Rank 1 Weighted Factorization for 3D Structure Recovery: Algorithms and Performance Analysis

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

Abstract—The paper describes the *rank 1 weighted factorization* solution to the *structure from motion* problem. This method recovers the 3D structure from the factorization of a data matrix that is rank 1 rather than rank 3. This matrix collects the estimates of the 2D motions of a set of feature points of the rigid object. These estimates are weighted by the inverse of the estimates error standard deviation so that the 2D motion estimates for "sharper" features, which are usually well-estimated, are given more weight, while the noisier motion estimates for "smoother" features are weighted less. We analyze the performance of the *rank 1 weighted factorization* algorithm to determine what are the most suitable 3D shapes or the best 3D motions to recover the 3D structure of a rigid object from the 2D motions of the features. Our approach is developed for the orthographic camera model. It avoids expensive singular value decompositions by using the power method and is suitable to handle dense sets of feature points and long video sequences. Experimental studies with synthetic and real data illustrate the good performance of our approach.

Index Terms—Factorization methods, structure from motion, image sequence analysis, rigid body motion, uncertainty in motion analysis, power method, weighted factorization.

1 Introduction

We propose the *rank 1 weighted factorization* algorithm to solve the *structure from motion* (SFM) problem for the orthographic camera model—recovering the 3D structure (3D shape and 3D motions) of a rigid object from the noisy estimates of the 2D motions across a monocular video sequence of point features of the object.

1.1 Brief Review of the Literature

The computer vision literature has widely addressed the problem of recovering 3D structure from a video sequence—the *structure from motion* (SFM) problem—since the strongest available cue in an image sequence is the 2D motion of the brightness pattern in the image plane. Applications range from robotics to digital video.

A number of approaches use only two or three consecutive frames. Most start by computing the 2D image motion, either in terms of a set of correspondences between feature points [1], or a dense optical flow map [2]. Then, the 3D motion and 3D shape are computed from the 2D motion estimates. Others overcome the ill-posedness inherent to the estimation of the 2D image motion by using only the normal component of the optical flow [3], or by estimating the 3D structure directly from the image intensity values [4], [5], without computing the 2D motion as an intermediate step.

When the scene is rigid, processing the whole video sequence can lead to a more accurate estimate of the 3D structure since it uses the 2D motion of the brightness pattern across a large set of frames. However, multiframe SFM is a challenge due to the nonlinearity and high-dimensionality in the problem. Existing approaches to multiframe SFM include: 1) nonlinear optimization methods, for example, a popular choice in computer vision is the Levenberg-Marquardt procedure [6], [7]; 2) recursive estimation techniques based on the extended Kalman-Bucy filter (EKBF) [8], [9]; or 3) linear subspace constraints that lead to the so-called factorization methods introduced by Tomasi and Kanade in the early 1990s [10], [11], [12].

The *factorization method* is an attractive approach to recover the 3D motion and 3D shape of a rigid object. The original formulation used the orthographic projection model that is known to be a good approximation to the perspective projection when the object is far from the camera. It tracks a set of N feature points across an image sequence of F frames and collects these trajectories in a $2F \times N$ measurement matrix $\mathbf{R}$. Due to the rigidity of the object, the measurement matrix $\mathbf{R}$ is rank 3 in a noiseless situation—it is the product of a $2F \times 3$ motion matrix by a $3 \times N$ shape matrix. The 3D motion of the camera and the 3D positions of the features are recovered by Singular Value Decomposition (SVD) of the measurement matrix $\mathbf{R}$.

The factorization method was later extended to the scaled-orthographic, or pseudo-perspective, and paraperspective projection models [13], [14]. Factorization-like algorithms were also proposed to address the full perspective projection model, see, for example, [15]. Other authors used correspondences between line segments [16]. We have extended in prior work the factorization method to work with surface patches [17], [18] rather than feature points. Morita and Kanade [19] proposed a recursive algorithm for

- *P.M.Q. Aguiar is with ISR—Institute for Systems and Robotics, IST—Instituto Superior Técnico, Torre Norte, Av. Rovisco Pais, 1049-001 Lisboa, Portugal. E-mail: aguiar@isr.ist.utl.pt.*
- *J.M.F. Moura is with the Electrical and Computer Engineering Department, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213-3890. E-mail: moura@ece.cmu.edu.*

Manuscript received 3 Jan. 2002; revised 30 Aug. 2002; accepted 13 Dec. 2002.
Recommended for acceptance by Z. Zhang.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115642.

0162-8828/03/$17.00 © 2003 IEEE Published by the IEEE Computer Society

the factorization method and reference [20] treated the multibody case.

1.2 Rank 1 Weighted Factorization

Rank 1 factorization. In this paper, we exploit a degree of freedom not yet used by existing factorization frameworks—the freedom to choose the relative alignment between the object and camera coordinate systems in *one* of the images in the sequence. We develop a method that recovers the SFM through the factorization of a *rank 1* matrix rather than a rank 3 matrix. We avoid altogether singular value decomposition (SVD) computations by performing the rank 1 factorization by the *power method* [21], a computationally much simpler method than the SVD of a rank 3 matrix. This reduces significantly the cost of the factorization method, which is highly relevant, in practice, where the dimensions of $\mathbf{R}$ can be very large. We develop our method for the orthographic camera model.

Weighted factorization. In practice, the entries of the matrix $\mathbf{R}$ that are the estimates of the 2D motions are noisy. Further, "sharper" features are usually easier to track than features with "smoother" spatial brightness. To accommodate these different levels of errors in the 2D motion estimates, [22] develops a two-step suboptimal algorithm that factors a rank 6 matrix. Another issue is occlusion. Poelman [13] considers reliability weights—when a feature is lost, it is given the weight of zero—and recovers the 3D structure by an iterative method, which may fail to converge. This iterative method was later extended to accommodate reliability weights other than zero or one [23].

We derive a *weighted* version of the *rank 1 factorization*. By choosing the weights to be time invariant, the *weighted* rank 1 factorization is equivalent to the *non*weighted rank 1 factorization of a *modified* data matrix: The resulting algorithm is noniterative *and* factors a matrix that is still rank 1.

Performance. An interesting theoretical, as well as practical, issue is to know what 3D motions are better suited to recover the 3D shape of an object, or what 3D shapes are better restored from the 2D motions. We answer these questions by analysis of the rank 1 factorization algorithm. We show, for example, that the shape is best retrieved from orthogonal views aligned with the longest and smallest axes of inertia of the object.

1.3 Paper Organization

In Section 2, we review the factorization approach to SFM. Section 3 details the two stages of the *rank 1 factorization* method—decomposition and normalization. In Section 4, we analyze how the 3D structure (motion and shape) affects the behavior of the decomposition and normalization stages. Section 5 extends our approach to accommodate different confidence weights associated with the feature points, introducing the *rank 1 weighted factorization* method. The appendix presents closed form expressions for the weights. In Section 6, we describe experiments that illustrate and demonstrate the performance of our methods. Section 7 concludes the paper. A summary of some results in this paper on the rank 1 factorization was presented in [24].

2 Structure from Motion: Factorization Approach

We consider a rigid body viewed by a camera; either the object, the camera, or both can move. Without loss of generality, we discuss a moving object and a static camera. We assume the orthographic camera model. We associate to the object and to the camera an object coordinate system (o.c.s.) and a camera coordinate system (c.c.s.) with axes labeled by x, y, and z, and u, v, and w, respectively. The plane defined by the axes u and v is the camera plane.

In this paper, the shape of the object is described by the 3D position (x_n, y_n, z_n) with respect to (wrt) the o.c.s. of a set of $n = 1, \ldots, N$ feature points. The 3D motion of the object is defined by specifying the position of the o.c.s. $\{x, y, z\}$ relative to the c.c.s. $\{u, v, w\}$, i.e., by specifying at each instant f a translation-rotation pair $\left(\boldsymbol{\tau}_f, \boldsymbol{\Theta}_f\right)$. The translation vector $\boldsymbol{\tau}_f = \left[t_{uf}, t_{vf}, t_{wf}\right]^T$ defines the coordinates of the origin of the o.c.s. wrt the c.c.s., and the rotation matrix $\boldsymbol{\Theta}_f$ orients the o.c.s. relative to the c.c.s.

At instant f, feature n has the following coordinates in the camera coordinate system,

$$1 \leq f \leq F,\ 1 \leq n \leq N:$$
$$\begin{bmatrix} u_{fn} \\ v_{fn} \\ w_{fn} \end{bmatrix} = \underbrace{\begin{bmatrix} i_{xf} & i_{yf} & i_{zf} \\ j_{xf} & j_{yf} & j_{zf} \\ k_{xf} & k_{yf} & k_{zf} \end{bmatrix}}_{\boldsymbol{\Theta}_f} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} + \underbrace{\begin{bmatrix} t_{uf} \\ t_{vf} \\ t_{wf} \end{bmatrix}}_{\boldsymbol{\tau}_f}. \tag{1}$$

The entries of $\boldsymbol{\Theta}_f$, i_{xf}, j_{xf}, k_{xf} are the direction cosines of the x-axis wrt each of the axis u, v, and w, and similarly for the remaining entries of $\boldsymbol{\Theta}_f$. Equation (1) encapsulates the rigid body assumption: The instantaneous rotation matrix $\boldsymbol{\Theta}_f$ and the translation vector $\boldsymbol{\tau}_f$ that define the rigid body motion at time f are the same for all features $1 \leq n \leq N$.

In the video sequence, only the projections on the image plane are available. The corresponding coordinates are given by the first two equations in (1),

$$1 \leq f \leq F,\ 1 \leq n \leq N:$$
$$\underbrace{\begin{bmatrix} u_{fn} \\ v_{fn} \end{bmatrix}}_{\mathbf{u}_{fn}} = \underbrace{\begin{bmatrix} i_{xf} & i_{yf} & i_{zf} \\ j_{xf} & j_{yf} & j_{zf} \end{bmatrix}}_{\boldsymbol{\Psi}_f} \underbrace{\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}}_{\mathbf{s}_n} + \underbrace{\begin{bmatrix} t_{uf} \\ t_{vf} \end{bmatrix}}_{\mathbf{t}_f}. \tag{2}$$

Expression (2) makes clear that, due to the orthographic camera model, the feature projections do not depend on the translational component t_{wf} along the w-axis, the axis perpendicular to the camera plane. The translational component of the motion that can be recovered under orthography is the translation parallel to the camera plane, represented by the vector $\mathbf{t}_f = \left[t_{uf}, t_{vf}\right]^T$.

Collecting the N vector-equations corresponding to instant f, we get the matrix equation

$$\begin{aligned} &1 \leq f \leq F: \\ &[\,\mathbf{u}_{f1} \quad \ldots \quad \mathbf{u}_{fN}\,] = \boldsymbol{\Psi}_f[\,\mathbf{s}_1 \quad \ldots \quad \mathbf{s}_N\,] + [\,\mathbf{t}_f \quad \ldots \quad \mathbf{t}_f\,] \end{aligned} \tag{3}$$

that again, using obvious notation, is written in matrix format

$$1 \leq f \leq F: \qquad \mathbf{U}_f^T = \boldsymbol{\Psi}_f \mathbf{S}^T + \mathbf{t}_f \mathbf{1}^T, \tag{4}$$

where $\mathbf{1} = [1, \ldots, 1]^T$ is an N-dimensional vector.

