
International Journal of Computer Vision 49(2/3), 101–116, 2002

Factorization with Uncertainty

P. ANANDAN
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA

MICHAL IRANI
Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel

Received March 22, 2001; Revised July 10, 2001; Accepted November 13, 2001

Abstract. Factorization using Singular Value Decomposition (SVD) is often used for recovering 3D shape and motion from feature correspondences across multiple views. SVD is powerful at finding the global solution to the associated least-square-error minimization problem. However, this is the correct error to minimize only when the x and y positional errors in the features are uncorrelated and identically distributed. But this is rarely the case in real data. Uncertainty in feature position depends on the underlying spatial intensity structure in the image, which has strong directionality to it. Hence, the proper measure to minimize is covariance-weighted squared-error (or the *Mahalanobis distance*). In this paper, we describe a new approach to covariance-weighted factorization, which can factor noisy feature correspondences with high degree of directional uncertainty into structure and motion. Our approach is based on transforming the raw-data into a covariance-weighted data space, where the components of noise in the different directions are uncorrelated and identically distributed. Applying SVD to the transformed data now minimizes a meaningful objective function in this new data space. This is followed by a linear but suboptimal second step to recover the shape and motion in the original data space. We empirically show that our algorithm gives very good results for varying degrees of directional uncertainty. In particular, we show that unlike other SVD-based factorization algorithms, our method does not degrade with increase in directionality of uncertainty, even in the extreme when only normal-flow data is available. It thus provides a unified approach for treating corner-like points together with points along linear structures in the image.

Keywords: factorization, structure from motion, directional uncertainty

1. Introduction

Factorization is often used for recovering 3D shape and motion from feature correspondences across multiple frames (Tomasi and Kanade, 1992; Poelman and Kanade, 1997; Quan and Kanade, 1996; Shapiro, 1995; Sturm and Triggs, 1996; Oliensis, 1999; Oliensis and Genc, to appear). Singular Value Decomposition (SVD) directly obtains the global minimum of the *total* (*orthogonal*) least-squares error (Van Huffel and Vandewalle, 1991; Kanatani, 1996) between the noisy data and the bilinear model involving motion of the camera and the 3D position of the points (*shape*). This is in contrast to iterative non-linear optimization methods which may converge to a local minimum. However, SVD assumes that the noise in the x and y positions of features are uncorrelated and have identical distributions. But, it is rare that positional errors of feature tracking algorithms are uncorrelated in their x and y coordinates. Quality of feature matching depends on the spatial variation of the intensity pattern around each feature. This affects the positional inaccuracy both in

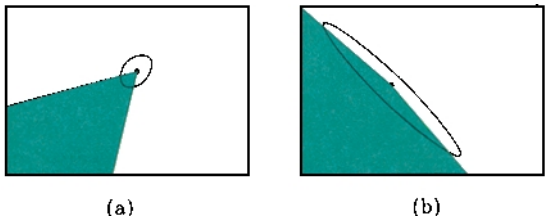

Figure 1. Directional uncertainty indicated by ellipse. (a) Uncertainty of a sharp corner point. The uncertainty in all directions is small, since the underlying intensity structure shows variation in multiple directions. (b) Uncertainty of a point on a flat curve, almost a straight line. Note that the uncertainty in the direction of the line is large, while the uncertainty in the direction perpendicular to the line is small. This is because it is hard to localize the point along the line.

the x and in the y components in a correlated fashion. This dependency can be modeled by *directional* uncertainty (Anandan, 1989) (which varies from point to point, as is shown in Fig. 1).

When the uncertainty in a feature position is isotropic, but different features have different variances, then *scalar-weighted SVD* can be used to minimize a weighted squared error measure (e.g., Aguiar and Moura, 1999). However, under directional uncertainty noise assumptions (which is the case in reality), the error minimized by SVD is no longer meaningful. The proper measure to minimize is the *covariance-weighted error* (the Mahalanobis distance). Kanatani (1996) and others (e.g., Leedan and Meer, 2000; Ben-Ezra et al., 2000; Matei and Meer, 2000; Morris et al., 1999; Morris and Kanade, 1998) have stressed the need to use Mahalanobis distance in various vision related estimation problems when the noise is data-dependent. However, most of the work on factorization of multiframe correspondences that uses SVD has not incorporated directional uncertainty (e.g., see Tomasi and Kanade, 1992; Poelman and Kanade, 1997; Aguiar and Moura, 1999; Sturm and Triggs, 1996).

The techniques that have incorporated directional uncertainty and minimized the Mahalanobis distance have not used the power of SVD to obtain a global minimum. For example, Morris and Kanade (1998) and Morris et al. (1999) have suggested a unified approach for recovering the 3D structure and motion from point and line features, by taking into account their directional uncertainty. However, they solve their objective function using an iterative non-linear minimization scheme. The line factorization algorithm of Quan and Kanade (1996) is SVD-based. However, it requires a preliminary step of 2D projective reconstruction, which is necessary for rescaling the line directions in the image before further factorization can be applied. This step is then followed by three sequential SVD minimization steps, each applied to different intermediate results. This algorithm requires at least seven different directions of lines.

In this paper we present a new approach to factorization, which introduces *directional uncertainty* into the SVD minimization framework. The input is the noisy positions of image features and their inverse covariance matrices which represent the uncertainty in the data. Following the approach of Irani (2002), we write the image position vectors as row vectors, rather than as column vectors as is typically done in factorization methods. This allows us to use the inverse covariance matrices to transform the input position vectors into a new data space (the "covariance-weighted space"), where the noise is uncorrelated and identically distributed. In the new covariance-weighted data space, corner points and points on lines all have the same reliability, and their new positional components are uncorrelated. (This is in contrast with the original data space, where corner points and points on lines had different reliability, and their x and y components were correlated.)

Once the data is thus transformed, we can apply SVD factorization to the covariance-weighted data. This is equivalent to minimizing the *Mahalanobis distance* in the original data space. However, the covariance-weighted data space has double the rank of the original data space. An additional suboptimal linear minimization step is needed to obtain the correct rank in the original data space. Despite this suboptimal linear step, the bulk of the rank reduction occurs during the preceding SVD step, leading to very good results in practice.

More importantly, our approach allows the recovery of 3D motion for all frames and the 3D shape for all points, even when the uncertainty of point position is highly elliptic (for example, point on a line). It can handle reliable corner-like point correspondences and partial correspondences of points on lines (e.g., normal flow), all within a single SVD-like framework. In fact, we can handle extreme cases when the only image data available is normal flow.

Irani (2002) used confidence-weighted subspace projection directly on spatio-temporal brightness derivatives, in order to constrain multi-frame

correspondence estimation. The confidences she used encoded directional uncertainty associated with each pixel. That formulation can be seen a special case of the covariance-weighted factorization presented in this paper.

Our approach thus combines the powerful SVD factorization technique with a proper treatment of directional uncertainty in the data. Different input features can have different directional uncertainties with different ellipticities (i.e., different covariance matrices). However, our algorithm is still slightly suboptimal. Furthermore, our approach does not allow for arbitrary changes in the uncertainty of a single feature over multiple frames. We are currently able to handle the case where the change in the covariance matrices of all of the image features can be modeled by a global 2D affine transformation, which varies from frame to frame.

The rest of the paper is organized as follows: Section 2 contains a short review of SVD factorization and formulates the problem for the case of directional uncertainty. Section 3 describes the transition from the raw data space, where noise is correlated and non-uniform, to the covariance-weighted data space, where noise is uniform and uncorrelated, giving rise to meaningful SVD subspace projection. Section 4 explains how the covariance-weighted data can be factored into 3D motion and 3D shape. Section 5 extends the solution presented in Sections 3 and 4 to a more general case when the directional uncertainty of a point changes across views. Section 6 provides experimental results and empirical comparison of our factorization method to other common SVD factorization methods. Section 7 concludes the paper. A shorter version of this paper appeared in Irani and Anandan (2000).

2. Problem Formulation

2.1. SVD Factorization

A set of P points are tracked across F images with coordinates $\{(u'_{fp}, v'_{fp}) \mid f = 1, \ldots, F, p = 1, \ldots, P\}$. The point coordinates are transformed to object-centered coordinates by subtracting their centroid. Namely, (u'_{fp}, v'_{fp}) is replaced by $(u_{fp}, v_{fp}) = (u'_{fp} - \bar{u}_f, v'_{fp} - \bar{v}_f)$ for all f and p, where $\bar{u}_f$ and $\bar{v}_f$ are the centroids of point positions in each frame: $\bar{u}_f = \frac{1}{P}\sum_p u'_{fp}$, $\bar{v}_f = \frac{1}{P}\sum_p v'_{fp}$.

Two $F \times P$ measurement matrices U and V are constructed by stacking all the measured correspondences as follows:

$$U = \begin{bmatrix} u_{11} & \cdots & u_{1P} \\ \vdots & & \vdots \\ u_{F1} & \cdots & u_{FP} \end{bmatrix}, \quad V = \begin{bmatrix} v_{11} & \cdots & v_{1P} \\ \vdots & & \vdots \\ v_{F1} & \cdots & v_{FP} \end{bmatrix}. \tag{1}$$

It was shown (Tomasi and Kanade, 1992; Poelman and Kanade, 1997; Shapiro, 1995) that when the camera is an affine camera (i.e., orthographic, weak-perspective, or paraperspective), and when there is no noise, then the rank of the $2F \times P$ matrix $W = \begin{bmatrix} U \\ V \end{bmatrix}$ is 3 or less, and can be factored into a product of a motion matrix M and a shape matrix S, i.e., $W = MS$, where:

$$M = \begin{bmatrix} M_U \\ M_V \end{bmatrix}_{2F\times 3}, \quad S = [s_1, \ldots, s_P]_{3\times P},$$
$$M_U = \begin{bmatrix} m_1^T \\ \vdots \\ m_F^T \end{bmatrix}_{F\times 3}, \quad M_V = \begin{bmatrix} n_1^T \\ \vdots \\ n_F^T \end{bmatrix}_{F\times 3}. \tag{2}$$

The rows of M encode the motion for each frame (rotation in the case of orthography), and the columns of S contain the 3D position of each point in the reconstructed scene.

