Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications*

Théodore Papadopoulo and Manolis I.A. Lourakis

INRIA Sophia Antipolis
2004 Route des Lucioles, BP 93
06902 SOPHIA-ANTIPOLIS Cedex
FRANCE

Abstract. The Singular Value Decomposition (*SVD*) of a matrix is a linear algebra tool that has been successfully applied to a wide variety of domains. The present paper is concerned with the problem of estimating the Jacobian of the SVD components of a matrix with respect to the matrix itself. An exact analytic technique is developed that facilitates the estimation of the Jacobian using calculations based on simple linear algebra. Knowledge of the Jacobian of the SVD is very useful in certain applications involving multivariate regression or the computation of the uncertainty related to estimates obtained through the SVD. The usefulness and generality of the proposed technique is demonstrated by applying it to the estimation of the uncertainty for three different vision problems, namely self-calibration, epipole computation and rigid motion estimation.

1 Introduction and Motivation

The SVD is a general linear algebra technique that is of utmost importance for several computations involving matrices. For example, some of the uses of SVD include its application to solving ordinary and generalized least squares problems, computing the pseudo-inverse of a matrix, assessing the sensitivity of linear systems, determining the numerical rank of a matrix, carrying out multivariate analysis and performing operations such as rotation, intersection, and distance determination on linear subspaces [10]. Owing to its power and flexibility, the SVD has been successfully applied to a wide variety of domains, from which a few sample applications are briefly described next. Zhang et al [42], for example, employ the SVD to develop a fast image correlation scheme. The problem of establishing correspondences is also addressed by Jones and Malik [14], who compare feature vectors defined by the responses of several spatial filters and use the SVD to determine the degree to which the chosen filters are independent of each other. Structure from motion is another application area that has greately benefited from the SVD. Longuet-Higgins [19] and later

* This work has been partially supported by the EEC under the TMR VIRGO research network.

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 554–570, 2000.
© Springer-Verlag Berlin Heidelberg 2000

Hartley [12] extract the translational and rotational components of rigid 3D motion using the SVD of the essential matrix. Tsai et al [36] also use the SVD to recover the rigid motion of a 3D planar patch. Kanade and co-workers [35, 28,27] assume special image formation models and use the SVD to factorize image displacements to structure and motion components. Using an SVD based method, Sturm and Triggs [34] recover projective structure and motion from uncalibrated images and thus extend the work of [35,28] to the case of perspective projection. The SVD of the fundamental matrix yields a simplified form of the Kruppa equations, on which self-calibration is based [11,20,21]. Additionally, the SVD is used to deal with important image processing problems such as noise estimation [15], image coding [40] and image watermarking [18]. Several parametric fitting problems involving linear least squares estimation, can also be effectively resolved with the aid of the SVD [2,37,16]. Finally, the latter has also proven useful in signal processing applications [31,26] and pattern recognition techniques such as neural networks computing [7,39] and principal components analysis [4].

This paper deals with the problem of computing the Jacobian of the SVD components of a matrix with respect to the elements of this matrix. Knowledge of this Jacobian is important as it is a key ingredient in tasks such as non-linear optimization and error propagation:

- Several optimization methods require that the Jacobian of the criterion that is to be optimized is known. This is especially true in the case of complicated criteria. When these criteria involve the SVD, the method proposed in this paper is invaluable for providing analytical estimates of their Jacobians. As will be further explained latter, numerical computation of such Jacobians using finite differences is not as straightforward as it might seem at a first glance.
- Computation of the covariance matrix corresponding to some estimated quantity requires knowledge of the Jacobians of all functions involved in the estimation of the quantity in question. Considering that the SVD is quite common in many estimation problems in vision, the method proposed in this work can be used in these cases for computing the covariance matrices associated with the estimated objects.

Paradoxically, the numerical analysis litterature provides little help on this topic. Indeed, a lot of studies have been made on the sensitivity of singular values and singular vectors to perturbations in the original matrix [33,10,6,38, 3], but these globally consider the question of perturbing the input matrix and derive bounds for the singular elements but do not deal with perturbations due to individual elements.

Thus, the method proposed here fills in an important gap, since, to the best of our knowledge, no similar method for SVD differentiation appears in the literature. The rest of this paper is organized as follows. Section 2 gives an analytical derivation for the computation of the Jacobian of the SVD and discusses practical issues related to its implementation in degenerate cases. Section 3 il-

lustrates the use of the proposed technique with three examples of covariance matrix estimation. The paper concludes with a brief discussion in section 4.

2 The Proposed Method

2.1 Notation and Background

In the rest of the paper, bold letters will be used for denoting vector and matrices. The transpose of matrix $\mathbf{M}$ is denoted by $\mathbf{M}^T$ and m_{ij} refers to the (i, j) element of $\mathbf{M}$. The i-th non-zero element of a diagonal matrix $\mathbf{D}$ is referred to by d_i, while $\mathbf{M}_i$ designates the i-th column of matrix $\mathbf{M}$.

A basic theorem of linear algebra states that any real $M \times N$ matrix $\mathbf{A}$ with $M \geq N$ can be written as the product of an $M \times N$ column orthogonal matrix $\mathbf{U}$, an $N \times N$ diagonal matrix $\mathbf{D}$ with non-negative diagonal elements (known as the *singular values*), and the transpose of an $N \times N$ orthogonal matrix $\mathbf{V}$ [10]. In other words,

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \sum_{i=1}^{N} d_i \mathbf{U}_i \mathbf{V}_i^T. \tag{1}$$

The singular values are the square roots of the eigenvalues of the matrix $\mathbf{A}\mathbf{A}^T$ (or $\mathbf{A}^T\mathbf{A}$ since these matrices share the same non-zero eigenvalues) while the columns of $\mathbf{U}$ and $\mathbf{V}$ (the *singular vectors*) correspond to the eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ respectively [17]. As defined in Eq.(1), the SVD is not unique since

- it is invariant to arbitrary permutations of the singular values and their corresponding left and right singular vectors. Sorting the singular values (usually by decreasing magnitude order) solves this problem unless there exist equal singular values.
- simultaneous changes in the signs of the vectors $\mathbf{U}_i$ and $\mathbf{V}_i$ do not have any impact on the leftmost part of Eq.(1). In practice, this has no impact on most numerical computations involving the SVD.

