Factorization with Uncertainty

Michal Irani[1] and P. Anandan[2]

[1] Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science, Rehovot 76100, Israel

[2] Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA

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. We empirically show that our new algorithm gives 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.

1 Introduction

Factorization is often used for recovering 3D shape and motion from feature correspondences across multiple frames [8,4,5,6,7]. Singular Value Decomposition (SVD) directly obtains the global minimum of the squared-error between the noisy data and the model. This is in contrast to iterative non-linear optimization methods which may converge to a local minimum. However, SVD requires that the noise in the x and y positions of features are uncorrelated and have identical distributions. But, it is rare that the 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 the x and in the y components in a correlated fashion. This dependency can be modeled by *directional* uncertainty (which varies from point to point, as is shown in Fig. 1).

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 539–553, 2000.

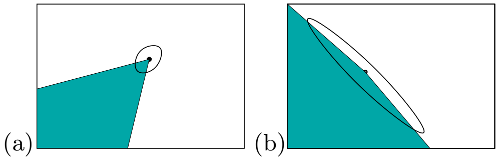

Fig. 1. *Directional uncertainty indicated by ellipse. (a) Uncertainty of a sharp corner point. The uncertainty in all directions is small, since the underlying intensity structure shows variation in multiple directions. (b) Uncertainty of a point on a flat curve, almost a straight line. Note that the 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.*

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 [1]. 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). This issue was either ignored by researchers [8,4,1, 7], or else was addressed using other minimization approaches [3,5]. Morris and Kanade [3] 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 [5] 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 [2], 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.)

We apply SVD factorization to the covariance-weighted data to obtain a global optimum. This minimizes the *Mahalanobis distance* in the original data space. However, the covariance-weighted data space has double the rank of the original data space. To obtain the required additional rank-halving, we use a least-squares minimization step within the double-rank subspace.

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 [2] 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 extends the use of 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 extension does not allow 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.

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 center of mass: (u'_{fp}, v'_{fp}) is replaced by $(u_{fp}, v_{fp}) = (u'_{fp} - \overline{u}_f, v'_{fp} - \overline{v}_f)$ for all f and p, where $\overline{u}_f$ and $\overline{v}_f$ are the centroids of point positions in each frame: $\overline{u}_f = \frac{1}{P}\sum_p u'_{fp}, \quad \overline{v}_f = \frac{1}{P}\sum_p v'_{fp}$.

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

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

It was shown [8,4,6] 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 $W = \begin{bmatrix} U \\ V \end{bmatrix}_{2F \times P}$ 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:

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

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.

When there are errors in the measurement matrix W, then each position $(u_{fp} \quad v_{fp})^T$ has a 2D noise vector associated with it

$$\mathcal{E}_{fp} = \begin{bmatrix} u_{fp} - m_f^T s_p \\ v_{fp} - n_f^T s_p \end{bmatrix}_{2\times 1}.$$

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

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

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

2.2 Scalar Uncertainty

The model in Section 2.1 (as well as in [8]) 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 $\{\sigma_p^2 \mid p = 1 \cdots P\}$, 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 $\sigma^{-1} = \mathrm{diag}(\sigma_1^{-1}, ..., \sigma_P^{-1})$, will minimize the correct error function:

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

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

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. Tracking algorithms introduce non-uniform correlated error in the tracked positions of points which depends 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. The covariance matrix determines an ellipse whose major and minor axes indicate the directional uncertainty in the location $(u_{fp} \quad v_{fp})^T$ of a point p in frame f (see Fig. 1, as well as [3] for some examples).

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:

$$\begin{aligned}\mathrm{Err}(M,S) &= \textstyle\sum_{f,p}(\mathcal{E}_{fp}^T Q_{fp} \mathcal{E}_{fp}) \\ &= \sum_{f,p} \left(\left[(u_{fp} - m_f^T s_p) \quad (v_{fp} - n_f^T s_p) \right] Q_{fp} \begin{bmatrix} u_{fp} - m_f^T s_p \\ v_{fp} - n_f^T s_p \end{bmatrix} \right) . \end{aligned} \tag{1}$$

Eq. (1) implies that in the case of directional uncertainty, the metric that we want to use in the minimization is the *Mahalanobis distance*, and not the *Frobenius (least-squares) norm*, which is the distance minimized by the SVD process.