In practice, the measured data is usually corrupted by noise. The standard approach is to model this noise as an additive stochastic random variable $\mathcal{E}_{fp}$ with a Gaussian probability density function. Thus the noisy measured position vector $(u_{fp}\ v_{fp})^T$ is modeled as:

$$\begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix} = \begin{bmatrix} m_f^T s_p \\ n_f^T s_p \end{bmatrix} + \mathcal{E}_{fp}. \tag{3}$$

When $\mathcal{E}_{fp}$ is modeled as an *isotropic* Gaussian random variable with a fixed variance σ^2, i.e., $\forall f\ \forall p\ \mathcal{E}_{fp} \sim N(0, \sigma^2 I_{2\times 2})$, then the maximum likelihood estimate is obtained by minimizing the squared error:

$$\mathrm{Err}_{\mathrm{SVD}}(M, S) = \sum_{f,p} \mathcal{E}_{fp}^T \mathcal{E}_{fp} = \|W - MS\|_F^2 \tag{4}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The *global minimum* to this non-linear problem is obtained by performing Singular Value Decomposition (SVD) on the measurement matrix: $W = A\Sigma B^T$, and setting to zero all but the three largest singular values in Σ, to get a

noise-cleaned matrix $\hat{W} = A\hat{\Sigma}B^T$. The recovered motion and shape matrices $\hat{M}$ and $\hat{S}$ are then obtained by: $\hat{M} = A\hat{\Sigma}^{1/2}$, and $\hat{S} = \hat{\Sigma}^{1/2}B$. Note that $\hat{M}$ and $\hat{S}$ are defined only up to an affine transformation.

2.2. Scalar Uncertainty

The model in Section 2.1 (as well as in Tomasi and Kanade (1992)) weights equally the contribution of each point feature to the final shape and motion matrices. However, when the noise $\mathcal{E}_{fp}$ is isotropic, but with different variances for the different points $\{\sigma_p^2 \mid p = 1, \ldots, P\}$, then $\mathcal{E}_{fp} \sim N(0, \sigma_p^2 I_{2\times 2})$. In such cases, applying SVD to the weighted-matrix $W_\sigma = W\sigma^{-1}$, where $\sigma^{-1} = \text{diag}(\sigma_1^{-1}, \ldots, \sigma_P^{-1})$, will minimize the correct error function:

$$\begin{aligned}\text{Err}_{\text{weighted-SVD}}(M, S) &= \sum \frac{\mathcal{E}_{fp}^T \mathcal{E}_{fp}}{\sigma_p^2} = \|(W - MS)\sigma\|_F \\ &= \|W_\sigma - MS_\sigma\|_F \end{aligned} \tag{5}$$

where $S_\sigma = S\sigma^{-1}$. Applying SVD-factorization to W_σ will give $\hat{M}$ and $\hat{S}_\sigma$, from which $\hat{S} = \hat{S}_\sigma\sigma$ can be recovered. This approach is known as weighted-SVD or weighted-factorization (Aguiar and Moura, 1999).

2.3. Directional Uncertainty

So far we have assumed that the noise in u_{fp} is uncorrelated with the noise in v_{fp}. In real image sequences, however, this is not the case. The uncertainty in the different components of the location estimate of an image feature will depend on the local image structure. For example, a corner point p will be tracked with high reliability both in u_{fp} and in v_{fp}, while a point p on a line will be tracked with high reliability in the direction of the gradient ("normal flow"), but with low reliability in the tangent direction (see Fig. 1). This leads to non-uniform correlated noise in u_{fp} and v_{fp}. We model the correlated noise $\mathcal{E}_{fp}$ by: $\mathcal{E}_{fp} \sim N(0, Q_{fp}^{-1})$ where Q_{fp} is the 2×2 *inverse* covariance matrix of the noise at point p in image-frame f (see Fig. 2). The covariance matrix determines an ellipse whose major and minor axes indicate the directional uncertainty in the location $(u_{fp}\ v_{fp})^T$ of a point p in frame f (see Fig. 1, as well as Morris and Kanade (1998) for some examples).[1]

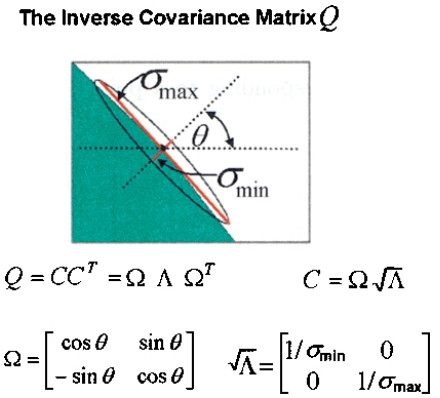

Figure 2. The inverse covariance matrix Q (and its square root matrix C) are defined by the orientation of the uncertainty ellipse and the degree of uncertainty along the major and minor axes.

Assuming that the noise at different points is independent, then the maximum likelihood solution is obtained by finding matrices M and S which minimize the following objective function:

$$\text{Err}(M, S) = \sum_{f,p} \left(\mathcal{E}_{fp}^T Q_{fp} \mathcal{E}_{fp}\right) \tag{6}$$

where:

$$\mathcal{E}_{fp} = \begin{bmatrix} u_{fp} - m_f^T s_p \\ v_{fp} - n_f^T s_p \end{bmatrix}.$$

Equation (6) implies that in the case of directional uncertainty, the metric that we want to use in the minimization is the *Mahalanobis distance*. When the noise in each of the data points is isotropic (as might be the case at a set of corner points), Q_{fp} are of the form $\lambda \mathbf{I}_{2\times 2}$ and the error reduces to the *Frobenius* (*least-squares*) *norm* of Eq. (5). This is the distance minimized by the standard SVD process, and is only meaningful when data consists entirely of points with isotropic noise.

Morris and Kanade (1998) have addressed this problem and suggested an approach to recovering M and S which is based on minimizing the Mahalanobis distance. However, their approach uses an iterative non-linear minimization scheme. In the next few sections we present our approach to SVD-based factorization, which minimizes the Mahalanobis error. Our approach combines the benefits of SVD-based factorization for getting a good solution, with the proper treatment of directional uncertainty. However, unlike (Morris and Kanade, 1998), our approach cannot handle *arbitrary*

changes in covariance matrices of a single feature over multiple frames. It can only handle frame-dependent 2D affine deformations of the covariance matrices across different views (see Section 5).

3. From the Raw-Data Space to the Covariance-Weighted Space

In this section we show how by transforming the noisy data (i.e., correspondences) from the raw-data space to a new covariance-weighted space, we can minimize the Mahalanobis distance defined in Eq. (6), while retaining the benefits of SVD minimization. This transition is made possible by rearranging the raw feature positions in a slightly modified matrix form: $[U \mid V]_{F\times 2P}$, namely the matrices U and V stacked horizontally (as opposed to vertically in $W = \begin{bmatrix} U \\ V \end{bmatrix}$, which is the standard matrix form used in the traditional factorization methods (see Section 2.1)). This modified matrix representation is necessary to introduce covariance-weights into the SVD process, and was originally proposed by Irani (2002).

For simplicity, we start by investigating the simpler case when the directional uncertainty of a point does not change over time (i.e., frames), namely, when the 2×2 inverse covariance matrix Q_{fp} of a point p is frame-independent: $\forall f\, Q_{fp} \equiv Q_p$. Later, in Section 5, we will extend the approach to handle the case when the covariance matrices undergo frame-dependent 2D-affine changes. Because Q_p is positive semi-definite, its eigenvalue decomposition has the form $Q_p = \Omega\Lambda\Omega^T$, where $\Omega_{2\times 2}$ is a real orthonormal matrix, and $\Lambda_{2\times 2} = \mathrm{diag}(\lambda_{\max}, \lambda_{\min})$. Also, $\lambda_{\max} = \frac{1}{\sigma^2_{\min}}$ and $\lambda_{\min} = \frac{1}{\sigma^2_{\max}}$, where $\sigma_{\max}$ and $\sigma_{\min}$ are the standard deviations of the uncertainty along the maximum and minimum uncertainty directions (see Fig. 2). Let $C_p = \Omega\Lambda^{\frac{1}{2}}$ and $[\alpha_{fp}\ \beta_{fp}]_{1\times 2} = [u_{fp}\ v_{fp}]_{1\times 2} C_{p_{2\times 2}}$. Therefore, α_{fp} is the component of $[u_{fp}\ v_{fp}]$ in the direction of the *highest* certainty (scaled by its certainty), and β_{fp} is the component in the direction of the *lowest* certainty (scaled by its certainty) (see Fig. 3).

For example, in the case of a point p which lies on a line, α_{fp} would correspond to the component in the direction perpendicular to the line (i.e., the direction of the *normal flow*), and β_{fp} would correspond to the component in the direction tangent the line (the direction of infinite uncertainty). In the case of a perfect line (i.e., zero certainty in the direction of the line), then $\beta_{fp} = 0$. When the position of a point can be determined with finite certainty in both directions (e.g., for corner points), then C_p is a regular matrix. Otherwise, when there is infinite uncertainty in at least one direction (e.g., as in lines or uniform image regions), then C_p is singular.