2.2 Computing the Jacobian of the SVD

Employing the definitions of section 2.1, we are interested in computing $\frac{\partial \mathbf{U}}{\partial a_{ij}}$, $\frac{\partial \mathbf{V}}{\partial a_{ij}}$ and $\frac{\partial \mathbf{D}}{\partial a_{ij}}$ for every element a_{ij} of the $M \times N$ matrix $\mathbf{A}$.
Taking the derivative of Eq. (1) with respect to a_{ij} yields the following equation

$$\frac{\partial \mathbf{A}}{\partial a_{ij}} = \frac{\partial \mathbf{U}}{\partial a_{ij}}\mathbf{D}\mathbf{V}^T + \mathbf{U}\frac{\partial \mathbf{D}}{\partial a_{ij}}\mathbf{V}^T + \mathbf{U}\mathbf{D}\frac{\partial \mathbf{V}}{\partial a_{ij}}^T \tag{2}$$

Clearly, $\forall\ (k, l) \neq (i, j)$, $\frac{\partial a_{kl}}{\partial a_{ij}} = 0$, while $\frac{\partial a_{ij}}{\partial a_{ij}} = 1$. Since $\mathbf{U}$ is an orthogonal matrix, we have the following:

$$\mathbf{U}^T\mathbf{U} = \mathbf{I} \Longrightarrow \frac{\partial \mathbf{U}}{\partial a_{ij}}^T \mathbf{U} + \mathbf{U}^T\frac{\partial \mathbf{U}}{\partial a_{ij}} = \Omega_{\mathbf{U}}^{ij\,T} + \Omega_{\mathbf{U}}^{ij} = \mathbf{0}, \tag{3}$$

where $\Omega_{\mathbf{U}}^{ij}$ is given by

$$\Omega_{\mathbf{U}}^{ij} = \mathbf{U}^T \frac{\partial \mathbf{U}}{\partial a_{ij}} \ . \tag{4}$$

From Eq. (3) it is clear that $\Omega_{\mathbf{U}}^{ij}$ is an antisymmetric matrix. Similarly, an antisymmetric matrix $\Omega_{\mathbf{V}}^{ij}$ can be defined for $\mathbf{V}$ as

$$\Omega_{\mathbf{V}}^{ij} = \frac{\partial \mathbf{V}}{\partial a_{ij}}^T \mathbf{V} \ . \tag{5}$$

Notice that $\Omega_{\mathbf{U}}^{ij}$ and $\Omega_{\mathbf{V}}^{ij}$ are specific to each differentiation $\frac{\partial}{\partial a_{ij}}$.

By multiplying Eq. (2) by $\mathbf{U}^T$ and $\mathbf{V}$ from the left and right respectively, and using Eqs. (4) and (5), the following relation is obtained:

$$\mathbf{U}^T \frac{\partial \mathbf{A}}{\partial a_{ij}} \mathbf{V} = \Omega_{\mathbf{U}}^{ij} \mathbf{D} + \frac{\partial \mathbf{D}}{\partial a_{ij}} + \mathbf{D} \Omega_{\mathbf{V}}^{ij}. \tag{6}$$

Since $\Omega_{\mathbf{U}}^{ij}$ and $\Omega_{\mathbf{V}}^{ij}$ are antisymmetric matrices, all their diagonal elements are equal to zero. Recalling that $\mathbf{D}$ is a diagonal matrix, it is easy to see that the diagonal elements of $\Omega_{\mathbf{U}}^{ij}\mathbf{D}$ and $\mathbf{D}\Omega_{\mathbf{V}}^{ij}$ are also zero. Thus, Eq. (6) yields the derivatives of the singular values as

$$\frac{\partial d_k}{\partial a_{ij}} = u_{ik} \ v_{jk} \ . \tag{7}$$

Taking into account the antisymmetry property, the elements of the matrices $\Omega_{\mathbf{U}}^{ij}$ and $\Omega_{\mathbf{V}}^{ij}$ can be computed by solving a set of 2×2 linear systems, which are derived from the off-diagonal elements of the matrices in Eq. (6):

$$\begin{cases} d_l \ \Omega_{\mathbf{U}\,kl}^{ij} + d_k \ \Omega_{\mathbf{V}\,kl}^{ij} = \ \ u_{ik} \ v_{jl} \\ d_k \ \Omega_{\mathbf{U}\,kl}^{ij} + d_l \ \Omega_{\mathbf{V}\,kl}^{ij} = - \ u_{il} \ v_{jk} \ \ , \end{cases} \tag{8}$$

where the index ranges are $k = 1 \ldots N$ and $l = i + 1 \ldots N$. Note that, since the d_k are positive numbers, this system has a unique solution provided that $d_k \neq d_l$. Assuming for the moment that $\forall \ (k, l), \ d_k \neq d_l$, the $\frac{N(N-1)}{2}$ parameters defining the non-zero elements of $\Omega_{\mathbf{U}}^{ij}$ and $\Omega_{\mathbf{V}}^{ij}$ can be easily recovered by solving the $\frac{N(N-1)}{2}$ corresponding 2×2 linear systems.

Once $\Omega_{\mathbf{U}}^{ij}$ and $\Omega_{\mathbf{V}}^{ij}$ have been computed, $\frac{\partial \mathbf{U}}{\partial a_{ij}}$ and $\frac{\partial \mathbf{V}}{\partial a_{ij}}$ follow as:

$$\frac{\partial \mathbf{U}}{\partial a_{ij}} = \mathbf{U} \Omega_{\mathbf{U}}^{ij} \ \ , \ \ \frac{\partial \mathbf{V}}{\partial a_{ij}} = -\mathbf{V} \Omega_{\mathbf{V}}^{ij} \, . \tag{9}$$

In summary, the desired derivatives are supplied by Eqs. (7) and (9).

2.3 Implementation and Practical Issues

In this section, a few implementation issues related to a practical application of the proposed method are considered.

