

# Stratification Approach for 3-D Euclidean Reconstruction of Nonrigid Objects From Uncalibrated Image Sequences

Guanghai Wang and Q. M. Jonathan Wu, *Member, IEEE*

**Abstract**—This paper addresses the problem of 3-D reconstruction of nonrigid objects from uncalibrated image sequences. Under the assumption of affine camera and that the nonrigid object is composed of a rigid part and a deformation part, we propose a stratification approach to recover the structure of nonrigid objects by first reconstructing the structure in affine space and then upgrading it to the Euclidean space. The novelty and main features of the method lies in several aspects. First, we propose a deformation weight constraint to the problem and prove the invariability between the recovered structure and shape bases under this constraint. The constraint was not observed by previous studies. Second, we propose a constrained power factorization algorithm to recover the deformation structure in affine space. The algorithm overcomes some limitations of a previous singular-value-decomposition-based method. It can even work with missing data in the tracking matrix. Third, we propose to separate the rigid features from the deformation ones in 3-D affine space, which makes the detection more accurate and robust. The stratification matrix is estimated from the rigid features, which may relax the influence of large tracking errors in the deformation part. Extensive experiments on synthetic data and real sequences validate the proposed method and show improvements over existing solutions.

**Index Terms**—Constrained power factorization (CPF), deformation weight constraint, motion analysis, nonrigid factorization, stratified 3-D reconstruction, structure from motion.

## I. INTRODUCTION

THREE-DIMENSIONAL reconstruction from image sequences is an important and essential task of computer vision. During the last two decades, many approaches have been proposed for different applications [16]. Among them, factorization-based methods are widely studied since these approaches uniformly deal with data sets in all images and achieve good robustness and accuracy [20]–[22], [25].

Manuscript received July 5, 2007. This work was supported in part by the Canada Research Chair Program, by the Ontario Centres of Excellence, and by the National Natural Science Foundation of China under Grant 60575015. This paper was recommended by Associate Editor P. Bhattacharya.

G. Wang is with Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, N9B 3P4, Canada, and also with National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China, and also with the Department of Control Engineering, Aviation University, Changchun 130022, China (e-mail: ghwangca@gmail.com).

Q. M. J. Wu is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: jwu@uwindsor.ca).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCB.2007.910534

The factorization method was first proposed by Tomasi and Kanade [22] in the early 1990s. The main idea of this algorithm is to simultaneously factorize the tracking matrix into motion and structure matrices by singular value decomposition (SVD) with a rank constraint. The algorithm assumes an orthographic projection model. It was later extended to weak perspective and paraperspective projection by Poelman and Kanade [20]. Christy and Horaud [5] extended it to the perspective camera model by iteratively performing the factorization under affine assumption. A similar method was also studied by Fujiki and Kurata [11]. In the case of uncalibrated cameras, Quan [21] proposed a self-calibration algorithm for affine cameras. Triggs [25] proposed a rank-4 factorization scheme to achieve the projective reconstruction, where the projective depths of the features were computed via epipolar geometry. The method was further studied by many researchers in [14], [17], and [18].

Most previous algorithms are based on finding a low-rank approximation of the tracking matrix via SVD factorization. Recently, Hartley and Schaffalitzky [15] proposed an alternate algorithm, named power factorization (PF), to solve the problem. The method is derived from the power method in matrix computation [12] and the sequential factorization method proposed by Morita and Kanade [19]. Vidal and Hartley [26] also proposed to combine the PF algorithm for motion segmentation. One noteworthy advantage of the PF algorithm is that it can handle the missing data problem in the tracking matrix.

All the above methods work only for rigid objects and static scenes. Whereas in the real world, many scenarios are nonrigid or dynamic. Examples include human faces that carry different expressions, lip movements, a walking person, moving vehicles, etc. Many extensions that stem from the factorization algorithm were proposed to relax the rigidity constraint. Costeira and Kanade [6] first discussed how to recover the motion and shape of several independent moving objects via factorization under orthographic projection. The problem was further discussed by Han and Kanade [13]. Basclé and Blake [1] proposed a method for factorizing facial expressions and poses based on a set of preselected basis images.

In the pioneer work on nonrigid factorization by Bregler *et al.* [4], it is demonstrated that the 3-D shape of many nonrigid objects can be expressed as a weighted linear combination of a set of shape bases. Then, the shape bases, weighting coefficients, and camera motions were simultaneously factorized under the rank constraint of the tracking matrix. Following this idea, the problem came to the attention of many researchers, such as

Brand [2], [3], Del Bue *et al.* [7]–[9], Torresani *et al.* [23], [24], and Xiao *et al.* [28]–[30], and has been widely studied.

Torresani *et al.* [24] introduced an iterative algorithm to optimize the recovered shape and motion. Brand [2] generalized the method via nonlinear optimization and proposed subspace flow to search for correspondences. These methods use an orthonormal (rotation) constraint for Euclidean reconstruction. It may cause ambiguity in the combination of shape bases. Xiao *et al.* [28] proposed the basis constraint to solve this ambiguity and provided a closed-form solution to the problem. However, it is difficult to automatically select the shape bases for contaminated data [3], [8].

Most nonrigid factorization methods are based on weak perspective assumption, which is an approximation to the real imaging conditions. It was extended to perspective projection with calibrated cameras by Wang *et al.* [27]. Xiao and Kanade [30] proposed a two-step method for perspective reconstruction of deformable objects from uncalibrated images. There are also some other extensions and methods for nonrigid reconstruction. Torresani *et al.* [23] proposed an algorithm for learning the time-varying shape using expectation maximization, where they modeled the shape motion as a rigid component combined with a nonrigid deformation. Del Bue *et al.* proposed to segment the rigid parts of the object directly from the tracking matrix either from a rank-3 constraint [7] or an epipolar constraint [8]. They, then, recover the nonrigid shape by a constrained nonlinear optimization process. However, the segmentation may be difficult in case different groups of features satisfy the rank constraint [7] or when the interframe movements are small [8]. Additionally, the final results of the nonlinear process greatly rely on the initial estimations.

In this paper, we try to solve the problem from a new viewpoint based on an uncalibrated affine camera model. We assume that some part of the nonrigid object does not deform with time and, thus, can be taken as rigid. Our main idea is to first recover the affine structure of the object and separate the rigid features from the deformation ones. Then, we estimate the transformation from affine to metric space by virtue of rigid features and stratify the affine structure to the Euclidean space. For this purpose, we propose a deformation weight constraint for nonrigid factorization and prove that the recovered structure and shape bases are transformation invariant under this constraint. We also propose a constrained PF (CPF) algorithm to factorize the tracking matrix in affine space. The proposed method overcomes some limitations and difficulties of previous methods.

The remaining parts of this paper are organized as follows. Some backgrounds on nonrigid factorization are reviewed in Section II. The deformation weight constraint is proposed and proved in Section III. In Section IV, we present the CPF algorithm. The strategies for deformation detection and Euclidean stratification are given in Section V. Some test results are presented in Section VI, followed by the conclusion of this paper in Section VII.

## II. BACKGROUND ON NONRIGID FACTORIZATION

Under the assumption of affine camera, the projection of a 3-D point  $\mathbf{X}_i = [X_i, Y_i, Z_i]^T$  to an image point  $\mathbf{x}_i = [u_i, v_i]^T$

can be modeled as  $\mathbf{x}_i = \mathbf{A}\mathbf{X}_i + \mathbf{t}$ , where  $\mathbf{A}$  is a  $2 \times 3$  rank-2 matrix, and  $\mathbf{t}$  is a translation vector. Under the affine projection, it is easy to verify that the centroid of a set of space points is projected to the centroid of their images. Therefore, if we register the coordinates of all the image points in each image to the centroid, the translation term can be eliminated, and the projection is simplified to

$$\mathbf{x}_i = \mathbf{A}\mathbf{X}_i. \quad (1)$$

Given a sequence of  $m$  video frames of a nonrigid object. Let  $\{\mathbf{x}_i^{(j)} \in \mathbb{R}^2\}_{i=1, \dots, n}^{j=1, \dots, m}$  be  $n$  tracked feature points across the sequence. We want to recover the shape  $\mathbf{S}^{(j)} = \{\mathbf{X}_i^{(j)} | i = 1, \dots, n\} \in \mathbb{R}^{3 \times n}$ , i.e., the corresponding 3-D coordinates of the features that are associated with each frame. Following the study of Bregler *et al.* [4], the nonrigid shape in the Euclidean space is approximated by a linear combination of  $k$  shape bases as

$$\mathbf{S}^{(j)} = \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}_l \quad (2)$$

where  $\mathbf{S}_l \in \mathbb{R}^{3 \times n}$ ,  $l = 1, \dots, k$  are the shape bases that embody the principal modes of the deformation, and  $\omega_l^{(j)} \in \mathbb{R}$  is the deformation weight (a perfect rigid object would correspond to the case of  $k = 1$  and  $\omega_l^{(j)} = 1$ ). Under the weak perspective assumption (1), we have

$$\begin{aligned} \mathbf{W}^{(j)} &= \mathbf{A}^{(j)} \left( \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}_l \right) \\ &= \left[ \omega_1^{(j)} \mathbf{A}^{(j)}, \dots, \omega_k^{(j)} \mathbf{A}^{(j)} \right]_{2 \times 3k} \mathbf{B}_{3k \times n} \end{aligned} \quad (3)$$