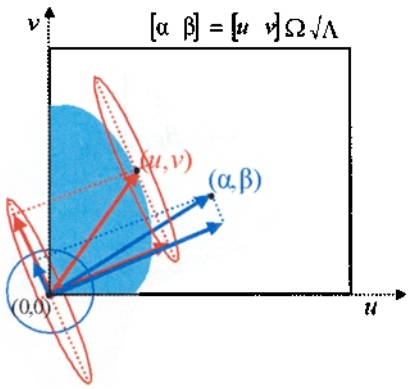

Figure 3. Using the notation from Fig. 2, $[u\ v]$ is projected onto the major and minor axes of the ellipse via the rotation matrix Ω. Each component is then scaled by its appropriate uncertainty using $\sqrt{\Lambda}$. This provides the covariance-weighted vector $[\alpha\ \beta]$, where α is the component in the direction of the highest certainty, and β is the component in the direction of the lowest certainty.

Let α_p, β_p, u_p and v_p be four $F \times 1$ vectors corresponding to a point p across all frames:

$$\alpha_p = \begin{bmatrix} \alpha_{1p} \\ \vdots \\ \alpha_{Fp} \end{bmatrix}, \quad \beta_p = \begin{bmatrix} \beta_{1p} \\ \vdots \\ \beta_{Fp} \end{bmatrix},$$

$$u_p = \begin{bmatrix} u_{1p} \\ \vdots \\ u_{Fp} \end{bmatrix}, \quad v_p = \begin{bmatrix} v_{1p} \\ \vdots \\ v_{Fp} \end{bmatrix}$$

then

$$[\alpha_p \quad \beta_p]_{F\times 2} = [u_p \quad v_p]_{F\times 2}\, C_{p_{2\times 2}}. \tag{7}$$

Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be two $F \times P$ matrices:

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1P} \\ \vdots & & \vdots \\ \alpha_{F1} & \cdots & \alpha_{FP} \end{bmatrix}_{F\times P} \quad \text{and}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1P} \\ \vdots & & \vdots \\ \beta_{F1} & \cdots & \beta_{FP} \end{bmatrix}_{F\times P} \tag{8}$$

then, according to Eq. (7):

$$[\alpha \mid \beta]_{F\times 2P} = [U \mid V]_{F\times 2P} C_{2P\times 2P} \tag{9}$$

where C is a $2P \times 2P$ matrix, constructed from all 2×2 matrices $C_p = \begin{bmatrix} c_{p1} & c_{p2} \\ c_{p3} & c_{p4} \end{bmatrix}$ $(p = 1, \ldots, P)$, as follows:

$$C = \left[\begin{array}{ccc|ccc} c_{11} & & 0 & c_{12} & & 0 \\ & \ddots & & & \ddots & \\ 0 & & c_{P1} & 0 & & c_{P2} \\ \hline c_{13} & & 0 & c_{14} & & 0 \\ & \ddots & & & \ddots & \\ 0 & & c_{P3} & 0 & & c_{P4} \end{array}\right]_{2P\times 2P}. \tag{10}$$

Note that matrix α contains the components of all point positions in their directions of highest certainty, and β contains the components of all point positions in their directions of lowest certainty. These directions vary from point to point and are independent. Furthermore, α_{fp} and β_{fp} are also independent, and the noise in those two components is now uncorrelated.

Let R denote the rank of $W = \begin{bmatrix} U \\ V \end{bmatrix}_{2F\times P}$ (when W is noiseless, and the camera is an affine camera, then $R \leq 3$; see Section 2.1). A review of different ranks R for different camera and world models can be found in Irani (2002). Then the rank of U and the rank of V is each at most R. Hence, the rank of $[U \mid V]_{F\times 2P}$ is at most $2R$ (for an affine camera, in the absence of noise, $2R \leq 6$). Therefore, according to Eq. (9), the rank of $[\alpha \mid \beta]$ is also at most $2R$.

The problem of minimizing the Mahalanobis distance of Eq. (6) can be restated as follows: Given noisy positions $\{(u_{fp}\ v_{fp})^T \mid f = 1, \ldots, F,\ p = 1, \ldots, P\}$, find new positions $\{(\hat{u}_{fp}\ \hat{v}_{fp})^T \mid f = 1, \ldots, F,\ p = 1, \ldots, P\}$ that minimize the following error function:

$$\begin{aligned} &\mathrm{Err}(\{(\hat{u}_{fp} \quad \hat{v}_{fp})^T\}) \\ &= \sum_{f,p} [(u_{fp} - \hat{u}_{fp}) \quad (v_{fp} - \hat{v}_{fp})] Q_{fp} \begin{bmatrix} u_{fp} - \hat{u}_{fp} \\ v_{fp} - \hat{v}_{fp} \end{bmatrix}. \end{aligned} \tag{11}$$

Because $Q_{fp} = Q_p = C_p C_p^T$, we can rewrite this error term as:

$$\begin{aligned} &= \sum_{f,p} ([(u_{fp} - \hat{u}_{fp}) \quad (v_{fp} - \hat{v}_{fp})] C_p) \\ &\quad \cdot ([(u_{fp} - \hat{u}_{fp}) \quad (v_{fp} - \hat{v}_{fp})] C_p)^T \\ &= \|[U - \hat{U} \mid V - \hat{V}]C\|_F^2 \\ &= \|[U \mid V]C - [\hat{U} \mid \hat{V}]C\|_F^2 \\ &= \|[\alpha \mid \beta] - [\hat{\alpha} \mid \hat{\beta}]\|_F^2 \end{aligned} \tag{12}$$

where $[\hat{U} \mid \hat{V}]$ is the $F \times 2P$ matrix containing all the $\{\hat{u}_{fp}, \hat{v}_{fp}\}$, and $[\hat{\alpha} \mid \hat{\beta}] = [\hat{U} \mid \hat{V}]C$.

Note, however, that in order to be a physically valid solution, $\hat{U}$ and $\hat{V}$ must satisfy the constraint

$$\begin{bmatrix} \hat{U} \\ \hat{V} \end{bmatrix} = \begin{bmatrix} \widehat{M_U} \\ \widehat{M_V} \end{bmatrix} \hat{S}, \tag{13}$$

for some motion matrices $\hat{M}_U$, $\hat{M}_V$, and shape matrix $\hat{S}$, i.e., $\begin{bmatrix} \hat{U} \\ \hat{V} \end{bmatrix}$ is a rank-R matrix. Hence,

$$\begin{aligned} [\hat{\alpha} \mid \hat{\beta}]_{F\times 2P} &= [\hat{M}_U \hat{S} \mid \hat{M}_V \hat{S}]C \\ &= [\hat{M}_U \mid \hat{M}_V]_{F\times 2R} \left[\begin{array}{c|c} \hat{S} & 0 \\ \hline 0 & \hat{S} \end{array}\right]_{2R\times 2P} C_{2P\times 2P}. \end{aligned} \tag{14}$$

Thus,

> Minimizing the *Mahalanobis distance* of Eq. (11) subject to Eq. (13) is equivalent to finding the rank-$2R$ matrix $[\hat{\alpha} \mid \hat{\beta}]$ closest to $[\alpha \mid \beta]$ in the *Frobenius norm* of Eq. (12) subject to Eq. (14).

4. Factoring Shape and Motion

In this section, we describe our algorithm to solve the constrained optimization problem posed at the end of Section 3. Our algorithm consists of two steps:

Step 1: Project the covariance-weighted data $[\alpha \mid \beta] = [U \mid V]C$ onto a $2R$-dimensional subspace (i.e., a rank-$2R$ matrix) $[\hat{\alpha} \mid \hat{\beta}]$ using SVD-based subspace projection. This step is guaranteed to obtain the closest $2R$-dimensional subspace because of the global optimum property of SVD.

This first step, although performs bulk of the projection of the noisy data from a high-dimensional space (the smaller of F and $2P$) to a much smaller $2R$ dimensional subspace (e.g., for an affine camera $2R \leq 6$), it does not guarantee the tighter rank R constraint of Eq. (13). To enforce this constraint, we perform a second step of the algorithm as described below.

Step 2: Starting with the matrix $[\hat{\alpha} \mid \hat{\beta}]$ obtained after Step 1, if C were an invertible matrix, then we could have recovered $[\hat{U} \mid \hat{V}]$ by: $[\hat{U} \mid \hat{V}] = [\hat{\alpha} \mid \hat{\beta}]C^{-1}$, and then proceeded with applying standard SVD to $\begin{bmatrix} \hat{U} \\ \hat{V} \end{bmatrix}$ to impose the rank-R constraint and recover $\hat{M}$ and $\hat{S}$. However, in general C is not invertible (e.g.,

because of points with high aperture problem). Imposing the rank-R constraint on $\hat{U} = \hat{M}_U \hat{S}$ and $\hat{V} = \hat{M}_V \hat{S}$ must therefore be done in the $[\hat{\alpha} \mid \hat{\beta}]$ space (i.e., without inverting C). As it was shown in Eq. (14):

$$[\hat{\alpha} \mid \hat{\beta}]_{F\times 2P} = [\hat{M}_U \hat{S} \mid \hat{M}_V \hat{S}]C = [\hat{M}_U \mid \hat{M}_V]_{F\times 2R} \left[\begin{array}{c|c} \hat{S} & 0 \\ \hline 0 & \hat{S} \end{array}\right]_{2R\times 2P} C_{2P\times 2P}.$$

Not every decomposition of $[\hat{\alpha} \mid \hat{\beta}]$ contains a shape matrix of the form $\left[\begin{array}{c|c} \hat{S} & 0 \\ \hline 0 & \hat{S} \end{array}\right]$. We try to find a decomposition of this form that is the closest approximation to the given $[\hat{\alpha} \mid \hat{\beta}]$.