Degenerate SVDs. Until now, the case where the SVD yields at least two identical singular values has been set aside. However, such cases occur often in practice, for example when dealing with the essential matrix (see section 3.3 ahead) or when fitting a line in 3D. Therefore, let us now assume that $d_k = d_l$ for some k and l. It is easy to see that these two singular values contribute to Eq. (1) with the term $d_l \left(\mathbf{U}_k \mathbf{V}_k^T + \mathbf{U}_l \mathbf{V}_l^T\right)$. The same contribution to $\mathbf{A}$ can be obtained by using any other orthonormal bases of the subspaces spanned by $(\mathbf{U}_k, \mathbf{U}_l)$ and $(\mathbf{V}_k, \mathbf{V}_l)$ respectively. Therefore, letting

$$\begin{aligned}
\mathbf{U}'_k &= \quad cos\alpha\, \mathbf{U}_k + sin\alpha\, \mathbf{U}_l \\
\mathbf{U}'_l &= -sin\alpha\, \mathbf{U}_k + cos\alpha\, \mathbf{U}_l \\
\mathbf{V}'_k &= \quad cos\alpha\, \mathbf{V}_k + sin\alpha\, \mathbf{V}_l \\
\mathbf{V}'_l &= -sin\alpha\, \mathbf{V}_k + cos\alpha\, \mathbf{V}_l \ ,
\end{aligned}$$

for any real number α, we have $\mathbf{U}_k \mathbf{V}_k^T + \mathbf{U}_l \mathbf{V}_l^T = \mathbf{U}'_k {\mathbf{V}'_k}^T + \mathbf{U}'_l {\mathbf{V}'_l}^T$. This implies that in this case, there exists a one dimensional family of SVDs. Consequently, the 2×2 system of Eqs. (8) must be solved in a least squares fashion in order to get only the component of the Jacobian that is "orthogonal" to this family. Of course, when more than two singular values are equal, all the 2×2 corresponding systems have to be solved simultaneously. The correct algorithm for all cases is thus:

- Group together all the 2×2 systems corresponding to equal singular values.
- Solve these systems using least squares.

This will give the exact Jacobian in non-degenerate cases and the "minimum norm" Jacobian when one or more singular values are equal.

Computational complexity. Assuming that the matrix $\mathbf{A}$ is $N \times N$ and non-degenerate for simplicity, it is easy to compute the complexity of the procedure for computing the Jacobian: For each pair (i, j), $i = 1 \ldots N$, $j = i + 1 \ldots N$, a total of $\frac{N(N-1)}{2}$ 2×2 linear systems have to be solved. In essence, the complexity of the method is $O(N^4)$ once the initial SVD has been carried out.

Computing the Jacobian using finite differences. The proposed method has been compared with a finite difference approximation of the Jacobian and same results have been obtained in the non-degenerate case (degenerate cases are more difficult to compare due to the non-uniqueness of the SVD). Although the ease of implementation makes the finite difference approximation more appealing for computing the Jacobian, the following points should also be taken into account:

- The finite difference method is more costly in terms of computational complexity. Considering again the case of an $N \times N$ non-degenerate matrix as in the previous paragraph, it is simple to see that such an approach requires

N^2 SVD computations to be performed (i.e. one for each perturbation of each element of the matrix). Since the complexity of the SVD operation is $O(N^3)$, the overall complexity of the approach is $O(N^5)$ which is an order of magnitude higher compared to that corresponding to the proposed method.
– Actually, the implementation of a finite difference approximation to a Jacobian is not as simple as it might appear. This is because even state of the art algorithms for SVD computation (eg LAPACK's *dgesvd* family of routines [1]) are "unstable" with respect to small perturbations of the input. By unstable, we mean that the signs associated with the columns $\mathbf{U}_i$ and $\mathbf{V}_i$ can change arbitrarily even with the slightest perturbation. In general, this is not important but it has strong effects in our case since the original and perturbed SVD do not return the same objects. Consequently, care has to be taken to compensate for this effect when the Jacobian is computed through finite differences.

3 Applications

In this section, the usefulness and generality of the proposed SVD differentiation method are demonstrated by applying it to three important vision problems. Before proceeding to the description of each of these problems, we briefly state a theorem related to error propagation that is essential for the developments in the subsections that follow. More specifically, let $\mathbf{x_0} \in \mathcal{R}^N$ be a measurement vector, from which a vector $\mathbf{y_0} \in \mathcal{R}^M$ is computed through a function $\mathbf{f}$, i.e. $\mathbf{y_0} = \mathbf{f}(\mathbf{x_0})$. Here, we are interested in determining the uncertainty of $\mathbf{y_0}$, given the uncertainty of $\mathbf{x_0}$. Let $\mathbf{x} \in \mathcal{R}^N$ be a random vector with mean $\mathbf{x_0}$ and covariance $\Lambda_\mathbf{x} = E[(\mathbf{x} - \mathbf{x_0})(\mathbf{x} - \mathbf{x_0})^T]$. The vector $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is also random and its covariance $\Lambda_\mathbf{y}$ up to first order is equal to

$$\Lambda_\mathbf{y} = \frac{\partial \mathbf{f}(\mathbf{x_0})}{\partial \mathbf{x_0}} \Lambda_\mathbf{x} \frac{\partial \mathbf{f}(\mathbf{x_0})}{\partial \mathbf{x_0}}^T, \tag{10}$$

where $\frac{\partial \mathbf{f}(\mathbf{x_0})}{\partial \mathbf{x_0}}$ is the derivative of $\mathbf{f}$ at $\mathbf{x_0}$. For more details and proof, the reader is referred to [8]. In the following, Eq. (10) will be used for computing the uncertainty pertaining to various entities that are estimated from images. Since image measurements are always corrupted by noise, the estimation of the uncertainty related to these entities is essential for effectively and correctly employing the latter in subsequent computations.

3.1 Self-Calibration Using the SVD of the Fundamental Matrix

The first application that we deal with is that of self-calibration, that is the estimation of the camera intrinsic parameters without relying upon the existence of a calibration object. Instead, self-calibration employs constraints known as the *Kruppa* equations, which are derived by tracking image features through an image sequence. More details regarding self-calibration can be found in [25,41,23,

20]. Here, we restrict our attention to a self-calibration method that is based on a simplification of the Kruppa equations derived from the SVD of the fundamental matrix. In the following paragraph, a brief description of the method is given; more details can be found in [20,22].