By centering the object coordinate system wrt the centroid of the object features in 3D, we have $\sum_n x_n = \sum_n y_n = \sum_n z_n = 0$. Likewise, centering the camera coordinate system wrt the centroid of the projections of the features on the image plane, we get $\sum_{m=1}^{N} \mathbf{u}_{fm} = \mathbf{0}$. With these centered coordinate systems, (4) is simply rewritten as

$$1 \leq f \leq F : \qquad \mathbf{U}_f^T = \mathbf{\Psi}_f \mathbf{S}^T. \tag{5}$$

We now collect the F matrix equations in the single matrix equation

$$\begin{bmatrix} \mathbf{U}_1^T \\ \vdots \\ \mathbf{U}_F^T \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi}_1 \\ \vdots \\ \mathbf{\Psi}_F \end{bmatrix} \mathbf{S}^T, \tag{6}$$

which we write compactly as

$$\overline{\mathbf{R}} = \overline{\mathbf{M}} \mathbf{S}^T. \tag{7}$$

The following terminology is common: $\overline{\mathbf{R}}$ is the measurement or data matrix, $\overline{\mathbf{M}}$ is the motion matrix, and $\mathbf{S}$ is the shape matrix. To summarize, these matrices are

$$\overline{\mathbf{R}} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1N} \\ v_{11} & v_{12} & \cdots & v_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{F1} & u_{F2} & \cdots & u_{FN} \\ v_{F1} & v_{F2} & \cdots & v_{FN} \end{bmatrix}, \quad \overline{\mathbf{M}} = \begin{bmatrix} i_{x1} & i_{y1} & i_{z1} \\ j_{x1} & j_{y1} & j_{z1} \\ \vdots & \vdots & \vdots \\ i_{xF} & i_{yF} & i_{zF} \\ j_{xF} & j_{yF} & j_{zF} \end{bmatrix}, \tag{8}$$
$$\mathbf{S} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix}.$$

The matrix format equation (7) was introduced by Tomasi and Kanade in their original work, see [10], [11], [12]. These references reduce the SFM problem to the following. The projections of the N features are tracked across the F frames, i.e., $\overline{\mathbf{R}}$ is measured. This $2F \times N$ matrix is rank deficient. In a noiseless situation, $\overline{\mathbf{R}}$ is rank 3, reflecting the high redundancy in the data due to the 3D rigidity of the object. The factorization approach of Tomasi and Kanade formulates the SFM problem as the minimization

$$\min_{\overline{\mathbf{M}},\mathbf{S}} \| \overline{\mathbf{R}} - \overline{\mathbf{M}} \mathbf{S}^T \|_F, \tag{9}$$

where the solution space is constrained by the structure of the matrix $\overline{\mathbf{M}}$. The notation $\|.\|_F$ represents the Frobenius norm [21]. Tomasi and Kanade [12] present a suboptimal solution to this factorization in two stages. The first stage, *decomposition stage*, solves $\overline{\mathbf{R}} = \overline{\mathbf{M}} \mathbf{S}^T$ in the least square (LS) sense by computing the SVD of $\overline{\mathbf{R}}$ and selecting the three largest singular values. From $\overline{\mathbf{R}} \simeq \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$, where $\mathbf{U}$ is $2F \times 3$, $\mathbf{\Sigma}$ is diagonal 3×3, and $\mathbf{V}^T$ is $3 \times N$, one solution is $\overline{\mathbf{M}} = \mathbf{U} \mathbf{\Sigma}^{\frac{1}{2}} \mathbf{A}$, $\mathbf{S}^T = \mathbf{A}^{-1} \mathbf{\Sigma}^{\frac{1}{2}} \mathbf{V}^T$, where $\mathbf{A}$ is a nonsingular 3×3 matrix. The second stage, *normalization stage*, computes $\mathbf{A}$ by approximating the constraints imposed by the structure of the matrix $\overline{\mathbf{M}}$. Although the overall result of the decomposition and normalization is suboptimal in an Euclidean (or metric) sense, it is interesting to note that the rank 3 algorithm of Tomasi and Kanade is the optimal affine reconstruction after decomposition of the 3D structure and camera motion.

3 Rank 1 Factorization

We derive now an alternative solution to the factorization in (7) and (9) by exploiting further the structure of the SFM problem. There is an additional degree of freedom that is not exploited in the rank 3 factorization algorithm and that is key to our development: The shape is invariant to the particular relation between the object coordinate system and the camera coordinate system. In other words, we can fix the orientation of the o.c.s. with respect to the c.c.s. in *one* of the images f—we call this image f the reference frame. To be specific, we make the o.c.s. and the c.c.s. parallel in the first frame $f = 1$. With this choice, the 3D x- and y-coordinates of each feature n equal the 2D u- and v-coordinates of the projection of this feature n in the camera plane in the first frame,

$$1 \leq n \leq N : \qquad x_n = u_{f=1,n} \qquad \text{and} \qquad y_n = v_{f=1,n}. \tag{10}$$

But, from (8), x_n and y_n, $1 \leq n \leq N$, are the first two columns of the 3D shape matrix $\mathbf{S}$ and, so, (10) means that these two first columns of $\mathbf{S}$ are known and given by the pixel coordinates of the features in frame 1.[1] We take advantage of this fact in our formulation of the SFM problem, the *rank 1 factorization*. By knowing two columns of $\mathbf{S}$, SFM is reduced to the following much simpler formulation: given the matrix $\overline{\mathbf{R}}$ of 2D motions, compute the matrix $\overline{\mathbf{M}}$ of 3D motions and the third column of the shape matrix $\mathbf{S}$, i.e., the coordinates $\{z_n, 1 \leq n \leq N\}$ of the features. In the original factorization method [10], [11], [12], the 3D structure is recovered up to a 3D rigid rotation. Our choice of the alignment of the coordinate systems is equivalent to fixing this 3D rigid rotation in such a way that the camera rotation matrix in the reference frame is the identity matrix.

Because we describe the unknown shape by the distances along the third dimension for each pixel of the image plane, this formulation seems to be in opposition to the idea behind the original factorization method as formulated by Tomasi and Kanade [10], [11], [12]. In their first paper [10], the factorization method is motivated by emphasizing that when the object is far from the camera the depth cannot be computed, and the 3D shape must be represented in terms of the set of coordinates $\{x_n, y_n, z_n\}$. We show here that, if the unknown shape is represented by the entities we really do not know, i.e., by the *relative depths* $\{z_n\}$, the solution to the problem is simplified. In a certain sense, we simplify the rank 3 factorization method by constraining the problem further, much like Azarbayejani and Pentland [9] used the fact that the coordinates along the axes defining the camera plane are known to simplify earlier approaches to computing rigid SFM [8] by extended Kalman-Bucy filtering.

We comment here on one aspect that may be a source of confusion regarding our approach. At first sight, it seems that we have artificially simplified the original SFM problem by introducing an arbitrary assumption, namely, that we force the 3D x- and y-coordinates of the features to be known, or that their 2D motion estimates in the reference frame are known

[1] In practice, most often, it is of course the other way around, we choose certain pixels in a reference frame as 2D features, track their motions, and reconstruct the 3D shape from their lifting to 3D space.

with no errors. In fact, we do not "arbitrarily" make such an assumption, rather, we exploit this fact. In other words, this is a "feature," not a "bug" of our method. First, in many applications in computer vision and in image processing, features are not selected in 3D space since the object is most likely not accessible, rather, they are selected indirectly by choosing appropriate pixels in a reference image of the video sequence, say frame $f = 1$. Doing so, the 2D positions of these features in this reference frame $f = 1$ are known, since, after all, we picked them: say, when choosing a pixel, for example, at position $u_{f=1,1} = 135$ and $v_{f=1,1} = 147$ in frame 1 as (the projection of) feature 1, we know exactly, with no error, the coordinates of this pixel in the first frame. By aligning the object coordinate system in frame 1 with the camera coordinate system, we make $x_{n=1} = u_{f=1,1} = 135$ and $y_{n=1} = v_{f=1,1} = 147$ and, similarly, with the other $N - 1$ features. Second, the 2D motions correspond to the *displacements* of pixels across frames, for example, between pixels in frames $f \geq 2$ and the reference frame $f = 1$. These displacements are estimated with errors. Our approach does work with these noisy estimates just like the rank 3 algorithm. It is not that the rank 1 method "arbitrarily" reduces the errors of the estimates of the x_n and y_n coordinates of the features. On the contrary, it exploits this additional structure and fixes these coordinates. The 2D projections u_{fn} and v_{fn} of the features $f \geq 2$, are still noisy and with errors. Finally, by extracting from prior knowledge the x- and y-coordinates of the features, we are left with estimating from the measurements the third coordinate z_n, $n = 1, \cdots, N$, a much simpler problem than the original problem of estimating *all* the 3D coordinates (x_n, y_n, z_n), for all the features $1 \leq n \leq N$.

On the other hand, there are applications where the features cannot be chosen arbitrarily in a reference frame, for example, because they are preselected from the 3D object (e.g., using markers). In such applications the x- and y-coordinates should not be assumed to be known with no errors. In Section 4, we study the performance of the algorithm in recovering the 3D structure in such applications.

Although nonlinear, the problem of estimating the matrix $\overline{\mathbf{M}}$ and the vector $\mathbf{z} = [z_1, \ldots, z_N]^T$ from the matrix $\overline{\mathbf{R}}$ has a specific structure: it is a bilinear constrained LS problem. The bilinear relation comes from (7), where the motion unknowns and the shape unknowns appear multiplied by each other, and the constraints are imposed by the pairwise orthonormality of the rows of the motion matrix $\overline{\mathbf{M}}$. Our solution is in two steps: the *decomposition stage* that solves the unconstrained bilinear problem; and the *normalization stage* that applies the orthogonal constraints.

3.1 Decomposition Stage

Analogously to the decomposition stage of the original rank 3 factorization [11], [12], which computes the optimal affine shape, the decomposition stage of the rank 1 factorization algorithm computes the optimal relative depth subspace.

We start by defining the matrices $\mathbf{R}$ and $\mathbf{M}$ by excluding from $\overline{\mathbf{R}}$ and $\overline{\mathbf{M}}$ in (8) the rows corresponding to frame 1, thus $\mathbf{R}$ is $2(F-1) \times N$ and $\mathbf{M}$ is $2(F-1) \times 3$. Then, write

$$\mathbf{M} = [\,\mathbf{M}_0 \quad \mathbf{m}_3\,] \quad \text{and} \quad \mathbf{S} = [\,\mathbf{x} \quad \mathbf{y} \quad \mathbf{z}\,] = [\,\mathbf{S}_0 \quad \mathbf{z}\,], \tag{11}$$

where 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 vectors $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ are the columns of $\mathbf{S}$. Let the vector spaces

$$\begin{aligned} \mathcal{S}_0 &= \text{range}\,(\mathbf{S}_0) = \text{span}\,\{\mathbf{x}, \mathbf{y}\} \quad \text{and} \\ \mathcal{S}_0^{\perp} &= \left\{\mathbf{u} : \mathbf{u}^T\mathbf{v} = 0, \quad \forall \mathbf{v} \in \mathcal{S}_0\right\} \end{aligned} \tag{12}$$

represent, respectively, the space spanned by the columns of $\mathbf{S}_0$ and its orthogonal complement. We decompose the relative depth vector $\mathbf{z}$ into the component $\mathbf{S}_0\mathbf{b}$ that belongs to $\mathcal{S}_0$ and the component $\mathbf{a}$ that belongs to $\mathcal{S}_0^{\perp}$,

$$\mathbf{z} = \mathbf{S}_0\mathbf{b} + \mathbf{a}, \qquad \text{with} \qquad \mathbf{a}^T\mathbf{S}_0 = [\,0 \quad 0\,]. \tag{13}$$

We use (11) and (13) to rewrite the matrix $\mathbf{R}$ in (7), obtaining

$$\mathbf{R} = \mathbf{M}_0\mathbf{S}_0^T + \mathbf{m}_3\mathbf{b}^T\mathbf{S}_0^T + \mathbf{m}_3\mathbf{a}^T. \tag{14}$$

The decomposition stage solves equation (14) with respect to the unknowns $\mathbf{M}_0$, $\mathbf{m}_3$, $\mathbf{b}$, and $\mathbf{a}$ as the unconstrained minimization

$$\min_{\mathbf{M}_0, \mathbf{m}_3, \mathbf{b}, \mathbf{a}} \left\| \mathbf{R} - \mathbf{m}_0\mathbf{S}_0^T - \mathbf{m}_3\mathbf{b}^T\mathbf{S}_0^T - \mathbf{m}_3\mathbf{a}^T \right\|_F. \tag{15}$$

We minimize (15) wrt $\mathbf{M}_0$ using the fact that the matrix $\mathbf{S}_0$ is known. Standard algebraic manipulations [21] and using the orthogonality between the vector $\mathbf{a}$ and the columns of the matrix $\mathcal{S}_0$, $\mathbf{a} \in \mathcal{S}_0^{\perp}$, see (13), lead to the estimate $\widehat{\mathbf{M}}_0$ of $\mathbf{M}_0$,

$$\widehat{\mathbf{M}}_0 = \mathbf{R}\mathbf{S}_0\left(\mathbf{S}_0^T\mathbf{S}_0\right)^{-1} - \mathbf{m}_3\mathbf{b}^T. \tag{16}$$

Replacing $\widehat{\mathbf{M}}_0$ in (14), we obtain the matrix $\widetilde{\mathbf{R}}$

$$\widetilde{\mathbf{R}} = \mathbf{R}\left[\mathbf{I} - \mathbf{S}_0\left(\mathbf{S}_0^T\mathbf{S}_0\right)^{-1}\mathbf{S}_0^T\right] = \mathbf{R}\mathbf{\Pi}_{\mathcal{S}_0^{\perp}}, \tag{17}$$

where $\mathbf{\Pi}_{\mathcal{S}_0^{\perp}}$ is the orthogonal projector onto the known subspace $\mathcal{S}_0^{\perp}$, see [21], given by

$$\mathbf{\Pi}_{\mathcal{S}_0^{\perp}} = \mathbf{I} - \mathbf{S}_0\left(\mathbf{S}_0^T\mathbf{S}_0\right)^{-1}\mathbf{S}_0^T. \tag{18}$$

The minimization in (15) becomes

$$\min_{\mathbf{m}_3, \mathbf{a}} \left\| \widetilde{\mathbf{R}} - \mathbf{m}_3\mathbf{a}^T \right\|_F. \tag{19}$$

We interpret (19) and (17). First, the solution for the vectors $\mathbf{m}_3$ and $\mathbf{a}$ in (19) is obtained from the rank 1 matrix that best approximates $\widetilde{\mathbf{R}}$. Second, only the (direction of the) component $\mathbf{a}$ of $\mathbf{z}$ is determined in the decomposition stage; the other component $\mathbf{b}$ of $\mathbf{z}$ that lies in the known subspace $\mathcal{S}_0$ is left undetermined at this stage. Finally, (17) says that the relevant information in the measurements $\mathbf{R}$ regarding the SFM problem is in the matrix $\widetilde{\mathbf{R}}$ that is the projection of the 2D motions onto the subspace orthogonal to $\mathcal{S}_0$ generated by $\mathbf{x}$ and $\mathbf{y}$.