where  $\mathbf{W}^{(j)} = [\mathbf{x}_1^{(j)}, \dots, \mathbf{x}_n^{(j)}] \in \mathbb{R}^{2 \times n}$  is the tracking matrix of the  $j$ th frame, and

$$\mathbf{B} = \begin{bmatrix} \mathbf{S}_1 \\ \dots \\ \mathbf{S}_k \end{bmatrix}$$

is the shape matrix that is composed of the  $k$  shape bases. Suppose all the  $n$  features are tracked across the  $m$  frames. Then, we can obtain the factorization equation of the tracking matrix by stacking all equations in (3) frame by frame. Hence

$$\mathbf{W}_{2m \times n} = \mathbf{M}_{2m \times 3k} \mathbf{B}_{3k \times n} \quad (4)$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_1^{(1)} & \dots & \mathbf{x}_n^{(1)} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_1^{(m)} & \dots & \mathbf{x}_n^{(m)} \end{bmatrix}$$

is the tracking matrix of the sequence, and

$$\mathbf{M} = \begin{bmatrix} \omega_1^{(1)} \mathbf{A}^{(1)} & \dots & \omega_k^{(1)} \mathbf{A}^{(1)} \\ \vdots & \ddots & \vdots \\ \omega_1^{(m)} \mathbf{A}^{(m)} & \dots & \omega_k^{(m)} \mathbf{A}^{(m)} \end{bmatrix}$$

is called the motion matrix. It is easy to see from (4) that the rank of  $\mathbf{W}$  is, at most,  $3k$  (usually  $2m$  and  $n$  are both larger than  $3k$ ). By performing SVD on the tracking matrix and imposing the rank constraint,  $\mathbf{W}$  may be factored as  $\hat{\mathbf{M}}_{2m \times 3k} \hat{\mathbf{B}}_{3k \times n}$ . However, the decomposition is not unique since it is only defined up to a nonsingular linear transformation matrix  $\mathbf{G} \in \mathbb{R}^{3k \times 3k}$  as  $\mathbf{W} = \mathbf{M}\mathbf{B} = (\hat{\mathbf{M}}\mathbf{G})(\mathbf{G}^{-1}\hat{\mathbf{B}})$ . If the transformation is known, then  $\mathbf{A}^{(j)}$ ,  $\mathbf{S}_l$ , and  $\omega_l^{(j)}$  can be directly recovered.

The computation of the transformation matrix  $\mathbf{G}$  is an extremely important and difficult step. Suppose the camera is calibrated and we are working with normalized images. Many researchers [2], [4], [24] utilize the rotation constraint of the motion matrix. Let  $\mathbf{Q} = \mathbf{G}\mathbf{G}^T$ , which is a symmetric positive definite matrix with  $(9k^2 + 3k)/2$  unknowns. Then, from  $\mathbf{M}\mathbf{M}^T = \hat{\mathbf{M}}\mathbf{G}\mathbf{G}^T\hat{\mathbf{M}}^T$ , we have the following linear constraints on  $\mathbf{Q}$ :

$$\begin{cases} \hat{\mathbf{M}}_{2i-1} \mathbf{Q} \hat{\mathbf{M}}_{2i-1}^T = \hat{\mathbf{M}}_{2i} \mathbf{Q} \hat{\mathbf{M}}_{2i}^T = \sum_{l=1}^k (\omega_l^{(i)})^2 \\ \hat{\mathbf{M}}_{2i-1} \mathbf{Q} \hat{\mathbf{M}}_{2i}^T = \hat{\mathbf{M}}_{2i} \mathbf{Q} \hat{\mathbf{M}}_{2i-1}^T = 0 \end{cases} \quad (5)$$

where  $\hat{\mathbf{M}}_i$  stands for the  $i$ th row of  $\hat{\mathbf{M}}$ . It appears that if we have enough features and frames, the matrix  $\mathbf{Q}$  can be linearly solved by stacking all the equations in (5). Then, the transformation  $\mathbf{G}$  may be factorized via eigendecomposition [2] or Cholesky decomposition [16]. However, as proved in [28], only the rotation constraints may be insufficient when the object deforms at varying speed; thus, a basis constraint is combined to solve the ambiguity. The main idea is to select  $k$  frames that include independent shapes and take them as a set of shape bases.

Nevertheless, the nonrigid factorization algorithm does not work as perfect as the rigid case due to the following reasons. First, it is difficult to automatically select the set of bases with noisy data. Second, the recovered motion matrix  $\mathbf{M} = \hat{\mathbf{M}}\mathbf{G}$  usually does not conform to the replicated block structure, as indicated in (4); therefore, a Procrustes analysis is needed [2], [4], [24], which may introduce additional errors to the final results. Third, the recovered matrix  $\mathbf{Q}$  is not positive definite in many cases due to image noise; thus, the correction matrix  $\mathbf{G}$  can be approximated by some methods [2], [21]. Moreover, the camera intrinsic parameters are difficult to estimate for nonrigid sequences.

### III. DEFORMATION WEIGHT CONSTRAINT

Previous methods utilize the orthonormal and basis constraints for nonrigid factorization, they do not have any constraint on the deformation weight. In this section, we will propose a new constraint to the deformation weight and prove that the relationship between the recovered shape bases and the structure remains invariant to the Euclidean and affine transformations if the constraint is satisfied.

*Deformation Weight Constraint:* In nonrigid factorization, the recovered deformation weights that correspond to each

frame should have equal and unit summation, i.e.,

$$\sum_{l=1}^k \omega_l^{(j)} = 1, \quad j = 1, \dots, m. \quad (6)$$

*Theorem 1:* When the structure  $\mathbf{S}^{(j)}$  of each frame and the shape bases  $\mathbf{S}_l$  are transformed by an arbitrary Euclidean transformation, the relationship in (2) remains invariant if and only if the deformation weight constraint is satisfied.

*Proof:* The transformations in the Euclidean space can be either a Euclidean transformation, i.e.,

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

or a similarity transformation, i.e.,

$$\mathbf{H}_s = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where  $\mathbf{R}$  is a  $3 \times 3$  orthonormal rotation matrix,  $\mathbf{t}$  is a 3-D translation vector,  $\mathbf{0}$  is a null 3-vector, and  $s$  is a similarity scalar. Without loss of generality, let us take  $\mathbf{H}_s$  as an example.

Suppose one set of the Euclidean shape bases and deformation weights are  $\mathbf{S}_l$  and  $\omega_l^{(j)}$ , respectively, the corresponding structure of the  $j$ th frame  $\mathbf{S}^{(j)}$  is given by (2). Under the similarity transformation  $\mathbf{H}_s$ , the shape bases and the structure are transformed to  $\mathbf{S}'_l$  and  $\mathbf{S}'^{(j)}$  as

$$\mathbf{S}'_l = s\mathbf{R}\mathbf{S}_l + \mathbf{T} \quad (7)$$

$$\mathbf{S}'^{(j)} = s\mathbf{R}\mathbf{S}^{(j)} + \mathbf{T} \quad (8)$$

where  $\mathbf{T} = [\mathbf{t}, \mathbf{t}, \dots, \mathbf{t}]$  is a  $3 \times n$  matrix. After a simple computation from (7) and (8), we have

$$\begin{aligned} \mathbf{S}'^{(j)} &= s\mathbf{R} \left( \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}_l \right) + \mathbf{T} \\ &= \sum_{l=1}^k \omega_l^{(j)} (s\mathbf{R}\mathbf{S}_l) + \mathbf{T} \\ &= \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}'_l + \left( 1 - \sum_{l=1}^k \omega_l^{(j)} \right) \mathbf{T}. \end{aligned} \quad (9)$$

It is clear that the transformed shape bases and shapes also satisfy the relation in (2) as  $\mathbf{S}'^{(j)} = \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}'_l$  if and only if  $\sum_{l=1}^k \omega_l^{(j)} = 1$ . ■

This theorem tells us that if the recovered deformation weight satisfies the constraint, the relationship between the structure and the shape bases of the object is invariant under any transformation in the Euclidean space.

*Theorem 2:* Let

$$\tilde{\mathbf{S}}_l = \begin{bmatrix} \mathbf{S}_l \\ \mathbf{1}^T \end{bmatrix}, \quad \tilde{\mathbf{S}}^{(j)} = \begin{bmatrix} \mathbf{S}^{(j)} \\ \mathbf{1}^T \end{bmatrix}$$

be the homogeneous forms of  $\mathbf{S}_l$  and  $\mathbf{S}^{(j)}$ , where  $\mathbf{1}$  stands for an  $n$ -vector with unit entities. Then, (2) can be written in homogeneous form as  $\tilde{\mathbf{S}}^{(j)} = \sum_{l=1}^k \omega_l^{(j)} \tilde{\mathbf{S}}_l$  if and only if the deformation weight constraint is satisfied.

*Proof:* The result is obvious. Since from

$$\sum_{l=1}^k \omega_l^{(j)} \tilde{\mathbf{S}}_l = \sum_{l=1}^k \omega_l^{(j)} \begin{bmatrix} \mathbf{S}_l \\ \mathbf{1}^T \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}_l \\ \sum_{l=1}^k \omega_l^{(j)} \mathbf{1}^T \end{bmatrix} \quad (10)$$

we can immediately have

$$\sum_{l=1}^k \omega_l^{(j)} \tilde{\mathbf{S}}_l = \begin{bmatrix} \mathbf{S}^{(j)} \\ \mathbf{1}^T \end{bmatrix} = \tilde{\mathbf{S}}^{(j)} \quad (11)$$

if and only if the deformation weight constraint is satisfied. ■

*Theorem 3:* When the structure and the shape bases are transformed by an arbitrary affine transformation, the relationship in (2) remains invariant if and only if the deformation weight constraint is satisfied.

