Three-Dimensional Modeling from Two-Dimensional Video

Pedro M. Q. Aguiar and José M. F. Moura, *Fellow, IEEE*

***Abstract*—This paper presents the *surface-based factorization* method to recover three-dimensional (3-D) structure, i.e., the 3-D shape and 3-D motion, of a rigid object from a two-dimensional (2-D) video sequence. The main ingredients of our approach are as follows:**

1) **we describe the unknown shape of the 3-D rigid object by polynomial patches;**
2) **projections of these patches in the image plane move according to parametric 2-D motion models;**
3) **we recover the parameters describing the 3-D shape and 3-D motion from the 2-D motion parameters by factorizing a matrix that is rank 1 in a noiseless situation.**

Our method is simultaneously an extension and a simplification of the original factorization method of Tomasi and Kanade [1]. We track regions where the 2-D motion in the image plane is described by a single set of parameters, avoiding the need to track a large number of pointwise features, in general, a difficult task. Then our method estimates the parameters describing the 3-D structure by factoring a rank 1 matrix, not rank 3 as in [1]. This allows the use of fast iterative algorithms to compute the 3-D structure that best fits the data. Experimental results with real-life video sequences illustrate the good performance of our approach.

***Index Terms*—Factorization, structure from motion, 3-D image and video processing, 3-D shape.**

I. Introduction

THE automatic generation of a three-dimensional (3-D) description of the real-world environment has received the attention of a large number of researchers. Target applications are found in several fields, including digital video, virtual reality, and robotics. The information source for a number of successful approaches to 3-D modeling has been a range image. This image, obtained from a usually expensive range sensor, provides the distance between the sensor and the environment in front of it, thus the range image itself contains explicit information about the 3-D structure of the environment. In this paper, we build 3-D models for rigid bodies from two-dimensional (2-D) video data, when no explicit 3-D information is given.

Manuscript received October 26, 1998; revised June 5, 2001. The work of P. M. Q. Aguiar was supported in part by INVOTAN. This work was supported in part by ONR under Grant N000 14-00-1-0593. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Michael R. Frater.

P. M. Q. Aguiar is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA and also with the Institute for Systems and Robotics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal (e-mail: aguiar@ece.cmu.edu; aguiar@isr.ist.utl.pt).

J. M. F. Moura is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890 USA (e-mail: moura@ece.cmu.edu).

Publisher Item Identifier S 1057-7149(01)08200-8.

A. Previous Related Work

The problem of recovering the 3-D structure (3-D shape and 3-D motion) from a 2-D video sequence has been widely considered by the computer vision community. Methods that infer 3-D shape from a single frame are based on cues such as shading and defocus. These methods fail to give reliable 3-D shape estimates for unconstrained real-world scenes. If no prior knowledge about the scene is available, the cue to estimating the 3-D structure is the 2-D motion of the brightness pattern in the image plane. For this reason, the problem is generally referred to as *structure from motion (SFM)*. The two major steps in SFM are usually as follows.

Step 1) Compute the 2-D motion in the image plane, either in the form of a dense field or in the form of a sparse set of correspondences (see [2] for a recent discussion on this topic).

Step 2) Estimate the 3-D shape and the 3-D motion from the computed 2-D motion.

Early approaches to SFM processed a single pair of consecutive frames and provided existence and uniqueness results to the problem of estimating 3-D motion and absolute depth from the 2-D motion in the camera plane between two frames [3]. Two-frame-based algorithms are highly sensitive to image noise and, when the object is far from the camera, i.e., at a large distance when compared to the object depth, they fail even at low-level image noise.

More recent research has been oriented toward the use of longer image sequences. An attractive tool to recursively extend two-frame algorithms to multiframe algorithms is the Kalman filter. A number of approaches used extended Kalman filter (EKF) to estimate 3-D structure from the 2-D motion across long video sequences [4]–[7]. The lack of guarantee of convergence for the EKF-based algorithms and the fact that those approaches did not truly enforce the 3-D rigidity of the scene over the sequence of images motivated a number of researchers to use nonlinear optimization methods to address the multiframe SFM problem in a batch way [8]–[11]. In general, these methods lead to complex and time-consuming algorithms.

In the early 1990s, Tomasi and Kanade [1] introduced the *factorization method*, an elegant method to solve multiframe SFM that avoids nonlinear optimization. They represent the 3-D shape by the 3-D position of a set of feature points. The 2-D projection of each feature point is tracked along the image sequence. The 3-D shape and motion are then estimated by factoring a measurement matrix whose entries are the set of trajectories of the feature point projections. Tomasi and Kanade

pioneered the use of linear subspace constraints in motion analysis. In fact, the key idea underlying the factorization method is the fact that the rigidity of the scene imposes that the measurement matrix lives in a low-dimensional subspace of the universe of matrices. Tomasi and Kanade have shown that the measurement matrix is a rank 3 matrix in a noiseless situation. Reference [1] uses the orthographic projection model. The factorization method was later extended to the scaled-orthography and para-perspective models [12] and to the multibody scenario [13].

A different approach to recovering 3-D structure from 2-D images, also denominated SFM, uses motion as the only cue, but rather than computing the 2-D motion in the image plane as as intermediate step, it attempts to compute the 3-D strucure directly from the image intensity values. Due to the complexity of the problem, these approaches have been so far restricted to the processing of only two or three consecutive frames [14]–[16] or to the use of a Kalman filter [17] or an iterative Levenberg–Marquardt minimization [18] to exploit rigidity across time.

B. Proposed Approach

The factorization method as developed by Tomasi and Kanade [1] relies on the matching of a set of point features along the image sequence. This task is difficult when processing noisy videos. In general, only distinguished points, as brightness corners, are used as "trackable" feature points. As a consequence, the approach of [1] does not provide dense depth estimates. Under our more general scenario, rather than describing the 3-D shape by the set of 3-D positions of the feature points, we parameterize the shape of the object surface and show that this parameterization induces a parametric model for the 2-D motion of the brightness pattern in the image plane. Instead of tracking pointwise features, we track larger regions where the image motion is described by a single set of parameters. For example, for scenes with polyhedral surfaces, each region corresponds to a flat surface patch and the 2-D image motion models reduce to the well-known affine motion model. The model parameters are computed by minimizing directly the differences in the intensity levels, leading to robust estimates [19]–[21]. Besides being particularly relevant in outdoor modeling of buildings with flat walls, our approach handles general shaped structures by approximating them by a piecewise planar surface or higher order polynomial surface. It is known that computer graphics methods using planar patches rather than points, provide usually much better quality 3-D shape reconstruction because they use, besides the 3-D relative depth at each point, the orientation of the surface at that point—an important clue to recover the shape. To recover in an expedite way the 3-D motion and 3-D shape parameters from the image motion parameters, we introduce the *surface-based factorization*, a generalization of the original factorization method that recovers the parameters describing the 3-D structure by factorizing a matrix that collects the 2-D motion parameters. We show that this matrix is rank 1 in a noiseless situation. The estimates of the 3-D motion parameters and the 3-D shape parameters are then obtained from the column vector and row vector whose outer product best matches the data in the matrix of 2-D motion parameters.

Another relevant feature of our method is its computational simplicity. There are two ways our method gains with respect to the original method of Tomasi and Kanade [1]. First, the surface-based representation leads to a much sparser parametric description for the 3-D surface than the feature points description: the number of patches required is in general significantly smaller than the number of feature points needed for similar levels of approximations. This reduces the computational effort because the number of patches to be tracked is much smaller and because the matrix to be factored is also much smaller. Second, by making an appropriate linear subspace projection, we find the unknown 3-D structure by factoring a matrix that is rank 1 in a noiseless situation, rather than a rank 3 matrix as in the original factorization method [1]. This allows the use of faster iterative algorithms to compute the matrix that best approximates the data.

C. Paper Overview

In Section II, we formulate the problem and motivate the *surface-based factorization* approach with a simple example. In Section III, we make explicit the relation between the 2-D image motion parameters and the 3-D structure parameters for objects whose 3-D shape is described by a piecewise polynomial surface. Section IV details our *surface-based factorization* method for recovering the 3-D shape and 3-D motion from the image motion parameters. Section V illustrates the approach with two real-life video clips and compares experimentally the computational cost of the feature-based specialization of our method with the one of the original factorization method [1]. Section VI concludes the paper. References [22] and [23] report parts of this work.

II. Problem Formulation

A. Model

We consider a rigid object $\mathcal{O}$ moving in front of a camera. The object $\mathcal{O}$ is described by its 3-D shape $\mathcal{S}$ and texture $\mathcal{T}$. The texture $\mathcal{T}$ represents the light received by the camera after reflecting on the object surface, i.e., the texture $\mathcal{T}$ is the object brightness as perceived by the camera. The texture depends on the object surface photometric properties, as well as on the environment illumination conditions. We assume that the texture does not change with time. The 3-D shape $\mathcal{S}$ is a representation of the surface of the object.