The rank 1 SVD solution to (19) is

$$\widetilde{\mathbf{R}} \simeq \mathbf{u}\sigma\mathbf{v}^T, \qquad \widehat{\mathbf{m}}_3 = \alpha\mathbf{u}, \qquad \hat{\mathbf{a}}^T = \frac{\sigma}{\alpha}\mathbf{v}^T, \tag{20}$$

where σ is the largest singular value of $\widetilde{\mathbf{R}}$, $\mathbf{u}$, and $\mathbf{v}$ are the corresponding left and right singular vectors, and α is a normalizing scalar different from zero. To compute $\mathbf{u}$, σ, and $\mathbf{v}$, we could perform the SVD of $\widetilde{\mathbf{R}}$; because $\widetilde{\mathbf{R}}$ is rank 1, it is much more efficient to use instead less expensive algorithms, in particular, we use the *power* method [21]. This makes the rank 1 decomposition stage much simpler than the decomposition step in the original factorization method of [12].

3.2 Normalization Stage

The normalization stage in the rank 1 factorization algorithm is also simpler than the one in references [11], [12] because the number of unknowns is three (α and $\mathbf{b} = [b_1, b_2]^T$) as opposed to the nine entries of a generic 3×3 normalization matrix. It follows by imposing the constraints that come from the structure of the matrix $\mathbf{M}$. From (2), (6), (7), and (8), the rows $\mathbf{i}_f^T = [i_{xf}, i_{yf}, i_{zf}]$ and $\mathbf{j}_f^T = [j_{xf}, j_{yf}, j_{zf}]$ of each block $\boldsymbol{\Psi}_f$ of $\mathbf{M}$ must be orthonormal,

$$\mathbf{i}_f^T\mathbf{i}_f = \mathbf{j}_f^T\mathbf{j}_f = 1, \qquad \text{and} \qquad \mathbf{i}_f^T\mathbf{j}_f = 0. \tag{21}$$

By replacing the estimate $\widehat{\mathbf{m}}_3$ given by (20) in (16), we get an estimate for $\widehat{\mathbf{M}}_0$. Replacing this estimate $\widehat{\mathbf{M}}_0$ of $\mathbf{M}_0$ as well as the estimate of $\widehat{\mathbf{m}}_3$ given in (20) in (11), we get the following estimate $\widehat{\mathbf{M}}$ of $\mathbf{M}$,

$$\begin{aligned}\widehat{\mathbf{M}} &= \left[\widehat{\mathbf{M}}_0 \quad \widehat{\mathbf{m}}_3\right] = \mathbf{N}\begin{bmatrix}\mathbf{I}_{2\times2} & \mathbf{0}_{2\times1}\\ -\alpha\mathbf{b}^T & \boldsymbol{\alpha}\end{bmatrix}, \text{where}\\ \mathbf{N} &= \left[\mathbf{R}\mathbf{S}_0\left(\mathbf{S}_0^T\mathbf{S}_0\right)^{-1} \quad \mathbf{u}\right].\end{aligned} \tag{22}$$

Denoting by $\widehat{\mathbf{n}}_i^T$ the row i of the matrix $\mathbf{N}$, the constraints (21) on the rows of $\widehat{\mathbf{M}}$ are expressed in terms of $\widehat{\mathbf{n}}_i^T$, α, and $\mathbf{b}$, as

$$\widehat{\mathbf{n}}_i^T\begin{bmatrix}\mathbf{I}_{2\times2} & -\alpha\mathbf{b}\\ -\alpha\mathbf{b}^T & \alpha^2(1+\mathbf{b}^T\mathbf{b})\end{bmatrix}\widehat{\mathbf{n}}_i = 1, \quad 1 \le i \le 2(F-1), \tag{23}$$

$$\widehat{\mathbf{n}}_{2j-1}^T\begin{bmatrix}\mathbf{I}_{2\times2} & -\alpha\mathbf{b}\\ -\alpha\mathbf{b}^T & \alpha^2(1+\mathbf{b}^T\mathbf{b})\end{bmatrix}\widehat{\mathbf{n}}_{2j} = 0, \quad 1 \le j \le F-1. \tag{24}$$

Rewrite $\widehat{\mathbf{n}}_i^T = \left[\mathbf{n}_i^T \; u_i\right]$, where u_i is the ith component of the vector $\mathbf{u}$. Replacing this definition of $\widehat{\mathbf{n}}_i^T$ in (23) and (24), after algebraic manipulations, these equations become

$$\left[-2u_i\mathbf{n}_i^T \;\; u_i^2\right]\begin{bmatrix}\alpha\mathbf{b}\\ \alpha^2\left(1+||\mathbf{b}||^2\right)\end{bmatrix} = 1 - ||\mathbf{n}_i||^2, \;\; 1 \le i \le 2(F-1), \tag{25}$$

$$\begin{aligned}&\left[-\left(u_{2j-1}\mathbf{n}_{2j}^T + u_{2j}\mathbf{n}_{2j-1}^T\right) \;\; u_{2j-1}u_{2j}\right]\begin{bmatrix}\alpha\mathbf{b}\\ \alpha^2\left(1+||\mathbf{b}||^2\right)\end{bmatrix}\\ &\quad = -\mathbf{n}_{2j-1}^T\mathbf{n}_{2j}, \quad 1 \le j \le F-1.\end{aligned} \tag{26}$$

The 3×1 normalization parameter vector $\boldsymbol{\alpha} = \left[\alpha \, \mathbf{b}^T\right]^T$ is now determined from the linear LS solution of the system of $3(F-1)$ equations (25) and(26). We collect these equations in matrix format and get

$$\boldsymbol{\Xi}\boldsymbol{\epsilon}(\boldsymbol{\alpha}) = \boldsymbol{\xi}, \tag{27}$$

where the 3×1 vector $\boldsymbol{\epsilon}(\boldsymbol{\alpha})$, the $3(F-1) \times 1$ vector $\boldsymbol{\xi}$, and the $3(F-1) \times 3$ matrix $\boldsymbol{\Xi}$ are

$$\boldsymbol{\epsilon}(\boldsymbol{\alpha}) = \left[\alpha\mathbf{b}^T \;\; \alpha^2(1+||\mathbf{b}||^2)\right]^T, \tag{28}$$

$$\boldsymbol{\xi} = \left[\begin{array}{c} 1-||\mathbf{n}_1||^2\\ \cdots\\ 1-||\mathbf{n}_{2(F-1)}||^2\\ \hline -\,\mathbf{n}_1^T\mathbf{n}_2\\ \cdots\\ -\mathbf{n}_{2j-1}^T\mathbf{n}_{2j}\\ \cdots\\ -\mathbf{n}_{2(F-1)-1}^T\mathbf{n}_{2(F-1)}\end{array}\right]^T, \tag{29}$$

$$\boldsymbol{\Xi} = \left[\begin{array}{cc} -2u_1\mathbf{n}_1^T & u_1^2\\ \vdots & \vdots\\ -2u_{2(F-1)}\mathbf{n}_{2(F-1)}^T & u_{2(F-1)}^2\\ \hline -\left(u_1\mathbf{n}_2^T + u_2\mathbf{n}_1^T\right) & u_1u_2\\ \vdots & \vdots\\ -\left(u_{2j-1}\mathbf{n}_{2j}^T + u_{2j}\mathbf{n}_{2j-1}^T\right) & u_{2j-1}u_{2j}\\ \vdots & \vdots\\ -\left(u_{2(F-1)-1}\mathbf{n}_{2(F-1)}^T + u_{2(F-1)}\mathbf{n}_{2(F-1)-1}^T\right) & u_{2(F-1)-1}u_{2(F-1)}\end{array}\right]. \tag{30}$$

We compute the least-squares solution of (25) and (26) by minimizing the cost function

$$C\left[\boldsymbol{\epsilon}(\boldsymbol{\alpha})\right] = \left[\boldsymbol{\Xi}\boldsymbol{\epsilon}(\boldsymbol{\alpha}) - \boldsymbol{\xi}\right]^T\left[\boldsymbol{\Xi}\boldsymbol{\epsilon}(\boldsymbol{\alpha}) - \boldsymbol{\xi}\right] \tag{31}$$

wrt $\boldsymbol{\alpha}$. The gradient of the cost function wrt $\boldsymbol{\alpha}$ is

$$\nabla_{\boldsymbol{\alpha}}C\left[\boldsymbol{\epsilon}(\boldsymbol{\alpha})\right] = \nabla_{\boldsymbol{\alpha}}\boldsymbol{\epsilon}(\boldsymbol{\alpha})\nabla_{\boldsymbol{\epsilon}}C\left[\boldsymbol{\epsilon}(\boldsymbol{\alpha})\right], \tag{32}$$

where the gradients of a scalar and of a vector used in (32) are defined as

$$\begin{aligned}\nabla_{\boldsymbol{\alpha}}C\left[\boldsymbol{\epsilon}(\boldsymbol{\alpha})\right] &= \frac{\partial C}{\partial\boldsymbol{\alpha}} = \left[\frac{\partial C}{\partial\alpha} \;\; \frac{\partial C}{\partial b_1} \;\; \frac{\partial C}{\partial b_2}\right]^T,\\ \left[\nabla_{\boldsymbol{\alpha}}\boldsymbol{\epsilon}(\boldsymbol{\alpha})\right]_{ij} &= \left[\frac{\partial\boldsymbol{\epsilon}}{\partial\boldsymbol{\alpha}}\right]_{ij} = \frac{\partial\epsilon_j}{\partial\alpha_i}, \qquad 1 \le i, j \le 3,\end{aligned} \tag{33}$$

with $\alpha_1 = \alpha$, $\alpha_2 = b_1$, and $\alpha_3 = b_2$.

To compute the linear LS solution of the system of (25) and (26), we equate to zero the gradient in (32), obtaining after substituting for $C[\alpha]$ and performing the derivatives

$$\begin{bmatrix}\mathbf{b}^T & 2\alpha(1+||\mathbf{b}||^2)\\ \alpha\mathbf{I}_{2\times2} & 2\alpha^2\mathbf{b}\end{bmatrix}\boldsymbol{\Xi}^T\left[\boldsymbol{\Xi}\boldsymbol{\epsilon}(\boldsymbol{\alpha}) - \boldsymbol{\xi}\right] = \mathbf{0}. \tag{34}$$

Since $\alpha \neq 0$, see (20), the first factor in the left-hand side is full rank. Equating then to zero the second factor in (34), the linear LS solution is, see [21],

$$\boldsymbol{\epsilon}_{\mathrm{LS}} = \left(\boldsymbol{\Xi}^T\boldsymbol{\Xi}\right)^{-1}\boldsymbol{\Xi}^T\boldsymbol{\xi}, \tag{35}$$

assuming that $\boldsymbol{\Xi}$ is rank 3.

The corresponding LS solution for the normalization parameter vector $\boldsymbol{\alpha} = \left[\alpha \; \mathbf{b}^T\right]^T$ is obtained by inverting (28) leading to

$$|\widehat{\alpha}| = \sqrt{\epsilon_{3LS} - \epsilon_{1LS}^2 - \epsilon_{2LS}^2}, \quad \widehat{b}_1 = \epsilon_{1LS}/\widehat{\alpha}, \quad \widehat{b}_2 = \epsilon_{2LS}/\widehat{\alpha}. \quad (36)$$

Clearly, these solutions exist and make sense if $\epsilon_{3LS} > \epsilon_{1LS}^2 - \epsilon_{2LS}^2$. We discuss in Section 4 when this fails.

Remark. Equation (36) determines only the magnitude of $\hat{\alpha}$, not its sign. This ambiguity is inherent to the orthographic projection model. Consider an object with relative depth $\overline{\mathbf{z}} = -\mathbf{z}$ (mirror reflection) and whose motion is such that the third column of the rotation matrix is $\overline{\mathbf{m}}_3 = -\mathbf{m}_3$. Then, from (13) and (14), the measurement matrix $\mathbf{R}$ is the same if $\overline{\mathbf{b}} = -\mathbf{b}$ and $\overline{\mathbf{a}} = -\mathbf{a}$. This causes a change in the sign of α, as seen from (20). See also, from (2), that the trajectories of the feature points for the two objects are the same. This is because all the quantities in the right-hand side of (2) are the same for the two scenarios, except for $\overline{i_{zf}} = -i_{zf}$, $\overline{j_{zf}} = -i_{zf}$, and $\overline{z}_n = -z_n$, which leave their products invariant,i.e., $\overline{i_{zf}}\overline{z}_n = i_{zf}z_n$ and $\overline{j_{zf}}\overline{z}_n = i_{zf}z_n$.