*Proof:* The transformation from Euclidean to affine space can be modeled by an affine transformation, i.e.,

$$\mathbf{H}_a = \begin{bmatrix} \mathbf{P} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where  $\mathbf{P}$  is an invertible  $3 \times 3$  matrix, and  $\mathbf{t}$  is a 3-D translation vector. We will use the result of Theorem 2 to prove the theorem. Let  $\mathbf{S}'_l$  and  $\mathbf{S}'^{(j)}$  be the transformed shape bases and shapes of  $\mathbf{S}_l$  and  $\mathbf{S}^{(j)}$ . Their relation can be written in homogeneous form as

$$\tilde{\mathbf{S}}'_l = \mathbf{H}_a \tilde{\mathbf{S}}_l \quad (12)$$

$$\tilde{\mathbf{S}}'^{(j)} = \mathbf{H}_a \tilde{\mathbf{S}}^{(j)}. \quad (13)$$

Then, from Theorem 2, we have

$$\tilde{\mathbf{S}}'^{(j)} = \mathbf{H}_a \sum_{l=1}^k \omega_l^{(j)} \tilde{\mathbf{S}}_l = \sum_{l=1}^k \omega_l^{(j)} \mathbf{H}_a \tilde{\mathbf{S}}_l = \sum_{l=1}^k \omega_l^{(j)} \tilde{\mathbf{S}}'_l. \quad (14)$$

It is clear that (14) can be written in inhomogeneous form as

$$\mathbf{S}'^{(j)} = \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}'_l \quad (15)$$

if and only if the deformation weight constraint is satisfied. ■

#### A. Geometrical Explanation of the Constraint

During nonrigid factorization, we register the image measurements to the centroid. Thus, the recovered shape bases, as well as the structure of each frame are also registered to their corresponding centroids. When the shape bases and the structure are subjected to a Euclidean/affine transformation, their centroids are deviated by a translation vector. With the deformation constraint, it is guaranteed that the translation terms that resulted from the combination of the shape bases are consistent with those of the transformed structures, such that

they are invariant to the transformation. The following example illustrates the effect of the deformation weight constraint.

Given a tracking matrix  $\mathbf{W}$ , suppose there are three frames and two shape bases. In the first case, we obtain the following factorization:

$$\mathbf{W} = \mathbf{M}\mathbf{B} = \begin{bmatrix} 0.2\mathbf{A}^{(1)} & 0.8\mathbf{A}^{(1)} \\ 0.4\mathbf{A}^{(2)} & 0.6\mathbf{A}^{(2)} \\ 0.6\mathbf{A}^{(3)} & 0.4\mathbf{A}^{(3)} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{11} \\ \mathbf{S}_{12} \end{bmatrix} \quad (16)$$

where the deformation weights satisfy the constraint in (6). Since the decomposition is not unique, as explained in Section II, they are defined up to a transformation matrix  $\mathbf{G}$ . Suppose

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & 0.5\mathbf{I}_3 \end{bmatrix}$$

where  $\mathbf{I}_3$  is a  $3 \times 3$  identity matrix. Then, we will have the second case of the factorization as follows:

$$\mathbf{W} = (\mathbf{M}\mathbf{G})(\mathbf{G}^{-1}\mathbf{B}) = \begin{bmatrix} 0.2\mathbf{A}^{(1)} & 0.4\mathbf{A}^{(1)} \\ 0.4\mathbf{A}^{(2)} & 0.3\mathbf{A}^{(2)} \\ 0.6\mathbf{A}^{(3)} & 0.2\mathbf{A}^{(3)} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{21} \\ \mathbf{S}_{22} \end{bmatrix} \quad (17)$$

where the new shape bases  $\mathbf{S}_{21} = \mathbf{S}_{11}$ ,  $\mathbf{S}_{22} = 2\mathbf{S}_{12}$ . The deformation weight constraint is no longer satisfied in this case. It is easy to verify that the corresponding structure of each frame is the same in both cases. We have

$$\begin{cases} \mathbf{S}^{(1)} = 0.2\mathbf{S}_{11} + 0.8\mathbf{S}_{12} = 0.2\mathbf{S}_{21} + 0.4\mathbf{S}_{22} \\ \mathbf{S}^{(2)} = 0.4\mathbf{S}_{11} + 0.6\mathbf{S}_{12} = 0.4\mathbf{S}_{21} + 0.3\mathbf{S}_{22} \\ \mathbf{S}^{(3)} = 0.6\mathbf{S}_{11} + 0.4\mathbf{S}_{12} = 0.6\mathbf{S}_{21} + 0.2\mathbf{S}_{22}. \end{cases} \quad (18)$$

However, if we impose the Euclidean transformation

$$\mathbf{H}_e = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

to the shape bases and structures, we have

$$\begin{aligned} \mathbf{S}'_{11} &= \mathbf{R}\mathbf{S}_{11} + \mathbf{T}; \quad \mathbf{S}'_{12} = \mathbf{R}\mathbf{S}_{12} + \mathbf{T} \\ \mathbf{S}'_{21} &= \mathbf{R}\mathbf{S}_{21} + \mathbf{T}; \quad \mathbf{S}'_{22} = \mathbf{R}\mathbf{S}_{22} + \mathbf{T} \end{aligned} \quad (19)$$

$$\mathbf{S}'^{(j)} = \mathbf{R}\mathbf{S}^{(j)} + \mathbf{T}, \quad j = 1, 2, 3 \quad (20)$$

where  $\mathbf{T} = [\mathbf{t}, \mathbf{t}, \dots, \mathbf{t}]$ . From (18)–(20), we can verify that

$$\begin{cases} \mathbf{S}'^{(1)} = 0.2\mathbf{S}'_{11} + 0.8\mathbf{S}'_{12} \neq 0.2\mathbf{S}'_{21} + 0.4\mathbf{S}'_{22} \\ \mathbf{S}'^{(2)} = 0.4\mathbf{S}'_{11} + 0.6\mathbf{S}'_{12} \neq 0.4\mathbf{S}'_{21} + 0.3\mathbf{S}'_{22} \\ \mathbf{S}'^{(3)} = 0.6\mathbf{S}'_{11} + 0.4\mathbf{S}'_{12} \neq 0.6\mathbf{S}'_{21} + 0.2\mathbf{S}'_{22}. \end{cases} \quad (21)$$

By comparing (18) and (21), we conclude that the relationship between the transformed structure and the transformed shape bases remains invariant if the deformation weight constraint is satisfied.

*Corollary 4:* In affine space, the nonrigid structure can also be written as a linear combination of a set of affine shape bases just as that in the Euclidean case. It is obvious that the affine solution can be upgraded to the Euclidean space and retain their combination relationship invariant if the deformation weight constraint is satisfied.

Corollary 4 suggests the feasibility of solving the nonrigid factorization by a stratification approach. In many cases, we have no knowledge about the camera parameters of an uncalibrated image sequence; thus, it is difficult to directly adopt the orthonormal constraint to compute the transformation matrix  $\mathbf{G}$  and recover the Euclidean structure. However, as shown in the subsequent sections, we may first decompose the tracking matrix in affine space and then stratify the solution to the Euclidean space.

#### IV. CPF-BASED AFFINE RECONSTRUCTION

The general PF method was proposed to find a low-rank approximation of a tracking matrix of static and rigid scenes [15]. In the case of nonrigid factorization, our objective is to recover the motion matrix  $\mathbf{M} \in \mathbb{R}^{2m \times 3k}$  and the shape matrix  $\mathbf{B} \in \mathbb{R}^{3k \times n}$ . However, the motion matrix recovered by general PF does not observe the replicated block structure of the motion matrix in (4); thus, it is defined up to a correction matrix  $\mathbf{G}$ , which is difficult to compute, as explained in Section II. We will now introduce a CPF algorithm to solve this problem. Let us decompose the motion matrix of (4) into two parts as follows:

$$\begin{aligned} \mathbf{M} &= \mathbf{\Omega} \otimes \mathbf{\Phi} \\ &= \begin{bmatrix} \omega_1^{(1)} \mathbf{E} & \cdots & \omega_k^{(1)} \mathbf{E} \\ \vdots & \ddots & \vdots \\ \omega_1^{(m)} \mathbf{E} & \cdots & \omega_k^{(m)} \mathbf{E} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(1)} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{(m)} & \cdots & \mathbf{A}^{(m)} \end{bmatrix} \end{aligned} \quad (22)$$

where  $\mathbf{\Omega}$  denotes the weighting matrix,  $\mathbf{\Phi}$  denotes the affine motion matrix,  $\mathbf{E}$  is a  $2 \times 3$  matrix with unit entries, and ' $\otimes$ ' stands for the element-by-element multiplication. The algorithm is summarized as follows.

*Algorithm 5 (CPF):* Given the tracking matrix  $\mathbf{W} \in \mathbb{R}^{2m \times n}$ , initial values  $\mathbf{\Phi}_0 \in \mathbb{R}^{2m \times 3k}$ , and  $\mathbf{B}_0 \in \mathbb{R}^{3k \times n}$ . Repeat the following three steps until the convergence of product  $(\mathbf{\Omega}_t \otimes \mathbf{\Phi}_t) \mathbf{B}_t$ .