The position and orientation of the object $\mathcal{O}$ relative to the camera at time instant f is represented by a vector $\mathbf{m_f}$. This vector codes a rotation-translation pair that takes values in the group of the rigid transformations of the space, the special Euclidean group SE(3). The 3-D structure obtained by applying the 3-D rigid transformation coded by the vector $\mathbf{m_f}$ to the object $\mathcal{O}$ is represented by $\mathcal{M}(\mathbf{m_f})\mathcal{O}$. The frame $\mathbf{I_f}$ captured at time f, $1 \leq f \leq F$, is modeled as a noisy observation of the projection of the object

$$\mathbf{I_f} = \mathcal{P}\left\{\mathcal{M}(\mathbf{m_f})\mathcal{O}\right\} + \mathbf{W_f}. \tag{1}$$

For simplicity, the observation noise $\mathbf{W_f}$ is zero mean, white, and Gaussian. We assume that $\mathcal{P}$ is the orthogonal projection operator that is known to be a good approximation to the perspective projection when the relative depth of the scene is small when compared to the distance to the camera. The *surface-based factorization* algorithm proposed in this paper is derived from the orthogonal projection model. Note, however, that it is easily extended to the scaled-orthography and the paraperspective projections by proceeding as [12] proposes for the original factorization method of Tomasi and Kanade [1].

The operator $\mathcal{P}$ returns the texture $\mathcal{T}$ as a real valued function defined over the image plane. This function is a nonlinear mapping that depends on the object shape $\mathcal{S}$ and the object position $\mathbf{m_f}$. The intensity level of the projection of the object at pixel $\mathbf{u}$ on the image plane is

$$\mathcal{P}\Big\{\mathcal{M}(\mathbf{m_f})\mathcal{O}\Big\}(\mathbf{u}) = \mathcal{T}\left(\mathbf{s_f}(\mathcal{S}, \mathbf{m_f}; \mathbf{u})\right) \tag{2}$$

where $\mathbf{s_f}(\mathcal{S}, \mathbf{m_f}; \mathbf{u})$ is the nonlinear mapping that lifts the point $\mathbf{u}$ on the image $\mathbf{I_f}$ to the corresponding point on the 3-D object surface. This mapping $\mathbf{s_f}(\mathcal{S}, \mathbf{m_f}; \mathbf{u})$ is determined by the object shape $\mathcal{S}$ and the position $\mathbf{m_f}$. To simplify the notation, we will usually write explicitly only the dependence on f, i.e., $\mathbf{s_f}(\mathbf{u})$.

Fig. 1 illustrates the lifting mapping $\mathbf{s_f}(\mathbf{u})$ and the direct mapping $\mathbf{u_f}(\mathbf{s})$ for the orthogonal projection of a 2-D object. The inverse mapping $\mathbf{u_f}(\mathbf{s})$ also depends on the object shape $\mathcal{S}$ and position $\mathbf{m_f}$ at frame f, but we will, again, usually show only explicitly the dependence on f. On the left of Fig. 1, the point $\mathbf{s}$ on the surface of the object projects onto $\mathbf{u_f}(\mathbf{s})$ on the image plane. On the right, pixel $\mathbf{u}$ on the image plane is lifted to $\mathbf{s_f}(\mathbf{u})$ on the object surface. We assume that the object does not occlude itself, i.e., we have $\mathbf{u_f}(\mathbf{s}_f(\mathbf{u})) = \mathbf{u}$ and $\mathbf{s_f}(\mathbf{u_f}(\mathbf{s})) = \mathbf{s}$. The mapping $\mathbf{u_f}(\mathbf{s})$, seen as a function of the frame index f, for a particular surface point $\mathbf{s}$, is the trajectory of the projection of that point in the image plane, i.e., it is the motion induced in the image plane.

We consider the estimation of the 3-D shape $\mathcal{S}$ and the 3-D motion $\{\mathbf{m_f}, \mathbf{1} \leq \mathbf{f} \leq \mathbf{F}\}$ given the video sequence $\{\mathbf{I_f}, \mathbf{1} \leq \mathbf{f} \leq \mathbf{F}\}$ of F frames. In [24] and [25], we discuss the *maximum-likelihood (ML)* estimate for this problem. There we show that, after eliminating the dependence on the texture $\mathcal{T}$, we are left with a cost function that depends on the *structure* (3-D shape $\mathcal{S}$ and 3-D motion $\{\mathbf{m_f}\}$) only through the *motion* induced in the image plane, i.e., through the 2-D motion mappings $\{\mathbf{u_f}(\mathbf{s})\}$. Recall that $\mathbf{u_f}(\mathcal{S}, \mathbf{m_f}; \mathbf{s})$ depends on the shape $\mathcal{S}$ and the motion $\mathbf{m_f}$. This makes clear why the problem we are addressing is referred to as SFM.

B. Approach

The *surface-based factorization* method uses a parametric description of the surface $\mathcal{S}$ of the rigid object in terms of a parameter vector $\mathbf{a}$, $\mathcal{S}(\mathbf{a})$. We exploit the constraints induced on the 2-D motion in the image plane by the projection operator, the rigidity of the object and the parameterization of the surface shape of the object. The constraints induced on the image motion enable us to parameterize the image motion mapping $\mathbf{u_f}(\mathbf{s})$ in terms of a parameter vector $\mathbf{d_f}$ as $\mathbf{u}(\mathbf{d_f}; \mathbf{s})$. The parameter vector $\mathbf{d_f}$ is directly related to the 3-D shape parameter vector $\mathbf{a}$ and the 3-D position $\mathbf{m_f}$, as will be shown below. Our approach follows these two stages. First, we estimate the parameters $\{\mathbf{d_f}\}$ by using a known numerical technique for image motion estimation. Then, we solve the inverse problem of going from the sequence of image motion parameters to the 3-D structure, i.e., we determine the 3-D shape parameter vector $\mathbf{a}$ and the sequence of 3-D positions $\{\mathbf{m_f}\}$, given the estimates $\{\hat{\mathbf{d}}_f\}$ of the image motion parameter vectors $\{\mathbf{d_f}\}$.

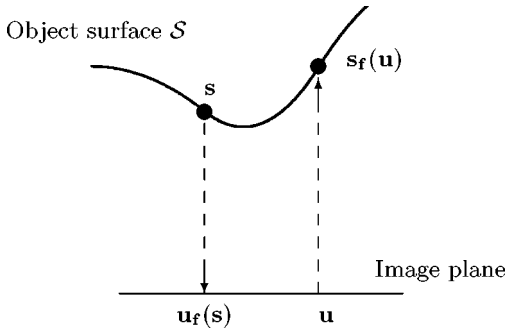

Fig. 1. Mappings $\mathbf{u_f}(\mathbf{s})$ and $\mathbf{s_f}(\mathbf{u})$.

Before addressing the general case, we illustrate our approach with a simple example: a parabolic patch moving in a 2-D world where the images are one-dimensional orthogonal projections. This scenario, although simpler than the 3-D world problem, reflects the very basic properties and difficulties of the SFM paradigm. Note that the 2-D scenario, illustrated in Fig. 2, corresponds to the real 3-D world, if we consider only one epipolar plane and assume that the motion occurs on that plane. The images are single scan-lines.

C. Example

Fig. 2 shows a parabolic patch $\mathcal{S}$ that moves with respect to a fixed camera. We attach a coordinate system to the object $\mathcal{S}$ given by the axes labeled by x and z. The 2-D object shape $\mathcal{S}(\mathbf{a})$ is described in terms of the parameter vector $\mathbf{a} = [a_0, a_1, a_2]^{\mathbf{T}}$, in the object coordinate system (o.c.s.), by the parabola

$$z(x) = z(\mathbf{a}; x) = a_0 + a_1 x + a_2 x^2. \tag{3}$$

To capture the motion of the object, we attach a different coordinate system to the camera given by the axes u and w (see Fig. 2). The u axis is the camera "plane." We define the 2-D motion of the object by specifying the position of the o.c.s. relative to the camera coordinate system (c.c.s.). The unconstrained motion of a rigid body can be described in terms of a time varying point translation and a rotation. Hence, the object position at time instant f is expressed in terms of $(t_{uf}, t_{wf}, \theta_f)$ where, as shown in Fig. 2, (t_{uf}, t_{wf}) are chosen to be the coordinates of the origin of the o.c.s. with respect to the c.c.s. (translational component of the 2-D motion) and θ_f is the orientation of the o.c.s. relative to the c.c.s. (rotational component of the motion).

At instant f, the point on the object with 2-D coordinates (x, z) in the o.c.s. has the following coordinates in the c.c.s.:

$$\begin{bmatrix} u_f \\ w_f \end{bmatrix} = \mathbf{\Theta_f} \begin{bmatrix} x \\ z \end{bmatrix} + \mathbf{t_f} = \begin{bmatrix} i_{xf} & i_{zf} \\ k_{xf} & k_{zf} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} t_{uf} \\ t_{wf} \end{bmatrix} \tag{4}$$

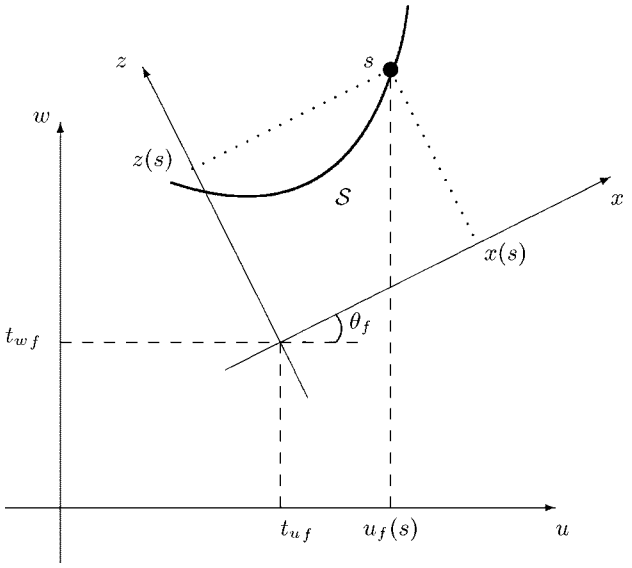

Fig. 2. Two-dimensional world: object and camera coordinate systems.

where $\mathbf{\Theta_f}$ is the rotation matrix for angle θ_f and $\mathbf{t_f}$ is the translation vector.