Let $\mathbf{S_F}$ be the vector formed by the parameters of the SVD of the fundamental matrix $\mathbf{F}$. The Kruppa equations in this case reduce to three linearly dependent constraints, two of which are linearly independent. Let $\pi_i(\mathbf{S_F}, \mathbf{K}), \quad i = 1 \dots 3$ denote those three equations as functions of the fundamental matrix $\mathbf{F}$ and the matrix $\mathbf{K} = \mathbf{A}\mathbf{A}^T$, where $\mathbf{A}$ is the 3×3 intrinsic calibration parameters matrix having the following well-known form [8]:

$$\mathbf{A} = \begin{bmatrix} \alpha_u & -\alpha_u \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

The parameters α_u and α_v correspond to the focal distances in pixels along the axes of the image, θ is the angle between the two image axes and (u_0, v_0) are the coordinates of the image principal point. In practice, θ is very close to $\frac{\pi}{2}$ for real cameras. The matrix $\mathbf{K}$ is parameterized with the unknown intrinsic parameters from Eq.(11) and is computed from the solution of a non-linear least squares problem, namely

$$\mathbf{K} = argmin_{\tilde{\mathbf{K}}} \sum_{i=1}^{N} \frac{\pi_1^2(\mathbf{S_{F}}_i, \tilde{\mathbf{K}})}{\sigma_{\pi_1}^2(\mathbf{S_{F}}_i, \tilde{\mathbf{K}})} + \frac{\pi_2^2(\mathbf{S_{F}}_i, \tilde{\mathbf{K}})}{\sigma_{\pi_2}^2(\mathbf{S_{F}}_i, \tilde{\mathbf{K}})} + \frac{\pi_3^2(\mathbf{S_{F}}_i, \tilde{\mathbf{K}})}{\sigma_{\pi_3}^2(\mathbf{S_{F}}_i, \tilde{\mathbf{K}})} \tag{12}$$

In the above equation, N is the number of the available fundamental matrices and $\sigma_{\pi_i}^2(\mathbf{S_{F}}_i, \mathbf{K})$ are the variances of constraints $\pi_i(\mathbf{S_F}, \mathbf{K}), \quad i = 1 \dots 3$, respectively, used to automatically weight the constraints according to their uncertainty. It is to the estimation of these variances that the proposed differentiation method is applied. More specifically, applying Eq.(10) to the case of the simplified Kruppa equations, it is straightforward to show that the variance of the latter is approximated by

$$\sigma_{\pi_i}^2(\mathbf{S_F},\ \mathbf{K}) = \frac{\partial \pi_i(\mathbf{S_F}, \mathbf{K})}{\partial \mathbf{S_F}} \frac{\partial \mathbf{S_F}}{\partial \mathbf{F}} \Lambda_{\mathbf{F}} \frac{\partial \mathbf{S_F}}{\partial \mathbf{F}}^T \frac{\partial \pi_i(\mathbf{S_F}, \mathbf{K})}{\partial \mathbf{S_F}}^T$$

In the above equation, $\frac{\partial \pi_i(\mathbf{S_F}, \mathbf{K})}{\partial \mathbf{S_F}}$ is the derivative of $\pi_i(\mathbf{S_F}, \mathbf{K})$ at $\mathbf{S_F}$, $\frac{\partial \mathbf{S_F}}{\partial \mathbf{F}}$ is the Jacobian of $\mathbf{S_F}$ at $\mathbf{F}$ and $\Lambda_{\mathbf{F}}$ is the covariance of the fundamental matrix, supplied as a by-product of the procedure for estimating $\mathbf{F}$ [5]. The derivative $\frac{\partial \pi_i(\mathbf{S_F}, \mathbf{K})}{\partial \mathbf{S_F}}$ is computed directly from the analytic expression for $\pi_i(\mathbf{S_F}, \mathbf{K})$, while $\frac{\partial \mathbf{S_F}}{\partial \mathbf{F}}$ is estimated using the proposed method for SVD differentiation. To quantitatively assess the improvement on the accuracy of the recovered intrinsic parameters that is gained by employing the covariances, a set of simulations has been conducted. More specifically, three rigid displacements of a virtual camera were simulated and a set of randomly chosen 3D points were projected on the simulated retinas. Following this, the resulting retinal points were contaminated by zero mean additive Gaussian noise. The noise standard deviation was increased from 0 to 4

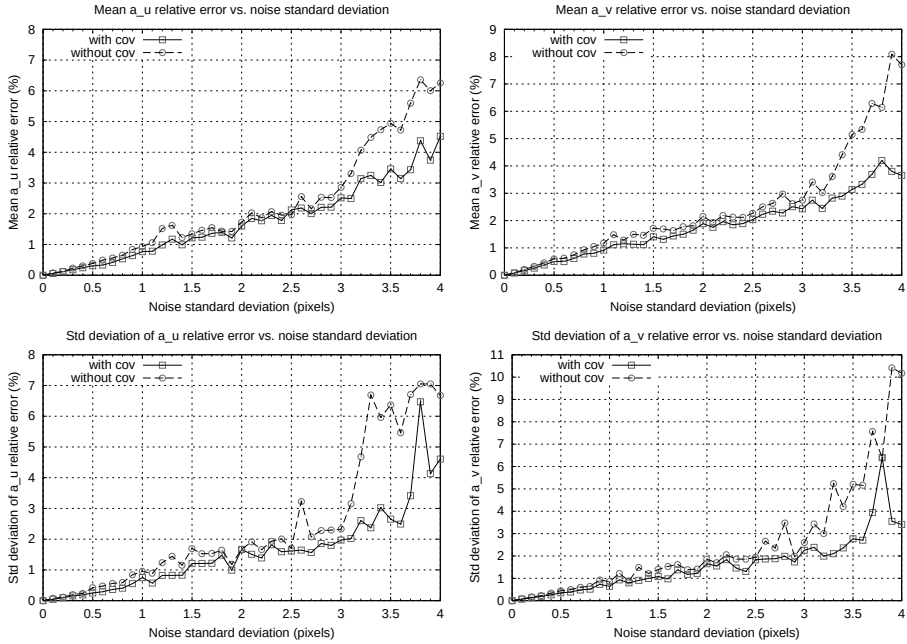

Fig. 1. The error in the recovered focal lengths in the presence of noise, with and without employing the covariance. Mean values are shown in the top row, standard deviations in the bottom.

pixels in steps of 0.1. A non-linear method [43] was then employed to estimate from the noisy retinal points the fundamental matrices corresponding to the simulated displacements. The estimates of the fundamental matrices serve as the input to self-calibration. To ensure that the recovered intrinsic calibration parameters are independent of the exact location of the 3D points used to form 2D correspondences, 100 experiments were run for each noise level, each time using a different random set of 3D points. More details regarding the simulation can be found in [20]. Figures 3.1 and 3.1 illustrate the mean and standard deviation of the relative error for the intrinsic parameters versus the standard deviation of the noise added to image points, with and without employing the covariances. When the covariances are not employed, the weights $\sigma^2_{\pi_i}(\mathbf{S_F}, \tilde{\mathbf{K}})$ in Eq.(12) are all assumed to be equal to one. Throughout all experiments, zero skew has been assumed, i.e. $\theta = \pi/2$ and $\tilde{\mathbf{K}}$ in Eq. (12) was parameterized using 4 unknowns. As is evident from the plots, especially those referring to the standard deviation of the relative error, the inclusion of covariances yields more accurate and more stable estimates of the intrinsic calibration parameters. Additional experimental results can be found in [20]. At this point, it is also worth mentioning that in the case of self-calibration, the derivatives $\frac{\partial \mathbf{S_F}}{\partial \mathbf{F}}$ were also computed analytically by using MAPLE to compute closed-form expressions for the SVD components

with respect to the elements of **F**. Using the computed expressions for the SVD, the derivative of the latter with respect to **F** was then computed analytically. As expected, the arithmetic values of the derivatives obtained in this manner were identical to those computed by the differentiation method proposed here.