Although, we use the orthographic projection model in deriving the rank 1 factorization method, our derivations are easily extended to more general camera models by proceeding as references [13], [14] do for the original factorization method of [11], [12].

4 Analysis of the Factorization Algorithm

We analyze the accuracy of the rank 1 approximation in the decomposition stage and discuss the situations that may cause its normalization stage to fail.

4.1 Influence of the 3D Structure on the Rank 1 Approximation

The decomposition stage estimates the motion vector $\mathbf{m}_3$ and the shape vector $\mathbf{a}$ from a noisy observation $\widetilde{\mathbf{R}} = \mathbf{m}_3\mathbf{a}^T + \widetilde{\mathcal{N}}$, but only up to a scale parameter α, see (20), i.e., it estimates the 1D linear subspaces of $\mathbf{m}_3$ and $\mathbf{a}$. The accuracy of these estimates improves as the ratio between the singular value $\lambda = \|\mathbf{m}_3\|\|\mathbf{a}\|$ of the noiseless component of $\widetilde{\mathbf{R}}$ and the singular value of its noise component increases, see [25]. This ratio is an equivalent signal to noise ratio (SNR). To increase this SNR, we either increase the "signal" λ or decrease the noise level. The noise corresponds to the errors induced in the 2D motion measurements provided by the tracking algorithm. We assume that we have no control over these and focus on how to maximize $\|\mathbf{m}_3\|$ and $\|\mathbf{a}\|$ by manipulating either the 3D rigid shape or the relative motion between the camera and the object. We assume that the object is stationary, only the camera moves.

Maximizing $\|\mathbf{m}_3\|$. The entries of $\mathbf{m}_3$ are the entries i_{zf} and j_{zf} of the rotation matrices that orient the camera coordinate system relative to the object coordinate system, i.e., the z-component of each orthonormal pair $\{\mathbf{i}_f, \mathbf{j}_f\}$ in $\mathbf{\Psi}_f$, see (2). Since we excluded from $\mathbf{M}$ in (8) the first two rows, which correspond to the reference frame, we have

$$\mathbf{m}_3 = \begin{bmatrix} i_{z2} & j_{z2} & i_{z3} & j_{z3} & \cdots & \cdots & i_{zF} & j_{zF} \end{bmatrix}^T. \quad (37)$$

Each pair of entries $\{i_{zf}, j_{zf}\}$ is constrained by

$$2 \leq f \leq F : \qquad i_{zf}^2 + j_{zf}^2 \leq 1 \quad (38)$$

since each $\left(i_{zf}, j_{zf}, k_{zf}\right)$ is the third column of a rotation matrix $\mathbf{\Theta}_f$, see (1), hence, an orthonormal vector. To maximize $\mathbf{m}_3$, we want (38) to be an equality for $f \geq 2$, which occurs when $k_{zf} = 0$, i.e., when at each frame f the optical axis of the camera is perpendicular to the z-axis. Since the object and camera coordinate systems coincide in the first frame, this condition means that the camera in frame f points in a direction that is perpendicular to the direction it pointed in frame 1. This is intuitively pleasing: The unknown z-coordinates of the feature points are most accurately estimated from their projections onto planes that are parallel to the z-axis, i.e., planes that are orthogonal to the image plane in the reference view. Further, since the analysis did not restrict in any way the 3D shape of the object, we conclude that the optimal position of the camera for all frames after frame 1 does not depend on the particular object shape. This camera placement strategy was arrived at by paying attention to the behavior of the rank 1 factorization algorithm alone. In practice, because conventional feature trackers only work well when the interframe displacements are kept small, one should use a camera trajectory that goes smoothly from the reference view to the orthogonal views. Also, when the goal is to refine the estimates of the 3D structure by using bundle-adjustment to minimize the reprojection error, one should place the cameras in a more evenly distributed configuration.

Maximizing $\|\mathbf{a}\|$. The vector $\mathbf{a} = \mathbf{\Pi}_{\mathcal{S}_0^\perp}\mathbf{z}$, see (13), is the component of the relative depth vector $\mathbf{z}$ in the subspace $\mathcal{S}_0^\perp$ that is orthogonal to the space spanned by the vectors $\mathbf{x}$ and $\mathbf{y}$. The magnitude $\|\mathbf{a}\|$ increases with the magnitude $\|\mathbf{z}\|$ and with the degree of orthogonality between $\mathbf{z}$ and the vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbf{S}_0$. The choice of the first view, the reference view, affects the magnitude $\|\mathbf{a}\|$ because it determines the object coordinate system and, so, affects $\mathbf{S}_0$ and the definition of $\mathbf{z}$, the third column in the shape matrix $\mathbf{S}$.

To determine the "best" reference view, we start with the SVD of the shape matrix $\mathbf{S}$

$$\mathbf{S} = \mathbf{U}_S\mathbf{\Sigma}_S\mathbf{V}_S^T. \quad (39)$$

If we change the reference view, the resulting shape matrix $\mathbf{S}^*$ is

$$\mathbf{S}^* = \mathbf{S}\mathbf{\Theta}, \quad (40)$$

where $\mathbf{\Theta}$ is a rotation matrix. Since $\mathbf{\Theta}$ is a unitary matrix, from (39), the SVD of $\mathbf{S}^*$ is

$$\mathbf{S}^* = \mathbf{U}_S\mathbf{\Sigma}_S\mathbf{V}_S^T\mathbf{\Theta}. \quad (41)$$

The magnitude $\|\mathbf{a}\|$ is maximized when the third column $\mathbf{z}$ of $\mathbf{S}^*$ is orthogonal to the first two and its norm $\|\mathbf{z}\|$ is the largest possible. Since the columns of $\mathbf{U}_S$ in (39) and (41) are orthonormal vectors $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$, the choice of $\mathbf{\Theta}$ must be such that the resulting $\mathbf{z}$ has the form $\mathbf{z} = \sigma_{i\max}\mathbf{u}_{i\max}$, where $\sigma_{i\max}$ is the largest singular value in $\mathbf{\Sigma}_S$ in (39) and (41), and $\mathbf{u}_{i\max}$ the corresponding singular vector. Assuming that the singular values in $\mathbf{\Sigma}_S$ are nondecreasingly ordered,[2] an optimal $\mathbf{\Theta}$ is such that $\mathbf{V}_S^T\mathbf{\Theta} = \mathbf{I}$. In this case, $\sigma_{i\max} = \sigma_3$. Since $\mathbf{V}_S$ is unitary, an optimal solution for the rotation matrix $\mathbf{\Theta}$ is then

$$\mathbf{\Theta} = \mathbf{V}_S. \quad (42)$$

This solution is not unique. The condition $\mathbf{z} = \sigma_{i\max}\mathbf{u}_{i\max} = \sigma_3\mathbf{u}_3$ restricts only two of the three degrees of freedom of $\mathbf{\Theta}$. The third degree of freedom, a rotation between the

[2] Usually, the singular values are decreasingly ordered. For commodity, we order them in the opposite way here.

vectors $\{\mathbf{u}_1, \mathbf{u}_2\}$ and $\{\mathbf{x}, \mathbf{y}\}$ does not affect the magnitude $\|\mathbf{a}\| = \|\mathbf{z}\| = \sigma_3^2$.

With the optimal rotation matrix $\boldsymbol{\Theta}$ in (42), the shape matrix $\mathbf{S}^*$ is simply given by

$$\mathbf{S}^* = \mathbf{U}_S \boldsymbol{\Sigma}_S = [\,\sigma_1 \mathbf{u}_1 \quad \sigma_2 \mathbf{u}_2 \quad \sigma_3 \mathbf{u}_3\,], \tag{43}$$

i.e., the optimal choice for the reference view corresponds to aligning the camera optical axis, the z-axis, with the object axis of smallest inertia (in this case, the inertial moment wrt the z-axis is given by $\sigma_1^2 + \sigma_2^2$).

This analysis provides a further distinction between the rank 1 and the rank 3 algorithms. If the object shape is almost planar (σ_1 close to zero) or almost linear (both σ_1 and σ_2 close to zero), the best rank 3 approximation to $\mathbf{R}$ is sensitive to the noise [13] and the original factorization method of [12] fails. This happens even when the average magnitude of the relative depths, given by σ_3^2, is high. In contrast, if σ_3^2 is large enough, the best rank 1 approximation to the matrix $\widetilde{\mathbf{R}}$ still performs well and captures the shape subspace (as we saw, the quality of the method is proportional to σ_3^2 and independent of σ_1 and σ_2).

4.2 Normalization Failure

If $\boldsymbol{\epsilon}_{\mathrm{LS}} = [\epsilon_{1\mathrm{LS}}, \epsilon_{2\mathrm{LS}}, \epsilon_{3\mathrm{S}}]^T$ determined by the LS solution of the system (27), see Section 3.2, is such that

$$\epsilon_{3\mathrm{LS}} < \epsilon_{1\mathrm{LS}}^2 + \epsilon_{2\mathrm{LS}}^2, \tag{44}$$

we cannot determine the scalar α and the vector $\mathbf{b} = [b_1, b_2]^T$ from (36), and the gradient $\nabla_{\boldsymbol{\alpha}} C[\boldsymbol{\epsilon}(\boldsymbol{\alpha})]$ in (32) and (34) is nonzero over the whole space where $\boldsymbol{\alpha}$ lives. This is a *failure* of the normalization stage as described by the least-squares solution method in Section 3.2, see also [12], where the normalization stage computes a normalization matrix $\mathbf{A}$ by factoring the estimate $\widetilde{\mathbf{B}}$ of an intermediate matrix $\mathbf{B} = \mathbf{A}\mathbf{A}^T$ that may fail to be nonnegative definite.

Since the cost function C in (31) is strictly nondecreasing and grows unbounded with $\|\boldsymbol{\alpha}\|$ (see the definition of $\boldsymbol{\epsilon}$ in (28)), the minimum of C with respect to $\boldsymbol{\alpha}$ occurs at the boundary, i.e., in the limit when α goes to zero. At the boundary, $\alpha = 0$, we have $\boldsymbol{\epsilon} = \mathbf{0}$, see (28), thus, from (31), the minimum value of the cost function C approaches $\boldsymbol{\xi}^T \boldsymbol{\xi}$. This is much larger than the small value for the minimum of C that we expect to obtain at the true value of the normalization parameter vector $\boldsymbol{\alpha}$. This indicates that the two-stage algorithm *decomposition-normalization* does not work and the matrix $\widetilde{\mathbf{R}}$ in (17), (19), and (20) is not well approximated by a rank 1 matrix.

The matrix $\widetilde{\mathbf{R}}$ is not well approximated by a rank 1 matrix in two situations. The first arises when the scene contains dramatic perspective effects. In this case, the rank of the corresponding noiseless $\widetilde{\mathbf{R}}$ is actually greater than 1 and the analysis should take this into account by adopting perspective rather than orthographic projection. The second situation occurs when the 3D shape of the object or its 3D motion cause $\widetilde{\mathbf{R}} = \mathbf{0}$ in a noiseless situation. In this case, the noiseless component $\widehat{\mathbf{R}}$ has rank 0. From (17), this happens in either of the two degenerate cases: The 3D motion is such that the third column of the matrix $\mathbf{M}$ is $\mathbf{m}_3 = \mathbf{0}$; or $\mathbf{a} = \mathbf{0}$, as when the 3D shape is planar, see (13). If $\mathbf{m}_3 = \mathbf{0}$, there is not enough information in the feature trajectories to recover the 3D structure. In spite of this, the images in the sequence can still be aligned by computing $\widehat{\mathbf{M}}_0$ according to (16), for example, by making

$$\widehat{\mathbf{M}}_0 = \mathbf{R}\mathbf{S}_0 \left(\mathbf{S}_0^T \mathbf{S}_0\right)^{-1}. \tag{45}$$

This confirms *analytically* what reference [13] found experimentally, namely, that the normalization matrix $\mathbf{A}$ used in the original method of [11], [12] is singular in the degenerate case where the measurement matrix $\mathbf{R}$ should be approximated by a lower rank matrix, in their case, rank less than 3.

However, if $\mathbf{a} = \mathbf{0}$, although the normalization method in Section 3.2 fails, the shape of the object is still recovered, in this case, theoretically, with no error. In fact, the shape is planar, its plane has been aligned with the reference frame, $\mathbf{z} = \mathbf{0}$, and $\mathbf{x}$ and $\mathbf{y}$ are known from this reference frame.