From (4), we see that the point (x, z) projects at time f on the image coordinate u_f given by

$$u_f = i_{xf}x + i_{zf}z + t_{uf}. \tag{5}$$

Expression (5) shows that the orthogonal projection is insensitive to the translation component t_{wf} of the object motion. This reflects the well-known fact that, under orthography, the absolute depth (distance from the camera to the object) cannot be estimated. Only the set of positions $\{\mathbf{m_f} = \{\mathbf{t_{uf}}, \theta_\mathbf{f}\}, \mathbf{1} \leq \mathbf{f} \leq \mathbf{F}\}$ can be estimated from the image sequence.

We now show how the mapping $\mathbf{u_f(s)}$, introduced above and illustrated in Fig. 1, is described parametrically. In the 2-D world, the mapping $\mathbf{u_f(s)}$ is written as $u_f(s)$ because it maps a scalar s to a scalar u. Choose the coordinate s, labeling the argument of the texture function $\mathcal{T}$ and representing in a unique way the generic point on the object surface (object contour in this case), to be the object coordinate x. We refer to s as the texture coordinate. A point with texture coordinate s on the object surface projects at time f, according to (5), to the image coordinate $u_f(s)$ given by

$$\begin{aligned} u_f(s) &= i_{xf}x(s) + i_{zf}z(s) + t_{uf} \\ &= i_{xf}s + i_{zf}\left(a_0 + a_1 s + a_2 s^2\right) + t_{uf} \end{aligned} \tag{6}$$

where $x(s)$ and $z(s)$ are the coordinates of the point s in the o.c.s. The equality $x(s) = s$ comes from the choice of the texture coordinate s and the expression for $z(s)$ comes from the parabolic shape [see (3)].

By defining the coefficients of the powers of s in (6) as

$$d_{f0} = i_{zf}a_0 + t_{uf}, \quad d_{f1} = i_{xf} + i_{zf}a_1, \quad d_{f2} = i_{zf}a_2 \tag{7}$$

we have the following parametric description for the image motion $u_f(s)$ in terms of the parameter vector $\mathbf{d_f} = [d_{f0}, d_{f1}, d_{f2}]^\mathbf{T}$

$$u_f(s) = u(\mathbf{d_f}; s) = d_{f0} + d_{f1}s + d_{f2}s^2. \tag{8}$$

The parameter vector $d_f = [d_{f0}, d_{f1}, d_{f2}]^\mathbf{T}$ describes the motion of the brightness pattern in the image plane, i.e., it describes the mapping $\mathbf{u_f(s)}$ introduced above (see Fig. 1).

With the parabolic patch, the steps of our approach to recover the 2-D structure, i.e., the shape parameters $\{a_0, a_1, a_2\}$ and the set of positions $\{t_{uf}, \theta_f, 1 \leq f \leq F\}$ are then summarized as follows.

Step 1) Given the image sequence of F frames, estimate the set of image motion parameters $\{d_{f0}, d_{f1}, d_{f2}, 1 \leq f \leq F\}$. This leads to $3F$ estimates $\{\widehat{d}_{f0}, \widehat{d}_{f1}, \widehat{d}_{f2}, 1 \leq f \leq F\}$.

Step 2) Invert (7), solving for the shape parameters $\{a_0, a_1, a_2\}$ and the $2F$ object positions $\{t_{uf}, \theta_f, 1 \leq f \leq F\}$, given the set of $3F$ estimates $\{\widehat{d}_{f0}, \widehat{d}_{f1}, \widehat{d}_{f2}\}$.

Step 1 is solved by using a known numerical technique to fit parametric models to the motion of the brightness patterns in the image plane. Step 2 leads in general to a nonlinear problem. Section IV details our approach to this problem. First, we obtain a closed-form solution for the estimate of the 3-D translation. Then, due to the structure of the orthogonal projection operator and the shape parameterization, we can express the dependence of $\mathbf{d_f} = \mathbf{d}(\mathbf{a}, \mathbf{m_f})$ for $1 \leq f \leq F$ on the vectors $\mathbf{a}$ and $\mathbf{m_f}$ in a bilinear matrix format as $\mathbf{R} = \mathbf{MS}^\mathbf{T}$, where the matrix $\mathbf{R}$ collects the image motion parameters $\{\mathbf{d_f}, \mathbf{1} \leq \mathbf{f} \leq \mathbf{F}\}$, $\mathbf{M}$ depends on the positions $\{\mathbf{m_f}, \mathbf{1} \leq \mathbf{f} \leq \mathbf{F}\}$ and $\mathbf{S}$ contains the shape parameter $\mathbf{a}$. The problem of estimating $\mathbf{a}$ and $\{\mathbf{m_f}, \mathbf{1} \leq \mathbf{f} \leq \mathbf{F}\}$ becomes how to find suitable factors $\mathbf{M}$ and $\mathbf{S}^\mathbf{T}$ for the factorization of the matrix $\mathbf{R}$. We will see how to solve this problem by computing only the largest singular value and the associated singular vector of a matrix $\widetilde{\mathbf{R}}$ that is easily obtained from $\mathbf{R}$.

Our general methodology can be used for any parametric shape description. The situations we are interested in are characterized by no prior knowledge about the object shape. For this kind of situations, a general shape model must be characterized by a local parameterization. The local shape parameterization induces a local parameterization for the motion in the image plane. In the following sections we detail our approach for a generic shape model locally parameterized: the piecewise polynomial functions.

III. Piecewise Polynomial Shape

The o.c.s. has axes labeled by x, y and z. The c.c.s. has axes labeled by u, v and w. We consider that the o.c.s. coincides with the c.c.s. on the first frame. The image plane is defined by the axes u and v.

A. Three-Dimensional Shape

The 3-D shape of the object is a parametric description of its surface. We consider objects whose shape is given

by a piecewise polynomial surface with N patches. The 3-D shape is described in terms of N sets of parameters $\{a^n_{00}, a^n_{10}, a^n_{01}, a^n_{11}, a^n_{20}, a^n_{02}, \ldots\}$, for $1 \leq n \leq N$, where

$$\begin{aligned} z = a^n_{00} &+ a^n_{10}(x - x^n_0) + a^n_{01}(y - y^n_0) \\ &+ a^n_{11}(x - x^n_0)(y - y^n_0) \\ &+ a^n_{20}(x - x^n_0)^2 + a^n_{02}(y - y^n_0)^2 + \cdots \end{aligned} \tag{9}$$

describes the shape of the patch n in the o.c.s. With respect to the representation of the polynomial patches, the parameters x^n_0 and y^n_0 can have any value, e.g., they can be made zero. We allow the specification of general parameters x^n_0, y^n_0 because the shape of a small patch n with support region (x, y) located far from the point (x^n_0, y^n_0) has a high sensivity with respect to the shape parameters. To minimize this sensivity, we choose for (x^n_0, y^n_0) the centroid of the support region of patch n. With this choice, we improve the accuracy of the 3-D structure recovery algorithm. In [25] we show that, by making (x^n_0, y^n_0) to be the centroid of the support region of patch n, we also improve the numerical stability of the algorithm that estimates the 2-D image motion parameters.

Expression (9) describes the 3-D shape with full generality in a local way—it is the Taylor series expansion of the relative depth $z(x, y)$ around the point $(x, y) = (x^n_0, y^n_0)$, for appropriate values of the set of shape parameters $\{a^n_{00}, a^n_{10}, a^n_{01}, a^n_{11}, a^n_{20}, a^n_{02}, \ldots\}$. We can recover the simpler feature-based shape description from the general 3-D shape described by (9) by making zero all the shape parameters, except for a^n_{00} that codes the relative depth of feature n, $z = a^n_{00}$. Expression (9) models also a special case of practical interest: the piecewise planar shapes. In this case, the planar patch n is described by the parameters $\{a^n_{00}, a^n_{10}, a^n_{01}\}$. This set of parameters codes the *orientation* of the planar patch, besides its position.