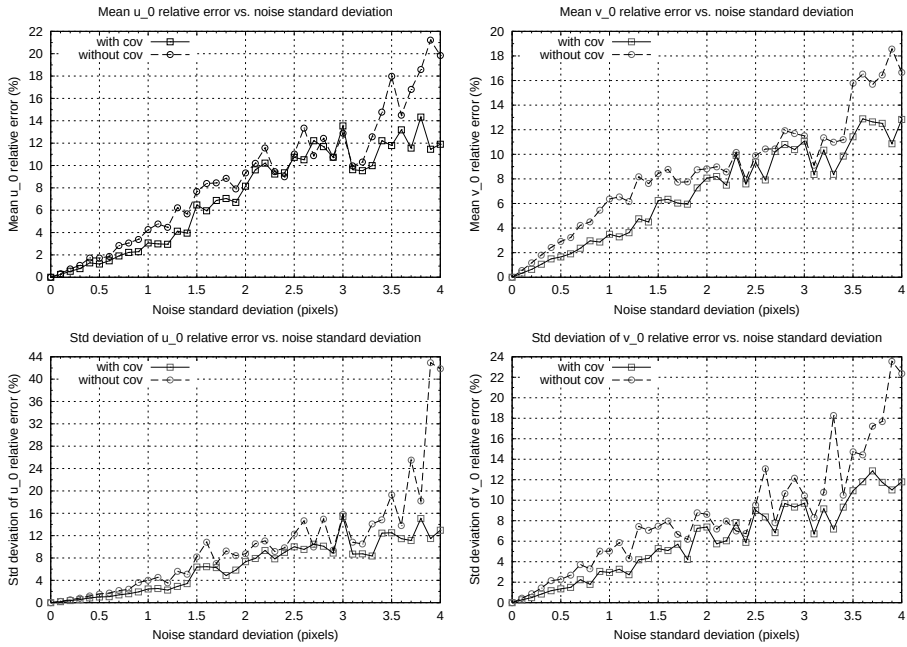

Fig. 2. The error in the recovered principal points in the presence of noise, with and without employing the covariance. Mean values are shown in the top row, standard deviations in the bottom.

3.2 Estimation of the Epipoles' Uncertainty

The epipoles of an image pair are the two image points defined by the projection of each camera's optical center on the retinal plane of the other. The epipoles encode information related to the relative position of the two cameras and have been employed in applications such as stereo rectification [30], self-calibration [25,41,23], projective invariants estimation [13,32] and point features matching [9]. Although it is generally known that the epipoles are hard to estimate accurately[1] [24], the uncertainty pertaining to their estimates is rarely quantified.

[1] This is particularly true when the epipoles lie outside the images.

Here, a simple method is presented that permits the estimation of the epipoles' covariance matrices based on the covariance of the underlying fundamental matrix.

Let $\mathbf{e}$ and $\mathbf{e}'$ denote the epipoles in the first and second images of a stereo pair respectively. The epipoles can be directly estimated from the SVD of the fundamental matrix $\mathbf{F}$ as follows. Assuming that $\mathbf{F}$ is decomposed as $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ and recalling that $\mathbf{F}^T\mathbf{e}' = \mathbf{F}\mathbf{e} = \mathbf{0}$, it is easy to see that $\mathbf{e}$ corresponds to the third column of $\mathbf{V}$, while $\mathbf{e}'$ is given by the third column of $\mathbf{U}$. The epipoles are thus given by two very simple functions $\mathbf{f_e}$ and $\mathbf{f_{e'}}$ of the vector $\mathbf{S_F}$ defined by the SVD of $\mathbf{F}$. More precisely, $\mathbf{f_e}(\mathbf{S_F}) = \mathbf{V_3}$ and $\mathbf{f_{e'}}(\mathbf{S_F}) = \mathbf{U_3}$, where $\mathbf{V_3}$ and $\mathbf{U_3}$ are the third columns of matrices $\mathbf{V}$ and $\mathbf{U}$ respectively. A direct application of Eq. (10) can be used for propagating the uncertainty corresponding to $\mathbf{F}$ to the estimates of the epipoles. Since this derivation is analogous to that in section 3.1, exact details are omitted and a study of the quality of the estimated covariances is presented instead.

First, a synthetic set of corresponding pairs of 2D image points was generated. The simulated images were 640×480 pixels and the epipoles $\mathbf{e}$ and $\mathbf{e}'$ were within them, namely at pixel coordinates (458.123, 384.11) and (526, 402) respectively. The set of generated points was contaminated by different amounts of noise and then the covariances of the epipoles estimated analytically using Eq. (10) were compared to those computed using a statistical method which approximates the covariances by exploiting the laws of large numbers. In simpler terms, the mean of a random vector $\mathbf{y}$ can be approximated by the discrete mean of a sufficiently large number N of samples, defined by $E_D[\mathbf{y}] = \frac{1}{N}\sum_{i=1}^{N}\mathbf{y}_i$ and the corresponding covariance by

$$\frac{N}{N-1}E_D[(\mathbf{y}_i - E_D[\mathbf{y}])(\mathbf{y}_i - E_D[\mathbf{y}])^T]. \tag{13}$$

Assuming additive Gaussian noise whose standard deviation increased from 0.1 to 2.0 in increments of 0.1 pixels, the analytically computed covariance estimates were compared against those produced by the statistical method. In particular, for each level of noise σ, 1000 noise-corrupted samples of the original corresponding pairs set were obtained by adding zero mean Gaussian noise with standard deviation σ to the original set of corresponding pairs. Then, 1000 epipole pairs were computed through the estimation of the 1000 fundamental matrices pertaining to the 1000 noise corrupted samples. Following this, the statistical estimates of the two epipole covariances were computed using Eq.(13) for $N = 1000$. To estimate the epipole covariances with the analytical method, the latter is applied to the fundamental matrix corresponding to a randomly selected sample of noisy pairs.