Because $[\hat{\alpha} \mid \hat{\beta}]_{F\times 2P}$ is a rank-$2R$ matrix, it can be written as a bilinear product of an $F \times 2R$ matrix H and a $2R \times 2P$ matrix G:

$$[\hat{\alpha} \mid \hat{\beta}]_{F\times 2P} = H_{F\times 2R} G_{2R\times 2P}. \tag{15}$$

This decomposition is not unique. For any invertible $2R \times 2R$ matrix D, $[\hat{\alpha} \mid \hat{\beta}] = (HD^{-1})(DG)$ is also a valid decomposition. We seek a matrix D which will bring DG into a form

$$DG = \left[\begin{array}{c|c} S & 0 \\ \hline 0 & S \end{array}\right] C \tag{16}$$

where S is an arbitrary $R \times P$ matrix. This is a linear system of equations in the unknown components of S and D. In general, this system does not have an exact solution (if it did, we would have an exact decomposition of $[\hat{\alpha} \mid \hat{\beta}]$ into the correct form). We therefore solve Eq. (16) in a least-squares sense to obtain $\hat{S}$ and $\hat{D}$. The final shape and motion matrices are then obtained as: $\hat{S}$ and $[\hat{M}_U \mid \hat{M}_V] := H\hat{D}^{-1}$ respectively. For more details on how $\hat{S}$ and $\hat{D}$ are recovered from DG, see Appendix A.

Our algorithm thus consists of two stpes. The first step, which performs the bulk of the optimization task (by taking the noisy high-dimensional data into the Rank-$2R$ subspace) is *optimal*. The second step is linear but suboptimal.[2] The optimal Rank-R solution to the original problem is not likely to lie within the Rank-$2R$ subspace computed in Step 1 of our algorithm. Although our algorithm is suboptimal, our empirical results presented in Section 6 indicate that our two-step algorithm accurately recovers the motion and shape, while taking into account varying degrees of directional uncertainty.

5. Frame-Dependent Directional Uncertainty

So far we have assumed that all frames share the same 2×2 inverse covariance matrix Q_p for a point p, i.e., $\forall f\, Q_{fp} \equiv Q_p$ and thus $C_{fp} \equiv C_p$. This assumption, however, is very restrictive, as image motion induces changes in these matrices. For example, a rotation in the image plane induces a rotation on C_{fp} (for all points p). Similarly, a scaling in the image plane induces a scaling in C_{fp}, and so forth for skew in the image plane. (Note, however, that a shift in the image plane does not change C_{fp}.)

The assumption $\forall f\, C_{fp} \equiv C_p$ was needed in order to obtain the separable matrix form of Eq. (9), thus deriving the result that the rank of $[\alpha \mid \beta]$ is at most $2R$. Such a separation can not be achieved for inverse covariance matrices Q_{fp} which change arbitrarily and independently. However, a similar result can be obtained for the case when all the inverse covariance matrices of all points change over time in a "similar way".

Let $\{Q_p \mid p = 1, \ldots, P\}$ be "reference" inverse covariance matrices of all the points (in Section 5.2 we explain how these are chosen). Let $\{C_p \mid p = 1, \ldots, P\}$ be defined such that $C_p C_p^T = Q_p$ (C_p is uniquely defined by the eigenvalue decomposition, same as defined in Section 3). In this section we show that if there exist 2×2 "deformation" matrices $\{A_f \mid f = 1, \ldots, F\}$ such that:

$$\forall p, \forall f : C_{fp} = A_f C_p, \tag{17}$$

then the approach presented in Sections 3 and 4 still applies.

Such 2×2 matrices $\{A_f\}$ can account for global 2D affine deformations in the image plane (rotation, scale, and skew). Note that while C_{fp} is different in every frame f and at every point p, they are not arbitrary. For a given point p, all its 2×2 matrices C_{fp} across *all views* share the same 2×2 reference matrix C_p (which captures the common underlying local image structure and degeneracies in the vicinity of p), while for a given frame (view) f, the matrices C_{fp} of *all points* within that view share the same 2×2 "affine" deformation A_f (which captures the common image distortion induced on the local image structure by the common camera motion). Of course, there are many scenarios in which Eq. (17) will not suffice to model the changes in the inverse covariance matrices. However, the formulation in Eq. (17) does cover a wide range of scenarios, and can be used as a *first-order approximation* to the actual changes in the inverse-covariance matrices in the more

general case. In Section 5.2 we discuss how we choose the matrices $\{C_p\}$ and $\{A_f\}$.

We next show that under the assumptions of Eq. (17), the rank of $[\alpha \mid \beta]$ is still at most $2R$. Let $[\alpha_{fp} \;\; \beta_{fp}]_{1\times 2} = [u_{fp} \; v_{fp}]_{1\times 2} C_{fp_{2\times 2}}$ (this is the same definition as in Section 3, only here we use C_{fp} instead of C_p). Then:

$$[\alpha_{fp} \quad \beta_{fp}] = [u_{fp} \quad v_{fp}] A_f C_p = [\tilde{u}_{fp} \quad \tilde{v}_{fp}] C_p \tag{18}$$

where $[\tilde{u}_{fp} \;\; \tilde{v}_{fp}] = [u_{fp} \;\; v_{fp}] A_f$. Let $\tilde{U}$ be the matrix of all $\tilde{v}_{fp}$ and $\tilde{V}$ be the matrix of all $\tilde{v}_{fp}$. Because C_p is shared by all views of the point p, then (just like in Eq. (9)):

$$[\alpha \mid \beta] = [\tilde{U} \mid \tilde{V}] C$$

where C is the same $2P \times 2P$ matrix defined in Section 3. Therefore the rank of $[\alpha \mid \beta]$ is at most the rank of $[\tilde{U} \mid \tilde{V}]$. We still need to show that the rank of $[\tilde{U} \mid \tilde{V}]$ is at most $2R$ (at most 6). According to the definition of $\tilde{u}_{fp}$ and $\tilde{v}_{fp}$:

$$\begin{bmatrix} \tilde{u}_{fp} \\ \tilde{v}_{fp} \end{bmatrix}_{2\times 1} = A^T_{f_{2\times 2}} \begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix}_{2\times 1} = A^T_{f_{2\times 2}} \begin{bmatrix} m^T_f \\ n^T_f \end{bmatrix}_{2\times R} s_{p_{R\times 1}}. \tag{19}$$

Let

$$A_f = \begin{bmatrix} a_{f1} & a_{f2} \\ a_{f3} & a_{f4} \end{bmatrix}_{2\times 2},$$

then

$$\begin{bmatrix} \tilde{U} \\ \tilde{V} \end{bmatrix}_{2F\times P} = A_{2F\times 2F} \begin{bmatrix} M_u \\ M_v \end{bmatrix}_{2F\times R} S_{R\times P}$$

where:

$$A_{2F\times 2F} = \left[\begin{array}{ccc|ccc} a_{11} & & 0 & a_{13} & & 0 \\ & \ddots & & & \ddots & \\ 0 & & a_{F1} & 0 & & a_{F3} \\ \hline a_{12} & & 0 & a_{14} & & 0 \\ & \ddots & & & \ddots & \\ 0 & & a_{F2} & 0 & & a_{F4} \end{array}\right] \tag{20}$$

This implies that the rank of $\left[\begin{smallmatrix} \tilde{U} \\ \tilde{V} \end{smallmatrix}\right]$ is at most R, and therefore the rank of $[\tilde{U} \mid \tilde{V}]$ is at most $2R$. Therefore, the rank of $[\alpha \mid \beta]$ is at most $2R$ even in the case of "affine-deformed" inverse covariance matrices.

5.1. The Generalized Factorization Algorithm

The factorization algorithm summarized in Section 4.1 can be easily generalized to handle the case of affine-deformed directional uncertainty. Given matrices $\{A_f \mid f = 1, \ldots, F\}$ and $\{C_p \mid p = 1, \ldots, P\}$, such that $C_{fp} = A_f C_p$, then the algorithm is as follows:

Step 0: For each point p and each frame f compute:

$$\begin{bmatrix} \tilde{u}_{fp} \\ \tilde{v}_{fp} \end{bmatrix}_{2\times 1} = A^T_{f_{2\times 2}} \begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix}_{2\times 1} \tag{21}$$

Steps 1 and 2: Use the same algorithm (Steps 1 and 2) as in Section 4.1 (with the matrices $\{C_p \mid p = 1, \ldots, P\}$), but apply it to the matrix $[\tilde{U} \mid \tilde{V}]$ instead of $[U \mid V]$. These two steps yield the matrices $\hat{S}$, $\tilde{M}_V$, and $\tilde{M}_V$, where

$$\begin{bmatrix} \tilde{m}^T_f \\ \tilde{n}^T_f \end{bmatrix}_{2\times R} = A^T_{f_{2\times 2}} \begin{bmatrix} \hat{m}^T_f \\ \hat{n}^T_f \end{bmatrix}_{2\times R}. \tag{22}$$

Step 3: Recover $\hat{M}_U$ and $\hat{M}_V$ by solving for all frames f:

$$\begin{bmatrix} \hat{m}^T_f \\ \hat{n}^T_f \end{bmatrix}_{2\times R} = \left(A^T_f\right)^{-1}_{2\times 2} \begin{bmatrix} \tilde{m}^T_f \\ \tilde{n}^T_f \end{bmatrix}_{2\times R}. \tag{23}$$

5.2. Choosing the Matrices A_f and C_p

Given a collection of inverse covariance matrices, $\{Q_{fp} \mid f = 1, \ldots, F, \; p = 1, \ldots, P\}$, Eq. (17) is not guaranteed to hold. However, we will look for the optimal collection of matrices $\{A_f \mid f = 1, \ldots, F\}$ and $\{C_p \mid p = 1, \ldots, P\}$ such that the error $\sum_{f,p} \|C_{fp} - A_f C_p\|$ is minimized (where $C_{fp} C^T_{fp} = Q_{fp}$). These matrices $\{A_f\}$ and $\{C_p\}$ can then be used in the generalized factorization algorithm of Section 5.1.