5 Rank 1 Weighted Factorization

The accuracy of the estimates of the 2D motions, i.e., the 2D displacements of the projections of the feature points, depends on the spatial variability of the brightness intensity pattern in the neighborhood of the feature point. The rank 1 factorization method of Section 3 weighs equally the contribution of each feature, regardless of the accuracy of the estimate of that feature's 2D motion. A more robust estimate of the 3D structure should weigh more heavily the estimate of the trajectory corresponding to a spatially "sharp" feature than the estimate of the trajectory corresponding to a feature with a more smooth texture. In this section, we develop the *rank 1 weighted factorization* method. We show that the weighted factorization approach carries no additional computational cost.

We develop a *Maximum Likelihood* (ML) estimation formulation for the rank 1 weighted factorization problem that accounts for the different noise levels in the 2D motion estimates. We model the errors in the estimates of the 2D motion vectors $\mathbf{u}_{fn}$ as additive zero mean Gaussian independent noises with covariance $\sigma_n^2 \mathbf{I}_{2\times 2}$. For each feature, we collect the noises in the motion estimates across the frames in a vector $\mathcal{N}_n$. We assume that the noise vectors $\{\mathcal{N}_n\}_{1\leq n\leq N}$ are statistically independent Gauss vectors with covariances $\sigma_n^2 \mathbf{I}$, $1 \leq n \leq N$. The variances σ_n^2, $1 \leq n \leq N$, are estimated from the spatial gradient of the image brightness pattern as given in (60) in the appendix.

First, we recenter the coordinate systems taking into account the different error variances σ_n^2. The o.c.s. is centered such that $\sum_n x_n/\sigma_n^2 = \sum_n y_n/\sigma_n^2 = \sum_n z_n/\sigma_n^2 = 0$. Then, the o.c.s is centered at the ML estimate of the translation along the camera plane

$$\forall f: \quad \widehat{\mathbf{t}}_f = \frac{\sum_{n=1}^N \mathbf{u}_{fn}/\sigma_n^2}{\sum_{n=1}^N 1/\sigma_n^2}. \tag{46}$$

Replacing the translation estimates (46) in (2), redefining the vector $\mathbf{u}_{fn}$ by their recentered versions as

$$\forall f, n: \quad \mathbf{u}_{fn} := \mathbf{u}_{fn} - \widehat{\mathbf{t}}_f, \tag{47}$$

and using these $\mathbf{u}_{fn}$ in the matrices $\mathbf{R}$, $\mathbf{M}$, and $\mathbf{S}$ as in (8), we obtain,

$$\mathbf{R} = \mathbf{M}\mathbf{S}^T + \mathcal{N}, \tag{48}$$

where $\mathcal{N} = [\mathcal{N}_1 \cdots \mathcal{N}_N]$ is the $2(F-1) \times N$ matrix collecting the independent Gauss noises in the 2D motion estimates. We whiten the measurements by inversely weighting each measurement by its noise variance. Define the N-dimensional weight vector $\mathbf{w}$ and the $2(F-1) \times N$ dimensional weight matrix $\mathbf{W}$ by

$$\mathbf{w} = \left[\frac{1}{\sigma_1} \cdots \frac{1}{\sigma_N}\right]^T \quad \text{and} \quad \mathbf{W} = \mathbf{1}\mathbf{w}^T, \tag{49}$$

where the all ones $2(F-1)$-dimensional vector $\mathbf{1} = [1, \ldots, 1]^T$. Representing the elementwise product or Hadamard matrix product of two matrices by $\odot$, the whitened measurements are then written as

$$\begin{aligned} \mathbf{R}_W &= \mathbf{R} \odot \mathbf{W} && (50) \\ &= \left(\mathbf{M}\mathbf{S}^T\right) \odot \mathbf{W} + \mathcal{N} \odot \mathbf{W} && (51) \\ &= \mathbf{M}\mathbf{S}_W^T + \mathcal{N} \odot \mathbf{W} && (52) \\ \mathbf{S}_W &= \operatorname{diag}(\mathbf{w})\mathbf{S}, && (53) \end{aligned}$$

where $\operatorname{diag}(\mathbf{w})$ is the $N \times N$ diagonal matrix whose diagonal entries are the entries of the vector $\mathbf{w}$. The step from (51) to (52) and the definition of $\mathbf{S}_W$ in (53) are allowed because the noises are stationary across the frames, i.e., the σ_n^2 are frame independent. This is an important assumption that allows the rank 1 *weighted* factorization to be conceptually similar to the rank 1 factorization algorithm, as we will see next. The ML estimation for the feature dependent noise generalizes the minimization in (9) to

$$\min_{\mathbf{M},\mathbf{S}} \left\|\mathbf{R}_W - \left(\mathbf{M}\mathbf{S}^T\right) \odot \mathbf{W}\right\|_F = \min_{\mathbf{M},\mathbf{S}_W} \left\|\mathbf{R}_W - \mathbf{M}\mathbf{S}_W^T\right\|_F. \tag{54}$$

The factorization on the left side of (54) does not lend itself to a direct solution. However, the factorization on the right side of (54), which is a consequence of (53), is conceptually similar to the one in Section 3: The modified measurement matrix $\mathbf{R}_W$ and the first two columns of the matrix $\mathbf{S}_W$ are known from (50) and (53), and the motion matrix $\mathbf{M}$ is the same matrix involved in the *rank 1 factorization* method of Section 3. The minimization on the right side of (54) is now accomplished by the rank 1 factorization procedure of Section 3 that computes the factor matrices $\widehat{\mathbf{M}}$ and $\widehat{\mathbf{S}_W}$. The final estimate $\widehat{\mathbf{S}}$ of the shape matrix is obtained by inverting (53)

$$\widehat{\mathbf{S}} = \operatorname{diag}(\mathbf{w})^{-1}\widehat{\mathbf{S}_W} = \operatorname{diag}(\sigma_1 \cdots \sigma_N)^{-1}\widehat{\mathbf{S}_W}. \tag{55}$$

Poelman [13] and Morris and Kanade [23] also consider reliability weights when estimating the matrices $\mathbf{M}$ and $\mathbf{S}$ using the original factorization method of [11], [10], [12]. In [13], [23], the weight matrix $\mathbf{W}$ has a general structure where each feature has a weight that is allowed to be time (frame) dependent. The step from (51) to (52) is no longer valid and the nice structure of the weighted minimization given by the right side in (54) is lost. These references propose an iterative solution that, as reported in [13], may fail to converge. This is not the case with our definition of the weight matrix $\mathbf{W}$. In fact, what we show is that the unconstrained bilinear problem given by the right side of (54) has, up to a scale factor, a single, global minimum when $\mathbf{W}$ has all equal rows or all equal columns. This is not true for a generic matrix $\mathbf{W}$, for which it is not possible to write the minimization (54) in the form of a factorization such as done on the right side of (54). For the general case, the existence of local minima makes using iterative numerical techniques nontrivial.

6 Experiments

We describe experiments that illustrate our methods. The experiments in Sections 6.1 to 6.4 use synthetic data. Section 6.1 illustrates the properties of the rank 1 matrix $\widetilde{\mathbf{R}}$. Section 6.2 evaluates the sensitivity of the rank 1 factorization to the observation noise level. Section 6.3 studies how positioning the camera influences the behavior of the rank 1 factorization algorithm. Section 6.5 compares the computational cost of the rank 1 and rank 3 factorization methods. Section 6.4 demonstrates the performance of the weighted factorization method. The experiments in Section 6.6 recover the 3D shape and 3D motion from real life video clips.

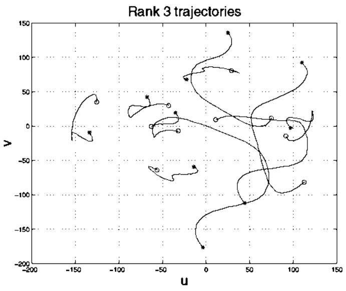


Fig. 1. Feature trajectories on the image plane.

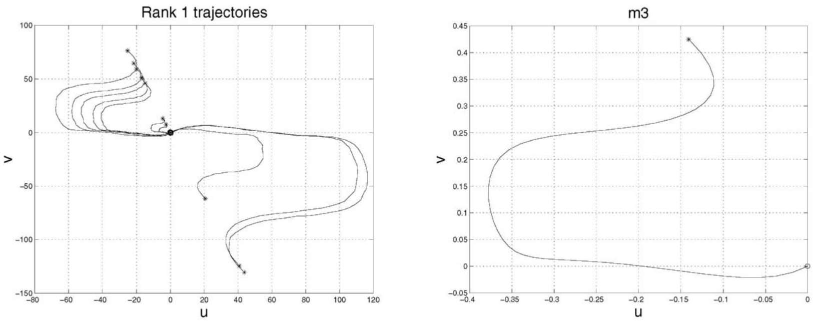
6.1 Rank 1 Factorization: Decoupling the 3D Motion from the 3D Shape

Object: 3D shape and 3D motion. We generated a rigid body by placing arbitrarily a set of 10 feature points inside a cube. The 3D rotational motion is simulated by a smooth time evolution of the Euler angles, which specify the orientation of the object coordinate system relative to the camera coordinate system.

Video sequence. We used perspective projection to project the features onto the image plane and generate a video sequence of 50 frames. The distance of the camera to the centroid of the set of feature points was set to a value high enough (approximately 10 times the maximum relative depth) such that orthographic projection is a good approximation to perspective projection. The lines in Fig. 1 are the trajectories described on the 2D image plane over the 50 frames by the projection of each frame, after adding Gaussian noise. In other words, these lines represent the columns of the matrix $\mathbf{R}$: The trajectory for feature n is the nth column of $\mathbf{R}$—it is the evolution of the image point $\left(\mathbf{R}_{2f-1,n}, \mathbf{R}_{2f,n}\right)$ across the frame index f, see (8). Each line starts with an "o" (initial position of the feature at frame $f = 1$) and ends with a "$\star$" (final position at frame f = 50).

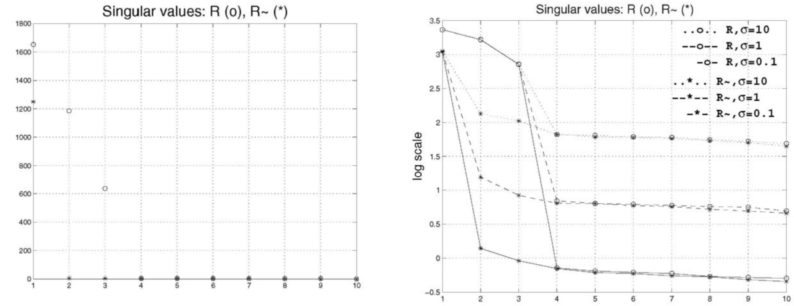
SFM. The challenge in the SFM problem arises because the 3D shape and the 3D motion of the rigid object are observed in a coupled way through the 2D motion on the image plane of the feature points (the trajectories in Fig. 1).

Rank 1 factorization. The matrix $\widetilde{\mathbf{R}}$ is computed by (17) from the data matrix $\mathbf{R}$. The left side plot of Fig. 2 represents the columns of $\widetilde{\mathbf{R}}$ in the same way as Fig. 1 plots $\mathbf{R}$, i.e., it shows the evolution of $\left(\widetilde{\mathbf{R}}_{2f-1,n}, \widetilde{\mathbf{R}}_{2f,n}\right)$ across the frame index f, for each feature n. All trajectories start at $(0,0)$, three trajectories develop to the right while seven expand to the left. Unlike the trajectories in Fig. 1, we see that all trajectories of Fig. 2 are scaled versions of the same shape. This is because the subspace projection of (17) eliminates the dependence of the trajectories on the x and y coordinates of the features. The fixed shape of the trajectories does not depend on the object shape. It is determined uniquely by the 3D motion of the object; it corresponds to the third



(a) (b)

Fig. 2. (a) Trajectories from the columns of matrix $\widetilde{\mathbf{R}}$. (b) The third column $\mathbf{m}_3$ of matrix $\mathbf{M}$.



(a) (b)

Fig. 3. (a) Singular values of matrices $\mathbf{R}$ and $\widetilde{\mathbf{R}}$. (b) Singular values of matrices $\mathbf{R}$ and $\widetilde{\mathbf{R}}$ in logarithmic scale for three levels of the observation noise standard deviation.

column of the matrix $\mathbf{M}$, the vector $\mathbf{m}_3$, see (17). This vector is represented in Fig. 2b. Comparing the trajectories in Fig. 2a with the plot in Fig. 2b, confirms that the trajectories of the features are all congruent: The scaling of each trajectory in matrix $\widetilde{\mathbf{R}}$ depends on the relative depth z of the corresponding feature point, see (19). When this scaling is negative, the corresponding trajectory is a reflection of the vector $\mathbf{m}_3$ with respect to the origin.

This experiment illustrates that the subspace projection of (17) decouples the 3D motion from the 3D shape. In contrast to the trajectories of $\mathbf{R}$ in Fig. 1, which are spaghetti like, for $\widetilde{\mathbf{R}}$, see Fig. 2, the 3D motion influences only the 2D shape of the trajectories, and the 3D shape influences only the magnitude of the trajectories.