Since the following derivations deal with a single surface patch, we will omit the super-index n in the shape parameters. To further simplify the notation, we define the vectors $\mathbf{a_1} = [a_{10}, a_{01}]^{\mathbf{T}}$, $\mathbf{s} = [x, y]^{\mathbf{T}}$, $\mathbf{s_0} = [x_0, y_0]^{\mathbf{T}}$, $\mathbf{a_2} = [a_{11}, a_{20}, a_{02}, \ldots]^{\mathbf{T}}$, and $\mathbf{p(s - s_0)} = [(x - x_0)(y - y_0), (x - x_0)^2, (y - y_0)^2, \ldots]^T$ and rewrite the shape of patch k as

$$z = a_{00} + \mathbf{a_1^T(s - s_0)} + \mathbf{a_2^T p(s - s_0)}. \tag{10}$$

The vectors $\mathbf{a}_2$ and $\mathbf{p(s - s_0)}$ are $P \times 1$ where P depends on the degree of the polynomial patch.

B. Three-Dimensional Motion

We define the 3-D motion of the object by specifying the position of the o.c.s. relative to the c.c.s. in terms of $\left(t_{uf}, t_{vf}, t_{wf}, \mathbf{\Theta_f}\right)$ where $\left(t_{uf}, t_{vf}, t_{wf}\right)$ are the coordinates of the origin of the o.c.s. with respect to the c.c.s. (3-D translation) and the matrix $\mathbf{\Theta_f}$ is the rotation matrix that determines the orientation of the o.c.s. relative to the c.c.s. (3-D rotation). A point with coordinates $[x, y, z]^T$ in the o.c.s. has the following coordinates in the c.c.s., at frame f

$$\begin{bmatrix} u_f \\ v_f \\ w_f \end{bmatrix} = \mathbf{\Theta_f} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_{uf} \\ t_{vf} \\ t_{wf} \end{bmatrix}. \tag{11}$$

Under orthography, the point with coordinates $[x, y, z]^T$ in the o.c.s. projects in frame f onto the image point $[u_f, v_f]^T$ given by

$$\begin{bmatrix} u_f \\ v_f \end{bmatrix} = \mathbf{M_f} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{t_f} \tag{12}$$

where the matrix $\mathbf{M_f}$ collects the first and second rows of the 3-D rotation matrix $\mathbf{\Theta_f}$ and the vector $\mathbf{t_f}$ contains the two components of the 3-D translation that can be recovered from the image sequence, $\mathbf{t_f} = [t_{uf}, t_{vf}]^{\mathbf{T}}$.

We now show how the 2-D motion induced in the image plane by the body-camera 3-D motion is described in terms of a set of parameters, and we relate the parameters of the 2-D motion model to the 3-D shape and 3-D motion parameters.

C. Image Motion

Consider a generic point in the object surface with coordinates $\mathbf{s} = [x, y]^{\mathbf{T}}$ and z given by (10). We denote by $\mathbf{u_f(s)} = [u_f(s), v_f(s)]^{\mathbf{T}}$ the trajectory of the projection of the point $\mathbf{s}$ in the image plane. Since we have chosen the coordinate systems to coincide on the first frame, we have $\mathbf{u}_1(\mathbf{s}) = \mathbf{s}$. At frame f, the point $\mathbf{s}$ projects according to (12), to the image point

$$\mathbf{u_f(s)} = \mathbf{N}_f\mathbf{s} + \mathbf{n}_f z + \mathbf{t_f} \tag{13}$$

where we have decomposed the matrix $\mathbf{M_f}$ as $\mathbf{M_f} = [\mathbf{N_f}, \mathbf{n_f}]$, where the 2×2 matrix $\mathbf{N_f}$ collects the first and second columns of the matrix $\mathbf{M_f}$, and the vector $\mathbf{n_f}$ is the third column of $\mathbf{M_f}$.

By inserting (10) into (13), we express the image displacement between frame 1 and frame f in terms of the 3-D shape and 3-D motion parameters. After simple manipulations, we obtain

$$\begin{aligned} \mathbf{u_f(s)} = \mathbf{N_f s_0} + \mathbf{n_f} a_{00} + \mathbf{t_f} + \left(\mathbf{N_f} + \mathbf{n_f a_1^T}\right)\mathbf{(s - s_0)} \\ + \mathbf{n_f a_2^T p(s - s_0)}. \end{aligned} \tag{14}$$

Denoting the 2×1 vector corresponding to the term independent of $\mathbf{s}$, the 2×2 matrix that multiplies $\mathbf{(s - s_0)}$ and the $2 \times P$ matrix that multiplies $\mathbf{p(s - s_0)}$ by, respectively

$$\begin{aligned} \mathbf{d_f} &= \mathbf{N_f s_0} + \mathbf{n_f} a_{00} + \mathbf{t_f} \\ \mathbf{D_f} &= \mathbf{N_f} + \mathbf{n_f a_1^T} \\ \mathbf{E_f} &= \mathbf{n_f a_2^T} \end{aligned} \tag{15}$$

we rewrite (14) as

$$\mathbf{u_f(s)} = \mathbf{d_f} + \mathbf{D_f(s - s_0)} + \mathbf{E_f p(s - s_0)}. \tag{16}$$

Expression (16) shows that the image coordinates at frame f, $\mathbf{u_f}$, of the points belonging to the object surface are parametric mappings of their image coordinates in frame 1, $\mathbf{u}_1 = \mathbf{s}$. The 2-D motion of the brightness pattern in the image plane is then described parametrically by (16). Expression (15) relates the parameters of the 2-D motion model for each surface patch, $\mathbf{d_f}$,

$\mathbf{D_f}$, and $\mathbf{E_f}$, to the 3-D motion parameters, $\mathbf{N_f}$, $\mathbf{n_f}$, and $\mathbf{t_f}$, and the 3-D shape parameters corresponding to that patch, a_{00}, $\mathbf{a}_1$, and $\mathbf{a}_2$. For the special case of piecewise planar surfaces, the 3-D shape of each patch is described by a_{00} and $\mathbf{a}_1$ and the 2-D motion in the image plane is given by $\mathbf{u_f(s)} = \mathbf{d_f} + \mathbf{D_f(s - s_0)}$, i.e., it is the affine motion model.

Except for particular 3-D motions, the 2-D motion in the image plane corresponding to different surface patches is described by different model parameterizations. The problem of estimating the support regions of the surface patches leads to the segmentation of the image motion field. Segmentation from motion has been widely addressed in the past [19], [21], [26]. In our experiments, we used two methods that lead to similar results. The first method simply slides a rectangular window across the image and detects abrupt changes in the motion parameters. The second method uses a quad-tree decomposition. We start by estimating the motion parameters considering the entire image as the support region. The region is recursively decomposed into smaller regions and the motion of each subregion is estimated. Then we associate regions with similar motion. To estimate the motion parameters within each region, we use the now widely known approach introduced in [27]. This approach uses a hierarchical Gauss-Newton method where the derivatives involved are computed from the image gradients. Another possible way to use our *surface-based factorization* method is to select *a priori* the support regions of the surface patches. This is the approach followed by the feature-based methods, where the features are selected *a priori*, based on the spatial variability of the brightness pattern.

IV. Suface-Based Factorization

The problem of inferring 3-D rigid structure from the image motion is formulated as estimating the 3-D motion parameters $\{\mathbf{N_f}, \mathbf{n_f}, \mathbf{t_f}, \mathbf{2 \le f \le F}\}$ and the 3-D shape parameters $\{a_{00}^n, \mathbf{a_1^n}, \mathbf{a_2^n}, \mathbf{1 \le n \le N}\}$ from the image motion parameters $\{\mathbf{d_f^n}, \mathbf{D_f^n}, \mathbf{E_f^n}, \mathbf{2 \le f \le F}, \mathbf{1 \le n \le N}\}$ by inverting the overconstrained set of equations of (15) for all the frames and all the surface patches. The super-index n above denotes the surface patch.

A. 3-D Structure from 2-D Motion

We start by estimating the translation vector $\mathbf{t_f}$. By choosing the o.c.s. in such a way that $\sum_n a_{00}^n = 0$ and the image origin in such a way that $\sum_n \mathbf{s_0^n} = [0,0]^T$, we obtain the least squares (LS) estimate for the translation vector $\mathbf{t_f}$ as the mean of the vectors $\{\mathbf{d_f^n}, \mathbf{1 \le n \le N}\}$

$$\widehat{\mathbf{t_f}} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{d_f}^n. \tag{17}$$

To eliminate the dependence of the image motion parameters on the translation, we replace the translation estimates into (15) and define a new set of parameters $\{\widetilde{\mathbf{d_f}}^n\}$ related to $\{\mathbf{d_f}^n\}$ by

$$\widetilde{\mathbf{d_f}}^n = \mathbf{d_f}^n - \frac{1}{\mathbf{N}} \sum_{m=1}^{N} \mathbf{d_f}^m. \tag{18}$$

Defining the $2 \times (P+3)$ matrix $\mathbf{R_f}$ and the $3 \times (P+3)$ matrix $\mathbf{S^T}$ as

$$\mathbf{R_f} = \begin{bmatrix} \widetilde{\mathbf{d_f}} & \mathbf{D_f} & \mathbf{E}_f \end{bmatrix}$$

and

$$\mathbf{S^T} = \begin{bmatrix} \mathbf{s_0} & \mathbf{I}_{2\times 2} & \mathbf{0}_{2\times \mathbf{P}} \\ a_{00} & \mathbf{a_1^T} & \mathbf{a_2^T} \end{bmatrix} \tag{19}$$

we rewrite the equation system (15) in matrix format as

$$\mathbf{R_f}^n = \mathbf{M_f S_n^T} \tag{20}$$

where the 2×3 matrix $\mathbf{M_f}$ contains the two first rows of the 3-D rotation matrix $\mathbf{\Theta_f}$ and the index n in $\mathbf{R_f^n}$ and $\mathbf{S_n^T}$ denotes the surface patch number. Expression (20) relates the image motion parameters at frame f and patch n, in matrix $\mathbf{R_f^n}$, to the 3-D rotation at frame f, in matrix $\mathbf{M_f}$, and the 3-D shape parameters for the patch n, in matrix $\mathbf{S_n^T}$.