1) Given  $\mathbf{\Phi}_{t-1}$  and  $\mathbf{B}_{t-1}$ , find  $\mathbf{\Omega}_t$  to minimize  $\|\mathbf{W} - (\mathbf{\Omega}_t \otimes \mathbf{\Phi}_{t-1}) \mathbf{B}_{t-1}\|_F^2$ , subject to the condition that  $\mathbf{\Omega}_t$  satisfies the deformation weight constraint.

2) Given  $\mathbf{\Phi}_{t-1}$  and  $\mathbf{\Omega}_t$ , find  $\mathbf{B}_t$  to minimize  $\|\mathbf{W} - (\mathbf{\Omega}_t \otimes \mathbf{\Phi}_{t-1}) \mathbf{B}_t\|_F^2$ .

3) Given  $\mathbf{\Omega}_t$  and  $\mathbf{B}_t$ , find  $\mathbf{\Phi}_t$  to minimize  $\|\mathbf{W} - (\mathbf{\Omega}_t \otimes \mathbf{\Phi}_t) \mathbf{B}_t\|_F^2$ , subject to the constraint that  $\mathbf{\Phi}_t$  is a block replicated matrix.

The main difference in Algorithm 5 with the general PF is that we divide the computation of  $\mathbf{M}$  into the computation of  $\mathbf{\Omega}$  and  $\mathbf{\Phi}$ . This makes it possible to directly combine the weight constraint and the replicated block structure of the motion matrix into the factorization.

It is quite easy to combine the weight constraint into the minimization scheme in step 1), since we can always set  $\omega_k^{(j)} = 1 - \sum_{l=1}^{k-1} \omega_l^{(j)}$ . To observe the block replicated structure of

$\mathbf{\Phi}$  in step 3), denote  $\mathbf{W}^{(j)} \in \mathbb{R}^{2 \times n}$  as the  $j$ th two-row of  $\mathbf{W}$ , which is the tracking matrix of the  $j$ th frame. Then, we have

$$\mathbf{W}^{(j)} = \mathbf{A}^{(j)} \mathbf{S}^{(j)} \quad \mathbf{S}^{(j)} = \sum_{l=1}^k \omega_l^{(j)} \mathbf{S}_l \quad (23)$$

where  $\omega_l^{(j)}$  can be obtained from  $\mathbf{\Omega}_t$  in step 1), and  $\mathbf{S}_l$  can be obtained from  $\mathbf{B}_t$  in step 2). Therefore,  $\mathbf{A}^{(j)}$  may be computed by the least squares as

$$\mathbf{A}^{(j)} = \mathbf{W}^{(j)} \left( \mathbf{S}^{(j)} \right)^T \left( \mathbf{S}^{(j)} \left( \mathbf{S}^{(j)} \right)^T \right)^{-1}. \quad (24)$$

Then, the block-replicated matrix  $\mathbf{\Phi}_t$  can be evaluated from  $\mathbf{A}^{(j)}$ , as indicated in (22).

*Determination of the Convergence:* A theoretical proof of the convergence of this algorithm is currently unavailable, even for the general PF in [15]. Nevertheless, extensive simulation tests show that the algorithm usually converges quickly when the rank of  $\mathbf{W}$  is close to  $3k$  and when good initial values are present. Specifically, suppose  $s_i$  is the  $i$ th largest singular value of  $\mathbf{W}$ . Then, the convergence is proportional to  $(s_{3k+1}/s_{3k})^{2t}$ .

There are several possible ways to determine the convergence of the algorithm. The most feasible way is to check the variation of the reprojected tracking matrix. Suppose  $\mathbf{W}_t = (\mathbf{\Omega}_t \otimes \mathbf{\Phi}_t) \mathbf{B}_t$  is the reprojected tracking matrix at the  $t$ th iteration, then the variation can be defined as

$$\delta = \|\mathbf{W}_t - \mathbf{W}_{t-1}\|_F^2. \quad (25)$$

*Initialization:* The solution of the algorithm is not unique and is defined up to an affine transformation. In the worst case, the recovered affine structure may be stretched or squeezed greatly along a certain direction due to bad initialization, which makes it difficult to detect the deformation features in the subsequent section. Usually, reasonable initial values of  $\mathbf{B}_0$  and  $\mathbf{\Phi}_0$  can simultaneously avoid the worst situation and improve convergence speed.

In our applications, we first utilize the rank-3 factorization method [15], [20], [21] to obtain a rigid approximation of the object as  $\mathbf{W} = \hat{\mathbf{M}} \hat{\mathbf{B}}$ , where  $\hat{\mathbf{M}} \in \mathbb{R}^{2m \times 3}$  is the rigid motion matrix,  $\hat{\mathbf{B}} \in \mathbb{R}^{3 \times n}$  is the rigid shape of the object. Under this estimation, we compute the mean reprojection error of each point across the sequence and denote it as  $e_r$ . Then, the initial values may be constructed as

$$\begin{aligned} \mathbf{\Phi}_0 &= [\hat{\mathbf{M}}, \dots, \hat{\mathbf{M}}]_{2m \times 3k} \\ \mathbf{B}_0 &= \begin{bmatrix} \hat{\mathbf{S}}_1 + \mathbf{e}_r^T \otimes \mathbf{N}_1 \\ \vdots \\ \hat{\mathbf{S}}_1 + \mathbf{e}_r^T \otimes \mathbf{N}_k \end{bmatrix}_{3k \times n} \end{aligned} \quad (26)$$

where  $\mathbf{e}_r^T \otimes \mathbf{N}_i$  is a reprojection-error-weighted shape balance matrix, and  $\mathbf{N}_i \in \mathbb{R}^{3 \times n}$  is a small random matrix. This term is used to ensure that initial shape bases are independent of each other. Usually, the motions recovered by rigid factorization are close to the real solutions; therefore, we directly use  $\hat{\mathbf{M}}$  to construct  $\mathbf{\Phi}_0$  and update the rotation matrix at the last

step in the algorithm, though the order of these steps can be interchanged. Experiments demonstrate that this initialization can usually give good results.

*Working With Missing Data:* It should be noted that each minimization step in the algorithm is equivalent to solving a set of equations by least squares, this can allow us to deal with the tracking matrix with missing data (i.e., some features are not tracked in some frames thus the corresponding entries in the tracking matrix are unavailable). In this case, the cost function in the algorithm can be modified as

$$\min \sum_{(i,j) \in \mathcal{A}} \left| W_{ij} - ((\mathbf{\Omega} \otimes \mathbf{\Phi})\mathbf{B})_{ij} \right|^2 \quad (27)$$

where  $W_{ij}$  denotes the  $(i, j)$ th element of the tracking matrix, and  $\mathcal{A}$  stands for the set of available entries in the tracking matrix. Thus, we can update  $\mathbf{\Omega}$ ,  $\mathbf{B}$ , and  $\mathbf{\Phi}$  using only available features in  $\mathbf{W}$  according to (27). This is a good attribute of the CPF algorithm, since it is hard to have all the features tracked across the whole sequence due to self-occlusion in real applications, although it is difficult to deal with missing data for SVD factorization.

## V. DEFORMATION DETECTION AND STRATIFICATION

### A. Deformation Detection Strategy

From the CPF algorithm, we can obtain the 3-D affine structure  $\mathbf{S}^{(j)}$  that is associated with each frame. For most nonrigid objects, some part of its structure is usually nondeformable and can be taken as rigid or near-rigid, whereas the remaining part is deformable. Our objective is to separate the rigid features  $\mathbf{S}_r^{(j)} \in \mathbb{R}^{3 \times n_1}$  from the deformation ones  $\mathbf{S}_n^{(j)} \in \mathbb{R}^{3 \times n_2}$ , where  $n_1 + n_2 = n$ . Since the structure of the rigid parts do not change with time, the rigid parts will be aligned well with each other if we register all the 3-D structures to a reference view by virtue of the rigid features, whereas the deformation part deforms with time and can be easily detected from the registration errors.

The registration in 3-D affine space is defined by an affine transformation. Let us take the first frame as a reference. Then, the transformation from the  $j$ th frame to the first frame, i.e.,

$$\mathbf{H}_a^{(j1)} = \begin{bmatrix} \mathbf{P}^{(j1)} & \mathbf{t}^{(j1)} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

can be computed from

$$\tilde{\mathbf{S}}_r^{(1)} = \mathbf{H}_a^{(j1)} \tilde{\mathbf{S}}_r^{(j)} \quad (28)$$

where the shapes are written in homogeneous forms. It is clear that the transformation  $\mathbf{H}_a^{(j1)}$  can be linearly computed from four pairs of point correspondences in a general position between the two structures. However, since we do not know whether the selected features are rigid or nonrigid, we adopt an iterative RANDOM SAMPLE CONSENSUS (RANSAC) paradigm [10] to compute the transformation. During each iteration, we randomly draw four pairs of corresponding 3-D features across the whole sequence and use these data sets to hypothesize the

transformation matrix  $\mathbf{H}_a^{(j1)}$  ( $j = 2, \dots, m$ ). We then register all structures to the reference frame. Suppose  $\mathbf{S}^{(j1)}$  is the transformed structure of  $\mathbf{S}^{(j)}$  to the reference frame. Then, the registration error is defined as the Euclidean distance between each pair of features. We have

$$\mathbf{e}^{(j)} = \text{diag} \left( \left( (\mathbf{S}^{(j1)} - \mathbf{S}^{(1)})^T (\mathbf{S}^{(j1)} - \mathbf{S}^{(1)}) \right) \right) \quad (29)$$

$$\mathbf{E} = \frac{1}{m-1} \sum_{j=2}^m \mathbf{e}^{(j)} \quad (30)$$

where  $\mathbf{e}^{(j)} \in \mathbb{R}^n$  stands for the error of each feature in frame  $j$ ,  $\text{diag}(\mathbf{X})$  stands for the main diagonal of matrix  $\mathbf{X}$ , and  $\mathbf{E} \in \mathbb{R}^n$  stands for the mean registration error of each feature in all frames. The transformation with the most supporting features (i.e., small registration error) must be the correct one.

