, Padova, Italy, with honors and the medal for the highest graduation score. He received the M.S. degree from the University of California at Berkeley, and the doctorate degree from the University of Padova in 1992.

Since 1993, he has been with the Dipartimento di Elettronica e Informatica, University of Padova, where he is currently a Ricercatore in communications and signal processing. His interests are in the field of multidimensional signal processing, video signal coding, fractal theory, and image coding.

Luciano Sbaiz was born in Udine, Italy, in 1968. He received the “Laurea in Ingegneria Elettronica” degree from the University of Padova, Padova, Italy, in 1993. Since 1994, he has worked toward the doctorate degree in electronics and telecommunications at the University of Padova.

The topic of his research is the application of computer vision motion estimation algorithms to object-oriented and model-based video coders. Besides video and image coding, he is interested in all topics regarding image and multidimensional signal processing.