In order to facilitate both the comparison and the graphical visualization of the estimated covariances, the concept of the hyper-ellipsoid of uncertainty is introduced next. Assuming that a $M \times 1$ random vector $\mathbf{y}$ follows a Gaussian distribution with mean $E[\mathbf{y}]$ and covariance $\Lambda_\mathbf{y}$, it is easy to see that the random vector χ defined by $\chi = {\Lambda_\mathbf{y}}^{-1/2}(\mathbf{y} - E[\mathbf{y}])$ follows a Gaussian distribution of mean

zero and of covariance equal to the $M \times M$ identity matrix $\mathbf{I}$. This implies that the random variable $\delta_{\mathbf{y}}$ defined as

$$\delta_{\mathbf{y}} = \chi^T \chi = (\mathbf{y} - E[\mathbf{y}])^T {\Lambda_{\mathbf{y}}}^{-1} (\mathbf{y} - E[\mathbf{y}])$$

follows a χ^2 (chi-square) distribution with r degrees of freedom, r being the rank of $\Lambda_{\mathbf{y}}$ [17,5]. Therefore, the probability that $\mathbf{y}$ lies within the k-hyper-ellipsoid defined by the equation

$$(\mathbf{y} - E[\mathbf{y}])^T {\Lambda_{\mathbf{y}}}^{-1} (\mathbf{y} - E[\mathbf{y}]) = k^2, \tag{14}$$

is given by the χ^2 cumulative probability function $P_{\chi^2}(k, r)$[2] [29].

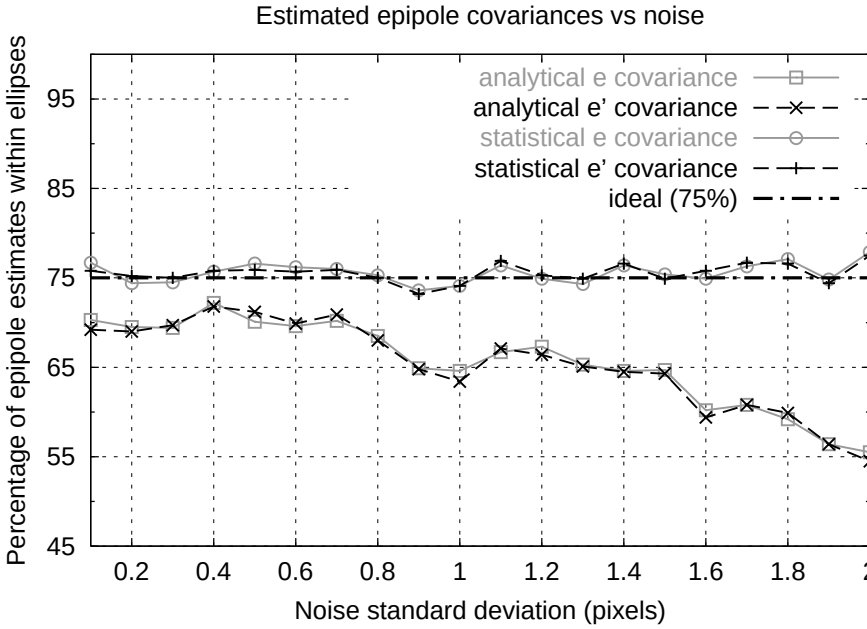

Fig. 3. The fraction of 1000 estimates of the two epipoles that lie within the uncertainty ellipses defined by the corresponding covariances computed with the analytical and statistical method. According to the χ^2 criterion, this fraction is ideally equal to 75%.

In the following, it is assumed that the epipoles are represented using points in the two dimensional Euclidean space $\mathcal{R}^2$ rather than in the embedding projective space $\mathcal{P}^2$. This is simply accomplished by normalizing the estimates of the epipoles obtained from the SVD of the fundamental matrix so that their

[2] The norm defined by the left hand side of Eq.(14) is sometimes referred to as the *Mahalanobis* distance.

third element is equal to one. Choosing the probability $P_{\chi^2}(k, r)$ of an epipole estimate being within the ellipsoid defined by the covariance to be equal to 0.75, yields $k = 1.665$ for $r = 2$. Figure 2 shows the fractions of the epipole estimates that lie within the uncertainty ellipses defined by the covariances computed with the analytical and statistical method, as a function of the noise standard deviation. Clearly, the fractions of points within the ellipses corresponding to the covariances computed with the statistical method are very close to the theoretical 0.75. On the other hand, the estimates of the covariances computed with the analytical method are satisfactory, with the corresponding fractions being over 0.65 when the noise does not exceed 1.5 pixels.

The difference between the covariances estimated by the analytical and statistical methods are shown graphically in Fig.2 for three levels of noise, namely 0.1, 1.0 and 2.0 pixels. Since the analytical method always underestimates the covariance, the corresponding ellipses are contained in the ellipses computed by the statistical method. Nevertheless, the shape and orientation of the ellipses computed with the analytical method are similar to these of the statistically computed ellipses.

3.3 Estimation of the Covariance of Rigid 3D Motion

The third application of the proposed SVD differentiation technique concerns its use for estimating the covariance of rigid 3D motion estimates. It is well known that the object encoding the translation and rotation comprising the 3D motion is the *essential matrix* $\mathbf{E}$. Matrix $\mathbf{E}$ is defined by $\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}$, where $\mathbf{T}$ and $\mathbf{R}$ represent respectively the translation vector and the rotation matrix defining a rigid displacement and $[\mathbf{T}]_\times$ is the antisymmetric matrix associated with the cross product:

$$[\mathbf{T}]_\times = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix}$$

There exist several methods for extracting estimates of the translation and rotation from estimates of the essential matrix. Here, we focus our attention to a simple linear method based on the SVD of $\mathbf{E}$, described in [19,12]. Assuming that the SVD of $\mathbf{E}$ is $\mathbf{E} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, there exist two possible solutions for the rotation $\mathbf{R}$, namely $\mathbf{R} = \mathbf{U}\mathbf{W}\mathbf{V}^T$ and $\mathbf{R} = \mathbf{U}\mathbf{W}^T\mathbf{V}^T$, where $\mathbf{W}$ is given by