Let E be a $2F \times 2P$ matrix which contains all the individual 2×2 matrices $\{C_{fp} \mid f = 1, \ldots, F, \; p = 1, \ldots, P\}$:

$$E = \left[\begin{array}{c|c|c} C_{11} & \cdots & C_{1P} \\ \hline \vdots & \ldots & \vdots \\ \hline C_{F1} & \cdots & C_{FP} \end{array}\right]_{2F\times 2P}. \tag{24}$$

When all the C_{fp}'s do satisfy Eq. (17), then the rank of E is 2, and it can be factored into the following two

rank-2 matrices:

$$E = \begin{bmatrix} \frac{A_1}{\vdots} \\ \hline A_F \end{bmatrix}_{2F\times 2} [C_1 \,|\, \cdots \,|\, C_N]_{2\times 2P}. \qquad (25)$$

When the entries of E (the matrices $\{C_{fp}\}$) do not exactly satisfy Eq. (17), then we recover an *optimal* set of $\{\hat{A}_f\}$ and $\{\hat{C}_p\}$ (and hence $\hat{C}_{fp} = \hat{A}_f \hat{C}_p$), by applying SVD to the $2F \times 2P$ matrix E, and setting to zero all but the two highest singular values. Note that $\{A_f\}$ and $\{C_p\}$ are determined only up to a global 2×2 affine transformation.

The technique described above assumes that the inverse covariance matrix Q_{fp} can be uniquely decomposed in the form $C_{fp}C_{fp}^T$. While this is true for points when the uncertainty is elliptic (i.e., the matrix Q_{fp} has unequal eigenvalues), C_{fp} is not unique when the uncertainty is *isotropic* (i.e., the eigenvalues are equal). This situation requires further exploration, but our current solution is to simply not include the isotropic points in E, and recover the frame-dependent affine transformations A_f purely from the elliptic data. These can then be used to recover the C_p for all data including the isotropic points.

6. Experimental Results

This section describes our experimental evaluation of the covariance weighted factorization algorithm described in this paper. We have applied the algorithm to synthetically generated data with ground truth, as well as to real data.

Using the synthetically generated data we demonstrate two key properties of this algorithm: (i) that its factorization of multi-frame position data into shape and motion is accurate regardless of the degree of ellipticity in the uncertainty of the data—i.e., whether the data consists of "corner-like" points, "line-like" points (i.e., points that lie on linear image structures), or both, and (ii) that in particular, the shape recovery is completely unhampered even when the positional uncertainty of a feature point along one direction is very large (even infinite, such as in the case of pure normal flow).[3] We also contrast its performance with two "bench-marks"—regular SVD (with no uncertainty taken into account; see Section 2.1) and scalar-weighted SVD, which allows a scalar (isotropic) uncertainty (see Section 2.2). We obtain a quantitative comparison of the different methods against ground truth under varying conditions.

We have also applied the algorithm to real data, to show that it can be used to recover *dense* 3D shape from real image sequences.

6.1. Experiments with Synthetic Data

In our experiments, we randomly generated 3D points and affine motion matrices to create ground-truth positional data of multiple features in multiple frames. We then added elliptic Gaussian noise to this data. We varied the ellipticity of the noise to go gradually from being fully circular to highly elliptic, up to the extreme case when the uncertainty at each point is infinite in one of the directions.

Specifically, we varied the shape of the uncertainty ellipse by varying the ellipticity parameter $r_\lambda = \sqrt{\lambda_{\max}/\lambda_{\min}}$ where $\lambda_{\max}$ and $\lambda_{\min}$ are the eigenvalues of the inverse covariance matrix Q (see Section 3). In the first set of experiments, the same value r_λ was used for all the points for a given run of the experiment. The orientation of the ellipse for each point was chosen independently at random. In addition, we included a set of trials in which $\lambda_{\min} = 0$ ($r_\lambda = \infty$) for all the points. This corresponds to the case when only "normal flow" information is available (i.e., infinite uncertainty along the tangential direction).

We ran 20 trials for each setting of the parameter r_λ. For each trial of our experiment, we randomly created a cloud of 100 3D-points, with uniformly distributed coordinates. This defined the ground-truth shape matrix S. We randomly created 20 affine motion matrices, which together define the ground-truth motion matrix M. The affine motion matrices were used to project each of the 100 points into the different views, to generate the noiseless feature positions.

For each trial run of the experiment, for each point in our input dataset, we randomly generated image positional noise with directional uncertainty as specified above. The noise in the direction of $\lambda_{\max}$ (the least uncertain direction) varied between 1% and 2% of the standard deviation of feature positions, whereas the noise in the direction of $\lambda_{\min}$ (the most uncertain direction), varied between 1% and 30% of the standard deviation of feature positions. For each point p in frame f, the generated noise vector ε_{fp} was added to the true position vector $(u_{fp}\ v_{fp})^T$ to create the noisy input matrices U and V.

The noisy input data was then fed to three algorithms: the covariance-weighted factorization algorithm described in this paper, the regular SVD algorithm, and the scalar-weighted SVD algorithm, for which the

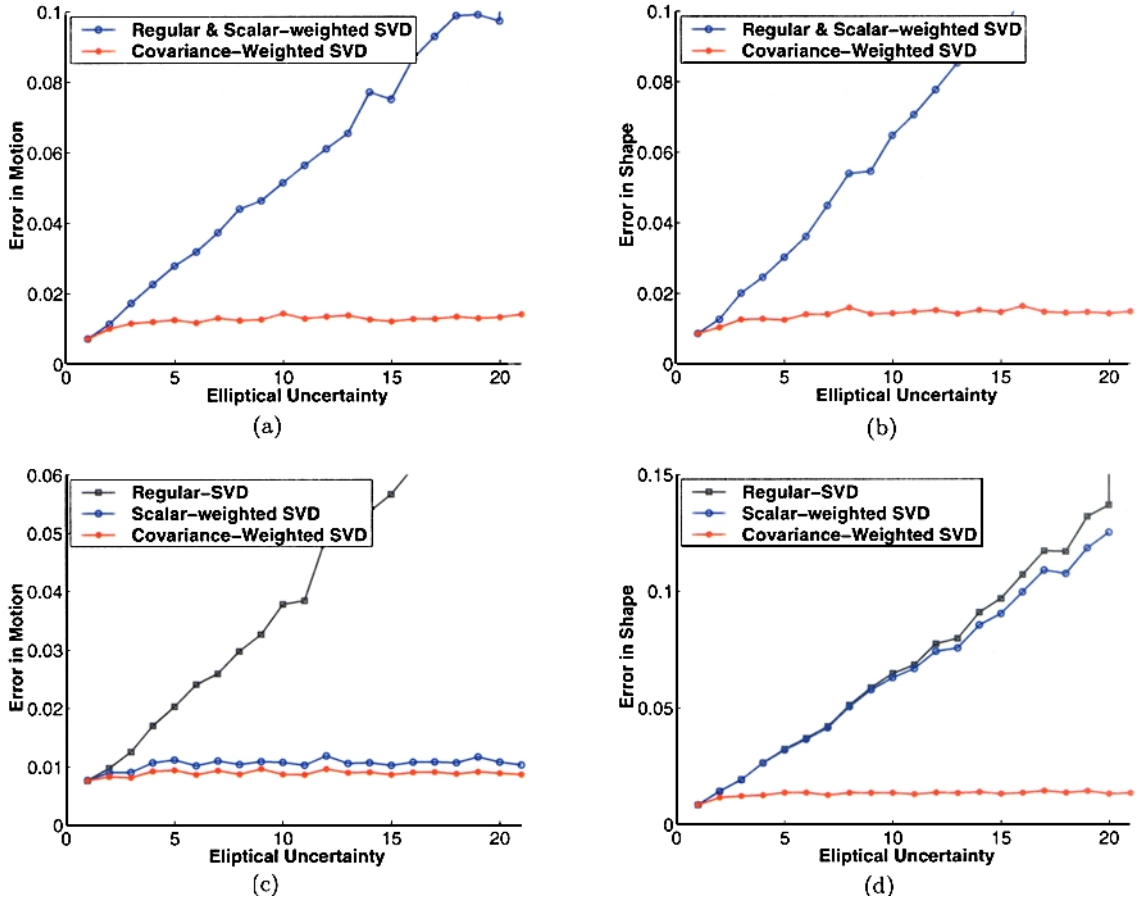

Figure 4. Plots of error in motion and shape w.r.t. ground truth for all three algorithms (Covariance-weighted SVD, scalar-weighted SVD, regular SVD). (a, b) Plots for the case when all points have the same elliptical uncertainty r_λ, which is gradually increased (a = motion error, b = shape error). (c, d) Plots for the case when half of the points have fixed circular uncertainty, and the other half have varying elliptical uncertainty (c = motion error, d = shape error). The displayed shape error in this case is the computed error for the group of elliptic points (the "bad" points).

scalar-weight at each point was chosen to be equal to $\sqrt{\lambda_{\max} * \lambda_{\min}}$ (which is equivalent to taking the determinant of the matrix C_{fp} at each point). Each algorithm outputs a shape matrix $\hat{S}$ and a motion matrix $\hat{M}$. These matrices were then compared against the ground-truth matrices S and M:

$$e_S = \frac{\|S - \hat{S}_N\|}{\|S\|} \qquad e_M = \frac{\|M - \hat{M}_N\|}{\|M\|}$$

where $\hat{S}_N$ and $\hat{M}_N$ are $\hat{S}$ and $\hat{M}$ after transforming them to be in the same coordinate system as S and M. These errors were then averaged over the 20 trials for each setting of the ellipticity parameter r_λ.