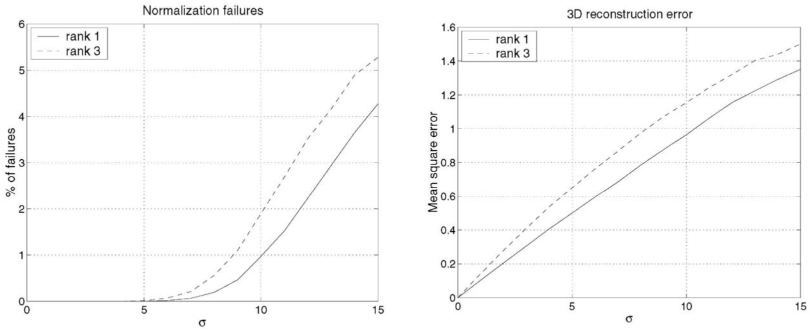
Singular values. Fig. 3a represents the 10 larger singular values of the matrices $\mathbf{R}$, marked with "o" and of $\widetilde{\mathbf{R}}$, marked with "*". Matrix $\mathbf{R}$ exhibits three significant singular values, while matrix $\widetilde{\mathbf{R}}$ is well described by only its largest singular value. To illustrate the influence of the observation noise, we used a logarithmic scale to represent in Fig. 3b the singular values of $\mathbf{R}$ and $\widetilde{\mathbf{R}}$ for three levels of noise. This plot shows that, as expected, the higher the noise level is, the more significant the noise singular values of $\mathbf{R}$ and $\widetilde{\mathbf{R}}$ are. Still, it is clear, even at these higher noise levels, that $\widetilde{\mathbf{R}}$ has one more significant singular value, while $\mathbf{R}$ has three.

6.2 Rank 1 versus Rank 3: Sensitivity to Noise

To evaluate the robustness of the rank 1 factorization algorithm to the noise affecting the 2D motion estimates, we run Monte-Carlo tests that compare the performances of the rank 1 factorization method and the original rank 3 factorization of [12] for several noise levels. We performed two sets of tests. The first set is more appropriate when the features are selected from the real 2D video. With these experiments, the 3D x_n- and y_n-coordinates of the feature n are the $u_{f=1,n}$ and $v_{f=1,n}$ coordinates of its projection in the reference frame, e.g., $f = 1$. In the remaining frames, $f \geq 1$, the 2D motion estimates of the projection of the feature are tracked with errors modeled as additive Gaussian noise with standard deviation σ. The second set of Monte-Carlo tests is appropriate when the features are preselected from the 3D object, for example, with visually distinctive marks placed at the points of interest. With these experiments, the projections $u_{f=1,n}$ and $v_{f=1,n}$ of the feature n in the reference frame are also synthesized with additive noise.

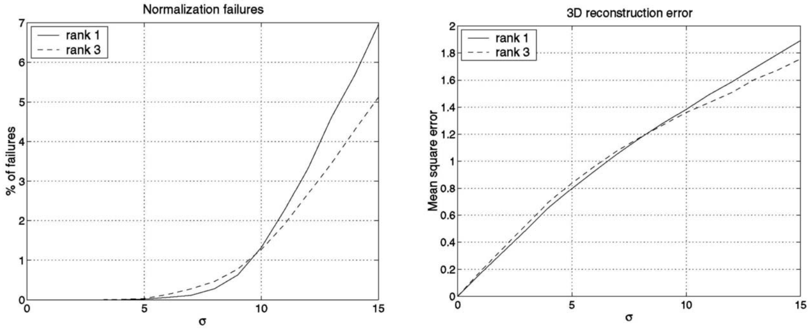
In our Monte-Carlo tests, we used randomly generated 3D structures with a number N of feature points ranging from 5 to 100 and a number F of frames ranging from 5 to 100. The results of the comparison of the rank 1 and rank 3 factorization methods were similar for all the 3D structures (3D shapes and 3D motions) tested. We now describe representative results obtained with a random 3D shape described by $N = 10$ feature points and a random 3D motion of the camera over $F = 10$ frames.

Selecting $\mathbf{x}$ and $\mathbf{y}$. Fig. 4a represents the percentage of normalization failures as a function of the noise standard deviation. As expected, the higher the noise level is, the more likely it is for the normalization stages to fail. We see that the percentage of failures of the rank 1 factorization (solid line) is smaller than the one of the original rank 3 factorization method (dotted line). Fig. 4b represents the average Euclidean error of the estimate of the 3D shape, computed over the runs when the normalization succeeded. We see that the rank 1 factorization leads to smaller errors



(a) (b)

Fig. 4. Percentage of failures and 3D reconstruction error as functions of the noise level.



(a) (b)

Fig. 5. The same plots as in Fig. 4 but now obtained when the feature projections onto the reference frame are noisy versions of their *x*- and *y*-coordinates.

(solid line) than the rank 3 factorization (dotted line). Figs. 4a and 4b agree with the fact that the rank 1 factorization approach exploits constraints in the SFM problem that are not taken into account by the original rank 3 factorization.

Estimating $\mathbf{x}$, $\mathbf{y}$. We run Monte-Carlo tests feeding the rank 1 factorization algorithm with noisy versions of the x- and y-coordinates of the features in the reference frame. The results are in Figs. 5a and 5b. We see that the performance of the methods is almost indistinguishable when the noise standard deviation $\sigma < 10$ and also that the original rank 3 factorization (dotted line) performs better than the rank 1 factorization (solid line) for levels of noise with $\sigma > 10$. In all these experiments, the image coordinates are in the interval $[-200, 200]$. Assuming the tracking errors in practical situations of interest are smaller than 10 pixels for images of 400×400 pixels, we conclude, in summary, that the behaviors of the rank 1 and rank 3 factorization methods are similar when processing real videos.

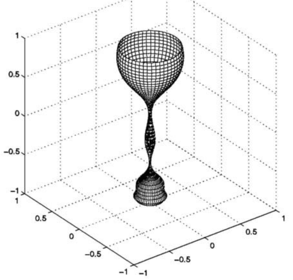


Fig. 6. Three-dimensional rigid shape.

6.3 Camera Positioning: Choice of Views

We describe two experiments that illustrate the predictions of Section 4 with respect to the camera trajectory and the reference frame viewing angle. We use a set of 50 features sampled from the surface of the synthetic object shown in Fig. 6.

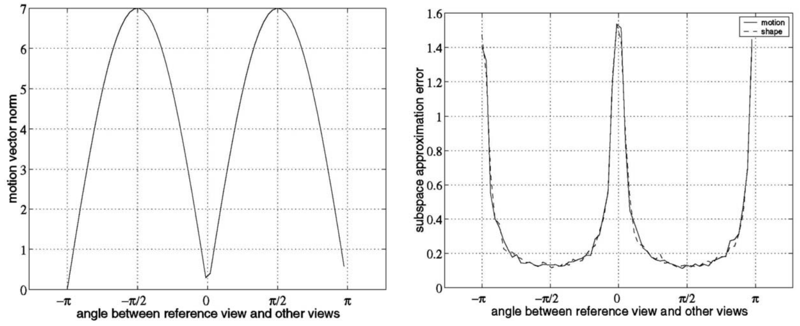
Camera trajectory. We fix the reference frame to be $f = 1$ (corresponding to an elevation[3] angle of zero) and created several trajectories by moving the camera around the object for $f = 2, \dots, F$. The elevation angle θ of the views $f = 2, \dots, F$ is kept constant in the course of the camera motion. The plot in Fig. 7a, computed from the ground truth, represents the norm $\|\mathbf{m}_3\|$ of the motion vector $\mathbf{m}_3$ as a function of the angle θ. The norm $\|\mathbf{m}_3\|$ is maximum when the views are orthogonal to the reference view—note that the maxima in Fig. 7a occur at $\theta = \pm\pi/2$. The minima, $\|\mathbf{m}_3\| = 0$, occur when the views are parallel to the reference frame—$\theta = 0$ or $\pm\pi$.

As expected from the analysis of Section 4, the norm $\|\mathbf{m}_3\|$ is maximum when the views $f = 2, \dots, F$ are orthogonal to the reference view, i.e., when $\theta = \pi/2 + k\pi$, and $\|\mathbf{m}_3\| = 0$ when they are parallel, i.e., when $\theta = k\pi$.

Using these trajectories, we synthesized noisy feature projections and applied the rank 1 factorization. Fig. 7b plots the estimation error for the shape and motion subspaces as functions of the angle θ. These errors are the angles[4] between

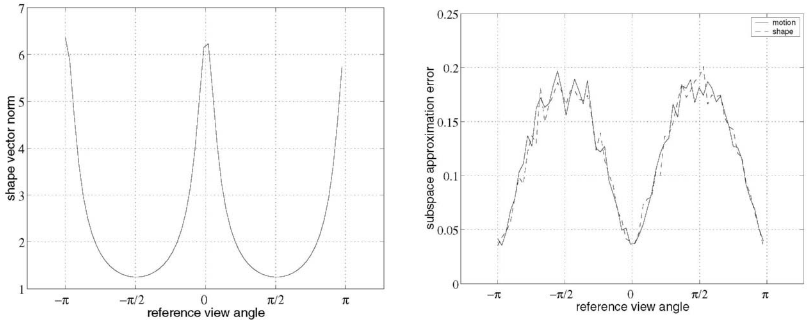
[3] The 3D orientation of the camera is commonly represented in terms of the three so-called Euler angles: elevation, compass, and twist, see [26]. The elevation angle is the angle between the optical axis and the horizontal plane.

[4] The angle between the 1D subspaces spanned by vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ is $\arccos\{\mathbf{s}_1^T\mathbf{s}_2/(\|\mathbf{s}_1\|\|\mathbf{s}_2\|)\}$.



(a) (b)

Fig. 7. $\|\mathbf{m}_3\|$ and subspace estimation errors as functions of the angle θ of the camera pose.



(a) (b)

Fig. 8. $\|\mathbf{a}\|$ and subspace estimation errors as functions of the reference view angle ϕ.

the ground truth subspaces and the subspaces recovered by the rank 1 factorization. We see that, as predicted by the analysis in Section 4, the errors are smaller when the views are close to being orthogonal ($\theta = \pm\pi/2$) to the reference view and larger when they are close to being parallel ($\theta = k\pi$) to the reference view.

Reference view. We then fixed the camera trajectory and used several reference frame viewing angles. Again, to make the analysis simpler, we chose the optical axis of the reference frame to be always in a vertical plane, i.e., in a plane containing the major axis of the object in Fig. 6 and varied its elevation angle. We denote by ϕ the angle between the optical axis and the major axis of the object, thus $\phi = 0$ corresponds to the top view. The plots in Fig. 8 represent, respectively, the norm $\|\mathbf{a}\|$ of the shape vector $\mathbf{a}$, and the subspace estimation errors, as functions of the angle ϕ. Again, these plots confirm the predictions of Section 4—$\|\mathbf{a}\|$ becomes larger, and the errors become smaller, as the reference view is "more aligned" with the axis of smallest inertia, i.e., when $\phi = k\pi$, and $\|\mathbf{a}\|$ becomes smaller, and the errors larger, as the reference view is "more orthogonal" to the axis of smallest inertia. Note that, in practice, in the limiting case of $\phi = k\pi$, the feature points may not be visible due to self-occlusion.

6.4 Rank 1 Weighted Factorization: 2D-Motion Errors with Different Variances

We now compare the rank 1 *weighted* factorization and the rank 1 factorization.

Object: 3D Shape and 3D Motions. The rigid shape is described by 21 features with coordinates x and y randomly located inside a square. The depth z is generated with a sinusoidal shape, see Fig. 10. The 3D translation is smooth and shown by the thick lines in Figs. 11a and 11b. The 3D rotational motion is synthesized by the time evolutions of the 6 entries of the 3D rotation matrix involved in the orthogonal projection, see (2). These are shown by the thick lines in Figs. 12a and 12b.

Video sequence. We used perspective projection to project the features onto the image plane and generated a sequence of 19 frames. The lens focal length parameter was set to a high value so that orthographic projection is a reasonable approximation. Fig. 9 shows the feature trajectories on the image plane, after adding Gaussian noise. We group the features in two sets: the "low noisy" set of 10 features and the "high noisy" set of 11 features, with noise variances of $\sigma_1^2 = 1$ and $\sigma_2^2 = 5$, respectively. As expected, the (estimates) of the trajectories corresponding to the features of the first subset are smooth, while the (estimates) of the trajectories corresponding to the features of the second subset have a much more jagged appearance, see Fig. 9.