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The translation is given by the third column of matrix $\mathbf{V}$, that is $\mathbf{T} = \mathbf{V}(0, 0, 1)^T$ with $|\mathbf{T}| = 1$. The two possible choices for $\mathbf{R}$ combined with the two possible signs of $\mathbf{T}$ yield four possible translation-rotation pairs, from which the correct solution for the rigid motion can be chosen based on the requirement that the visible 3D points appear in the front of both camera viewpoints [19]. In the

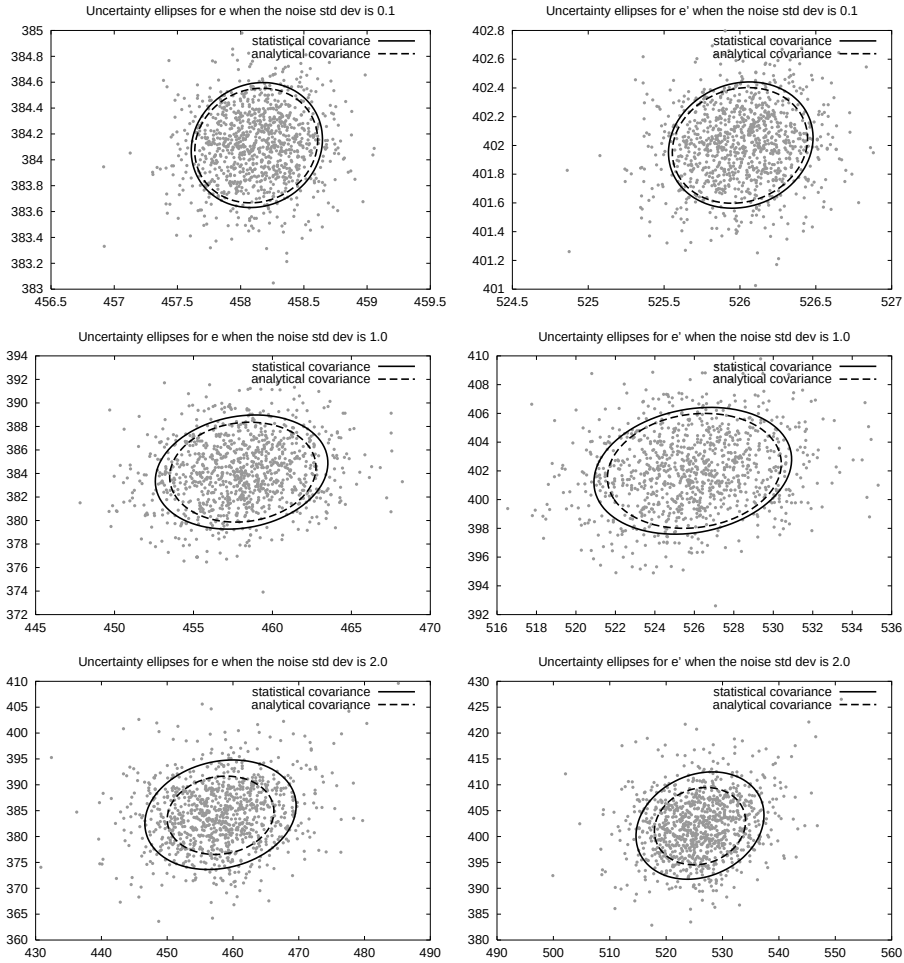

Fig. 4. The ellipses defined by Eq. (14) using $k = 0.75$ and the covariances of the two epipoles computed by the statistical and analytical method. The left column corresponds to $\mathbf{e}$, the right to $\mathbf{e}'$. The standard deviation of the image noise is 0.1 pixels for the first row, 1.0 and 2.0 pixels for the middle and bottom rows respectively. Both axes in all plots represent pixel coordinates while points in the plots marked with grey dots correspond to estimates of the epipoles obtained during the statistical estimation process.

following, the covariance corresponding only to the first of the two solutions for rotation will be computed; the covariance of the second solution can be computed in a similar manner.

Supposing that the problem of camera calibration has been solved, the essential matrix can be recovered from the fundamental matrix using

$$\mathbf{E} = \mathbf{A}^T \mathbf{F} \mathbf{A},$$

where $\mathbf{A}$ is the intrinsic calibration parameters matrix. Using Eq.(10), the covariance of $\mathbf{E}$ can be computed as

$$\Lambda_{\mathbf{E}} = \frac{\partial(\mathbf{A}^T\mathbf{F}\mathbf{A})}{\partial \mathbf{F}} \Lambda_{\mathbf{F}} \frac{\partial(\mathbf{A}^T\mathbf{F}\mathbf{A})}{\partial \mathbf{F}}^T,$$

where $\frac{\partial(\mathbf{A}^T\mathbf{F}\mathbf{A})}{\partial \mathbf{F}}$ is the derivative of $\mathbf{A}^T\mathbf{F}\mathbf{A}$ at $\mathbf{F}$ and $\Lambda_{\mathbf{F}}$ is the covariance of $\mathbf{F}$ [5]. The derivative of $\mathbf{A}^T\mathbf{F}\mathbf{A}$ with respect to the element f_{ij} of $\mathbf{F}$ is equal to

$$\frac{\partial(\mathbf{A}^T\mathbf{F}\mathbf{A})}{\partial f_{ij}} = \mathbf{A}^T \frac{\partial \mathbf{F}}{\partial f_{ij}} \mathbf{A}$$

Matrix $\frac{\partial \mathbf{F}}{\partial f_{ij}}$ is such that all its elements are zero, except from that in row i and column j which is equal to one. Given the covariance of $\mathbf{E}$, the covariance of $\mathbf{R}$ is then computed from

$$\Lambda_{\mathbf{R}} = \frac{\partial(\mathbf{U}\mathbf{W}\mathbf{V}^T)}{\partial \mathbf{E}} \Lambda_{\mathbf{E}} \frac{\partial(\mathbf{U}\mathbf{W}\mathbf{V}^T)}{\partial \mathbf{E}}^T$$

The derivative of $\mathbf{U}\mathbf{W}\mathbf{V}^T$ with respect to the element e_{ij} of $\mathbf{E}$ is given by

$$\frac{\partial(\mathbf{U}\mathbf{W}\mathbf{V}^T)}{\partial e_{ij}} = \frac{\partial \mathbf{U}}{\partial e_{ij}} \mathbf{W}\mathbf{V}^T + \mathbf{U}\mathbf{W} \frac{\partial \mathbf{V}^T}{\partial e_{ij}}$$

The derivatives $\frac{\partial \mathbf{U}}{\partial e_{ij}}$ and $\frac{\partial \mathbf{V}^T}{\partial e_{ij}}$ in the above expression are computed with the aid of the proposed differentiation method.