Figure 4(a) and 4(b) display the errors in the recovered motion and shape for all three algorithms as a function of the degree of ellipticity in the uncertainty $r_\lambda = \sqrt{\lambda_{\max}/\lambda_{\min}}$. In this particular case, the behavior of regular SVD and scalar-weighted SVD is very similar, because all points within a single trial (for a particular finite r_λ), have the same confidence (i.e., the same scalar-weight r_λ). Note how the error in the recovered shape and motion increases rapidly for the regular SVD and for the scalar-weighted SVD, while the covariance-weighted SVD consistently retains very high accuracy (i.e., very small error) in the recovered shape and motion. The error is kept low and uniform even when the elliptical uncertainty is infinite ($r_\lambda = \infty$; i.e., when only normal-flow information is available). This point is out of the displayed range of this graph, but is visually displayed (for a similar experiment) in Fig. 5.

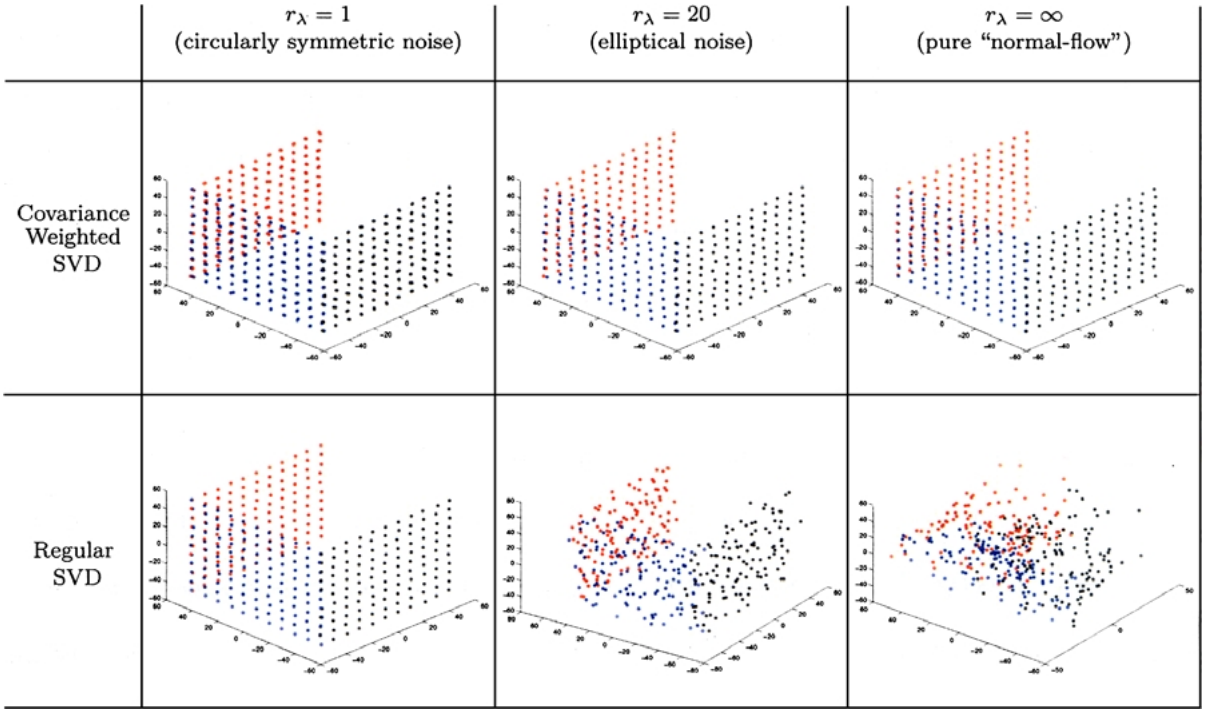

Figure 5. Reconstructed shape of the cube by the Covariance-weighted SVD (top row) vs. the regular SVD (bottom row). For visibility sake, only 3 sides of the cube are displayed. The quality of shape reconstruction of the covariance weighted factorization method does not degrade with the increase in the degree of ellipticity, while in the case of regular SVD, it degrades rapidly.

In the second set of experiments, we divided the input set of points into two equal subsets of points. For one subset, we maintained a circular uncertainty through all the runs (i.e., for those points $r_\lambda = 1$), while for the other subset we gradually varied the shape of the ellipse in the same manner as in the previous experiment above (i.e., for those points r_λ is varied from 1 to ∞). In this case, the quality of the *motion* reconstruction for the scalar-weighted SVD showed comparable results (although still inferior) to the covariance-weighted SVD (see Fig. 4(c)), and significantly better results than the regular SVD. The reason for this behavior is that "good" points (with $r_\lambda = 1$) are weighted highly in the scalar-weighted SVD (as opposed to the regular SVD, where all points are weighted equally). However, while the recovered *shape* of the circularly symmetric ("good") points is quite accurate and degrades gracefully with noise, the error in shape for the "bad" elliptical points (points with large r_λ) increases rapidly with the increase of r_λ, both in the scalar-weighted SVD and in the regular SVD. The error in shape for this group of points (i.e., half of the total number of points) is shown in Fig. 4(d). Note how, in contrast, the covariance-weighted SVD maintains high quality of reconstruction both in the motion and in shape.

In order to *visualize* the results (i.e., visually compare the shape reconstructed by the different algorithms for different types of noise), we repeated these experiments, but this time instead of applying it to a random shape, we applied it to a well defined shape—a cube. We used randomly generated affine motion matrices to determine the positions of 726 cube points in 20 different views, then corrupted them with random noise as before. Sample displays of the reconstructed cube by covariance-weighted algorithm vs. the regular SVD algorithm are shown in Fig. 5 for three interesting cases: case of circular Gaussian noise $r_\lambda = 1$ for all the points (first column of Fig. 5), case of elliptic Gaussian noise with $r_\lambda = 20$ (second column of Fig. 5), and the case of pure "normal flow", when $\lambda_{\min} = 0$ ($r_\lambda = \infty$) (third column of Fig. 5). (For visibility sake, only 3 sides of the cube are displayed). The covariance-weighted SVD (top row) consistently maintains high accuracy of shape recovery, even in the case of pure normal-flow. The shape reconstruction

obtained by regular SVD (bottom row), on the other hand, degrades severely with the increase in the degree of elliptical uncertainty. Scalar-weighted SVD reconstruction was not added here, because when all the points are equally reliable, then scalar-weighted SVD coincides with regular-SVD (see Fig. 4(b)), yet it is not defined for the case of infinite uncertainty (because then all the weights are equal to zero).

6.2. *Experiments with Real Data*

Methods that recover 3D shape and motion using SVD-based factorization usually rely on careful selection of feature points which can be reliably matched across all images. This limits the 3D reconstruction to a small set of points (usually corner points).

One of the benefits of the covariance-weighted factorization presented in this paper is that it can handle data with any level of ellipticity and directionality in their uncertainty, ranging from reliable corner points to points on lines or curves, to points where only normal flow information is available. In other words, given dense flow-fields and the directional uncertainty associated with each pixel (those can be estimated from the local intensity derivatives), a *dense* 3D shape can be recovered using the covariance-weighted factorization.

Such an example is shown in Fig. 6. A scene was imaged by a hand-held camera. The camera moved forward in the first few frames and then moved sideways in the remaining frames (this is the "block" sequence from Kumar et al. (1994)). Because the scene was imaged from a short distance and with a relatively wide field-of-view, the original sequence contained strong projective effects. Therefore, the multi-frame correspondences span a non-linear variety (Anandan and Avidan, 2000), i.e., they do not reside in a low-dimensional linear subspace (as opposed to the case of an affine camera). All factorization methods assume that the correspondences reside in a linear subspace. Therefore, in order to eliminate this non-linearity, the sequence was first aligned with respect to the ground plane (the carpet).

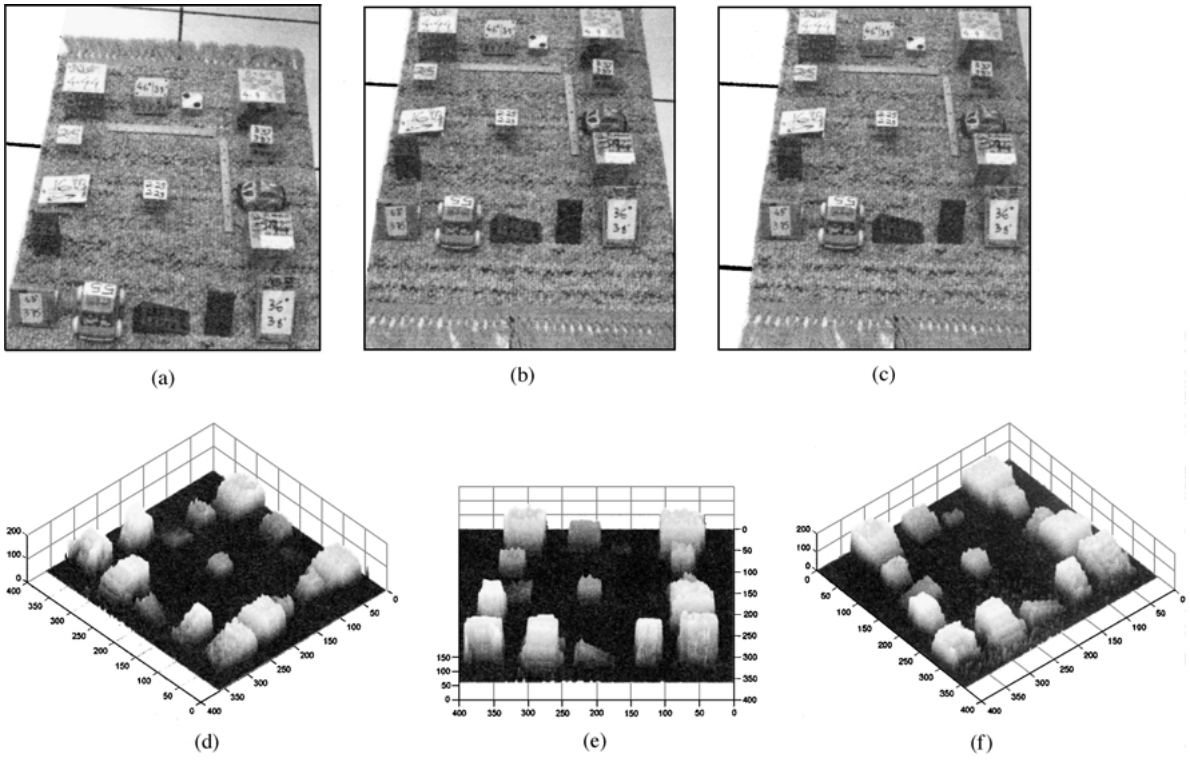

Figure 6. Dense shape recovery from a real sequence using covariance-weighted factorization. (a, b, c) Three out of seven images obtained by a hand-held camera. The camera moved forward in the first few frames and then moved sideways in the remaining frames. (This is the "block" sequence from Kumar et al. (1994)). (d, e, f) The recovered shape relative to the ground plane (see text) displayed from three different viewing angles.