Results. We applied both the nonweighted rank 1 factorization of Section 3 and the rank 1 *weighted* factorization of

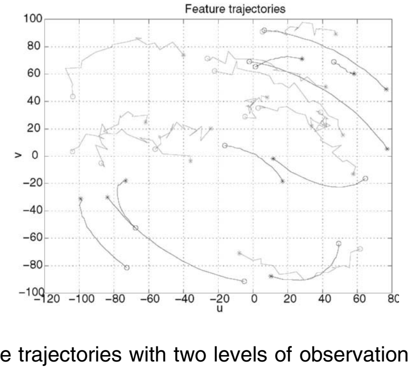


Fig. 9. Feature trajectories with two levels of observation noise.

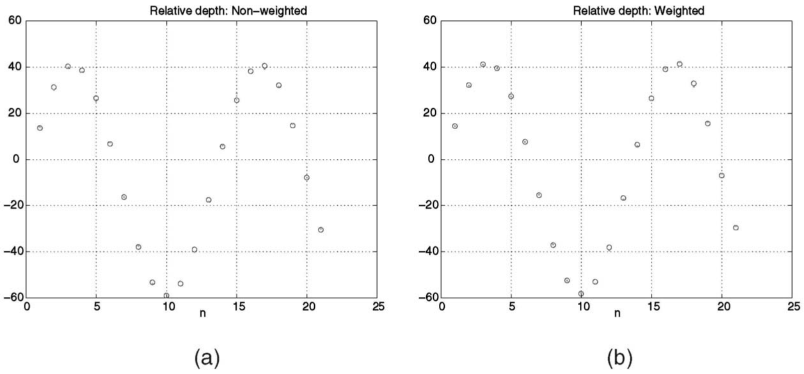


Fig. 10. Estimates (points ".") and true values (circles "o") of the relative depth. (a) Rank 1 nonweighted factorization method. (b) Rank 1 weighted factorization method.

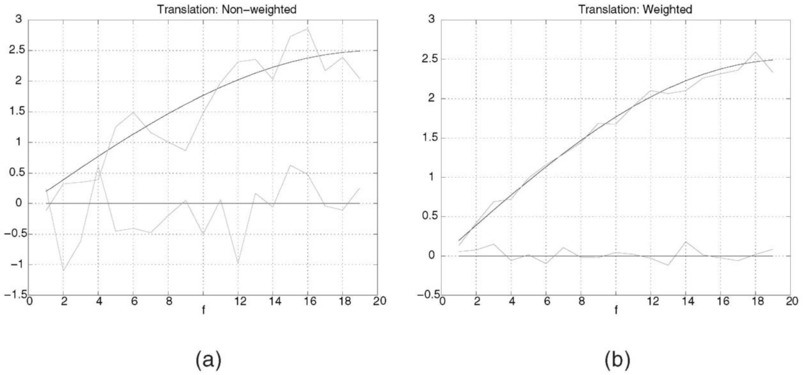


Fig. 11. Estimates (thin lines) and true value (thick lines) of the translation along the camera plane. (a) Rank 1 nonweighted factorization . (b) Rank 1 weighted factorization method.

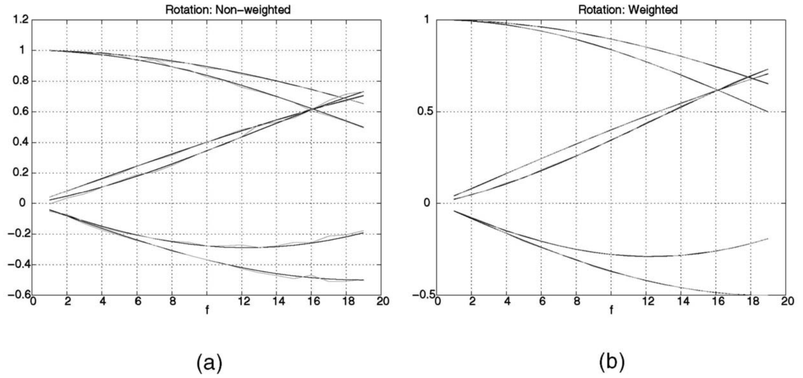
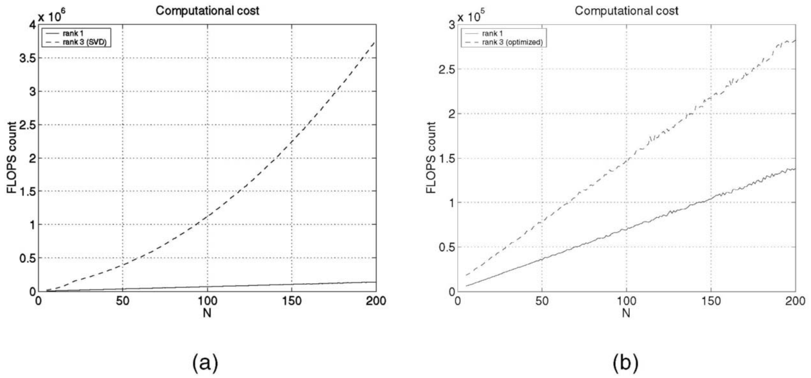


Fig. 12. Estimates (thin lines) and true value (thick lines) of the six entries of the 3D rotation matrix involved in the projection. (a) Nonweighted factorization. (b) Rank 1 weighted factorization.

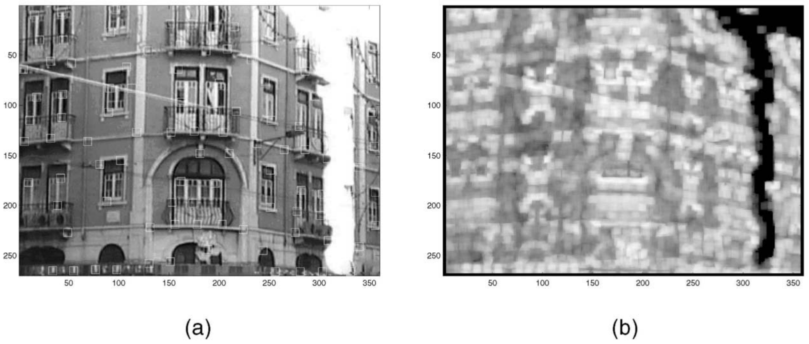
Section 5 to the feature trajectories of Fig. 9. The estimates of the 3D shape and 3D motion are shown in Figs. 10a and 10b, 11a and 11b, and 12a and 12b, superimposed to the true values. Figs. 10a, 11a, and 12a represent the nonweighted estimates and Figs. 10b, 11b, and 12b represent the rank 1 weighted factorization results. We see from Figs. 11 and 12 that the 3D motion estimates obtained through the rank 1 weighted factorization method is more accurate than the ones obtained without taking into account the different noise levels. This is particularly true for the translation estimates as can be seen by comparing Figs. 11a and 11b. The difference is still noticeable for the estimates of the entries of the 3D rotation matrix, see Fig. 12; these differences have a much larger impact originating much larger differences in the feature projections because the projections are the multiplication of the 3D rotation matrix by the 3D position of the features, see (2).

The 3D shape estimates represented by the relative depths are shown in Figs. 10a and 10b. They show that the weighted estimate of the 3D shape is slightly more accurate than the nonweighted estimate—note that the depth estimates (points ".") are usually at the center of their true values (circles "o") in Fig. 10b, while the nonweighted estimates (points ".") are usually off-centered with respect to their true values (circles "o") in Fig. 10a. A second point of note is that the accuracy of the weighted estimate of the relative depth of a given feature reflects the level of the observation noise for the trajectory of the projection of that feature—with reference to Fig. 10b, the weighted estimates of the relative depths of the subset of features observed with



(a) (b)

Fig. 13. Computational cost of the rank 1 factorization (solid line) and rank 3 factorization (dashed line). The rank 3 factorization was computed using the (a) SVD and (b) a power method.



(a) (b)

Fig. 14. The building video sequence. (a) First frame. (b) "Trackability" of the feature candidates.

higher level of noise (the last 11 features in Fig. 10b) are less accurate than the estimates of the relative depths of the subset of the first 10 features.

6.5 Computational Cost

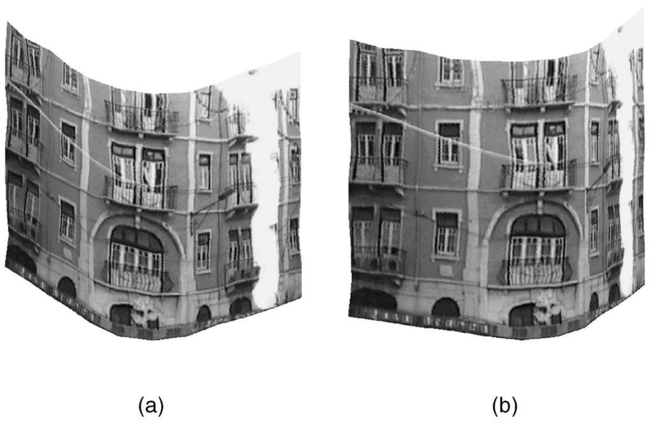
We now compare the computational costs of our rank 1 factorization algorithm and the original rank 3 factorization method of [12]. We counted the MatLab floating point operations (FLOPS) for both algorithms with $F = 50$ frames and a number N of feature points ranging from 5 to 200. Fig. 13a represents the FLOPS count as a function of N. From this plot, we see that the computational cost of the rank 1 factorization (solid line) is much smaller than that of the rank 3 factorization. This was expected since the cost of the rank 3 factorization algorithm is dominated by the computation of the SVD while the rank 1 factorization algorithm uses a power method. We have also implemented an optimized version of the rank 3 factorization that uses a power method to compute the best rank 3 approximation. The FLOPS count for this optimized algorithm, represented in Fig. 13b (dashed line), is also always larger than that of the rank 1 factorization (solid line) by a factor of approximately 2.

6.6 Real-Life Video Clips

Building sequence. We processed a sequence of 30 frames showing a building with two planar walls meeting along a smooth (round) edge. The first frame of the video sequence is shown in Fig. 14a. The figure also shows, marked as white squares, the 100 features selected for processing. Fig. 14b represents the "trackability" of the feature candidates. This image is obtained from the spatial evolution of the condition number of the matrix needed to estimate the 2D motions, see [17] for the details. The brighter a point is, the more reliable its tracking is. We choose the feature points by selecting the peaks of this image. We assign to each feature the confidence weight computed as detailed in the appendix, see (60).

We tracked the feature points by matching the intensity pattern of each feature along the sequence. Using the rank 1 weighted factorization, we recovered the 3D motion and the relative depth of the feature points from the set of feature trajectories, as described in Sections 3 and 5. Fig. 15 shows two perspective views of the reconstructed 3D shape with the texture mapped on it. The angle between the walls is clearly seen and the round edge is also well-reconstructed.

CMU's hotel sequence. Fig. 16 shows frames 1 and 50 from the CMU's hotel video sequence. On the left image,



(a) (b)

Fig. 15. Three-dimensional shape and texture reconstructed from the building video sequence.

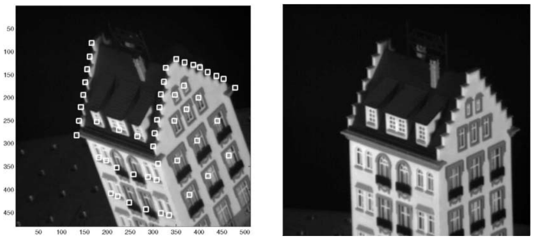


Fig. 16. Two frames from the hotel sequence.

taken as the reference frame, we marked with white squares the 50 feature points used by the *rank 1 factorization* algorithm. In this video sequence, the camera undergoes a slow rotation around the object.

To illustrate the relevance of the view selection, we run our algorithm with two distinct sets of 12 frames. In the first experiment, we used consecutive frames, thus all the views had very similar orientation. The reconstructed 3D shape is shown in Fig. 17a. In the second experiment, we selected one frame for each eight video frames, thus the orientation of the last view was very distinct from the first one (although not orthogonal). The better quality of the 3D reconstruction obtained with this sparse view selection, shown in Fig17b, confirms our theoretical analysis.

7 Conclusion

We developed a new *rank 1 weighted factorization* approach for the recovery of 3D structure from 2D motion. Our method reduces the problem to the factorization of a rank 1 matrix that we compute using the power method, avoiding any expensive singular value decomposition of large matrices. The rank 1 factorization method is computationally considerably simpler than the original rank 3 factorization algorithm, see Figs. 13a and 13b. We extended the rank 1 factorization method to the rank 1 *weighted* factorization method to account for different variances in the errors of the features 2D motion estimates without significant additional computational cost. We present explicit analytical expressions for the weights needed by the method—the inverses of the error variances of the 2D motion estimates. We provide extensive analysis of the algorithm. The paper studies the impact of the actual 3D shape and 3D motion on the performance of the rank 1 factorization algorithm. We show that the optimal choice for the reference view is to align the camera optical axis with the object axis of smallest inertia and that in subsequent views the camera optical axis should be orthogonal to its position in the reference view. Experimental results with synthetic data and real videos illustrate our approach.