Regarding the covariance of $\mathbf{T}$, let $\mathbf{V_3}$ denote the vector corresponding to the third column of $\mathbf{V}$. The covariance of translation is then simply

$$\Lambda_{\mathbf{T}} = \frac{\partial \mathbf{V_3}}{\partial \mathbf{E}} \Lambda_{\mathbf{E}} \frac{\partial \mathbf{V_3}}{\partial \mathbf{E}}^T,$$

with $\frac{\partial \mathbf{V_3}}{\partial \mathbf{E}}$ being again computed using the proposed method.

4 Conclusions

The Singular Value Decomposition is a linear algebra technique that has been successfully applied to a wide variety of domains that involve matrix computations. In this paper, a novel technique for computing the Jacobian of the SVD components of a matrix with respect to the matrix itself has been described. The usefulness of the proposed technique has been demonstrated by applying it to the estimation of the uncertainty in three different practical vision problems, namely self-calibration, epipole estimation and rigid 3D motion estimation.

Acknowledgments The authors are very grateful to Prof. Rachid Deriche for his continuous support during this work.

References

1. E. Anderson, Z. Bai, C. Bishof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688, second edition, 1994.
2. K. Arun, T. Huang, and S. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, Sept. 1987.
3. Å. Björck. *Numerical methods for least squares problems*. SIAM, 1996.
4. T. Chen, S. Amari, and Q. Lin. A unified algorithm for principal and minor components extraction. *Neural Networks*, 11(3):385–390, 1998.
5. G. Csurka, C. Zeller, Z. Zhang, and O. Faugeras. Characterizing the uncertainty of the fundamental matrix. *CVGIP: Image Understanding*, 68(1):18–36, Oct. 1997.
6. J. Demmel and K. Veselić. Jacobi's method is more accurate than QR. *SIAM Journal on Matrix Analysis and Applications*, 13:1204–1245, 1992.
7. K. Diamantaras and S. Kung. Multilayer neural networks for reduced-rank approximation. *IEEE Trans. on Neural Networks*, 5(5):684–697, 1994.
8. O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
9. N. Georgis, M. Petrou, and J. Kittler. On the correspondence problem for wide angular separation of non-coplanar points. *Image and Vision Computing*, 16:35–41, 1998.
10. G. Golub and C. V. Loan. *Matrix computations*. The John Hopkins University Press, Baltimore, Maryland, second edition, 1989.
11. R. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):133–135, Feb. 1997.
12. R. I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision*, pages 579–587, Santa Margherita, Italy, May 1992. Springer-Verlag.
13. R. I. Hartley. Cheirality invariants. In *Proceedings of the ARPA Image Understanding Workshop*, pages 745–753, Washington, DC, Apr. 1993. Defense Advanced Research Projects Agency, Morgan Kaufmann Publishers, Inc.
14. D. Jones and J. Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708, 1992.
15. B. N. K. Konstantinides and G. Yovanof. Noise estimation and filtering using block-based singular value decomposition. *IEEE Trans. on Image Processing*, 6(3):479–483, 1997.
16. K. Kanatani. Analysis of 3-D rotation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):543–54, 1994.
17. K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996.
18. R. Liu and T. Tan. A new SVD based image watermarking method. In *Proc. of the 4th Asian Conference on Computer Vision*, volume I, pages 63–67, Jan. 2000.

19. H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
20. M. I. Lourakis and R. Deriche. Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3D measurements. Research Report 3748, INRIA Sophia-Antipolis, Aug. 1999.
21. M. I. Lourakis and R. Deriche. Camera self-calibration using the Kruppa equations and the SVD of the fundamental matrix: The case of varying intrinsic parameters. Research report, INRIA Sophia-Antipolis, 2000. In preparation.
22. M. I. Lourakis and R. Deriche. Camera self-calibration using the singular value decomposition of the fundamental matrix. In *Proc. of the 4th Asian Conference on Computer Vision*, volume I, pages 403–408, Jan. 2000.
23. Q.-T. Luong and O. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *The International Journal of Computer Vision*, 22(3):261–289, 1997.
24. Q.-T. Luong and O. Faugeras. On the determination of epipoles using cross-ratios. *CVGIP: Image Understanding*, 71(1):1–18, July 1998.
25. S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *The International Journal of Computer Vision*, 8(2):123–152, Aug. 1992.
26. M. Moonen and B. D. Moor, editors. *SVD and Signal Processing III: Algorithms, Analysis and Applications*. Elsevier, Amsterdam, 1995.
27. T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):858–867, 1997.
28. C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, 1997.
29. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
30. S. Roy, J. Meunier, and I. Cox. Cylindrical rectification to minimize epipolar distortion. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 393–399, San Juan, Puerto Rico, June 1997. IEEE Computer Society.
31. L. Scharf. The SVD and reduced rank signal processing. *Signal Processing*, 25(2):113–133, 1991.
32. A. Shashua and N. Navab. Relative affine structure - canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:873–883, 1996.
33. G. W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review*, 15(4):727–764, Oct. 1973.
34. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, pages 709–720, Cambridge, UK, Apr. 1996.
35. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *The International Journal of Computer Vision*, 9(2):137–154, 1992.
36. R. Tsai, T. Huang, and W. Zhu. Estimating three-dimensional motion parameters of a rigid planar patch, II: singular value decomposition. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 30(4):525–534, 1982.
37. S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

38. R. J. Vaccaro. A second-order perturbation expansion for the svd. *SIAM Journal on Matrix Analysis and Applications*, 15(2):661–671, Apr. 1994.
39. C. Wu, M. Berry, S. Shivakumar, and J. McLarty. Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition. *Machine Learning*, 21(1-2), 1995.
40. J. Yang and C. Lu. Combined techniques of singular value decomposition and vector quantization for image coding. *IEEE Trans. on Image Processing*, 4(8):1141–1146, 1995.
41. C. Zeller and O. Faugeras. Camera self-calibration from video sequences: the Kruppa equations revisited. Research Report 2793, INRIA, Feb. 1996.
42. M. Zhang, K. Yu, and R. Haralick. Fast correlation registration method using singular value decomposition. *International Journal of Intelligent Systems*, 1:181–194, 1986.
43. Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, Oct. 1995.