The plane alignment removes most of the projective effects (which are captured by the plane homography), and the residual planar-parallax displacements can be well approximated by a linear subspace with very low dimensionality (Irani, 2002; Oliensis and Genc, 2001). For more details see Appendix B.

We used seven images of the "block sequence" and aligned them with respect to the ground plane (the carpet). We computed a dense parallax displacement field between one of the frames (the "reference frame") and each of the other six frames using a multi-scale (coarse-to-fine) Lucas & Kanade flow algorithm (1981). This algorithm produces dense and noisy correspondences. The algorithm also computes a 2×2 inverse-covariance matrix at *each pixel* based on the local spatial image derivatives. We use these inverse-covariance matrices along with the noisy estimated dense correspondences as input to our covariance-weighted factorization algorithm.[4] The recovered 3D structure is shown in Fig. 6.

Note that unlike most standard factorization methods, which obtain only a "point cloud reconstruction" (i.e., the 3D structure of a sparse collection of highly distinguishable image features), our approach can recover a dense 3D shape. No careful prior feature extraction is necessary. All pixels are treated within a single framework according to their local image structure, regardless of whether they are corner points, points along lines, etc.

7. Conclusion

In this paper we have introduced a new algorithm for performing covariance-weighted factorization of multiframe correspondence data into shape and motion. Unlike the regular SVD algorithms which minimize the Frobenius norm error in the data, or the scalar-weighted SVD which minimizes a scalar-weighted version of that norm, our algorithm minimizes the *covariance weighted* error (or the Mahalanobis distance). This is the proper measure to minimize when the uncertainty in feature position is directional. Our algorithm transforms the raw input data into a covariance-weighted data space, and applies SVD in this transformed data space, where the Frobenius norm now minimizes a meaningful objective function. This SVD step projects the covariance-weighted data to a $2R$-dimensional subspace. We complete the process with an additional sub-optimal linear estimation step to recover the rank R shape and motion estimates.

A fundamental advantage of our algorithm is that it can handle input data with any level of ellipticity in the directional uncertainty—i.e., from purely circular uncertainty to highly elliptical uncertainty, even including the case of points along lines where the uncertainty along the line direction is infinite. It can also simultaneously use data which contains points with different levels of directional uncertainty. We empirically show that our algorithm recovers shape and motion accurately, even when the more conventional SVD algorithms perform poorly. However, our algorithm cannot handle *arbitrary* changes in the uncertainty of a single feature over multiple frames (views). It can only account for frame dependent 2D affine deformations in the covariance matrices.

Appendix A: Recovering S and M

In this appendix we explain in detail how to obtain the decomposition of DG into the matrix structure described in Eq. (16), and thus solve for S and D.

Eq. (16) states that:

$$DG = \left[\begin{array}{c|c} S & 0 \\ \hline 0 & S \end{array}\right] C$$

Let the four $P \times P$ quadrants of the $2P \times 2P$ matrix C be denoted by the four diagonal matrices C_1, C_2, C_3, C_4:

$$C = \left[\begin{array}{c|c} C_1 & C_2 \\ \hline C_3 & C_4 \end{array}\right]_{2P \times 2P}.$$

Similarly, let $D = \left[\begin{array}{c|c} D_1 & D_2 \\ \hline D_3 & D_4 \end{array}\right]_{2R \times 2R}$ and $G = \left[\begin{array}{c|c} G_1 & G_2 \\ \hline G_3 & G_4 \end{array}\right]_{2R \times 2P}$. Then we get the following four matrix equations:

$$\begin{cases} D_1 G_1 + D_2 G_3 = S C_1 \\ D_1 G_2 + D_2 G_4 = S C_2 \\ D_3 G_1 + D_4 G_3 = S C_3 \\ D_3 G_2 + D_4 G_4 = S C_4. \end{cases} \tag{26}$$

These equations are *linear* in the unknown matrices D_1, D_2, D_3, D_4 and S. This set of equations can, in principle, be solved directly as a huge linear set of equations with $4R^2 + RP$ unknowns. But there are relatively few *global unknowns* (the $4R^2$ elements of D) and a huge number of independent *local unknowns* (the RP unknown elements of the matrix S, which are the shape components of the individual P image points. This may accumulate to hundreds of thousands of unknowns).

Instead of directly solving this huge system of linear equations, we can solve this system much more efficiently by doing the following: Using the fact that C_1, C_2, C_3, C_4 are diagonal (hence commute with each other), we can eliminate S and obtain the following three linearly independent matrix equations in the four unknown $R \times R$ matrices D_1, D_2, D_3, D_4.

$$\begin{aligned} &D_1(G_1C_2 - G_2C_1) + D_2(G_3C_2 - G_4C_1) = 0 \\ &D_3(G_1C_4 - G_2C_3) + D_4(G_3C_4 - G_4C_3) = 0 \\ &D_1(G_1C_3) + D_2(G_3C_3) - D_3(G_1C_1) - D_4(G_3C_1) = 0. \end{aligned} \tag{27}$$

This homogeneous set of equations is highly overdetermined ($3RP$ equations in $4R^2$ unknowns, where $R \ll P$). It can be linearly solved to obtain the unknown D_i's. Note that the choice of D_i's is not unique. This is because S is not unique, and can only be determined up to an $R \times R$ affine transformation. To solve for D, the R eigenvectors of the R smallest eigenvalues of the normal equations associated with the homogeneous system in Eq. (27) were used.

Now that D has been recovered, we proceed to estimate $\hat{M}$ and $\hat{S}$. Recovering the motion is straightforward: $[\hat{M}_U \mid \hat{M}_V] = HD^{-1}$ where H is defined in Eq. (15). To recover the shape $\hat{S}$, we can proceed in two ways: We can either linearly solve Eq. (16), or else linearly solve Eq. (14). Equation (14) goes back to the cleaned up input measurement data with the appropriate covariance-weighting, and is therefore preferable to Eq. (16), which uses intermediate results. Note however, that since the columns of S are independent of each other, the constraint from Eq. (14) can be used to solve for the values of S on a point-by-point basis using only local information, as shape is a local property. So once again, we resort to a very small set of equations for recovering each component of S.

Appendix B: Factorization of Planar Parallax Displacements

In the real experiment of Fig. 6 in Section 6.2 we applied the covariance-weighted factorization to the residual planar-parallax displacements after plane alignment. To make the paper self contained, we briefly rederive here the linear subspace approximation of planar-parallax displacements. For more details on the "Plane + Parallax" decomposition see Irani et al. (1998), Irani and Anandan (1996), Irani et al. (1999), Kumar et al. (1994), Sawhney (1994), Shashua and Navab (1994), Irani et al. (1997), Criminisi et al. (1998) and Triggs (2000). For more details on the linear subspace approximation of planar-parallax displacements see Irani (2002) and Oliensis and Genc (2001).

Let Π be an arbitrary planar surface in the scene, which is visible in all frames. After plane alignment the residual planar-parallax displacements between the reference frame and any other plane-aligned frame f ($f = 1, \ldots, F$) are (see Kumar et al. (1994) and Irani et al. (1999)):

$$\begin{bmatrix} \mu_{fp} \\ \nu_{fp} \end{bmatrix} = -\frac{\gamma_p}{1 + \gamma_p \epsilon_{Z_f}} \left(\epsilon_{Z_f} \begin{bmatrix} u_p \\ v_p \end{bmatrix} - \begin{bmatrix} \epsilon_{U_f} \\ \epsilon_{V_f} \end{bmatrix} \right) \tag{28}$$

where (u_p, v_p) are the coordinates of a pixel in the reference frame, $\gamma_p = \frac{H_p}{Z_p}$ represents its 3D structure, H_p is the perpendicular distance (or "height") of the point i from the reference plane Π, and Z_p is its depth with respect to the reference camera. $(\epsilon_{U_f}, \epsilon_{V_f}, \epsilon_{Z_f})$ denotes the camera translation up to a (unknown) projective transformation (i.e., the scaled epipole in projective coordinates). The above formulation is true both for the calibrated case as well as for the uncalibrated case. The residual image motion of Eq. (28) is due only to the *translational* part of the camera motion, and to the *deviations* of the scene structure from the planar surface. All effects of rotations and of changes in calibration within the sequence are captured by the homography (e.g., see Irani and Anandan, 1996; Irani et al., 1999; Triggs, 2000). The elimination of the homography (via image warping) reduces the problem from the general uncalibrated unconstrained case to the simpler case of pure translation with fixed (unknown) calibration.

Although the original sequence may contain large rotations and strong projective effects, resulting in a non-linear variety, this non-linearity is mostly captured by the plane homography. The residual planar-parallax displacements can be approximated well by a linear subspace with very low dimensionality.