Morris and Kanade [3] 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[1]. However, unlike [3], 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 Raw-Data Space to 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. (1), while still retaining the benefits of SVD minimization. In particular, we will show that *minimizing the Frobenius Norm in the new data space* (e.g., via SVD) *is equivalent to minimizing the Mahalanobis distance in the raw-data space*. 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 [2], who used it for applying confidence-weighted subspace projection to spatio-temporal brightness derivatives for computing optical flow across multiple frames.

[1] When directional uncertainty is used, the centroids $\{\overline{u}_f\}$ and $\{\overline{v}_f\}$ defined in Section 2.1, are the covariance-weighted means over all points of $\{u_{fp}\}$ and $\{v_{fp}\}$ in frame f.

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-*in*dependent: $\forall f\; Q_{fp} \equiv Q_p$. 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 $\Omega_{2\times2}$ is a real orthonormal matrix, and $\Lambda_{2\times2} = \text{diag}(\lambda_{max}, \lambda_{min})$. Let $C_p = \Omega \Lambda^{\frac{1}{2}}$ and $[\alpha_{fp} \quad \beta_{fp}]_{1\times2} = [u_{fp} \quad v_{fp}]_{1\times2} {C_p}_{2\times2}$. Therefore, α_{fp} is the component of $[u_{fp} \quad v_{fp}]$ 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). 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.

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

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

then

$$[\alpha_p \quad \beta_p]_{F\times2} = [u_p \quad v_p]_{F\times2} \; {C_p}_{2\times2} \,. \tag{2}$$

Let α and β be two $F \times P$ matrices:

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

then, according to Eq. (2):

$$[\alpha \mid \beta]_{F\times2P} = [U \mid V]_{F\times2P} \; C_{2P\times2P} \tag{3}$$

where C is a $2P$ x $2P$ matrix, constructed from all 2 x 2 matrices $C_p = \begin{bmatrix} c_{p_1} & c_{p_2} \\ c_{p_3} & c_{p_4} \end{bmatrix}$ $(p = 1 \cdots P)$, as follows:

$$C = \left[\begin{array}{cc|cc} c_{11} & 0 & c_{12} & 0 \\ & \ddots & & \ddots \\ 0 & c_{P1} & 0 & c_{P2} \\ \hline c_{13} & 0 & c_{14} & 0 \\ & \ddots & & \ddots \\ 0 & c_{P3} & 0 & c_{P4} \end{array}\right]_{2P\times2P} .$$

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 *un*correlated. This will be shown and used below.

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 [2]. 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. (3), the rank of $[\alpha \mid \beta]$ is also at most $2R$.

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

$$\text{Err}\Big(\left\{(\widehat{u}_{fp} \quad \widehat{v}_{fp})^T\right\}\Big) = \sum_{f,p} \left[(u_{fp} - \widehat{u}_{fp}) \quad (v_{fp} - \widehat{v}_{fp})\right] Q_{fp} \begin{bmatrix} u_{fp} - \widehat{u}_{fp} \\ v_{fp} - \widehat{v}_{fp} \end{bmatrix} . \quad (4)$$

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

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

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

Minimizing the *Mahalanobis distance* of Eq. (4) is equivalent to finding the rank-$2R$ matrix $[\widehat{\alpha} \mid \widehat{\beta}]$ closest to $[\alpha \mid \beta]$ in the *Frobenius norm.*

This minimization can be done by applying SVD subspace projection to the matrix $[\alpha \mid \beta]$, to obtain the optimal $[\widehat{\alpha} \mid \widehat{\beta}]$. This is done by applying SVD to the known $[\alpha \mid \beta]$ matrix, and setting to zero all but the highest $2R$ singular values. However, note that although optimal, $[\widehat{\alpha} \mid \widehat{\beta}] = [\widehat{U} \mid \widehat{V}]C$ is in general a rank-$2R$ matrix, and does not guaranty that $\widehat{W} = \begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix}$ is a rank-R matrix. In Section 4 we show how we complete the process by making the transition from the optimal rank-$2R$ matrix $[\widehat{\alpha} \mid \widehat{\beta}]$ to the rank-R solution $\widehat{W} = \begin{bmatrix} \widehat{M_U} \\ \widehat{M_V} \end{bmatrix} \widehat{S}$.