Appendix

Tracking feature points estimates their 2D motion in the image plane. This has been widely addressed by the computer vision community. Usually motion is estimated by minimizing the sum of the square difference of the intensities over a spatial region, e.g., [27]. This minimization is accomplished by using a Gauss-Newton method [28]. We follow this approach, see [17] for the details. In this appendix, we present an expression for the variance of the estimation error in terms of the spatial gradient of the brightness pattern. This expression is used to compute the weights involved in the rank 1 weighted factorization method described in Section 5. Consider a generic image motion model parameterized by the $p \times 1$ parameter vector $\mathbf{p}$. The Gauss-Newton updates for the estimate $\widehat{\mathbf{p}}$ of $\mathbf{p}$ is $\widehat{\mathbf{p}} = \mathbf{p}_0 + \widehat{\delta_p}$, where $\widehat{\delta_p}$ is the solution of the linear system

$$\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)\,\widehat{\delta_p} = \gamma_{\mathcal{R}}(\mathbf{p}_0), \tag{56}$$

$$\begin{aligned} \boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0) &= \iint_{\mathcal{R}} \nabla_{\mathbf{p}}\mathbf{d}^T(\mathbf{p}_0)\mathbf{i}_{xy}\mathbf{i}_{xy}^T\nabla_{\mathbf{p}}\mathbf{d}(\mathbf{p}_0)\,dx\,dy, \\ \gamma_{\mathcal{R}}(\mathbf{p}_0) &= -\iint_{\mathcal{R}} i_t(\mathbf{p}_0)\nabla_{\mathbf{p}}\mathbf{d}^T(\mathbf{p}_0)\mathbf{i}_{xy}\,dx\,dy. \end{aligned} \tag{57}$$

In (57), $i_t(\mathbf{p}_0; x, y)$ is the image temporal derivative, the $\mathbf{i}_{xy}(x, y) = \left[i_x(x, y), i_y(x, y)\right]^T$ contains the spatial derivatives, and the $p \times 2$ matrix $\nabla_{\mathbf{p}}\mathbf{d}^T(\mathbf{p}_0; x, y)$ is, see [27], [28],

$$\nabla_{\mathbf{p}}\mathbf{d}^T(\mathbf{p}_0; x, y) = \left[\nabla_{\mathbf{p}}d_x(\mathbf{p_0}; \mathbf{x}, \mathbf{y}) \quad \nabla_{\mathbf{p}}\mathbf{d_y}(\mathbf{p_0}; \mathbf{x}, \mathbf{y})\right], \tag{58}$$

where $[d_x, d_y]$ is the image displacement. The good convergence of the Gauss-Newton iterates depends on the value of the condition number of $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$, thus it is usual to select features based on this value, see [29], [17]. In [17], we show that, to first-order approximation, the estimate $\widehat{\mathbf{p}}$ is unbiased and the error covariance matrix $\boldsymbol{\Sigma}_p$ is given by

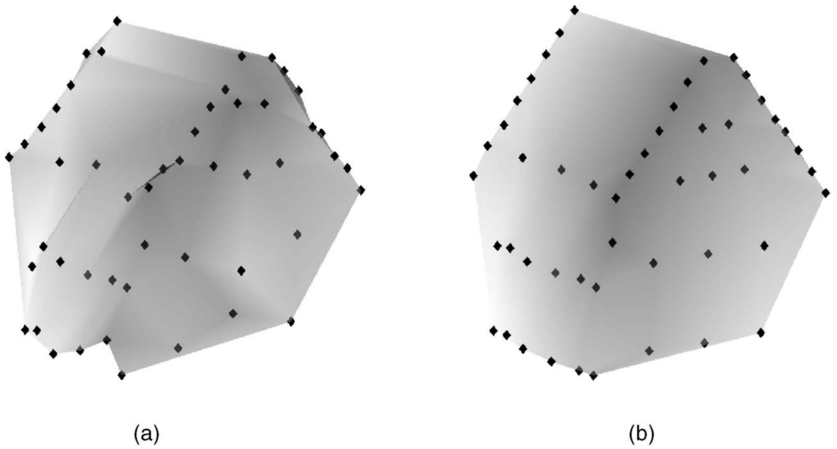


Fig. 17. Estimated 3D shape of the hotel. (a) Bad selection of views. (b) Good selection of views.

$$\boldsymbol{\Sigma}_p = \sigma_t^2 \boldsymbol{\Gamma}_{\mathcal{R}}^{-1}(\mathbf{p}_a). \tag{59}$$

Expression (59) provides an inexpensive way to compute the reliability of the motion estimates. The matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_a)$ is, in general, unknown because it depends on the actual value $\mathbf{p}_a$ of the unknown $\mathbf{p}$. However, an available approximation to $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_a)$ is the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ used in the iterative estimation algorithm. We note that, when the motion model is linear in the motion parameters, as it is the case with the majority of motion models used in practice, $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p})$ becomes independent of the vector $\mathbf{p}$ because the derivatives of the displacement $\mathbf{d}(\mathbf{p})$ involved in (57) do not depend on the motion parameters. In this case, the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ does not change along the iterative estimation algorithm. The matrix $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ depends uniquely on the image region $\mathcal{R}$ and $\boldsymbol{\Gamma}_{\mathcal{R}}(\mathbf{p}_0)$ will be denoted simply by $\boldsymbol{\Gamma}_{\mathcal{R}}$. Since the noise variance σ_t^2 is considered to be constant, we measure the error covariance for different regions by comparing the corresponding matrices $\boldsymbol{\Gamma}_{\mathcal{R}}^{-1}$. For example, the mean square Euclidean distance between the true vector $\mathbf{p}_a$ and the estimated vector $\widehat{\mathbf{p}}$, denoted by σ_p^2, is proportional to the trace of the matrix $\boldsymbol{\Gamma}_{\mathcal{R}}^{-1}$. The covariance matrix of the estimation error for the translational motion model is given by (59) after replacing $\boldsymbol{\Gamma}_{\mathcal{R}}$, see [17]. The mean square error σ_p^2 of the displacement estimate (in the sense of the Euclidean distance) is given by the trace of the covariance matrix $\boldsymbol{\Sigma}_p$ and expressed in terms of image gradients as

$$\sigma_p^2 = \sigma_t^2 \frac{\int_{\mathcal{R}} i_y^2\,dx\,dy + \int_{\mathcal{R}} i_x^2\,dx\,dy}{\int_{\mathcal{R}} i_x^2\,dx\,dy\ \int_{\mathcal{R}} i_y^2\,dx\,dy - \left(\int_{\mathcal{R}} i_x i_y\,dx\,dy\right)^2}. \tag{60}$$

In Section 5, we use the estimate of the mean square error σ_p^2 given by (60) to weigh motion estimates corresponding to different features.

Acknowledgments

This work was partially supported by the US Office of Naval Research grant no. N000 14-00-1-0593. The work of P.M.Q. Aguiar was partially supported by INVOTAN. The authors would like to thank the reviewers and the associate editor for their comments that greatly improved the manuscript.

References

[1] O. Faugeras, *Three-Dimensional Computer Vision.* Cambridge, Mass.: MIT Press, 1993.

[2] B. Horn and B. Chunck, "Determining Optical Flow," *Artificial Intelligence,* vol. 17, pp. 185-203, 1981.

[3] T. Brodsky, C. Fermuller, and Y. Aloimonos, "Shape from Video," *IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 146-151, 1999.

[4] B.K.P. Horn and E.J. Weldon Jr., "Direct Methods for Recovering Motion," *Kluwer Int'l J. Computer Vision,* vol. 2, no. 1, pp. 51-76, June 1988.

[5] G.P. Stein and A. Shashua, "Model-Based Brightness Constraints: On Direct Estimation of Structure and Motion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 9, pp. 992-1015, Sept. 2000.

[6] J. Weng, N. Ahuja, and T.S. Huang, "Optimal Motion and Structure Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 864-884, June 1993.

[7] R. Szeliski and S. Kang, "Recovering 3D Shape and Motion from Image Streams Using Nonlinear Least squares," *J. Visual Comm. and Image Representation,* vol. 5, no. 1, 1994.

[8] T. Broida and R. Chellappa, "Estimating the Kinematics and Structure of a Rigid Object from a Sequence of Monocular Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 6, June 1991.

[9] A. Azarbayejani and A.P. Pentland, "Recursive Estimation of Motion, Structure, and Focal Length," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 6, 1995.

[10] C. Tomasi and T. Kanade, "Shape and Motion without Depth," *Proc. IEEE Int'l Conf. Computer Vision,* June 1990.

[11] C. Tomasi, "Shape and Motion from Image Streams: A Factorization Method," PhD thesis, Carnegie Mellon Univ., Pittsburgh, Pa., 1991.

[12] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: A Factorization Method," *Int'l J. Computer Vision,* vol. 9, no. 2, 1992.

[13] C.J. Poelman, "A Paraperspective Factorization Method for Shape and Motion Recovery," PhD thesis, Carnegie Mellon Univ., Pittsburgh, Pa., 1995.

[14] C.J. Poelman and T. Kanade, "A Paraperspective Factorization Method for Shape and Motion Recovery," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 3, Mar. 1997.

[15] P. Sturm and B. Triggs, "A Factorization Based Algorithm for Multi-Image Projective Structure and Motion," *Proc. European Conf. Computer Vision,* vol. 2, pp. 709-720, Apr. 1996.

[16] L. Quan and T. Kanade, "A Factorization Method for Affine Structure from Line Correspondences," *IEEE Conf. Computer Vision and Pattern Recognition,* June 1996.

[17] P.M.Q. Aguiar, "Rigid Structure from Video," PhD thesis, Instituto Superior Técnico, Lisboa, Portugal, Jan. 2000, available at http://www.isr.ist.utl.pt/aguiar.

[18] P.M.Q. Aguiar and J.M F. Moura, "Three-Dimensional Modeling from Two-Dimensional Video," *IEEE Trans. Image Processing,* vol. 10, no. 10, pp. 1541-1551, Oct. 2001.

[19] T. Morita and T. Kanade, "A Sequential Factorization Method for Recovering Shape and Motion from Image Streams," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 8, pp. 858-867, Aug. 1997.

[20] J.P. Costeira and T. Kanade, "A Factorization Method for Independently Moving Objects," *In'l J. Computer Vision,* vol. 29, no. 3, pp. 159-179, 1998.

[21] G.H. Golub and C.F. Van Loan, "Matrix Computations," Johns Hopkins Series in Math. Sciences, third ed. The Johns Hopkins Univ. Press, 1989.

[22] M. Irani and P. Anandan, "Factorization with Uncertainty," *Proc. European Conf. Computer Vision,* vol. 1, pp. 539-553, June 2000.

[23] D.D. Morris and T. Kanade, "A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertainty Models," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 696-702, 1998.

[24] P.M.Q. Aguiar and J.M.F. Moura, "Factorization as a Rank 1 Problem," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 178-184, June 1999.

[25] L.L. Scharf, "Statistical Signal Processing—Detection, Estimation, and Time Series Analysis," *Electrical and Computer Engineering: Digital Signal Processing,* Addison-Wesley, 1991.

[26] N. Ayache, *Artificial Vision for Mobile Robots.* Cambridge, Mass.: The MIT Press, 1991.

[27] G.D. Hager and P.N. Belhumeur, "Efficient Region Tracking with Parametric Models of Geometry and Illumination," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 10, Oct. 1998.

[28] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation," *Proc. European Conf. Computer Vision,* pp. 237-252, May 1992.

[29] J. Shi and C. Tomasi, "Good Features to Track," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 593-600, 1994.

Pedro M.Q. Aguiar received the PhD 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 engineering at IST. His main research interests are in image analysis and computer vision. He is a member of the IEEE.

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 MSc, EE, and the DSc degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and 1975, respectively. He has been a professor of electrical and computer engineering at Carnegie Mellon University since 1986. In 1999-2000, he was a visiting professor of electrical engineering at MIT. Prior to this, he was on the faculty of IST (1975-1984). He was Genrad associate professor of electrical engineering and computer science (visiting) at MIT (1984-1986), and a visiting research scholar at the University of Southern California (Department of Aerospace Engineering, summers 1978-1981). His research interests include statistical signal processing and telecommunications, image processing, and video representations. He has published more than 230 technical contributions, he is the coeditor of two books, he holds five patents on image and video processing, and digital communications with the US Patent Office, and has given numerous invited seminars at US and European universities and laboratories. Dr. Moura served as Vice-President for Publications for the IEEE Signal Processing Society (SPS) and was a member of the Board of Governors of the same society (2000-2002). He was also Vice-President for Publications for the IEEE Sensors Council (2000-2002). He is on the editorial board of the *IEEE Proceedings*. He chairs the IEEE TAB Transactions Committee (2002-2003). He was the Editor-in-Chief for the *IEEE Transactions in Signal Processing* (1975-1999). He has been a member of several technical committees of the SPS. He was on the IEEE Press Board (1991-95). 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, Sigma Xi, AMS, AAAS, IMS, and SIAM.