B. Factorization

There are $N(F-1)$ matrix equations like (20): one for each surface patch $1 \le n \le N$ and each frame $2 \le f \le F$. To make explicit the entire set of equations that arise from considering every patch and every frame, we define the $2(F-1) \times N(P+3)$ matrix $\mathbf{R}$ of image motion parameters, the $2(F-1) \times 3$ matrix $\mathbf{M}$ of 3-D rotation parameters,and the $3 \times N(P+3)$ matrix $\mathbf{S^T}$ of 3-D shape parameters as

$$\mathbf{R} = \begin{bmatrix} \mathbf{R_2^1} & \mathbf{R_2^2} & \cdots & \mathbf{R_2^N} \\ \mathbf{R_3^1} & \mathbf{R_3^2} & \cdots & \mathbf{R_3^N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R_F^1} & \mathbf{R_F^2} & \cdots & \mathbf{R_F^N} \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M_2} \\ \mathbf{M_3} \\ \vdots \\ \mathbf{M_f} \end{bmatrix}, \qquad \mathbf{S^T} = \begin{bmatrix} \mathbf{S_1^T} & \mathbf{S_2^T} & \cdots & \mathbf{S_N^T} \end{bmatrix} \tag{21}$$

and write the relation between the image motion parameters and the 3-D structure parameters as

$$\mathbf{R} = \mathbf{M S^T}. \tag{22}$$

Expression (22) is an extension of the expression derived by Tomasi and Kanade [1] for the feature-based approach to the SFM problem. Expression (22), unlike the one in [1], accommodates regions whose shape is parameterized rather than described by a single point. In fact, the measurement matrix involved in the feature-based approach described in [1] is composed by the columns of the matrix $\mathbf{R}$ in (21) and (22) that contain the parameters $\{\widetilde{\mathbf{d}}_f^n, 1 \le n \le N, 2 \le f \le F\}$ [see (21) and (19)]. For this reason, we call the matrix $\mathbf{R}$ in (21) and (22) the *surface-based measurement matrix*. The shape matrix involved in the feature-based factorization of [1] is composed by the columns of the matrix $\mathbf{S^T}$ in (21) and (22) that contain the parameters $\{\mathbf{s_0^n}, \mathbf{a_{00}^n}, \mathbf{1 \le n \le N}\}$ [see (21) and (19)]. For the special case of piecewise planar shapes, the submatrices $\mathbf{R_f^n}$ and $\mathbf{S_n^T}$ in (19) and (20) are simplified —they have only the first three columns. In this case, the surface-based measurement matrix $\mathbf{R}$ is $2(F-1) \times 3N$ and the shape matrix $\mathbf{S^T}$ is $3 \times 3N$.The

motion matrix $\mathbf{M}$ in (21) and (22) is the same matrix that appears in the feature-based factorization of [1].

Expression (22) shows that the surface-based measurement matrix $\mathbf{R}$ of the image motion parameters is rank deficient. In a noiseless situation, the surface-based measurement matrix $\mathbf{R}$ is rank 3 reflecting the high redundancy in the image motion parameters, due to the rigidity of the object. Thus, the surface-based measurement matrix $\mathbf{R}$ has the same rank of the measurement matrix involved in the feature-based factorization of [1]. The problem of estimating the 3-D shape and 3-D motion parameters amounts to finding suitable factors of the surface-based measurement matrix $\mathbf{R}$. The 3-D shape matrix $\mathbf{S}$ and the 3-D motion matrix $\mathbf{M}$ are then the solution of

$$\min_{\mathbf{M},\mathbf{S}} \left\|\mathbf{R}-\mathbf{M}\mathbf{S}^{\mathbf{T}}\right\|_F \tag{23}$$

where the rows of the matrices $\mathbf{M}$ and $\mathbf{S}^{\mathbf{T}}$ are restricted to have the special structure of (21) – the rows of $\mathbf{M}$ are restricted to have: 1) unit norm and 2) row $2i-1$ orthogonal to row $2i$; and the first two rows of $\mathbf{S}^{\mathbf{T}}$ are given by (19).

The problem of estimating the matrix $\mathbf{M}$ and the third row of $\mathbf{S}^{\mathbf{T}}$ from the matrix $\mathbf{R}$, although nonlinear, has a specific structure: it is a bilinear constrained LS problem. The bilinear relation comes from (22), where the motion unknowns and the shape unknowns appear multiplied by each other and the constraints are imposed by the orthonormality of the rows of the matrix $\mathbf{M}$. This specific structure enables us to solve the nonlinear problem by using a computationally simple decomposition-normalization approach. The decomposition stage solves the unconstrained bilinear problem, leading to a solution up to a scale factor. The normalization stage determines the scale factor by approximating the constraints.

C. Decomposition

Define $\mathbf{M}=[\mathbf{M}_0,\mathbf{m}_3]$ and $\mathbf{S}=[\mathbf{S_0},\mathbf{s}]$. The matrices $\mathbf{M}_0$ and $\mathbf{S_0}$ contain the first two columns of the matrices $\mathbf{M}$ and $\mathbf{S}$, respectively, the vector $\mathbf{m}_3$ is the third column of $\mathbf{M}$ and the vector $\mathbf{s}$ is the third column of $\mathbf{S}$. We decompose the vector $\mathbf{s}$ into the component that belongs to the space spanned by the columns of $\mathbf{S_0}$ and the component orthogonal to this space as

$$\mathbf{s}=\mathbf{S_0}\mathbf{b}+\mathbf{a}, \qquad \text{with} \qquad \mathbf{a}^{\mathbf{T}}\mathbf{S_0}=[0 \quad 0]. \tag{24}$$

We rewrite the matrix $\mathbf{R}$ by inserting (24) in (22), obtaining

$$\mathbf{R}=\mathbf{M}_0\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}+\mathbf{m}_3\mathbf{b}^{\mathbf{T}}\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}+\mathbf{m}_3\mathbf{a}^{\mathbf{T}}. \tag{25}$$

The decomposition stage solves the matrix (25) with respect to the unknowns $\mathbf{M}_0$, $\mathbf{m}_3$, $\mathbf{b}$, and $\mathbf{a}$, ignoring the constraints imposed by the structure of the matrix $\mathbf{M}$. We formulate this problem as the unconstrained minimization

$$\min_{\mathbf{M}_0,\mathbf{m}_3,\mathbf{b},\mathbf{a}} \left\|\mathbf{R}-\mathbf{M}_0\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}-\mathbf{m}_3\mathbf{b}^{\mathbf{T}}\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}-\mathbf{m}_3\mathbf{a}^{\mathbf{T}}\right\|_F. \tag{26}$$

Since we know the matrix $\mathbf{S_0}$, we eliminate the dependence of (26) on $\mathbf{M}_0$ by solving the linear LS for $\mathbf{M}_0$ in terms of the other variables. We get

$$\widehat{\mathbf{M}}_0=\mathbf{R}\mathbf{S}_0\left(\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}\mathbf{S}_0\right)^{-1}-\mathbf{m}_3\mathbf{b}^{\mathbf{T}} \tag{27}$$

where we used the *Moore-Penrose pseudoinverse* [28] and the orthogonality between the vector $\mathbf{a}$ and the columns of the matrix $\mathbf{S_0}$ [see (24)]. By replacing $\widehat{\mathbf{M}}_0$ given by (27) in (26), we get

$$\min_{\mathbf{m}_3,\mathbf{a}} \left\|\widetilde{\mathbf{R}}-\mathbf{m}_3\mathbf{a}^{\mathbf{T}}\right\|_F$$

where

$$\widetilde{\mathbf{R}}=\mathbf{R}\left[\mathbf{I}-\mathbf{S_0}\left(\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}\mathbf{S_0}\right)^{-1}\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}\right]. \tag{28}$$

We see that the decomposition stage does not determine $\mathbf{b}$. This is because the component of $\mathbf{s}$ that lives in the space spanned by the columns of $\mathbf{S_0}$ does not affect the space spanned by the columns of the entire matrix $\mathbf{S}$ and the decomposition stage restricts only this latter space.

The solution for the vectors $\mathbf{m}_3$ and $\mathbf{a}$ is given by the rank 1 matrix that best approximates $\widetilde{\mathbf{R}}$. In a noiseless situation, $\widetilde{\mathbf{R}}$ is rank 1, since we would get

$$\widetilde{\mathbf{R}}=\mathbf{m}_3\mathbf{a}^{\mathbf{T}} \tag{29}$$

by replacing $\mathbf{R}$, given by (25), in (28). By computing the largest singular value of $\widetilde{\mathbf{R}}$ and the associated singular vectors, we get

$$\widetilde{\mathbf{R}}\simeq\mathbf{u}\sigma\mathbf{v}^{\mathbf{T}}, \qquad \widehat{\mathbf{m}}_3=\alpha\mathbf{u}, \qquad \hat{\mathbf{a}}^{\mathbf{T}}=\frac{\sigma}{\alpha}\mathbf{v}^{\mathbf{T}} \tag{30}$$

where α is a normalizing scalar different from zero.