4 Factoring Shape and Motion

The process of finding the rank-$2R$ $[\widehat{\alpha} \mid \widehat{\beta}]$, as outlined in Section 3, does not yet guarantee that the corresponding $\widehat{U}$ and $\widehat{V}$ can be decomposed into rank-R matrices as follows: $\begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix} = \widehat{M}\widehat{S} = \begin{bmatrix} \widehat{M_U} \\ \widehat{M_V} \end{bmatrix} \widehat{S}$. In this section we complete the process and recover $\widehat{M}$ and $\widehat{S}$ by enforcing this matrix constraint on $\widehat{U}$ and $\widehat{V}$.

Note that if C were an invertible matrix, then we could have recovered $[\widehat{U} \mid \widehat{V}] = [\widehat{\alpha} \mid \widehat{\beta}]C^{-1}$, and then proceeded with applying standard SVD to $\begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix}$ to impose the rank-R constraint and recover $\widehat{M}$ and $\widehat{S}$. However, C is in general not invertible (e.g., because of points with high aperture problem). Imposing the rank-R constraint on $\widehat{U} = \widehat{M_U}\widehat{S}$ and $\widehat{V} = \widehat{M_V}\widehat{S}$ must therefore be done in the $[\widehat{\alpha} \mid \widehat{\beta}]$ space (i.e., without inverting C):

$$[\widehat{\alpha} \mid \widehat{\beta}]_{F \times 2P} = [\widehat{M_U}\widehat{S} \mid \widehat{M_V}\widehat{S}]C = [\widehat{M_U} \mid \widehat{M_V}]_{F \times 2R} \left[\begin{array}{c|c} \widehat{S} & 0 \\ \hline 0 & \widehat{S} \end{array}\right]_{2R \times 2P} C_{2P \times 2P} \ . \quad (5)$$

Not every decomposition of $[\widehat{\alpha} \mid \widehat{\beta}]$ has the matrix form $\left[\begin{array}{c|c} S & 0 \\ \hline 0 & S \end{array}\right]$. However, if we are able to decompose $[\widehat{\alpha} \mid \widehat{\beta}]$ into the matrix form of Eq. (5), then the resulting $\widehat{M} = \begin{bmatrix} \widehat{M_U} \\ \widehat{M_V} \end{bmatrix}$ and $\widehat{S}$ (which can be determined only up to an affine transformation) will provide the desired rank-R solution.

Because $[\widehat{\alpha} \mid \widehat{\beta}]_{F \times 2P}$ 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:

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

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

$$DG = \left[\begin{array}{c|c} S & 0 \\ \hline 0 & S \end{array}\right] C \quad (6)$$

where S is an arbitrary $R \times P$ matrix. This is a linear system of equations in the unknown components of S and D. We therefore linearly solve for S and D, from which the desired solution is obtained by: $\widehat{S} := S$ and $[\widehat{M_U} \mid \widehat{M_V}] := HD^{-1}$.

4.1 Summary of the Algorithm

We summarize the steps of the algorithm:

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) $[\widehat{\alpha} \mid \widehat{\beta}]$ (for an affine camera $2R \leq 6$). This step is guaranteed to obtain the closest $2R$-dimensional subspace because of the global optimum property of SVD.

Step 2: Further enforce the rank-R solution by enforcing that $\begin{bmatrix} \widehat{U} \\ \widehat{V} \end{bmatrix} = \begin{bmatrix} \widehat{M_U} \\ \widehat{M_V} \end{bmatrix} \widehat{S}$. This additional subspace projection is achieved within the $[\alpha \mid \beta]$ space, and is obtained with simple least squares minimization applied to the *linear* set of equations (6).