When the following relation holds:

$$\gamma_p \epsilon_{Z_f} \ll 1 \tag{29}$$

then Eq. (28) reduces to:

$$\begin{bmatrix} \mu_{fp} \\ \nu_{fp} \end{bmatrix} = -\gamma_p \left(\epsilon_{Z_f} \begin{bmatrix} u_p \\ v_p \end{bmatrix} - \begin{bmatrix} \epsilon_{U_f} \\ \epsilon_{V_f} \end{bmatrix} \right), \tag{30}$$

which is bilinear in the motion and shape. The condition in Eq. (29) ($\gamma_p \epsilon_{Z_f} = \frac{H_p}{Z_p} \epsilon_{Z_f} \ll 1$), which gave rise to

the bilinear form of Eq. (30), is satisfied if at least one of the following two conditions holds:

Either: (i) $H_p \ll Z_p$, namely, *the scene is shallow* (i.e., the distance H_p of the scene point from the reference plane Π is much smaller than its distance Z_p from the camera. This condition is usually satisfied if the plane lies within the scene, and the camera is not too close to it),

Or: (ii) $\epsilon_{Z_f} \ll Z_p$, namely, *the forward translational motion of the camera is small relative to its distance from the scene*, which is often the case within short temporal segments of real video sequences.

We next show that the planar-parallax displacements of Eq. (30) span a low-dimensional linear subspace (of rank at most 3). Equation (30) can be rewritten as a *bilinear* product:

$$\begin{bmatrix} \mu_{fp} \\ \nu_{fp} \end{bmatrix}_{2\times 1} = \begin{bmatrix} m_f \\ n_f \end{bmatrix}_{2\times 3} s_p{}_{3\times 1}$$

where

$$s_p = [\gamma_p \quad -\gamma_p u_p \quad -\gamma_p v_p]^T$$

is a *point-dependent* column vector ($p = 1, \ldots, P$), and

$$m_f = \begin{bmatrix} \epsilon_{U_f} & \epsilon_{Z_f} & 0 \end{bmatrix}$$
$$n_f = \begin{bmatrix} \epsilon_{V_f} & 0 & \epsilon_{Z_f} \end{bmatrix}$$

are *frame-dependent* row vectors ($f = 1, \ldots, F$). Therefore, all planar parallax displacements of all points across all (plane-aligned) frames can be expressed as a bilinear product of matrices:

$$\begin{bmatrix} \frac{\mu}{\nu} \end{bmatrix}_{2F\times P} = \begin{bmatrix} \frac{M_U}{M_V} \end{bmatrix}_{2F\times 3} S_{3\times P} \qquad (31)$$

Equation (31) implies that $\text{rank}\left(\begin{bmatrix} \frac{\mu}{\nu} \end{bmatrix}\right) \leq 3$. Note that this rank constraint was derived for point *displacements* (as opposed to point positions).

A similar approach to factorization of translational motion after cancelling the rotational component can be found in Oliensis (1999) and Oliensis and Genc (2001). A different approach to factorization of planar parallax displacements can be found in Triggs (2000). The latter approach is a rank 1 factorization and makes no approximations to the parallax displacements. However, it assumes prior computation of the projective depths (scale factors) at each point.

Acknowledgments

The authors would like to thank Moshe Machline for his help in the real-data experiments. The work of Michal Irani was supported by the Israel Science Foundation (Grant no. 153/99) and by the Israeli Ministry of Science (Grant no. 1229).

Notes

1. When directional uncertainty is used, the centroids $\{\bar{u}_f\}$ and $\{\bar{v}_f\}$ defined in Section 2.1, are the covariance-weighted means in frame f: $\bar{u}_f = (\sum_p Q_{fp})^{-1} \sum_p (Q_{fp} u_{fp})$ and $\bar{v}_f = (\sum_p Q_{fp})^{-1} \sum_p (Q_{fp} v_{fp})$. Note that the centering the data in this fashion adds a weak correlation between all the data points. This is true for all factorization algorithms that employ this strategy, including ours. However, we ignore this issue in this paper, since our main focus is the extension of the standard SVD algorithms to handle directional uncertainty.
2. This is analogous to the situation described by Tomasi and Kanade (1992), where the orthogonality constraint on the motion matrix is imposed in a suboptimal second step following the optimal SVD-based subspace projection step.
3. The fact that we can recover structure and motion purely from normal flow may be a bit counter intuitive. However, it is evident that the motion for any pair of frames implicitly provides an epipolar line constraint, while the normal flow for a point provides another line constraint. The intersection of these two lines uniquely defines the position of the point and its corresponding shape. However, the epipolar line is unknown, and in two views there are not enough constraints to uniquely recover the shape and the motion from normal flow. When three or more views are available and the camera centers are not colinear, there is an adequate set of normal flow constraints to uniquely determine all the (epipolar) lines (and the motion of the cameras) and the shape of all points. This has been previously demonstrated for iterative techniques in Hanna and Okamoto (1993), Stein and Shashua (2000), and Irani et al. (1999). In particular, Stein and Shashua (2000) also prove that under general conditions, for the case of three frames the structure and motion can be uniquely recovered from normal flow. The method proposed in our paper also combines normal-flow constraints with implicit epipolar constraints (captured by the motion matrix M) to provide dense structure and motion, but in a non-iterative way using global SVD-based minimization.
4. The covariance weighted factorization algorithm can be equally applied to pixel displacements as to point positions, since both reside in low-dimensional linear subspaces (see Appendix B).

References

Aguiar, P.M.Q. and Moura, J.M.F. 1999. Factorization as a rank 1 problem. *IEEE Computer Vision and Pattern Recognition Conference* 9, A:178–184.

Anandan, P. 1989. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310.

Anandan, P. and Avidan, S. 2000. Integrating local affine into global perspective images in the joint image space. In *European Conference on Computer Vision*, Dublin, pp. 907–921.

Ben-Ezra, M., Peleg, S., and Werman, M. 2000. Real-time motion analysis with linear programming. *International Journal of Computer Vision*, 78:32–52.

Criminisi, A., Reid, I., and Zisserman, A. 1998. Duality, rigidity and planar parallax. In *European Conference on Computer Vision*, Freiburg.

Hanna, K. and Okamoto, N.E. 1993. Combining stereo and motion for direct estimation of scene structure. In *International Conference on Computer Vision*, Berlin, Germany, pp. 357–365.

Irani, M. 2002. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194 (shorter version appeared in *International Conference on Computer Vision*, 1999, pp. 626–633).

Irani, M. and Anandan, P. 1996. Parallax geometry of pairs of points for 3d scene analysis. In *European Conference on Computer Vision*, Cambridge, UK, pp. 17–30.

Irani, M. and Anandan, P. 2000. Factorization with uncertainty. In *European Conference on Computer Vision*, Dublin, pp. 539–553.

Irani, M., Anandan, P., and Cohen, M. 1999. Direct recovery of planar-parallax from multiple frames. In *Vision Algorithms: Theory and Practice Workshop*, Corfu.

Irani, M., Anandan, P., and Weinshall, D. 1998. From reference frames to reference planes: Multi-view parallax geometry and applications. In *European Conference on Computer Vision*, Freiburg.

Irani, M., Rousso, B., and Peleg, S. 1997. Recovery of ego-motion using region alignment. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(3):268–272.

Kanatani, K. 1996. *Statistical Optimization for Geometric Computation: Theory and Practice*. North-Holland: Amsterdam, The Netherlands.

Kumar, R., Anandan, P., and Hanna, K. 1994. Direct recovery of shape from multiple views: A parallax based approach. In *Proc. 12th International Conference on Pattern Recognition*, Elsevier Science: Amsterdam, The Netherlands, pp. 685–688.

Leedan, Y. and Meer, P. 2000. Heteroscedastic regression in computer vision: Problems with bilinear constraint. *International Journal on Computer Vision*, 37(2):127–150.

Lucas, B.D. and Kanade, T. 1981. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pp. 121–130.

Matei, B. and Meer, P. 2000. A general method for errors-in-variables problems in computer vision. *IEEE Computer Vision and Pattern Recognition Conference*, 2:18–25.

Morris, D. and Kanade, T. 1998. A unified factorization algorithm for points, line segments and planes with uncertain models. *International Conference on Computer Vision*, pp. 696–702.

Morris, D., Kanatani, K., and Kanade, T. 1999. Uncertainty modeling for optimal structure from motion. In *Vision Algorithms: Theory and Practice Workshop*, Corfu, pp. 33–40.

Oliensis, J. 1999. A multi-frame structure-from-motion algorithm under perspective projection. *International Journal of Computer Vision*, 34(2/3):163–192.

Oliensis, J. and Genc, Y. 2001. Fast and accurate algorithms for projective multi-image structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):546–559.

Poelman, C.J. and Kanade, T. 1997. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:206–218.

Quan, L. and Kanade, T. 1996. A factorization method for affine structure from line correspondences. *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 803–808.

Sawhney, H. 1994. 3D geometry from planar parallax. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Shapiro, L.S. 1995. *Affine Analysis of Image Sequences*. Cambridge University Press: Cambridge, UK.

Shashua, A. and Navab, N. 1994. Relative affine structure: Theory and application to 3d reconstruction from perspective views. In *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, pp. 483–489.

Stein, G.P. and Shashua, A. 2000. Model-based brightness constraints: On direct estimation of structure and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):992–1015.

Sturm, P. and Triggs, B. 1996. A factorization based algorithm for multi-image projective structure and motion. *European Conference on Computer Vision*, 2:709–720.

Tomasi, C. and Kanade, T. 1992. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154.

Triggs, W. 2000. Plane + parallax, tensors, and factorization. In *European Conference on Computer Vision*, Dublin, pp. 522–538.

Van Huffel, S. and Vandewalle, J. 1991. *The Total Least Squares Problem*. SIAM: Philadelphia, PA.