To compute $\mathbf{u}$, σ, and $\mathbf{v}$, we could use *singular value decomposition (SVD)*, but the rank deficiency of $\widetilde{\mathbf{R}}$ enables the use of a less expensive algorithm known as the *power method* [28]. This makes our decomposition stage simpler than the one in the original factorization method of Tomasi and Kanade [1]. In fact, the matrix $\widetilde{\mathbf{R}}$ in (28) is equal to the matrix $\mathbf{R}$ multiplied by the orthogonal projector onto the orthogonal complement of the space spanned by the columns of $\mathbf{S_0}$. This projection reduces the rank of the problem from 3 (matrix $\mathbf{R}$) to 1 (matrix $\widetilde{\mathbf{R}}$).

D. Normalization

In this stage, we compute the scalar α and the vector $\mathbf{b}$ by imposing the constraints that come from the structure of the matrix $\mathbf{M}$.

By replacing the estimate $\widehat{\mathbf{m}}_3$, given by (30), in (27), we get for the estimate $\widehat{\mathbf{M}}$

$$\widehat{\mathbf{M}}=\left[\widehat{\mathbf{M}}_0 \quad \widehat{\mathbf{m}}_3\right]=\mathbf{N}\begin{bmatrix}\mathbf{I}_{2\times2} & \mathbf{0}_{2\times1}\\ -\alpha\mathbf{b}^{\mathbf{T}} & \alpha\end{bmatrix}$$

where

$$\mathbf{N}=\left[\mathbf{R}\mathbf{S}_0\left(\mathbf{S}_{\mathbf{0}}^{\mathbf{T}}\mathbf{S}_0\right)^{-1} \quad \mathbf{u}\right]. \tag{31}$$

The constraints imposed by the structure of the matrix $\mathbf{M}$ are the unit norm of each row and the orthogonality between the consecutive rows. In terms of $\mathbf{N}$, α, and $\mathbf{b}$, the constraints are then

$$\begin{aligned}\mathbf{n}_{\mathbf{i}}^{\mathbf{T}}\begin{bmatrix}\mathbf{I}_{2\times2} & -\alpha\mathbf{b}\\ -\alpha\mathbf{b}^{\mathbf{T}} & \alpha^2(1+\mathbf{b}^{\mathbf{T}}\mathbf{b})\end{bmatrix}\mathbf{n}_i&=1\\ \mathbf{n}_{2j-1}^{T}\begin{bmatrix}\mathbf{I}_{2\times2} & -\alpha\mathbf{b}\\ -\alpha\mathbf{b}^{\mathbf{T}} & \alpha^2(1+\mathbf{b}^{\mathbf{T}}\mathbf{b})\end{bmatrix}\mathbf{n}_{2j}&=0\end{aligned} \tag{32}$$

where $1\le i\le 2(F-1)$, $1\le j\le F-1$, and $\mathbf{n}_{\mathbf{i}}^{\mathbf{T}}$ denotes the row i of the matrix $\mathbf{N}$.

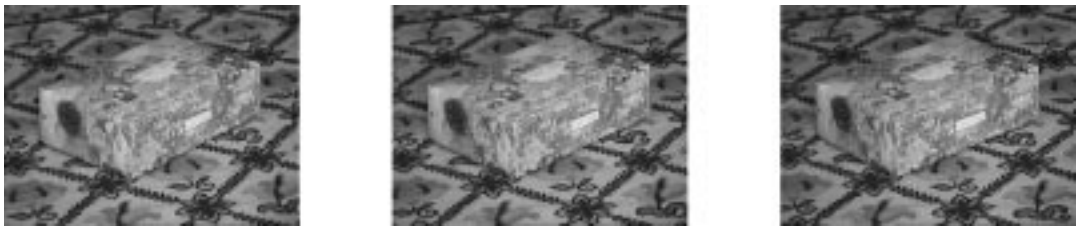

Fig. 3. Three consecutive frames of the box video sequence.

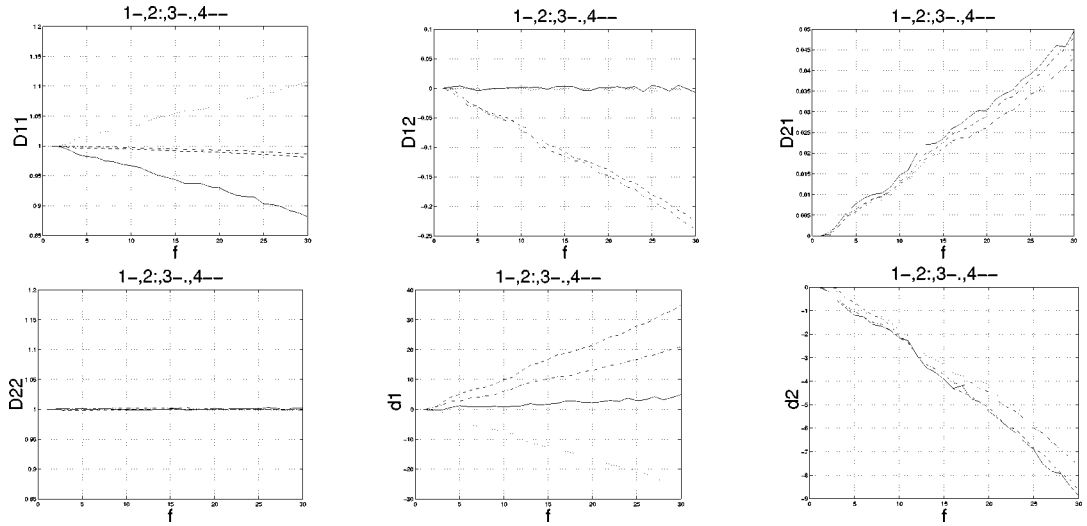

Fig. 4. Estimates of the image motion parameters in the 2×2 matrix $\mathbf{D_f^n}$ and the 2×1 vector $\mathbf{d_f^n}$. From left to right, top to bottom, $\mathbf{D}_{11}$, $\mathbf{D}_{12}$, $\mathbf{D}_{21}$, $\mathbf{D}_{22}$, $\mathbf{d}_1$, and $\mathbf{d}_2$.

We compute the normalization parameters α and $\mathbf{b}$ from the linear LS solution of the system of (32), in an analogous way to [1] and [25]. The normalization stage is also simpler than in the original factorization method [1] because the number of unknowns is three (α and $\mathbf{b} = [\mathbf{b_1}, \mathbf{b_2}]^{\mathbf{T}}$) as opposed to the nine entries of a generic 3×3 normalization matrix.

V. Experiments

In this section, we describe three experiments that illustrate the validity of our approach. We use the *surface-based factorization* method to analyze two real-life video clips. Finally, we demonstrate experimentally the computational savings of our approach when applied to the feature-based case.

A. Box Sequence

In this experiment, we taped a video sequence of 30 frames showing a box over a carpet with a hand-held camera. Fig. 3 shows three consecutive frames of the box video sequence. The 3-D shape of the scene is well described in terms of four planar patches. One corresponds to the floor and the other three correspond to the three visible faces of the box. The camera motion was approximately a rotation around the box.

We start by estimating the parameters describing the 2-D motion of the brightness pattern in the image plane. For planar patches, the 2-D motion in the image plane is described by the affine motion model. The plots in Fig. 4 represent the time evolution of the affine motion parameters. The six affine motion parameters are the entries of the 2×2 matrix $\mathbf{D_f^n}$ and the 2×1 vector $\mathbf{d_f^n}$ introduced in Section III [see (16)]. The top four plots of Fig. 4 represent the entries of $\mathbf{D_f^n}$ as a function of f for each of the four planar patches. The bottom two plots represent $\mathbf{d_f^n}$. The planar patches are identified as follows: the solid line corresponds to patch 1 (the left side vertical face of the box in the frames of Fig. 3); the dotted line corresponds to patch 2 (the right side vertical face of the box); the dash-dotted line corresponds to patch 3 (the top of the box); the dashed line corresponds to patch 4 (the floor). The plots show that the evolution of the set of affine parameters is distinct for each surface patch, in particular, see the evolution of $\mathbf{D}_{11}$, $\mathbf{D}_{12}$, and $\mathbf{d}_1$.

From the affine motion parameters of Fig. 4, we recover the 3-D structure of the scene by using the *surface-based factorization* method described in Section IV. After computing the 3-D structure parameters, we recover the texture of each surface patch by averaging the video frames co-registered according to the recovered 3-D structure. Fig. 5 shows six perspective views of the reconstructed 3-D shape with the scene texture mapped on it. The spatial limits of the planar patches were determined in the following way. The angles between the planar patches are correctly recovered. Each edge that links two visible patches was computed from the intersection of the planes corresponding to the patches. Each edge that is not in the intersection of two visible patches was computed by fitting a line to the boundary that separates two regions with different 2-D motion parameters. Note that the success of our method does not depend on an accurate segmentation of the planar patches. As pointed out in Section III, we can even select arbitrarily the support region of the patches, provided the region size is enough to enable the estimation of the 2-D image motion parameters.