The RANSAC algorithm selects samples with uniform probability, which is computationally expensive, particularly for large data. Suppose the fraction of the rigid features is  $\gamma$ . Then, the trial number  $N$  can be determined from

$$N = \frac{\log(1-P)}{\log(1-\gamma^4)} \quad (31)$$

where  $P$  is the probability that all the randomly selected four sets of samples are rigid features. For example, if we set  $P = 0.99$  and  $\gamma = 0.4$ , then the trials should be  $N = 178$ . However, as we have obtained the mean reprojection error  $\mathbf{e}_r$  in the initialization step of Section IV, one may speed up the efficiency of the algorithm by incorporating this prior information into the drawing process. It is clear that the points with small reprojection errors are more likely to belong to the rigid parts; thus, they are given a higher drawing probability. Whereas the features with large errors are given lower probability. Suppose the fraction of rigid features was increased to  $\gamma' = 0.7$  by taking out of the features with large reprojection errors. Then, the trials will be reduced to  $N' = 17$  under the same probability  $P = 0.99$ . Thus, more than 90% of the computation cost may be saved in this way.

After the correct transformation matrix  $\mathbf{H}_a^{(j1)}$  is recovered, we can register all structures to the reference frame and compare their registration error. Since the structure of deformation parts varies from frame to frame, the mean registration error of these features is much larger than that of the rigid ones. Thus, the deformation is easily distinguished, as shown in the tests. Previous methods tried to detect the deformation from 2-D measurements [7], [8]. This is a difficult problem since the constraint for segmentation in 2-D are prone to be violated by noise. Whereas, in 3-D space, we have more geometrically meaningful information, and the errors of the deformation parts are accumulated frame by frame. Thus, more accurate and robust results are expected.

### B. Stratification to the Euclidean Space

We have now separated the rigid features  $\mathbf{S}_r^{(j)}$  from the deformation ones  $\mathbf{S}_n^{(j)}$ . For uncalibrated rigid objects, there

are many good methods to recover the Euclidean shape and motions [14], [20], [21]. Thus, we can obtain the Euclidean structure  $\mathbf{S}_{er}$  of the rigid parts via one of these methods. From Corollary 4, we know that the affine solution can be stratified to the Euclidean space via the affine transformation  $\mathbf{H}_a^{(j)}$ , which can be computed as follows:

$$\tilde{\mathbf{S}}_{er} = \mathbf{H}_a^{(j)} \tilde{\mathbf{S}}_r^{(j)}. \quad (32)$$

In (32), the stratification matrix is estimated only from the rigid parts. Thus, the influence caused by larger tracking errors of the nonrigid features may be relaxed, since feature tracking is a difficult problem, particularly for deformable features due to the absence of a disambiguating geometric constraint. After  $\mathbf{H}_a^{(j)}$  is recovered, the deformation parts, as well as the whole structure, can be stratified to the Euclidean space from

$$\tilde{\mathbf{S}}_{en}^{(j)} = \mathbf{H}_a^{(j)} \tilde{\mathbf{S}}_n^{(j)} \quad \tilde{\mathbf{S}}_e^{(j)} = \mathbf{H}_a^{(j)} \tilde{\mathbf{S}}^{(j)} \quad (33)$$

where  $\tilde{\mathbf{S}}_{en}^{(j)}$  and  $\tilde{\mathbf{S}}_e^{(j)}$  stand for the Euclidean shape of the deformation parts and the whole structure, respectively. The solution of (33) is suboptimal since the stratification matrix is computed only from the rigid features. Thus, a global optimization scheme is followed after stratification. Suppose the motion matrix and structure after stratification are  $\mathbf{A}_e^{(j)}$  and  $\mathbf{S}_e^{(j)}$ , respectively, these parameters may be optimized by minimizing the image reprojection residuals

$$f(\mathbf{A}_e^{(j)}, \mathbf{S}_e^{(j)}) = \min \|\mathbf{W} - \bar{\mathbf{W}}\|_F^2 \quad (34)$$

where  $\bar{\mathbf{W}}$  denotes the reprojected tracking matrix. The minimization process is also termed as bundle adjustment in a computer vision society that can be solved via the Newton or the Levenberg–Marquardt iteration method, as given in [16].

*Implementation Outline:* The implementation details of the proposed method are summarized as follows.

- 1) For the given tracking matrix, perform rigid factorization and construct the initial values according to (26).
- 2) Compute the affine shape bases and structure according to the CPF Algorithm 5.
- 3) Separate the rigid features from the deformation ones from the 3-D affine structure via the RANSAC algorithm.
- 4) Compute the stratification matrix from (32), and stratify the structure from affine to the Euclidean space.
- 5) Perform the optimization scheme to the global structure and motion according to (34).

## VI. EXPERIMENTAL EVALUATIONS

### A. Tests With Synthetic Data

We generated a synthetic cube with three visible surfaces in the space, whose dimensions was  $10 \times 10 \times 10$ , with nine evenly distributed points on each edge. There were three sets of moving points ( $17 \times 3$  points) on the adjacent surfaces of the cube that move on the surfaces at a constant speed, as shown in Fig. 1. The object was composed of 90 rigid points and

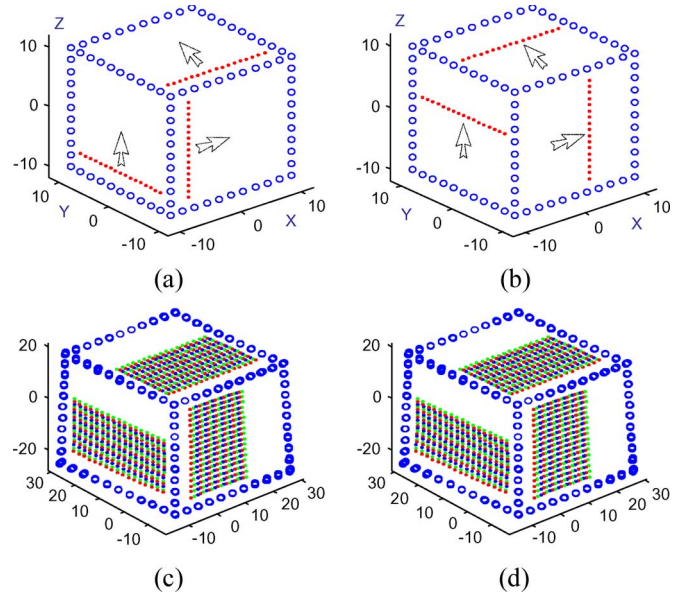


Fig. 1. Synthetic data and reconstruction results. (a) and (b) Synthetic cubes that correspond to the first and last frames, where the rigid and moving points are denoted by circles and dots, respectively. (c) Registered 3-D structures of the 20 frames obtained by the proposed method. (d) Registered 3-D structures by the method of SVD + RB.

51 deformation points. We generated 20 frames with different camera parameters by perspective projection. The image size is  $500 \times 500$ , whereas the distance of the camera to the object was set at about 12 times the object size such that the imaging condition was close to affine.

*Reconstruction Results and Evaluations:* During the test, one-pixel Gaussian noise was added to the images. We recovered the 3-D Euclidean structures of the 20 frames by the proposed stratification algorithm and automatically registered all structures to the first frame via RANSAC. The result is shown in Fig. 1(c). We also performed a comparison with the SVD-based method with rotation and basis constraints (SVD + RB) [28], as shown in Fig. 1(d). One may see from the results that the deformation structure is correctly recovered by both methods.

It should be noted that the recovered structures in Fig. 1(c) and (d) are defined up to a 3-D similarity transformation with the ground truth. For evaluation, we computed the similarity matrix by virtue of the point correspondences between the recovered structure and the ground truth; then, the structure was registered with the ground truth. We calculated the distances between all the corresponding point pairs. Fig. 2 shows the mean and standard deviation of the distances that are associated with each frame at two different noise levels. As a comparison, we also give the results obtained by the SVD-based method with the rotation constraint only (SVD + R) [4] and the method with both the rotation and basis constraints (SVD + RB) [28]. As shown in Fig. 2, it is clear that the proposed method outperforms the two SVD-based methods.