Note that the Rank-R subspace obtained by the second step is contained inside the Rank-$2R$ subspace obtained in the first step. We cannot prove that the *optimal* Rank-R solution is guaranteed to lie within this Rank-$2R$ subspace. However, the bulk of the optimization task is performed in Step 1, which takes the noisy high-dimensional data into the Rank-$2R$ subspace in an *optimal* fashion. Moreover, both steps of our algorithm are linear. 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. (3), 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 $\{Q_p \mid p = 1 \cdots P\}$ be "reference" inverse covariance matrices of all the points (in Section 5.2 we explain how these are chosen). Let $\{C_p \mid p = 1 \cdots P\}$ 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:

$$\forall p, \forall f\colon\ C_{fp} = A_f C_p\ , \tag{7}$$

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. (7) will not suffice to model the changes in the inverse covariance matrices. However, the formulation in Eq. (7) 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. (7), the rank of $[\boldsymbol{\alpha} \mid \boldsymbol{\beta}]$ is still at most $2R$. Let $[\alpha_{fp} \quad \beta_{fp}]_{1\times2} = [u_{fp} \quad v_{fp}]_{1\times2} {C_{fp}}_{2\times2}$ (this is the same definition as in Section 3, only here we use C_{fp} instead of C_p). Then:

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

where $[\widetilde{u}_{fp} \quad \widetilde{v}_{fp}] = [u_{fp} \quad v_{fp}]A_f$. Let $\widetilde{U}$ be the matrix of all $\widetilde{u}_{fp}$ and $\widetilde{V}$ be the matrix of all $\widetilde{v}_{fp}$. Because C_p is shared by all views of the point p, then (just like in Eq. (3)):

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

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

$$\begin{bmatrix}\widetilde{u}_{fp}\\ \widetilde{v}_{fp}\end{bmatrix}_{2\times1} = {A_f}^T_{2\times2}\begin{bmatrix}u_{fp}\\ v_{fp}\end{bmatrix}_{2\times1} = {A_f}^T_{2\times2}\begin{bmatrix}m_f^T\\ n_f^T\end{bmatrix}_{2\times R} {s_p}_{R\times1}\ .$$

Let $A_f = \begin{bmatrix}a_{f1} & a_{f2}\\ a_{f3} & a_{f4}\end{bmatrix}_{2\times2}$, then

$$\begin{bmatrix}\widetilde{U}\\ \widetilde{V}\end{bmatrix}_{2F\times P} = \left[\begin{array}{cc|cc} a_{11} & 0 & a_{13} & 0\\ & \ddots & & \ddots\\ 0 & a_{F1} & 0 & a_{F3}\\ \hline a_{12} & 0 & a_{14} & 0\\ & \ddots & & \ddots\\ 0 & a_{F2} & 0 & a_{F4}\end{array}\right]_{2F\times2F} \begin{bmatrix}M_u\\ M_v\end{bmatrix}_{2F\times R} S_{R\times P}\ .$$

This implies that the rank of $\begin{bmatrix}\widetilde{U}\\ \widetilde{V}\end{bmatrix}$ is at most R, and therefore the rank of $[\widetilde{U} \mid \widetilde{V}]$ is at most $2R$. Therefore, the rank of $[\boldsymbol{\alpha} \mid \boldsymbol{\beta}]$ 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 $\{A_f \mid f = 1 \cdots F\}$ and $\{C_p \mid p = 1, \cdots P\}$, 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:

$$\begin{bmatrix} \widetilde{u}_{fp} \\ \widetilde{v}_{fp} \end{bmatrix}_{2\times 1} = {A_f}^T_{2\times 2} \begin{bmatrix} u_{fp} \\ v_{fp} \end{bmatrix}_{2\times 1} .$$

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

$$\begin{bmatrix} \widetilde{m}_f^T \\ \\ \widetilde{n}_f^T \end{bmatrix}_{2\times R} = {A_f}^T_{2\times 2} \begin{bmatrix} \widehat{m}_f^T \\ \\ \widehat{n}_f^T \end{bmatrix}_{2\times R} .$$

Step 3: Recover $\widehat{M_U}$ and $\widehat{M_V}$ by solving for all frames f:

$$\begin{bmatrix} \widehat{m}_f^T \\ \\ \widehat{n}_f^T \end{bmatrix}_{2\times R} = (A_f^T)^{-1}_{2\times 2} \begin{bmatrix} \widetilde{m}_f^T \\ \\ \widetilde{n}_f^T \end{bmatrix}_{2\times R} .$$