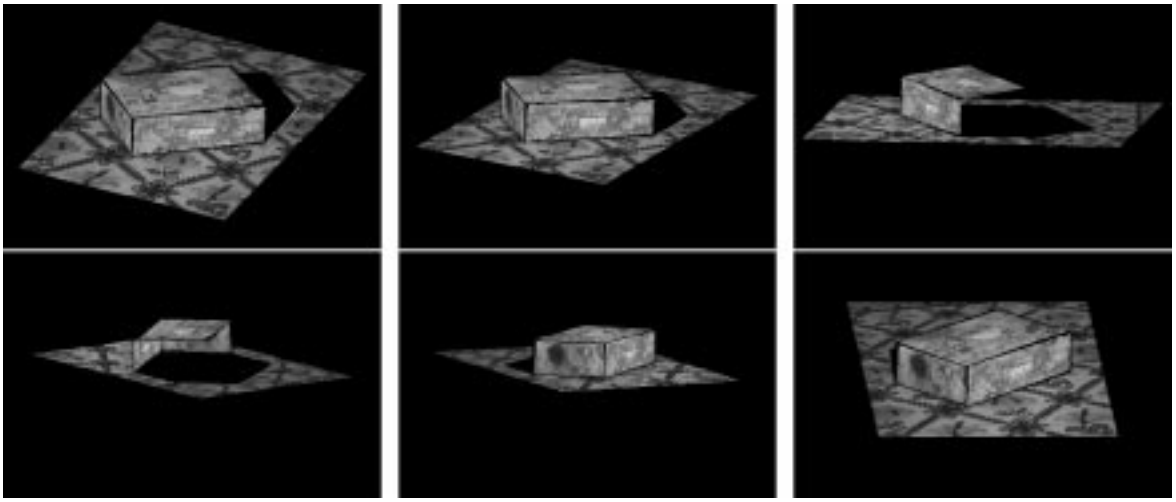

Fig. 5. Perspective views of the 3-D shape and texture reconstructed from the box sequence.

Fig. 6. Two frames from the pedestal sequence.

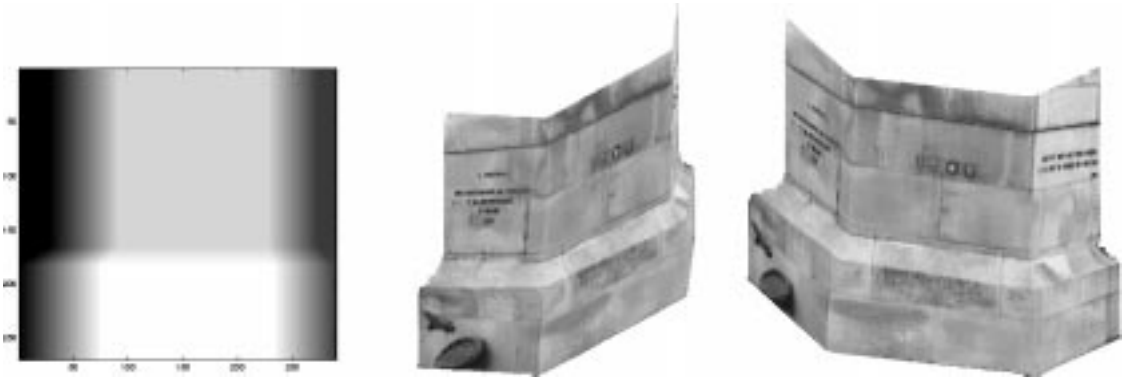

Fig. 7. Relative depth and reconstructed 3-D shape and texture.

B. Pedestal Sequence

In the second experiment a video shows a pedestal with nine patches. Fig. 6 shows two frames. We derive from the estimated 3-D shape parameters the relative depth of the pedestal shown on the left image of Fig. 7. In this image, the brightness level of a pixel codes the relative depth of that pixel, the brighter the pixel, the closer it is to the camera in reference frame 1. We reconstructed the 3-D shape from the depth map and superimposed the texture extracted from the sequence. Two perspective views of the reconstructed 3-D shape are shown on the center and rightmost images of Fig. 7. The nine planar patches of the pedestal are clearly seen as well as the angles between them. These two images represent two different views obtained by rotating the 3-D model. Other views are generated in a similar way.

C. Computational Cost

To compare the computational cost of our rank 1 matrix factorization algorithm with the rank 3 matrix factorization method originally proposed by Tomasi and Kanade [1], we specialize our approach to the feature-based case that is the only case addressed in [1]. We generated a set of N feature points randomly located inside a cube. The 3-D rotational motion was simulated by synthesizing a smooth time evolution for the Euler angles that specify the orientation of the o.c.s. relative to the c.c.s. We use the perspective projection model to project the features onto the image plane. The distance of the camera to the centroid of the set of feature points was set to a value high enough such that orthographic projection is a valid approximation. We ran the experiment described for a fixed number of $F = 50$ frames and a number of N feature points varying from 10 to 100; and

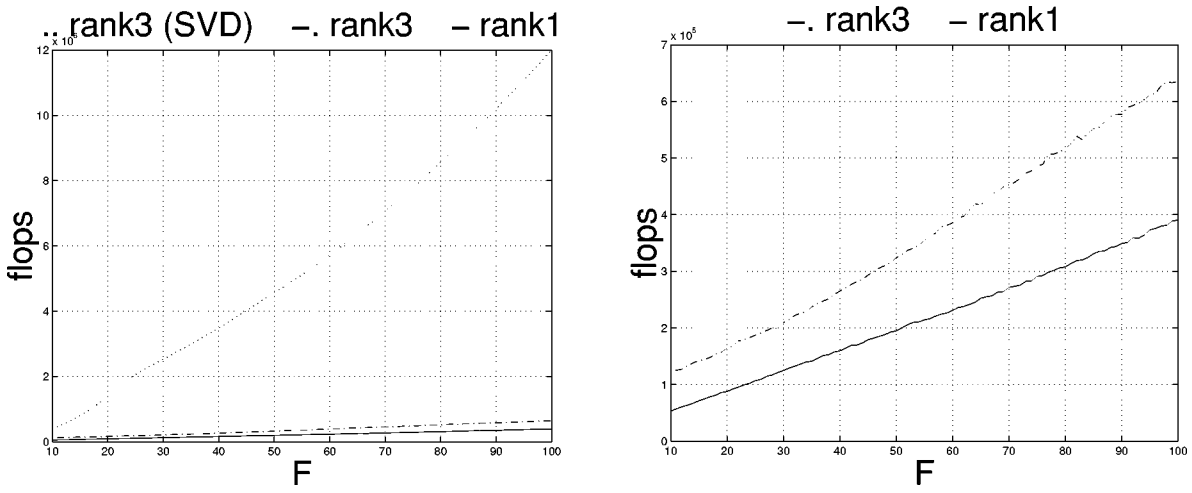

Fig. 8. MatLab FLOPS count as a function of the number of frames.

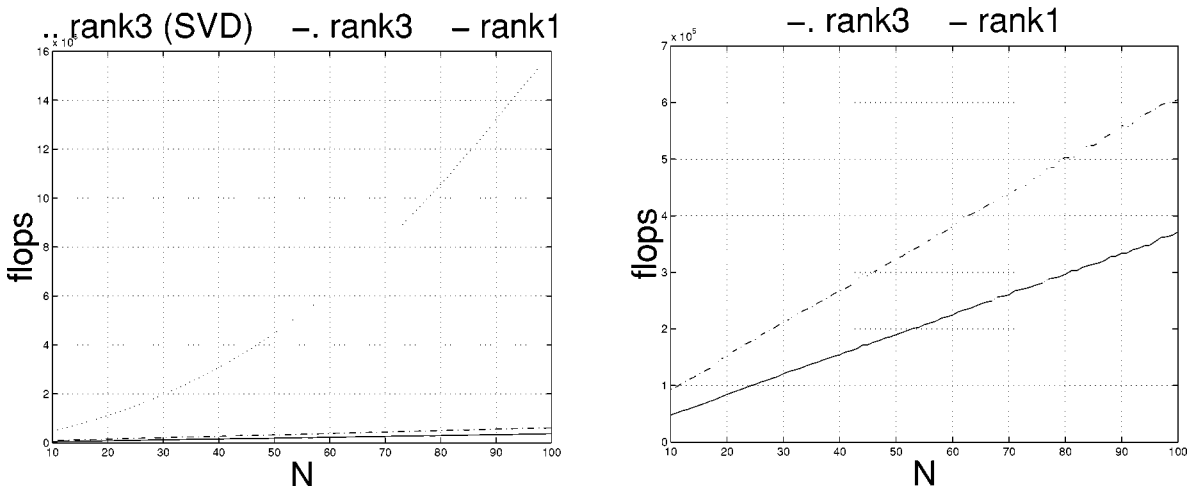

Fig. 9. MatLab FLOPS count as a function of the number of feature points.

for a fixed number of $N = 50$ feature points and a number of F frames varying from 10 to 100. We computed the average number of MatLab floating point operations (FLOPS) over 1000 tests for each experiment.