*Convergence Property of the CPF Algorithm:* We tested the convergence rate in three cases: First, we used all data without added noise. Secondly, we randomly deleted 20% points from the tracking matrix. Thirdly, we added two-pixel Gaussian noise to the imaged features. At each iteration  $t$ , we recorded

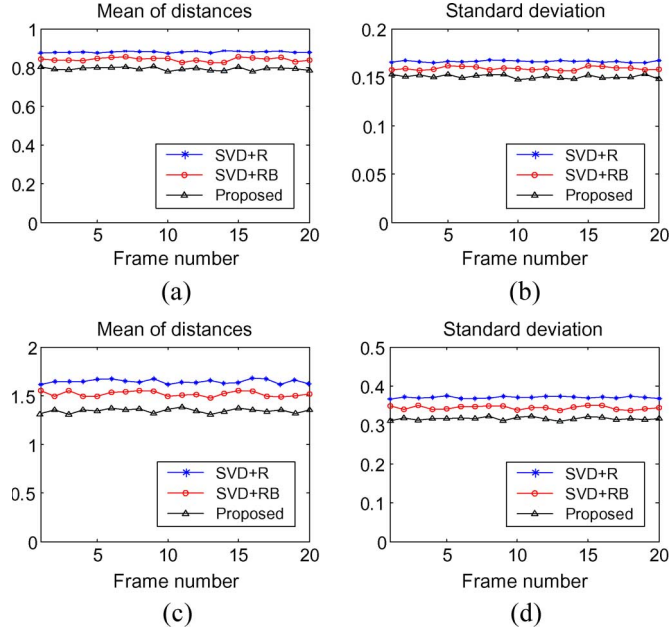


Fig. 2. Performance evaluation with the ground truth. We registered the reconstructed structure to the ground truth and compared the mean and standard deviation of the distances between the recovered structure and its ground truth associated with each frame. (a) and (b) Mean and standard deviation of the errors with one-pixel Gaussian noise. (c) and (d) Results with two-pixel Gaussian noise.

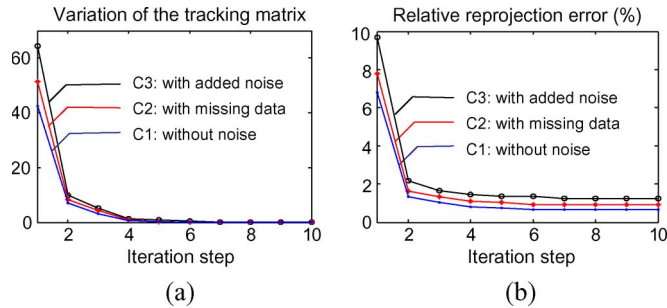


Fig. 3. Convergence property of the CPF algorithm in three conditions. (a) Variation of the reprojected tracking matrix at each iteration. (b) Relative reprojection error at each iteration.

the variation (25) of the reprojected tracking matrix and the relative reprojection error defined by

$$\mathbf{E}_{\text{rep}} = \|\mathbf{W} - (\mathbf{\Omega}_t \otimes \mathbf{\Phi}_t)\mathbf{B}_t\|_F^2 / \|\mathbf{W}\|_F^2 \times 100 (\%). \quad (35)$$

The results are shown in Fig. 3. It can be seen from these tests that the CPF algorithm quickly converges, even with some missing data and measurement errors. However, the algorithm may fail when the proportion of missing entries exceeds a certain level. One may also find that a small relative reprojection error still exists after convergence, even for noise-free data. This is because the images are formulated by perspective projection rather than affine projection. The residual error will vanish if the images are generated by affine projection. For the computation time of one iteration, the CPF algorithm takes 0.062 s on an Intel Pentium IV personal computer with a 3.6-GHz central processing unit, programmed with Matlab 6.5. Whereas the SVD-based method [28] only takes 0.017 s. It is clear that the CPF algorithm takes much more computation time than the

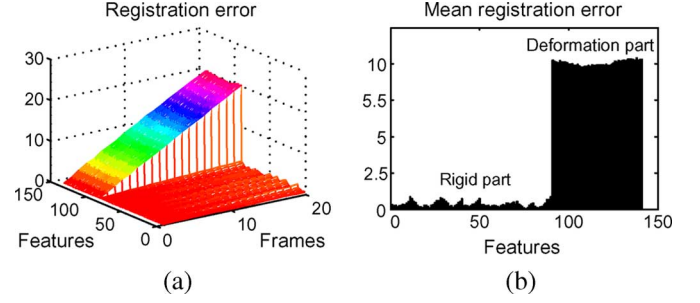


Fig. 4. (a) Registration error  $e^{(j)}$  of the features in each frame. (b) Mean registration error  $\mathbf{E}$  of every feature across the sequence.

TABLE I  
MISCLASSIFICATION ERROR WITH RESPECT TO DIFFERENT NOISE LEVELS AND DEFORMATION/RIGID FEATURE RATIOS OUT OF 100 TRIALS

Noise level	0	0.5	1	1.5	2	
Ratio	51/90	0.00	0.00	0.05	0.37	0.59
	51/60	0.00	0.06	0.15	0.42	0.83
	51/30	0.09	0.21	0.54	1.05	1.77

SVD-based method. However, the algorithm is still very fast for most offline applications.

*Deformation Detection:* From the CPF algorithm, we can recover the affine structure and shape bases. All the structures are automatically registered to the first view via the RANSAC algorithm, and the registration error (29) of the features in each frame and the mean error (30) of every feature across the sequence are shown in Fig. 4, where the first 90 features belong to the rigid part. We can see that the deformation part is easy to detect from the mean registration error in 3-D space.

The detection strategy may be affected by the ratio of nonrigid features over rigid ones, noise level, threshold value, deformation amplitude, etc. We studied the misclassification error (the number of misclassified features) with respect to the noise level and the ratio of nonrigid features. The results are shown in Table I, where we vary the number of rigid features from 90 to 30, whereas the number of nonrigid features is fixed to 51; the noise level is varied from 0 to 2 pixels. The values in Table I are evaluated from 100 independent tests. In real applications, the threshold is experimentally determined based on the distribution of the mean registration error (30). We usually avoid the misclassification of deformation features into rigid ones by reducing the threshold such that the stratification matrices may be more accurately recovered.

## B. Tests With Real Sequences

We tested the proposed methods on several real image sequences, and the results obtained from four such sequences are reported in this paper. All sequences in the test were captured by a Canon Powershot G3 camera, except the Franck sequence that was downloaded over the Internet.

*Test on Grid Sequence:* The sequence is composed of 12 images with a resolution of  $1024 \times 768$ . The first and the last frames are shown in Fig. 5. The background of the sequence is made up of two orthogonal sheets with square grids, which are used as the ground truth for evaluation. On

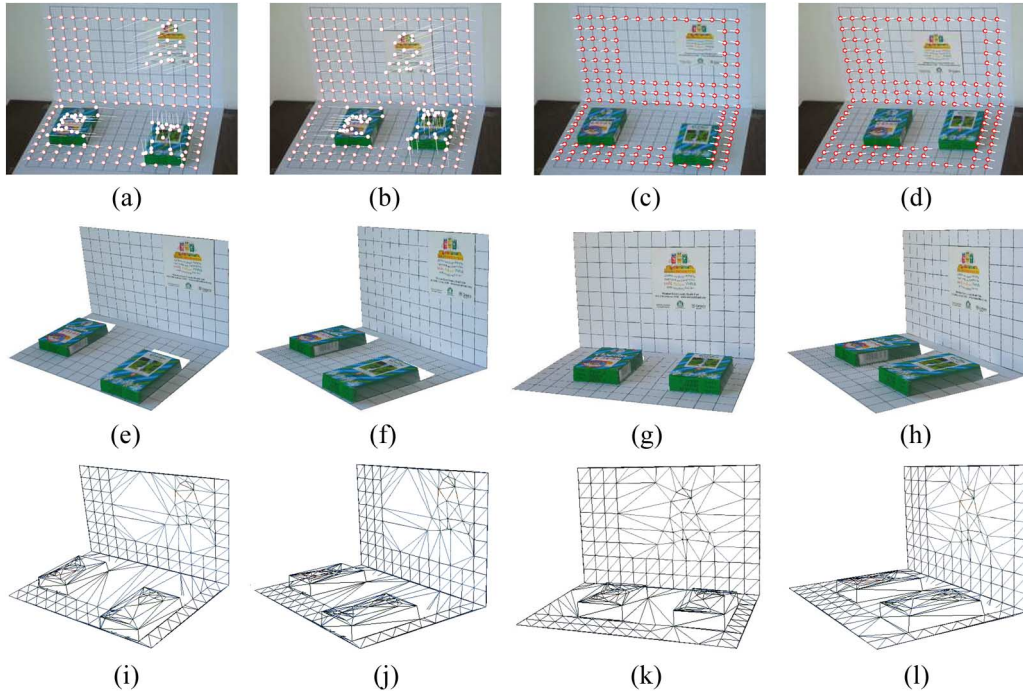


Fig. 5. Reconstruction results of the grid sequence. (a) and (b) First and last frames from the sequence overlaid with 206 tracked features and the relative disparities shown in white lines. (c) and (d) Two frames overlaid with the automatically detected rigid features. (e)–(h) Reconstructed VRML models of the two frames shown in different viewpoints with texture mapping. (i)–(l) Corresponding wireframes of the VRML models.

the two orthogonal surfaces, there are three moving objects that linearly move in three directions. As shown in Fig. 5(a) and (b), 206 tracked features are interactively established across the sequence, where 140 features belong to the static background, and 66 features belong to the three moving objects. All the static features are automatically detected by the RANSAC scheme, as shown in Fig. 5(c) and (d). We recovered the metric structure of the scenario by the proposed method. Fig. 5(e)–(l) shows the reconstructed virtual reality modeling language (VRML) models and the corresponding triangulated wireframes of the two frames at different viewpoints. We can see from the results that the dynamic structure is correctly recovered. The static and moving features are automatically separated by the proposed method.