5.2 Choosing the Matrices A_f and C_p

Given a collection of inverse covariance matrices, $\{Q_{fp} \mid f = 1 \cdots F,\ p = 1 \cdots P\}$, Eq. (7) is not guaranteed to hold. However, we will look for the optimal collection of matrices $\{A_f \mid f = 1 \cdots F\}$ and $\{C_p \mid p = 1 \cdots P\}$ such that the error $\sum_{f,p} \|C_{fp} - A_f C_p\|$ is minimized (where $C_{fp} C_{fp}^T = Q_{fp}$). 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 $\{C_{fp} \mid f = 1 \cdots F,\ p = 1 \cdots P\}$:

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

When all the C_{fp}'s do satisfy Eq. (7), then the rank of E is 2, and it can be factored into the following two rank-2 matrices:

$$E = \left[\begin{array}{c} A_1 \\ \hline \vdots \\ \hline A_F \end{array}\right]_{2F\times 2} \quad [C_1 \mid \cdots \mid C_N]_{2\times 2P} .$$

When the entries of E (the matrices $\{C_{fp}\}$) do not exactly satisfy Eq. (7), then we recover an *optimal* set of $\{\widehat{A}_f\}$ and $\{\widehat{C}_p\}$ (and hence $\widehat{C}_{fp} = \widehat{A}_f \widehat{C}_p$), 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.

6 Experimental Results

This section describes our experimental evaluation of the covariance weighted factorization algorithm described in this paper. In particular, 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 direction of pure normal flow). 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 uncertainty (see Section 2.2). We performed experiments with synthetically generated data, in order to obtain a quantitative comparison of the different methods against ground truth under varying conditions.

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 parameter $r_\lambda = \sqrt{\lambda_{max}/\lambda_{min}}$, where λ_{max} and λ_{min} correspond to the major and minor axes of the uncertainty ellipse (these are the eigenvalues of the covariance matrix of the noise in feature positions). 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 ϵ_{fp} with directional uncertainty as specified above. The noise in the direction of λ_{max} (the least uncertain direction) varied between 1% and 2% of the feature positions, whereas the noise in the direction of λ_{min} (the most uncertain direction), varied between 1% and 30% of the feature positions. This noise vector 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 algorithm: the covariance-weighted factorization algorithm described in this paper, the regular SVD algorithm, and the scalar-weighted SVD algorithm, for which the 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: $e_S = \frac{||S-\hat{S}_N||}{||S||}$ $e_M = \frac{||M-\hat{M}_N||}{||M||}$ where $\hat{S}_N$ and $\hat{M}_N$ are $\hat{S}$ and $\hat{M}$ after transforming them to be in the same

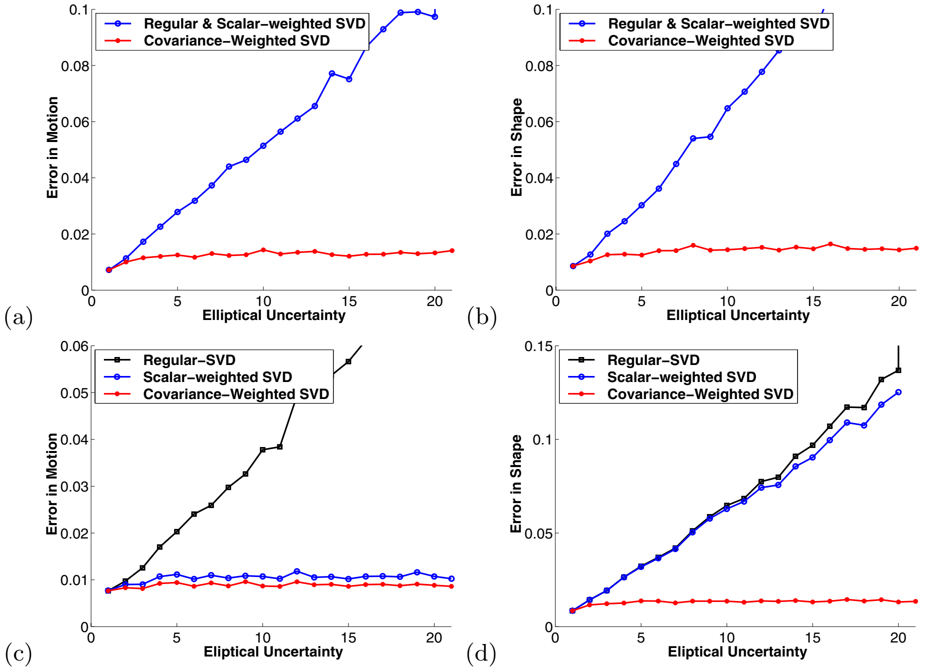

Fig. 2. *Plots of error in motion and shape w.r.t. ground truth for all three algorithms (Covariance-weighted SVD, scalar-weighted SVD, regular SVD). (a,b) Plots for the case when all points have the similar elliptical uncertainty, which is gradually increased (a = motion error, b = shape error). (c,d) 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).*