For each experiment, we estimate the 3-D shape and 3-D motion by using three methods: 1) the original factorization method [1] that computes the SVD of the measurement matrix $\mathbf{R}$; 2) the same method but computing the factorization of the rank 3 matrix $\mathbf{R}$ by using an algorithm detailed in [25], which is based on the *power method* [28]; and 3) our formulation of the factorization as a rank 1 problem. The reason why we include Method 2 in the experiment is because it is the fastest way available to compute the rank 3 matrix factorization. Figs. 8 and 9 plot the average number of FLOPS as a function of the number of frames and the number of feature points. The number of FLOPS are marked with dotted lines for Method 1, dash-dotted lines for Method 2, and solid lines for Method 3. The left plots show the three curves, while the right plots show only the curves for Methods 2 and 3 using a different vertical scale, for better visualization.

From the left side plots, we see that the number of FLOPS is much larger for the original factorization method than for our method. This is due to the high computational cost of the SVD. The computational gain factor is approximately 20 when processing 50 frames and 50 feature points and even larger when processing a larger number of features and/or a large number of frames. From the right side plots, we see that the number of FLOPS increases approximately linearly with both the number of frames and the number of feature points, for both iterative Methods 2 and 3. The rate of increase is lower for the factorization of the rank 1 matrix $\widetilde{\mathbf{R}}$ than the rank 3 matrix $\mathbf{R}$, by a factor of approximately 2. This is because both the decomposition and normalization stages in Method 3 are simpler than the ones in Method 2. In all of the experiments, the performance of the three methods in terms of the accuracy of the estimates of the 3-D structure is the same.

VI. Conclusion

We presented a new approach for the estimation of 3-D rigid shape and 3-D motion from a 2-D video sequence. We describe the 3-D shape by a parameterized representation. We show how this parametric representation induces a parametric representation for the 2-D image motion. Our method recovers the 3-D shape and 3-D motion by first estimating the image motion parameters. The rigidity of the 3-D shape along the image sequence leads to a highly constrained problem when estimating the 3-D structure parameters from the image motion parameters. These constraints are expressed in terms of a matrix that is rank

1 in a noiseless situation. Our method is based on the factorization of this matrix, leading to a computationally very simple algorithm. Its good performance is illustrated by two experiments with real-life video. In summary, there are two ways our method gains with respect to the original factorization method of Tomasi and Kanade [1]. First, a rank 1 factorization is simpler than a rank 3 factorization, as shown by Figs. 8 and 9. Second, planar patches (or in general higher order polynomial patches) lead to much sparser parametric descriptions for the 3-D surface than the feature points description, i.e., the number of patches required is in general significantly smaller than the number of feature points needed for similar levels of approximations, what this means is that the computational effort is reduced because the number of patches to be tracked is much smaller and because the matrix to be factored is also much smaller.

REFERENCES

[1] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, 1992.

[2] R. Chellapa and S. Srinivasan, "Structure from motion: Sparse versus dense correspondence methods," in *Proc. IEEE ICIP*, vol. 2, 1999, pp. 492–499.

[3] R. Tsai and T. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 1, 1984.

[4] T. Broida and R. Chellappa, "Recursive estimation of 3-D motion from a monocular image sequence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 4, 1990.

[5] S. Soatto, P. Perona, R. Frezza, and G. Picci, "Recursive motion and structure estimations with complete error characterization," in *Proc. IEEE CCVPR*, June 1993, pp. 428–433.

[6] J. Thomas, A. Hansen, and J. Oliensis, "Understanding noise: The critical role of motion error in scene reconstruction," in *Proc. IEEE ICCV*, 1993, pp. 325–329.

[7] A. Azarbayejani and A. P. Pentland, "Recursive estimation of motion, structure and focal length," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, June 1995.

[8] M. Spetsakis and Y. Aloimonos, "A multi-frame approach to visual motion perception," *Int. J. Comput. Vis.*, vol. 6, no. 3, pp. 245–255, 1991.

[9] T. Broida and R. Chellappa, "Estimating the kinematics and structure of a rigid object from a sequence of monocular images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, June 1991.

[10] J. Weng, N. Ahuja, and T. S. Huang, "Optimal motion and structure estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 864–884, Sept. 1993.

[11] R. Szeliski and S. Kang, "Recovering 3-D shape and motion from image streams using nonlinear least squares," *J. Vis. Commun. Image Represent.*, vol. 5, no. 1, 1994.

[12] C. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, Mar. 1997.

[13] J. P. Costeira and T. Kanade, "A factorization method for independently moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, 1998.

[14] B. K. P. Horn and E. J. Weldon , "Direct methods for recovering motion," *Int. J. Comput. Vis.*, vol. 2, no. 1, pp. 51–76, 1988.

[15] D. Michaels, "Exploiting continuity-in-time in motion vision ," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1992.

[16] G. P. Stein and A. Shashua, "Model-based brightness constraints: On direct estimation of structure and motion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 992–1015, Sept. 2000.

[17] J. Heel, "Direct estimation of structure and motion from multiple frames," Mass. Inst. Technol., Cambridge, MIT AI Lab. Memo 1190, 1990.

[18] T. Brodsky, C. Fermuller, and Y. Aloimonos, "Shape from video," in *Proc. IEEE CCVPR*, vol. 2, Fort Collins, CO, 1999, pp. 146–151.

[19] H. Zheng and S. D. Blostein, "Motion-based object segmentation and estimation using the MDL principle," *IEEE Trans. Image Processing*, vol. 4, Sept. 1995.

[20] S. Mann and R. Piccard, "Video orbits of the projective groups: A simple approach to featureless estimation of parameters," *IEEE Trans. Image Processing*, vol. 6, Sept. 1997.

[21] M. Chang, M. Tekalp, and M. Sezan, "Simultaneous motion estimation and segmentation," *IEEE Trans. Image Processing*, vol. 6, Sept. 1997.

[22] P. M. Q. Aguiar and J. M. F. Moura, "Video representation via 3-D shaped mosaics," in *Proc. IEEE ICIP*, Chicago, IL, Oct. 1998.

[23] ——, "A fast algorithm for rigid structure from image sequences," in *Proc. IEEE ICIP*, Kobe, Japan, Oct. 1999.

[24] ——, "Maximum likelihood inference of 3-D structure from image sequences," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. New York: Springer–Verlag, 1999.

[25] P. M. Q. Aguiar, "Rigid structure from video ," Ph.D. dissertation, Instituto Superior Técnico, Lisboa, Portugal, 2000.

[26] S. Ayer, "Sequential and competitive methods for estimation of multiple motions," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 1995.

[27] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. ECCV*, Italy, May 1992, pp. 237–252.

[28] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed, ser. Johns Hopkins Series in Mathematical Sciences. Baltimore, MD: Johns Hopkins Univ. Press, 1989.

Pedro M. Q. Aguiar received the Ph.D. degree in electrical and computer engineering from Instituto Superior Técnico (IST), Technical University of Lisbon, Portugal, in 2000.

He is presently a Researcher at the Institute for Systems and Robotics (ISR), Lisbon and an Assistant Professor of Electrical and Computer Enginnering at IST.

His main research interests are in image analysis and computer vision.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the engenheiro electrotécnico degree in 1969 from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc. and E.E. degrees in 1973 and the D.Sc. degree in electrical engineering and computer science in 1975, all from the Massachusetts Institute of Technology (MIT), Cambridge.

In 1986, he joined Carnegie Mellon University, Pittsburgh, PA, as a Professor of electrical and computer engineering. Other appointments have included Visiting Professor of electrical engineering at the Massachusetts Institute of Technology, Cambridge, (1999–2000), Professor Auxiliar, Professor Agregado, and Professor Catedrático at IST (1975–1984), Visiting Research Scholar at the University of Southern California (Department of Aerospace Engineering, Summers 1978–1981), and Genrad Associate Professor of EECS (Visiting) at M.I.T. (1984–1986). His research interests include statistical signal processing and telecommunications, image processing, and video representations. He has published over 230 technical contributions, coedited two books, and holds four patents on image and video processing and digital communications with the U.S. Patent Office. He has given numerous invited seminars at U.S. and European universities and laboratories.

Dr. Moura's service with the IEEE Signal Processing Society includes being Vice-President of Publications (2000–2002), Elected Member of the Board of Governors (1999–2002), Chair of the Publications Board (2000–2002), Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (1995–1999), Member of several technical committees, Member of program committees for numerous conferences and workshops, and Associate Editor for RANSACTIONS ON SIGNAL PROCESSING (1988–1992) and IEEE SIGNAL PROCESSING LETTERS (1995–1999). His other service with IEEE includes Vice-President of Publications of the Sensors Council (2000–2002), Member of the Editorial Board of IEEE PROCEEDINGS (1999–present), and Member of the IEEE Press Board (1991–1995). He was Guest Coeditor of the IEEE TRANSACTIONS ON INFORMATION THEORY August 2000 Special Issue. He is a Fellow of the IEEE and Corresponding Member of the Academy of Sciences of Portugal (Section of Sciences). He is affiliated with several IEEE societies, SPIE, Sigma Xi, AMS, AAAS, IMS, and SIAM. In 2000, he received the IEEE Millenium Medal.