For more performance evaluation and comparison, we first computed the relative reprojection errors ( $\mathbf{E}_{rep}$ ) by the proposed method and the two SVD-based methods. Then, we computed the angle between the two orthogonal sheets of the background, the length ratio of the two diagonals of each square and the angle formed by the two diagonals. The mean errors of these three values are denoted by  $\mathbf{E}_{angb}$ ,  $\mathbf{E}_{ratio}$ , and  $\mathbf{E}_{angd}$ , respectively. The comparative results obtained by the proposed and the two SVD-based methods are listed in Table II, which shows that the proposed method performs better than the SVD-based methods, as expected.

*Test on the Scarf Sequence:* There were 15 images in the sequence, and the scarf was pressed so that its structure deforms to a certain extent during shooting. The image resolution was  $1024 \times 768$ . We established the initial correspondences by the method in [31] and interactively deleted some outliers. As shown in Fig. 6, 2986 features were tracked across the sequence. Fig. 6 shows the reconstructed VRML models and

TABLE II  
PERFORMANCE COMPARISON AND EVALUATION OF THE PROPOSED METHOD WITH RESPECT TO THE SVD-BASED METHODS

Sequence	Method	$\mathbf{E}_{rep}$	$\mathbf{E}_{angb}$	$\mathbf{E}_{ratio}$	$\mathbf{E}_{angd}$
Grid	Proposed	4.293	0.025	0.108	0.012
	SVD+RB	5.662	0.041	0.147	0.016
	SVD+R	5.817	0.049	0.153	0.024
Scarf	Proposed	3.275	0.038	0.093	0.007
	SVD+RB	4.946	0.060	0.126	0.011
	SVD+R	5.173	0.066	0.149	0.026
	Rigid	8.268	0.141	0.286	0.055

the wireframes of the two frames from different viewpoints. The recovered structures were visually plausible and realistic.

The background of the sequence is two orthogonal grid sheets just as those in the grid sequence. We also perform a comparative evaluation on the reprojection error, angle between the two sheets, the length ratio, and the angle of the two diagonals of each grid. The results are also listed in Table II, from which we can see that the reconstruction error obtained by the proposed method is smaller than that of the two SVD-based methods.

For the scarf sequence, the deformation is small across the sequence. We assume that the scarf is rigid, and we utilize the rigid SVD factorization algorithm [20] to recover its structure. Then, we compute the errors of  $\mathbf{E}_{rep}$ ,  $\mathbf{E}_{angb}$ ,  $\mathbf{E}_{ratio}$ , and  $\mathbf{E}_{angd}$ . As shown in Table II, these errors greatly increase if we assume that the sequence is rigid.

*Test on the Franck Sequence:* The sequence was downloaded from the European Working Group on Face and Gesture Recognition ([www-prima.inrialpes.fr/FGnet/](http://www-prima.inrialpes.fr/FGnet/)). We selected 60 frames with various facial expressions for the experiment. The resolution was  $720 \times 576$ , and there were 68 tracked

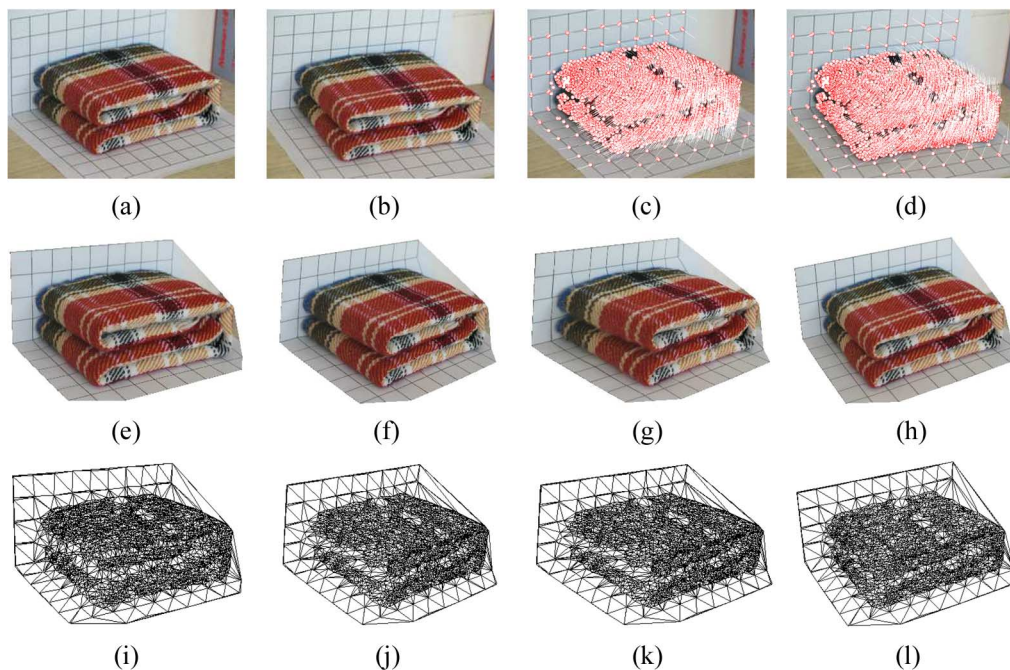


Fig. 6. Reconstruction results of the scarf sequence. (a) and (b) Two frames from the sequence. (c) and (d) The 2986 tracked features of the two frames with relative disparities shown in white lines. (e)–(h) Reconstructed VRML models of the two frames shown in different viewpoints with texture mapping. (i)–(l) Corresponding wireframes of the VRML models.

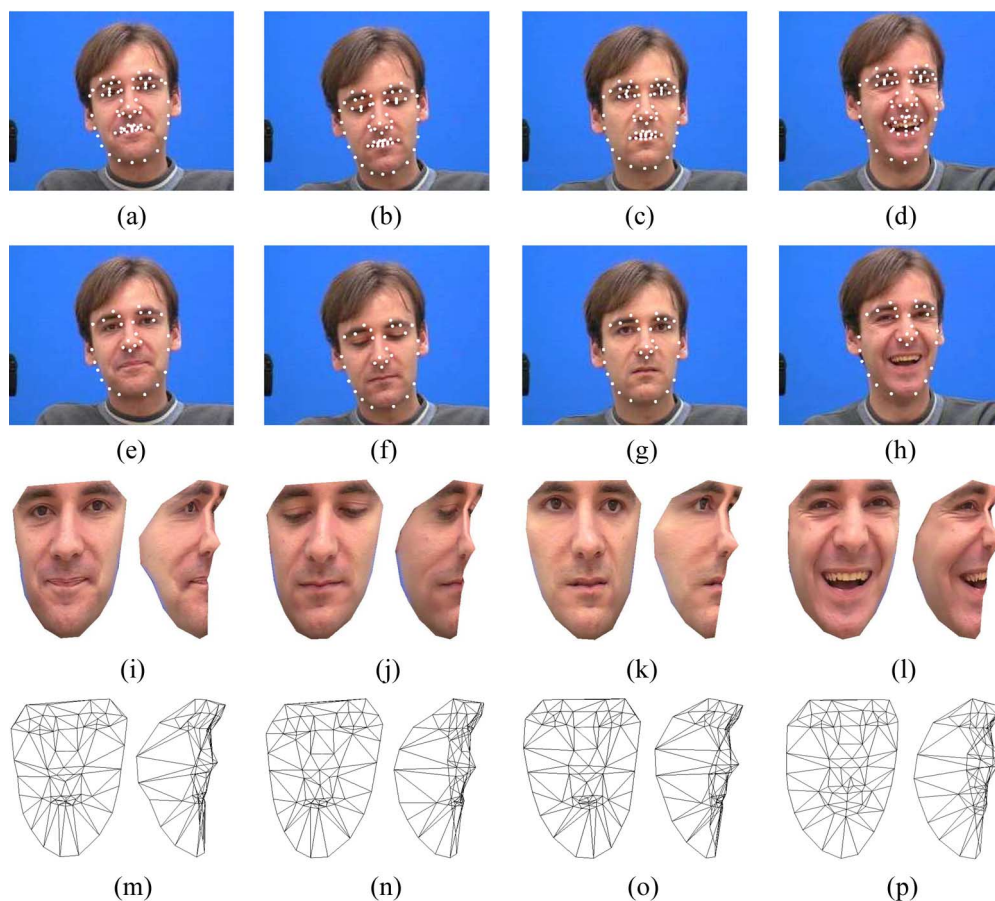


Fig. 7. Reconstruction results of the Franck sequence. (a)–(d) Four frames from the sequence overlaid with 68 tracking features. (e)–(h) Four frames overlaid with the automatically detected rigid features. (i)–(l) Front and side views of the corresponding reconstructed VRML models with texture mapping. (m)–(p) Corresponding triangulated wireframe models.