coordinate system as S and M. These errors were then averaged over the 20 trials for each setting of the parameter r_λ.

Fig. 2.a and 2.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). 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. 3.

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 reconstruction *motion* for the scalar-weighted SVD showed comparable results (although still inferior) to the covariance-weighted SVD (see Fig. 2.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. 2.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. 3 for three interesting cases: case of circular Gaussian noise $r_\lambda = 1$ for all the points (Figs. 3.a and 3.d), case of elliptic Gaussian noise with $r_\lambda = 20$ (Figs. 3.b and 3.e), and the case of pure "normal flow", when $\lambda_{min} = 0$ ($r_\lambda = \infty$) (Figs. 3.c and 3.f). (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. 2.b), yet it is not defined for the case of infinite uncertainty (because then all the weights are equal to zero).

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 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

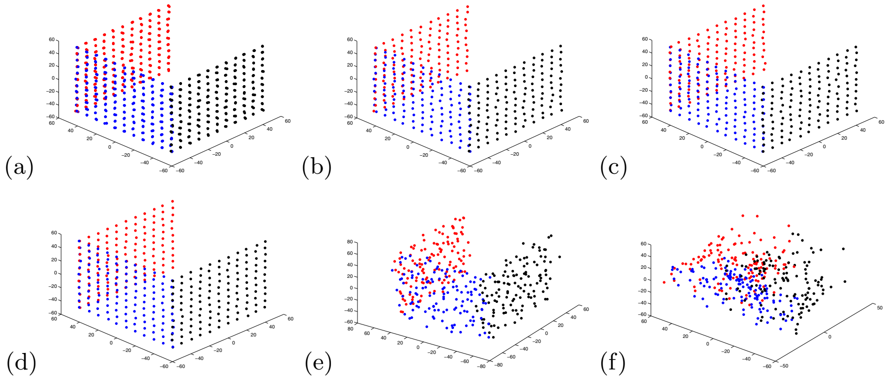

Fig. 3. *Reconstructed shape of the cube by the Covariance-weighted SVD (top row) vs. the regular SVD (bottom row). For visibility sake, only 3 sides of the cube are displayed. (a,d) case of circularly symmetric noise. (b,e) case of elliptical noise with ratio $r_\lambda = 20$. (c,f) case of pure "normal flow" (only line-like features) $r_\lambda = \infty$. Note that 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.*

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.

References

1. Aguiar, P. M. Q. and Moura, J. M. F., Factorization as a rank 1 problem, *CVPR'99*, pp. 178–184.
2. Irani, M., Multi-frame optical flow estimation using subspace constraints, *ICCV'99*, pp. 626–633, 1999.
3. Morris, D. D. and Kanade, T., A unified factorization algorithm for points, line segments and planes with uncertain models, *ICCV'98*, pp. 696–702, 1998.
4. Poelman, C. J. and Kanade, T., A paraperspective factorization method for shape and motion recovery, *IEEE Trans. PAMI*, Vol. 19, pp. 206–218, 1997.
5. Quan, L. and Kanade, T., A factorization method for affine structure from line correspondences, *CVPR'96*, pp. 803–808, San Francisco, CA, June 1996.
6. Shapiro, L. S., *Affine Analysis of Image Sequences*, Cambridge University Press, Cambridge, UK, 1995.
7. Sturm, P. and Triggs, B., A factorization based algorithm for multi-image projective structure and motion, *ECCV'96*, Vol. II, pp. 709–720.
8. Tomasi, C. and Kanade, T., Shape and motion from image streams under orthography: a factorization method, *IJCV*, Vol. 9, 1992, pp. 137–154.