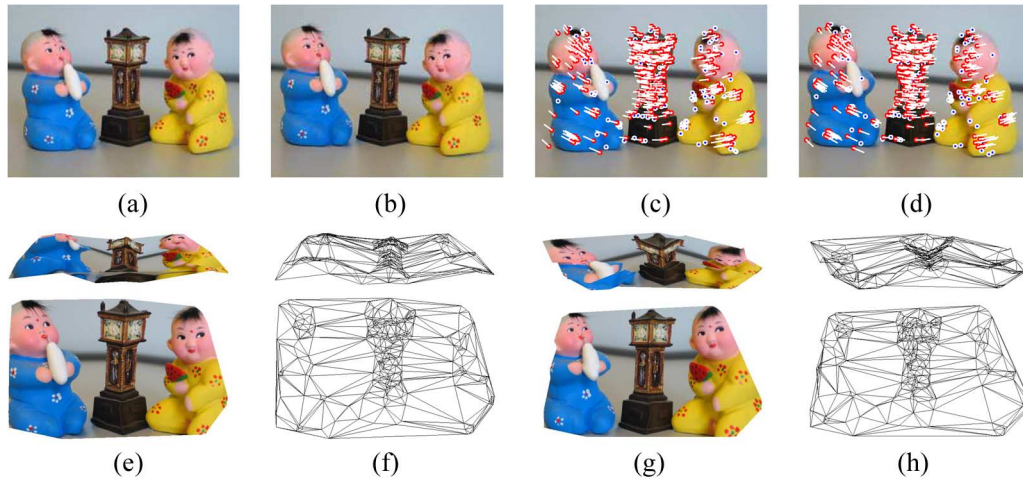


Fig. 8. Reconstruction results of the toy sequence with 12% missing data. (a) and (b) Two frames from the sequence. (c) and (d) There are 427 and 439 detected features in the two frames, respectively, where 352 points are correctly tracked, as shown with relative disparities in white lines; the other points are not matched between the two views, as shown in black dots with white circles. (e) and (g) Reconstructed VRML models of the two frames shown in different viewpoints with texture mapping. (f) and (h) Corresponding wireframes of the VRML models.

features across the sequence. Fig. 7 shows the automatically detected rigid features, the reconstructed VRML models, and triangulated wireframes of four frames by the proposed method. We can see from the results that different facial expressions are correctly recovered. The results could be used for visualization and recognition. However, the structure of some feature points was not very accurate due to tracking errors. In comparison, the relative reprojection errors by the proposed method and the two SVD-based methods are 5.758, 6.425, and 6.653, respectively.

*Test on Toy Sequence:* There are 25 frames in the sequence, and the image resolution is  $1024 \times 768$ . The scene is composed of three rigid objects, where the clock tower is fixed and the two clay babies move slowly during shooting. The matching is also done by the method in [31]. The feature tracking is very difficult for this sequence due to the smooth texture of the clay babies. Fig. 8(a)–(d) shows the detected and matched features of the two frames. In this test, instead of using the features being tracked across the whole sequence, we use all those being tracked across more than 20 frames. Thus, there are about 12% missing data in the tracking matrix. We performed the CPF algorithm on the tracking matrix and recovered the affine structure of the scene; then, we automatically detected the static features and upgraded the solution to the Euclidean space. Fig. 4(a) and (b) shows the reconstructed 3-D structures and wireframes of the two frames from different viewpoints. One may notice from the results that some 3-D points are not accurate due to the tracking errors. However, the structures of the scene are largely reasonable. The relative reprojection error by the proposed method is 7.164. We do not have the results of the SVD-based algorithm for this sequence due to the missing data problem in the tracking matrix.

## VII. CONCLUSION

In this paper, we first proposed the deformation weight constraint to ensure the invariant relationship between the recovered shape bases and structures. Then, we presented the CPF algorithm to recover the deformation structure in affine

space. The algorithm overcomes some limitations of previous SVD-based methods: it is easy to implement and can work with missing data. Based on the 3-D affine structures, we proposed a RANSAC-based strategy to detect and separate the rigid features from the deformation ones and stratified the solution from affine to the Euclidean space by virtue of the rigid features. Experiments with synthetic data and real sequences validate the proposed method and show the improvements over the SVD-based methods. The registration and detection strategy may fail when the rigid features only amount for a small portion of the whole features. We are currently working on this problem.

## REFERENCES

- [1] B. Basclé and A. Blake, "Separability of pose and expression in facial tracing and animation," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 323–328.
- [2] M. Brand, "Morphable 3D models from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2001, vol. 2, pp. 456–463.
- [3] M. Brand, "A direct method for 3D factorization of nonrigid motion observed in 2D," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, vol. 2, pp. 122–128.
- [4] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, vol. 2, pp. 690–696.
- [5] S. Christy and R. Horaud, "Euclidean shape and motion from multiple perspective views by affine iterations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 11, pp. 1098–1104, Nov. 1996.
- [6] J. Costeira and T. Kanade, "A multibody factorization method for independent moving objects," *Int. J. Comput. Vis.*, vol. 29, no. 3, pp. 159–179, Sep. 1998.
- [7] A. Del Bue, X. Lladó, and L. de Agapito, "Non-rigid face modelling using shape priors," in *Proc. 2nd Int. Workshop AMFG*, 2005, pp. 97–108.
- [8] A. Del Bue, X. Lladó, and L. de Agapito, "Non-rigid metric shape and motion recovery from uncalibrated images using priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, vol. 1, pp. 1191–1198.
- [9] A. Del Bue, F. Smeraldi, and L. Agapito, "Non-rigid structure from motion using non-parametric tracking and non-linear optimization," in *Proc. IEEE Workshop Articulated Nonrigid Motion*, 2004, pp. 8–15.
- [10] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
- [11] J. Fujiki and T. Kurata, "Recursive factorization method for the paraperspective model based on the perspective projection," in *Proc. ICPR*, 2000, vol. 1, pp. 406–410.

- [12] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1983.
- [13] M. Han and T. Kanade, "Reconstruction of a scene with multiple linearly moving objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, vol. 2, pp. 542–549.
- [14] M. Han and T. Kanade, "Creating 3D models with uncalibrated cameras," in *Proc. IEEE Comput. Soc. Workshop Appl. Comput. Vis.*, 2000, pp. 178–185.
- [15] R. Hartley and F. Schaffalitzky, "Power factorization: 3D reconstruction with missing or uncertain data," in *Proc. Australia-Japan. Adv. Workshop Comput. Vis.*, 2003, pp. 1–9.
- [16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [17] A. Heyden, R. Berthilsson, and G. Sparr, "An iterative factorization method for projective structure and motion from image sequences," *Image Vis. Comput.*, vol. 17, no. 13, pp. 981–991, Nov. 1999.
- [18] S. Mahamud and M. Hebert, "Iterative projective reconstruction from multiple views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, vol. 2, pp. 430–437.
- [19] T. Morita and T. Kanade, "A sequential factorization method for recovering shape and motion from image streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 8, pp. 858–867, Aug. 1997.
- [20] C. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 206–218, Mar. 1997.
- [21] L. Quan, "Self-calibration of an affine camera from multiple views," *Int. J. Comput. Vis.*, vol. 19, no. 1, pp. 93–105, Jul. 1996.
- [22] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, Nov. 1992.
- [23] L. Torresani, A. Hertzmann, and C. Bregler, "Learning non-rigid 3D shape from 2D motion," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.
- [24] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler, "Tracking and modeling non-rigid objects with rank constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2001, vol. 1, pp. 493–500.
- [25] B. Triggs, "Factorization methods for projective structure and motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1996, pp. 845–851.
- [26] R. Vidal and R. Hartley, "Motion segmentation with missing data using power factorization and GPCA," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 2, pp. 310–316.
- [27] G. Wang, Y. Tian, and G. Sun, "Modelling nonrigid object from video sequence under perspective projection," in *Proc. ACII*, 2005, vol. 3784, pp. 64–71.
- [28] J. Xiao, J.-X. Chai, and T. Kanade, "A closed-form solution to non-rigid shape and motion recovery," in *Proc. Eur. Conf. Comput. Vis.*, 2004, vol. 4, pp. 573–587.
- [29] J. Xiao and T. Kanade, "Non-rigid shape and motion recovery: Degenerate deformations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, vol. 1, pp. 668–675.
- [30] J. Xiao and T. Kanade, "Uncalibrated perspective reconstruction of deformable structures," in *Proc. Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1075–1082.
- [31] J. Yao and W. Cham, "Feature matching and scene reconstruction from multiple widely separated views," Dept. Electron. Eng., Chinese Univ. Hong Kong, Hong Kong, Tech. Rep., 2005.



**Guanghai Wang** received the M.S. degree in control theory and applications from the Jilin University of Technology, Changchun, China, in 2000 and the Ph.D. degree in pattern recognition and intelligent systems from the Chinese Academy of Sciences, Beijing, in 2004.

From March 2003 to March 2004, he was a Research Assistant with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong, where he was also a Visiting Scholar from March 2005 to September 2005. He is currently

a Research Fellow with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. He is also with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, and the Department of Control Engineering, Aviation University, Changchun, China. His current research interests include structure from motion, camera calibration, artificial intelligence, and robot localization and navigation. He is the author of about 30 peer-reviewed papers published in major journals and conference proceedings.



**Q. M. Jonathan Wu** (M'92) received the Ph.D. degree in electrical engineering from the University of Wales, Cardiff, U.K., in 1990.

In 1995, he joined the National Research Council of Canada, Ottawa, ON, Canada, where he was a Senior Research Officer and Group Leader. He is currently a full Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON. He is a holder of the Canada Research Chair in automotive sensors and sensing systems. He is the author of over 100 published scientific papers in the areas of computer vision, neural networks, fuzzy systems, robotics, microsensors and actuators, and integrated microsystems. His current research interests include 3-D image analysis, active video object extraction, vision-guided robotics, sensor analysis and fusion, wireless sensor networks, and integrated microsystems.

Dr. Wu is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS.