

Perspective Nonrigid Shape and Motion Recovery

Richard Hartley¹ and René Vidal²

¹ Australian National University and NICTA, Canberra, ACT, Australia

² Center for Imaging Science, Johns Hopkins University, Baltimore, MD, USA

Abstract. We present a closed form solution to the nonrigid shape and motion (NRSM) problem from point correspondences in multiple perspective uncalibrated views. Under the assumption that the nonrigid object deforms as a linear combination of K rigid shapes, we show that the NRSM problem can be viewed as a reconstruction problem from multiple projections from \mathbb{P}^{3K} to \mathbb{P}^2 . Therefore, one can linearly solve for the projection matrices by factorizing a multifocal tensor. However, this projective reconstruction in \mathbb{P}^{3K} does not satisfy the constraints of the NRSM problem, because it is computed only up to a projective transformation in \mathbb{P}^{3K} . Our key contribution is to show that, by exploiting algebraic dependencies among the entries of the projection matrices, one can upgrade the projective reconstruction to determine the affine configuration of the points in \mathbb{R}^3 , and the motion of the camera relative to their centroid. Moreover, if $K \geq 2$, then either by using calibrated cameras, or by assuming a camera with fixed internal parameters, it is possible to compute the Euclidean structure by a closed form method.

1 Introduction

Structure from motion (SfM) refers to the problem of reconstructing a 3-D rigid scene from multiple 2-D images taken by a moving camera. This is a well studied problem in computer vision (see for instance [1,2]), which has found numerous applications in image-based modeling, human-computer interaction, robot navigation, vision-based control, etc.

A fundamental limitation of classical SfM algorithms is that they cannot be applied to scenes containing nonrigid objects, such as scenes containing articulated motions, facial expressions, hand gestures, etc. This has motivated the development of a family of methods where a moving *affine calibrated* camera observes a nonrigid shape that deforms as a linear combination of K rigid shapes with time varying coefficients [3,4,5,6,7,8]. This assumption allows one to recover nonrigid shape and motion (NRSM) using extensions of the classical rigid factorization algorithm of Tomasi and Kanade [9]. For instance, Bregler et al. [5] use multiple matrix factorizations to enforce orthonormality constraints on camera rotations. Brand [3] uses a non-linear optimization method called flexible factorization. Torresani et al. [7] use a trilinear optimization algorithm that alternates between the computation of shape bases, shape coefficients, and camera rotations. Xiao et al. [8] provide a characterization of the space of ambiguous solutions as well as a closed form solution by enforcing additional *shape constraints* on the shape bases. Their solution not only applies to shapes of full rank three, but can also be extended to degenerate rank one and two shapes, as shown in [10].

An important assumption made by these approaches is that the projection model is *affine* and the camera is *calibrated*. One way of extending affine methods to the projective case is to alternate between the estimation of the projective depths and the estimation of shape and motion, similarly to the Sturm and Triggs algorithm [11]. This approach was indeed explored in [12] for the NRSM problem. However, it is well known that iterative schemes are often very sensitive to initialization. In the rigid case the projective depths can be initialized using algebraic methods based on two-view geometry. In the nonrigid case, the situation is obviously not as straightforward, and hence the method of [12] simply assumes the initial depths to be all equal to one. To the best of our knowledge, the only existing algebraic solution to the perspective NRSM problem can be found in [13], where it is shown that the problem is solvable for a number of views F in the range $(3K + 1)/2 \leq F \leq (3K + 1)$. However, the algorithm for computing shape and motion relies on the factorization of a quintifocal tensor, and is applicable only in the case of two shape bases seen in five calibrated perspective views.

In this paper, we present a closed form solution to nonrigid shape and motion recovery for an arbitrary number of shape bases K and an arbitrary number F of perspective uncalibrated views in the range $(3K + 1)/2 \leq F \leq (3K + 1)$. Our solution exploits the fact that the NRSM problem can be viewed as a reconstruction problem from \mathbb{P}^{3K} to \mathbb{P}^2 where the projection matrices have a particular structure. As shown in [14], the camera projections associated with any reconstruction problem from \mathbb{P}^n to \mathbb{P}^m can be computed in closed form from the factorization of a multifocal tensor. However, the projection matrices computed by this method do not necessarily conform with the particular structure of the NRSM problem, because they are computed up to a projective transformation in \mathbb{P}^{3K} only. The main contribution of our work is to show that one can solve for the projective transformation, and hence for the camera matrices, shape basis, and shape coefficients, in closed form using linear algebraic techniques which do not require the use of iteration. More specifically, we show that the NRSM problem can be solved as follows:

1. Linearly compute a multifocal tensor from point correspondences in multiple views of a nonrigid object.
2. Factorize the multifocal tensor into $\mathbb{P}^{3K} \rightarrow \mathbb{P}^2$ projection matrices, defined up to a common projective transformation of \mathbb{P}^{3K} .
3. Compute a normalizing projective transformation by enforcing internal constraints on the projection matrices.
4. Compute the camera matrices, shape basis and shape coefficients from the normalized projection matrices.

Using this method, we find the following results, when the number of shape bases is $K \geq 2$.

1. The structure of the point set may be determined in each frame, up to an affine transformation common to all frames. This is in contrast with the classic reconstruction problem with a single shape basis, where the structure may be computed only up to a projective transformation.
2. If the cameras are calibrated, or have constant internal parameters, then the Euclidean shape may be determined by closed form or linear techniques.

3. Since the points are potentially moving (within the space spanned by the K shape bases), it is possible to determine the camera motion only relative to the moving points, and up to an individual scaling in each frame. This is the only ambiguity of the reconstruction (other than a choice of the affine or Euclidean coordinate frame). If the points are assumed to be centred at the origin, then the camera motion is uniquely determined apart from a scale within the affine or Euclidean coordinate frame.

Paper Contributions. This paper gives the first non-iterative solution for the general nonrigid perspective structure-from-motion problem. Because of the deterministic nature of the algorithm, it is guaranteed to find the correct solution at least for noise-free data. This is not the case with previous iterative algorithms. (For the difficulties involved with such iterative methods, see for instance [15].) Further, our analysis allows us to discover the fundamental ambiguities and limitations of NRSM, both in the affine and perspective cases. Our results clarify and complete the previous results on the ambiguities of affine NRSM given in [16,13].

2 Nonrigid Shape and Motion Problem

Notation. We make extensive use of the Kronecker or tensor product $A \otimes B$, where A and B are matrices. This tensor product is given by

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix},$$

where the a_{ij} are the elements of A . A basic property is that $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ whenever the dimensions are compatible so that this equality makes sense. Consequently, if A and B are square, then $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.

We use the notation $\text{stack}(\dots)$ to represent the matrix or vector created by stacking its arguments (matrices or vectors) vertically.

Bold font (\mathbf{X}, \mathbf{x}) is used to represent vectors (one-dimensional arrays) and typewriter font (A, W, \dots) to represent matrices (two-dimensional arrays). Given a homogeneous vector, such as \mathbf{x} or \mathbf{X} , the corresponding non-homogeneous vector is denoted with a hat, such as $\hat{\mathbf{x}}$ or $\hat{\mathbf{X}}$. Notation such as $\Pi_{a:b}$ represents rows a to b of Π .

Finally, for inline representation of simple matrices, we use the notation $[a, b ; c, d]$, where the elements are listed in row major order, rows separated by a semi-colon.

Problem statement. Let $\{\mathbf{x}_{fp} \in \mathbb{P}^2 \mid p = 1, \dots, P ; f = 1, \dots, F\}$ be the perspective projections of P (possibly moving) 3-D points $\{\mathbf{X}_{fp} \in \mathbb{P}^3\}$ onto F frames from a moving camera. Let $P_f = [M_f \ \mathbf{t}_f] \in \mathbb{R}^{3 \times 4}$ be the *camera matrix* associated with frame f . Then

$$\lambda_{fp} \mathbf{x}_{fp} = P_f \mathbf{X}_{fp}, \tag{1}$$

where λ_{fp} is an unknown scale factor, called *projective depth*. It follows that

$$W = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} \cdots \lambda_{1P}\mathbf{x}_{1P} \\ \vdots \\ \lambda_{F1}\mathbf{x}_{F1} \cdots \lambda_{FP}\mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} P_1\mathbf{X}_1 \\ \vdots \\ P_F\mathbf{X}_F \end{bmatrix}, \quad (2)$$

where $\mathbf{X}_f = [\mathbf{X}_{f1} \ \mathbf{X}_{f2} \ \cdots \ \mathbf{X}_{fP}] \in \mathbb{R}^{4 \times P}$ is called the *structure matrix* and is formed from the homogeneous coordinates of all the P points in the f -th frame.

The *structure from motion problem* (SfM) refers to the problem of recovering the camera matrices P_f , and the structure matrices \mathbf{X}_f from measurements of the image point trajectories \mathbf{x}_{fp} . Without some restriction on the moving 3-D points, the SfM problem is of course not solvable.

When the P points lie on a rigid stationary object, the structure matrices are equal, that is $\mathbf{X}_1 = \mathbf{X}_2 = \cdots = \mathbf{X}_F = \mathbf{X}$. Hence, given the depths one can factorize W into a motion matrix $\Pi \in \mathbb{R}^{3F \times 4}$ and a structure matrix $\mathbf{X} \in \mathbb{R}^{4 \times P}$ as $W = \Pi\mathbf{X}$. This rank constraint has been the basis for all factorization-based algorithms, e.g. [9,11]. In fact, one can solve the SfM problem by alternating between the estimation of the depths, and the estimation of motion and structure [17], though care must be taken to avoid converging to trivial solutions [15].

In this paper we study the case where the 3-D points lie on a nonrigid object, thereby allowing the 3-D points \mathbf{X}_{fp} to move as a function of time. As suggested in [3,4,5,6,7], we assume that the P points deform as a linear combination of a fixed set of K rigid shape bases with time varying coefficients. That is, $\hat{\mathbf{X}}_f = \sum_{k=1}^K c_{fk}\hat{\mathbf{B}}_k$, where the matrix $\hat{\mathbf{X}}_f = [\hat{\mathbf{X}}_{f1} \cdots \hat{\mathbf{X}}_{fP}] \in \mathbb{R}^{3 \times P}$ is the *object shape* at frame f , the matrices $\{\hat{\mathbf{B}}_k = [\hat{\mathbf{B}}_{k1} \cdots \hat{\mathbf{B}}_{kP}] \in \mathbb{R}^{3 \times P}\}$ are the *shape bases* and $\{c_{fk} \in \mathbb{R}\}$ are the *shape coefficients*.

Under this deformation model, the projection equation (1) can be rewritten as a projection equation from \mathbb{P}^{3K} to \mathbb{P}^2 of the form

$$\lambda_{fp}\mathbf{x}_{fp} = M_f \sum_{k=1}^K (c_{fk}\hat{\mathbf{B}}_{kp}) + \mathbf{t}_f = [c_{f1}M_f \cdots c_{fK}M_f \ \mathbf{t}_f] \begin{bmatrix} \hat{\mathbf{B}}_{1p} \\ \vdots \\ \hat{\mathbf{B}}_{Kp} \\ 1 \end{bmatrix} = \Pi_f \mathbf{B}_p. \quad (3)$$

Therefore, the matrix of image measurements W in (2) can be factorized into the product of a motion matrix $\Pi \in \mathbb{R}^{3F \times (3K+1)}$ and a basis matrix $\mathbf{B} \in \mathbb{R}^{(3K+1) \times P}$ as

$$W = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} \cdots \lambda_{1P}\mathbf{x}_{1P} \\ \vdots \\ \lambda_{F1}\mathbf{x}_{F1} \cdots \lambda_{FP}\mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} c_{11}M_1 \cdots c_{1K}M_1 & \mathbf{t}_1 \\ \vdots & \vdots \\ c_{F1}M_F \cdots c_{FK}M_F & \mathbf{t}_F \end{bmatrix} \begin{bmatrix} \hat{\mathbf{B}}_1 \\ \vdots \\ \hat{\mathbf{B}}_K \\ \mathbf{1}^\top \end{bmatrix} = \Pi \mathbf{B}. \quad (4)$$

Note that the motion matrix Π has the form $\Pi = [\text{diag}(M_1, \dots, M_F)(\mathbf{C} \otimes \mathbf{I}_3) \mid \mathbf{t}]$, where $\mathbf{t} = \text{stack}(\mathbf{t}_1, \dots, \mathbf{t}_F)$. Furthermore, given the factorization in this form, we may read off the camera matrices $P_f = [M_f \mid \mathbf{t}_f]$ and the 3-D points from

$$\text{stack}(\hat{\mathbf{X}}_1, \dots, \hat{\mathbf{X}}_F) = (\mathbf{C} \otimes \mathbf{I}_3)\text{stack}(\hat{\mathbf{B}}_1, \dots, \hat{\mathbf{B}}_K). \quad (5)$$

Note here, however, a basic ambiguity: the individual projection matrices can be determined from Π only up to independent scale factors, since scaling M_f can be balanced by a corresponding inverse scaling to the corresponding row of the coefficient matrix C .

Iterative methods. The rank constraint implied by (4) has been the basis for existing projective NRSM algorithms. As shown in [12], when the depths are known, the shape coefficients and shape basis may be computed from the factorization of W using a factorization technique similar to that in [8] for affine cameras. In [12], they solve the perspective reconstruction problem by alternately solving for the depths and the shape and motion parameters, in a similar way to [17]. In this paper, we seek an alternative purely algebraic solution to the problem that does not rely on any iterative optimization. In doing so, we are able to determine exactly what it is possible to compute uniquely, and what are the unavoidable ambiguities.

3 Nonrigid Shape and Motion Recovery

In this section, we propose a closed form solution to the NRSM problem from multiple perspective views. The key to our approach is to observe from equation (3) that the NRSM problem is a particular case of a reconstruction problem from \mathbb{P}^{3K} to \mathbb{P}^2 . This interpretation will allow us to solve directly for the motion matrix Π in (4) up to a projective transformation in $\mathbb{P}^{3K \times 3K}$, as we will show in §3.1. We will then propose an extremely simple linear algorithm for recovering the unknown projective transformation, hence the original camera matrices in $\mathbb{P}^{3 \times 2}$, shape bases, and shape coefficients.

3.1 Recovery of the Projection Matrices $\mathbb{P}^{3K} \rightarrow \mathbb{P}^2$

While factorization methods such as [9,18,8] are commonly used in affine reconstruction problems involving affine or orthographic cameras, they are not so useful for reconstruction from perspective cameras, since they require iterative estimation of the depth values [11,17]. For such problems an alternative is to use tensor-based methods. The standard methods used for rigid structure and motion problems involve the fundamental matrix, trifocal or quadrifocal tensors [1]. It was shown in [14] that these tensor based methods can be extended to projections between projective spaces \mathbb{P}^n and \mathbb{P}^m of arbitrary dimensions with $n > m$. We will rely heavily on this method. In the particular case of relevance to the current problem, $n = 3K$ and $m = 2$.

In brief, given a suitable number of projections $\mathbb{P}^n \rightarrow \mathbb{P}^m$, we may compute a tensor that relates the coordinates of matching image points \mathbf{x}_{fp} in \mathbb{P}^m . This tensor may be computed linearly, and from it the set of projection matrices Π_f may be extracted using non-iterative techniques. Subsequently, points \mathbf{B}_p in \mathbb{P}^n may be computed by triangulation such that $\lambda_{fp}\mathbf{x}_{fp} = \Pi_f\mathbf{B}_p$. Here, points \mathbf{B}_p and the corresponding image points \mathbf{x}_{fp} are expressed in homogeneous coordinates and the λ_{fp} are unknown scale factors, which do not need to be known for this reconstruction to be computed.

One may stack the projection matrices Π_f as well as the points \mathbf{x}_{fp} on top of each other and form an equation

$$W = \text{stack}(\Pi_1, \dots, \Pi_F)[\mathbf{B}_1 \dots \mathbf{B}_P] = \Pi B, \quad (6)$$

which is of exactly the same form as the type of decomposition formulated in (4). It was shown in [14] that this factorization ΠB is unique except for the (non-significant) multiplication of each of the camera matrices Π_f by an arbitrary scale factor k_f and except for modifying ΠB to $\Pi A A^{-1} B$, where $A \in \mathbb{R}^{(3K+1) \times (3K+1)}$ is an invertible matrix. This is exactly analogous to the affine ambiguity inherent in affine factorization algorithms. However, here the matrix A represents a *projective* transformation, since we are using homogeneous coordinates. Thus, using tensors, we may achieve a similar factorization in the projective case as that computed by linear methods in the affine case. The only difference is that the number of views that may be used is restricted.

In the case of projective nonrigid motion, the image projection may be expressed as $\Pi_f : \mathbb{P}^{3K} \rightarrow \mathbb{P}^2$ and a factorization $W = \Pi B$ may be computed from any number of views between $(3K + 1)/2$ and $3K + 1$ (see [13]) using the tensor method. However, this does not produce a solution of the particular required form, given in (4) and it is impossible to extract the individual $\mathbb{P}^3 \rightarrow \mathbb{P}^2$ projection matrices immediately. We need to do some more work to find a matrix A that transforms each Π_f into the correct form. However, as will be seen, we gain from this since the remaining ambiguity is only affine or Euclidean (for calibrated cameras). Thus affine or Euclidean reconstruction is possible. How we enforce the correct form on the projection matrices Π_f will be the main focus of the rest of this paper.

3.2 Recovery of the Projective Transformation

As a result of our analysis in the previous subsection, at this point we have computed a projection matrix $\Pi \in \mathbb{R}^{3F \times (3K+1)}$. Our task is to transform this projection matrix by a matrix $A \in \mathbb{R}^{(3K+1) \times (3K+1)}$ such that ΠA is of the form $[\text{diag}(M_1, \dots, M_F)(C \otimes I_3) \mid \mathbf{t}]$ given in (4). To that end, we use the following steps.

Step 1. We assume that the matrix Π is full rank and, without loss of generality, that the top $3K \times 3K$ block of Π is non-singular. Hence, if we multiply Π by A_1 , where $A_1^{-1} = [\Pi_{1:K}; \mathbf{0}^\top, 1]$, we arrive at a matrix of a new form in which

$$(\Pi A_1)_{1:K} = [I_K \otimes I_3 \ \mathbf{0}]. \tag{7}$$

At this point, the first K row-blocks (in the block-representation) of ΠA_1 are of the desired form, but the remaining rows may be arbitrary.

Step 2. We multiply the matrix ΠA_1 by the block-diagonal matrix A_2 , given by $A_2 = \text{diag}(M_{K+1,1}, \dots, M_{K+1,K}, 1)^{-1}$. Here the matrices $M_{K+1,k}$, $k = 1, \dots, K$, are obtained from the $(K + 1)$ -st row-block of ΠA_1 . Under a suitable assumption of genericity, these matrices will be non-singular, as will be seen in the proof of Theorem 1 below. This results in a matrix such that

$$(\Pi A_1 A_2)_{1:K+1} = \text{diag}(M_1, \dots, M_K, I_3) \begin{bmatrix} I_K \otimes I_3 & \mathbf{0} \\ I_3 \cdots I_3 & \mathbf{t}_{K+1} \end{bmatrix}, \tag{8}$$

where now the first $K + 1$ row-blocks of $\Pi A_1 A_2$ are in the desired form and the $(K + 1)$ -st row-block contains only identity matrices.

Step 3. We are left with enforcing that the remaining $F - K - 1$ row-blocks of $\Pi A_1 A_2$ have the desired algebraic structure by multiplying by a further matrix A_3 . In order to

preserve the block diagonal structure of the top $3K \times 3K$ block of $\Pi A_1 A_2$, we can only multiply by a matrix A_3 whose top $3K \times 3K$ is also block diagonal. Therefore, we seek a matrix $A_3 = [\text{diag}(N_1, \dots, N_K), \mathbf{0}; \mathbf{s}_1^\top \cdots \mathbf{s}_K^\top, 1]$. In order for the $(K + 1)$ -st row-block to remain as identity matrices, it is easily verified that $N_k = I_3 - \mathbf{t}_{K+1} \mathbf{s}_k^\top$, so we need only compute the values of each \mathbf{s}_k .

For some $f > K + 1$, let M_{fk} be the matrix in position (f, k) of $\Pi A_1 A_2$, and \mathbf{t}_f be the vector in position $(f, K + 1)$. By multiplication by A_3 , M_{fk} is transformed to $M'_{fk} = M_{fk}(I_3 - \mathbf{t}_{K+1} \mathbf{s}_k^\top) + \mathbf{t}_f \mathbf{s}_k^\top$, which we may write as $M_{fk} + \mathbf{v}_{fk} \mathbf{s}_k^\top$, where the only unknown is \mathbf{s}_k . Our requirement on the form of the resulting matrix $\Pi A_1 A_2 A_3$ is that for each $f > K + 1$ and $k > 1$ we have $c_{fk}^{-1} M'_{fk} = c_{f1}^{-1} M'_{f1}$ for some coefficients c_{fk} . This leads to equations

$$c_{fk}^{-1} (M_{fk} + \mathbf{v}_{fk} \mathbf{s}_k^\top) = c_{f1}^{-1} (M_{f1} + \mathbf{v}_{f1} \mathbf{s}_1^\top) \tag{9}$$

in which the unknowns are the vectors $\mathbf{s}_1, \dots, \mathbf{s}_K$ and the coefficients c_{fk}^{-1} . Note that these equations are not linear. However, they may be written in the form

$$\frac{c_{f1}}{c_{fk}} (\mathbf{m}_{fk} + \mathbf{V}_{fk} \mathbf{s}_k) = \mathbf{m}_{f1} + \mathbf{V}_{f1} \mathbf{s}_1 \tag{10}$$

for suitable known matrices $\mathbf{V}_{f1}, \mathbf{V}_{fk} \in \mathbb{R}^{9 \times 3}$ and vectors $\mathbf{m}_{f1}, \mathbf{m}_{fk} \in \mathbb{R}^9$. Multiplying this equation by a matrix $\Gamma_{fk} \in \mathbb{R}^{5 \times 9}$ such that $\Gamma_{fk} \mathbf{m}_{fk} = 0$ and $\Gamma_{fk} \mathbf{V}_{fk} = 0$ leads to $5(F - K)K$ linear equations in \mathbf{s}_1 of the form $\Gamma_{fk} \mathbf{V}_{f1} \mathbf{s}_1 = -\Gamma_{fk} \mathbf{m}_{f1}$. Once \mathbf{s}_1 is known, one may rearrange (9) so that the equations become linear in the remaining \mathbf{s}_k and coefficients c_{fk}/c_{f1} . Notice that there are many alternative ways of solving the equations in (9). Experimentation showed the the current method performs on par with other techniques.

3.3 Recovery of the Camera Matrices and of the Nonrigid Shape

After applying the transformation $A_1 A_2 A_3$ to Π , we obtain a matrix that is nominally of the desired form $\Pi' = [\text{diag}(M_1, \dots, M_F)(C \otimes I_3) \mid \mathbf{t}]$. Indeed, the first $K + 1$ row blocks will be exactly of the desired form. However, because of measurement noise, the remaining blocks, corresponding to projections Π'_f , $f = K + 2, \dots, F$, will not be, so we need to correct this.

Consider a fixed frame $f > K + 1$, and let $\Pi'_f = [M_{f1}, \dots, M_{fK} \mid \mathbf{t}_f]$. This matrix will be nominally of the form $[c_{f1} M_f, \dots, c_{fK} M_f \mid \mathbf{t}_f]$, but will be corrupted by noise. Each correspondence $M_{fk} = c_{fk} M_f$ may be seen as a set of 9 bilinear equations in the variables c_{fk} and M_f . We arrange the entries of all the M_{fk} into a matrix $E_{9 \times K}$, one column for the entries of each M_{fk} . The set of all equations (for a fixed f) may then be written as $E_{9 \times K} = \mathbf{m}_f \mathbf{c}_f^\top$ where $\mathbf{c}_f^\top = (c_{f1}, \dots, c_{fK})$ and \mathbf{m}_f is the vector of entries of the matrix M_f . We can then solve for \mathbf{m}_f and \mathbf{c}_f by computing the best rank-1 approximation of $E_{9 \times K}$. Vectors \mathbf{m}_f and \mathbf{c}_f are computed up to a reciprocal scale ambiguity, which is all that is possible, as remarked previously.

By this method, we compute all M_f for $f > K + 1$ and the corresponding shape coefficients c_{fk} . The resulting matrix Π'' will be exactly in the required true form. A solution for the shape bases B and projective depths λ_{fp} is then obtained by linear triangulation using equation (4). Finally, the nonrigid shape is given by $\hat{X} = (C \otimes I_3) \hat{B}$, and the camera matrices are $P_f = [M_f \mid \mathbf{t}_f]$.

4 Algorithm Justification

In the previous section, a method was given for transforming the matrix Π to the required form given in (4). However, there is no justification given that the resulting camera matrices and nonrigid shape will correspond to the ground truth. For instance, we did not show that the equations in (9) have a unique solution for the vectors s_k , hence the matrix A_3 may not be unique. In this section we show that, under suitable assumptions, the resulting product $\Pi A_1 A_2 A_3$ is unique.

To that end, we make various definitions. A matrix $\Pi = \text{stack}(\Pi_1, \dots, \Pi_F)$ is said to be in *true form* if it is of the form $[\text{diag}(M_1, \dots, M_K)(C \otimes I_3) \mid \mathbf{t}]$, where all matrices M_f are invertible. A matrix is said to be in *canonical form* if it is of the form given in (8) with $\mathbf{t}_{K+1} \neq \mathbf{0}$, and in *true-canonical form* if it satisfies both conditions. We now state an important result.

Theorem 1. *Let $\Pi = \text{stack}(\Pi_1, \dots, \Pi_F)$ be a motion matrix, and assume that there exists A such that ΠA is in true form. Subject to possible reordering of the rows Π_f of Π and under suitable assumptions of genericity, there exists a matrix A' such that $\Pi A'$ is in true-canonical form. Furthermore, the true-canonical form is unique (for a fixed ordering of the rows Π_f).*

The meaning of the assumption of genericity will be made clear in the proof. Broadly speaking, it means that the motion of the camera is sufficiently general and independent of the shape deformation, and that the shape space is indeed K -dimensional, spanned by the K shape bases. In addition we assume that we can find $K + 1$ frames such that no K of the corresponding shape matrices \widehat{X}_f are linearly dependent. We will order the frames so that these $K + 1$ frames are numbered $1, \dots, K + 1$. The first K shapes will serve as the K shape bases.

Granted the truth of this theorem, the algorithm in the previous section will lead to the correct and unique solution. In particular, the matrix A_3 used in step 3 must lead to a solution in true-canonical form. Therefore, the set of linear equations solved will have a unique solution.

The proof of Theorem 1 given here is of necessity brief. In a possible expanded version of this paper we can give more details, and in particular an exact analysis of the required genericity conditions.

Existence. For the existence part of the proof, it is clear that it is enough to show the existence of a matrix A' that transforms a true form matrix to one in true-canonical form. The steps of the proof follow the steps 1–2 of the algorithm of §3.2, except that we start with a matrix of the form $\Pi = [\text{diag}(M_1, \dots, M_F)(C \otimes I_3) \mid \mathbf{t}]$.

In the first step, the required transformation matrix will be of the form $[C_{1:K} \otimes I_3, \mathbf{t}_{1:K}; \mathbf{0}^\top, 1]^{-1}$. This will exist as long as $C_{1:K}$ is invertible, which is the generic case, meaning that the shape matrices $\{\widehat{X}_f \mid f = 1, \dots, K\}$ span the complete shape space. If not, then we can reorder the frames so that this is so.

After the first step, the matrix remains in true form. Therefore, row $K + 1$ will be of the form $\Pi_{K+1} = [c_1 M, \dots, c_K M, \mathbf{t}_{K+1}]$. We require that $\mathbf{t}_{K+1} \neq \mathbf{0}$, otherwise we rearrange the matrices Π_f so that this is so. The transformation matrix $A_2 = \text{diag}(c_1 M, \dots, c_K M, 1)^{-1}$ will transfer the matrix into canonical form. Observe that M is

invertible by our assumption that the matrix is in true form. If one of the c_k is zero, this means that the shape $\widehat{\mathbf{X}}_{K+1}$ at frame $K+1$ is in a space spanned by a proper subset of the shape bases $\widehat{\mathbf{B}}_k$, which we rule out by an appeal to genericity. Since we started with a matrix in true form, after these two steps, the matrix is now in true-canonical form. This completes the existence part of the proof.

Uniqueness. For the uniqueness part of the proof, consider a possible transformation \mathbf{A}_3 , which transforms a matrix Π in true-canonical form to $\Pi' = \Pi\mathbf{A}_3$, also in true-canonical form. By the same argument as in §3.2, the matrix \mathbf{A}_3 must be of the form $\mathbf{A}_3 = [\text{diag}(N_1, \dots, N_K), \mathbf{0}; \mathbf{s}_1^\top \dots \mathbf{s}_K^\top, \lambda]$, with $\lambda \neq 0$ and $N_k = \mathbf{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_k^\top$. Applying this transform to the f -th row-block $\Pi_f = [c_{f1}\mathbf{M}_f, \dots, c_{fK}\mathbf{M}_f, \mathbf{t}_f]$ of Π , results in a new block with entries $M'_{fk} = c_{fk}\mathbf{M}_f(\mathbf{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_k^\top) + \mathbf{t}_f\mathbf{s}_k^\top$. Since this new row-block must be in true form, for any two indices $1 \leq j, k \leq K$, there must exist constants c'_{fk} and c'_{fj} such that $c'_{fk}M'_{fk} = c'_{fj}M'_{fj}$. This leads to

$$c'_{fj} (c_{fk}\mathbf{M}_f(\mathbf{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_k^\top) + \mathbf{t}_f\mathbf{s}_k^\top) = c'_{fk} (c_{fj}\mathbf{M}_f(\mathbf{I}_3 - \mathbf{t}_{K+1}\mathbf{s}_j^\top) + \mathbf{t}_f\mathbf{s}_j^\top), \quad (11)$$

which may be rewritten as

$$(c'_{fj}c_{fk} - c_{fj}c'_{fk})\mathbf{M}_f = c'_{fk}(\mathbf{t}_f - c_{fj}\mathbf{M}_f\mathbf{t}_{K+1})\mathbf{s}_j^\top - c'_{fj}(\mathbf{t}_f - c_{fk}\mathbf{M}_f\mathbf{t}_{K+1})\mathbf{s}_k^\top. \quad (12)$$

Since \mathbf{M}_f is a matrix of rank 3, and the two terms on the right are of rank 1, this is impossible, unless $c'_{fj}c_{fk} - c_{fj}c'_{fk} = 0$ and

$$c_{fj}(\mathbf{t}_f - c_{fk}\mathbf{M}_f\mathbf{t}_{K+1})\mathbf{s}_k^\top = c_{fk}(\mathbf{t}_f - c_{fj}\mathbf{M}_f\mathbf{t}_{K+1})\mathbf{s}_j^\top, \quad (13)$$

where we have used the fact that $c_{fj}/c_{fk} = c'_{fj}/c'_{fk}$ to replace c'_{fk} by c_{fk} . Since the factorization of a rank-1 matrix is unique up to scaling the two factors, this relationship means that one of the following conditions must be true.

1. $\mathbf{t}_{K+1} = \mathbf{0}$. However, this is ruled out by hypothesis.
2. $\mathbf{t}_f = \mathbf{0}$. If this is so for all $f > K+1$ this implies that the position of these cameras are dependent on the position of the first K cameras. This is not a generic camera motion.
3. The vectors \mathbf{t}_f and $\mathbf{M}_f\mathbf{t}_{K+1}$ are linearly dependent. However, if this is true for all $f > K+1$, then it implies that $\mathbf{M}_f^{-1}\mathbf{t}_f$ is a multiple of \mathbf{t}_{K+1} . This is a non-generic camera motion, since $-\mathbf{M}_f^{-1}\mathbf{t}_f$ is the position of the camera at frame f .
4. Finally, if $\mathbf{M}_f\mathbf{t}_{K+1}$ and \mathbf{t}_f are linearly independent, then in order for the vectors $\mathbf{t}_f - c_{fk}\mathbf{M}_f\mathbf{t}_{K+1}$ and $\mathbf{t}_f - c_{fj}\mathbf{M}_f\mathbf{t}_{K+1}$ to differ only by a scale factor, it is necessary that $c_{fj} = c_{fk}$. This means that the f -th row-block of Π must be of the form $\Pi_f = [c_f\mathbf{I}_3, c_f\mathbf{I}_3, \dots, c_f\mathbf{I}_3, \mathbf{t}_f]$, with all the coefficients c_{fk} the same along this row. Apart from a constant scale, these are the same set of coefficients as for the $(K+1)$ -st row-block, which means that the shape of the scene is the same for this frame as for frame $K+1$. If this is true for all $f > K+1$, it implies that the object has the same shape, and does not deform for all of the frames $K+1$ to F .

If on the other hand, the deformation of the scene is generic, then none of the conditions given above can be fulfilled. In this case the canonical form is uniquely determined. This concludes the proof.

5 Affine and Euclidean Shape Reconstruction

Having established the correctness of the proposed reconstruction algorithm, we now turn to the question of uniqueness of the reconstructed shapes. In particular, we show that, even though there are ambiguities in the reconstruction of shape bases and shape coefficients, the reconstructed shape is actually unique. Moreover, we will show that when $K \geq 2$, one recover the shape up to an affine transformation, which represents a significant improvement with respect to the case $K = 1$, where one can only recover the shape up to a projective transformation.

Affine Shape Reconstruction. As a consequence of the proof of the uniqueness result of Theorem 1, if Π and Π' are two matrices in true form, then they are related by $\Pi' = \Pi A$, where A is some product of matrices of the form

$$A_1 = \begin{bmatrix} C' \otimes I_3 & 0 \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} I_3 \otimes I_3 & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} I_3 \otimes M & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad (14)$$

(not the same as the matrices A_1, A_2, A_3 in §3.2) and their inverses. Here C' has dimension a $K \times K$, and the product matrix A may be written as $A = [C' \otimes M, \mathbf{t}; \mathbf{0}^\top, 1]$.

In the factorization of $\widehat{W} = \Pi B$, the inverse transformations are applied to B . The first of these transformations causes a change of the shape bases through linear combinations. However, it does not change the shape of the points \mathbf{X}_{fp} . To see this, observe that the corresponding change to \widehat{B} is to replace it by $(C' \otimes I_3)^{-1} \widehat{B}$. At the same time, the coefficients in the representation (4) of Π are multiplied by C' . However, from (5) X is unchanged by this operation, since $\widehat{X} = (C \otimes I_3) \widehat{B} = (C \otimes I_3)(C' \otimes I_3)(C' \otimes I_3)^{-1} \widehat{B}$ so the matrix $(C' \otimes I_3)$ cancels with its inverse, leaving \widehat{X} unchanged.

The other two transformations effect an affine transformation of the shape bases. By an application of the transformation A_2 each of the shape bases may be translated so that the points it consists of have their centroid at the origin. The resulting reconstruction will be called “centred”. Since each of the shape bases is centred at the origin, so will the sets of points X_f at any other frame, since they are linear combinations of the shape bases. If desired, the reconstruction ΠB may be centred by applying a transformation $\Pi \rightarrow \Pi A_4$ and $B \rightarrow A_4^{-1} B$, where $A_4^{-1} = [I_K \otimes I_3, -\mathbf{w}; \mathbf{0}^\top, 1]$, and $\widehat{\mathbf{w}} = \text{stack}(\widehat{\mathbf{w}}_1, \dots, \widehat{\mathbf{w}}_K)$ is made up of the centroids $\widehat{\mathbf{w}}_k$ of the points $\widehat{B}_{k1}, \dots, \widehat{B}_{kP}$ in each shape basis \widehat{B}_k . A centred reconstruction is unique up to a common linear transformation of all of the shape bases and a corresponding transformation of each of the camera matrices.

We see that the reconstruction is unique except for the following ambiguities.

1. Individual scaling applied to each frame independently, as pointed out in §2.
2. Individual translations of each of the shape bases. Thus, there is a K -fold translation ambiguity in the global reconstruction over all frames. This ambiguity may be removed by computing a centred reconstruction, or by assuming that the first K translations are zero. However, observe that the obtained translations do not necessarily correspond to the ground truth.
3. An overall linear transformation. In the case of calibrated cameras, this is an overall global rotation with respect to a global coordinate frame, which of course can not be determined.

Euclidean Shape Reconstruction. If the cameras are calibrated, we may assume that they are of the form $P_i = [R_i \mid -R_i \mathbf{t}_i]$, where each of the R_i is a rotation. In this case, any initially computed motion matrix Π will be equivalent (under multiplication by A) to a *Euclidean true form* motion matrix (4), meaning all the M_f are rotations. Furthermore, the details of the existence part of Theorem 1 show that Π is then equivalent to a Euclidean true-canonical form matrix. Since the true-canonical form is unique, this shows that Euclidean reconstruction is possible and unique. Furthermore, the algorithm of §3 will naturally lead virtually without modification to the correct Euclidean solution. The details are simple to verify.

Autocalibration. It is interesting and somewhat surprising that for $K \geq 2$ our algorithm gives an affine reconstruction even from uncalibrated cameras. This contrasts with the rigid motion case ($K = 1$), where the reconstruction is only projective. It is easily seen that the affine reconstruction is easily upgraded to a Euclidean reconstruction using standard linear autocalibration techniques. Indeed in the standard method of stratified reconstruction and autocalibration the upgrade from projective to affine reconstruction is difficult, but to upgrade from affine to Euclidean, given mild assumptions on common parameters of the cameras is simple and linear. Details may be found in [1].

6 Experiments

Synthetic Data. We first evaluate our algorithm on synthetically generated data. The $K = 2, 3$ shape bases are generated by randomly drawing P 2-D points uniformly on $[-1, 1] \times [-1, 1]$ and then scaling these points with a depth uniformly drawn in the range of 100-400 units of focal length (u.f.l.). The shape coefficients are also randomly drawn from a uniform distribution in $[-1, 1]$. The 3-D points are then generated by taking a linear combination of the shape bases with the shape coefficients. These points are rotated and translated according to rigid-body motions with a random axis of rotation and a random direction of translation. $F = 4$ to 6 perspective views are obtained by projecting these points onto an image with 1000×1000 pixels. Zero-mean Gaussian noise with a standard deviation of $\sigma \in [0, 2]$ pixels is added to the so-obtained point correspondences.

We evaluate the accuracy of our algorithm with respect to four factors: amount of noise, number of shape bases, number of frames, and number of point correspondences. The performance measures are the angles between the estimated rotations and translations ($\hat{M}_f, \hat{\mathbf{t}}_f$) and the ground truth (M_f, \mathbf{t}_f),

$$\theta_M = \frac{1}{F} \sum_{f=1}^F \arccos((\text{trace}(\hat{M}_f^\top M_f) - 1)/2), \quad \theta_t = \frac{1}{F - K} \sum_{f=K+1}^F \arccos(\hat{\mathbf{t}}_f^\top \mathbf{t}_f),$$

averaged over 1000 trials. Note that, due to the reconstruction ambiguities, we assume that the first K translations are zero. Moreover, recall that the remaining translations are computed up to one scale factor per frame, hence the choice of the angle between the true and estimated translation as an error measure.

Figure 1 shows average error versus amount of noise plots for several choices of the parameters. The number of points is chosen either as $P = 200$, or as twice the

minimum number of points needed to reconstruct the multifocal tensor, i.e., $P \geq 3^F / \prod_{f=1}^F (3 - \pi_f)$, where $\pi_f \in \{1, 2\}$ defines the tensor profile for the f -th frame. As expected, the error increases with the amount of noise and reduces with the number of points correspondences. However, the error does not necessarily reduce as the number of frames increases. When $K = 2$, this can be seen by comparing the curves for $(F, P) = (4, 200)$ and $(F, P) = (4, 82)$, with those for $(F, P) = (5, 200)$ and $(F, P) = (5, 62)$, respectively. This is because the number of unknowns in the multifocal tensor increases exponentially with the number of frames, and a number of nonlinear constraints on the entries of the tensor are neglected when computing and factorizing this tensor using linear techniques. Notice also by comparing the curves for $(K, F, P) = (2, 5, 62)$ and $(K, F, P) = (3, 5, 486)$ that the error reduces as the number of shape bases increases. However, the improvement comes at the cost of increasing the number of points needed. Indeed, when the number of points is increased from 62 to 200, the performances for $(K, F) = (2, 5)$ and $(K, F) = (3, 5)$ are comparable. Finally, notice also that the best existing affine algorithm by Xiao et al. [8] does not perform well on perspective data. This algorithm requires a minimum of $F \geq K^2 + K$ images, so we only evaluate it for $(K, F) = (2, 6)$. Our algorithm, on the other hand, requires a minimum number of frames of $F \geq (3K + 1)/2$.

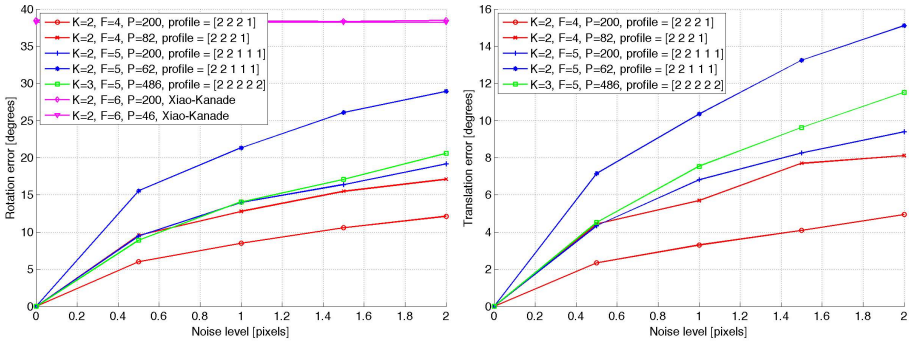


Fig. 1. Reconstruction errors as a function of noise, number of shape basis, number of frames, and number of point correspondences

Real Data. We now test the performance of our algorithm on a video sequence containing two hands moving in front of a static background shown in Fig. 2. The sequence is taken from [13], and consists of $F = 5$ views taken by a moving camera observing 8 points on the static background and another 32 points on the gesturing hands. The 8-point algorithm was used to compute the ground truth camera motion from the 8 static points. We then applied our algorithm and the algebraic algorithm of [13] for $K = 2$ shape basis and $F = 5$ views. We chose the first image as the reference. The errors in the estimation of the rotations are shown in Table 6. Note that our algorithm outperforms that in [13] for 3 out of 4 frames. Translation errors are not computed, as with real sequences one cannot assume zero translations for the first K frames.



Fig. 2. Frames 1-5 of a sequence of gesturing hands used in [13]

Table 1. Errors in the estimation of the rotations for a sequence of gesturing hands

Frame	2	3	4	5
Quintifocal method [13]	0.1644°	5.9415°	2.5508°	54.5860°
Our method	5.5174°	0.6773°	0.1642°	27.1583°

7 Discussion and Conclusions

We have presented several theoretical results pertaining to the nonrigid shape and motion problem from multiple perspective views. Most notably, we have shown that a highly multilinear problem admits a closed form, linear solution. Furthermore, we highlighted several similarities and differences between the rigid and nonrigid case.

While our theoretical framework does provide an algorithm for solving the reconstruction problem, we did not explore algorithmic aspects in this paper, such as robustness to noise or outliers. The reader can see that our proposed method is very simple, involving essentially a series of matrix multiplications. Each one of those steps can be made robust. We argue that the real bottleneck with the current method is not in our approach, but rather in the tensor estimation and factorization approach of [14]. Improving on the robustness of these methods is an interesting avenue for future research.

Acknowledgments. Richard Hartley has been supported by NICTA, which is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. René Vidal has been supported by startup funds from JHU, by grants NSF CAREER IIS-0447739, NSF EHS-0509101, and ONR N00014-05-10836, and by contract JHU APL-934652.

References

1. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge (2004)
2. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: *An Invitation to 3D Vision: From Images to Geometric Models*. Springer, Heidelberg (2003)
3. Brand, M.: Morphable 3D models from video. In: *Conference on Computer Vision and Pattern Recognition*, pp. 456–463 (2001)
4. Brand, M., Bhotika, R.: Flexible flow for 3D nonrigid tracking and shape recovery. In: *Conference on Computer Vision and Pattern Recognition*, pp. 315–322 (2001)
5. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: *Conference on Computer Vision and Pattern Recognition*, pp. 2690–2696 (2000)

6. Torresani, L., Bregler, C.: Space-time tracking. In: European Conference on Computer Vision, pp. 801–812 (2002)
7. Torresani, L., Yang, D., Alexander, E., Bregler, C.: Tracking and modeling non-rigid objects with rank constraints. In: Conference on Computer Vision and Pattern Recognition, pp. 493–500 (2001)
8. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision* 67, 233–246 (2006)
9. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography. *International Journal of Computer Vision* 9, 137–154 (1992)
10. Xiao, J., Kanade, T.: Non-rigid shape and motion recovery: Degenerate deformations. In: Conference on Computer Vision and Pattern Recognition, pp. 668–675 (2004)
11. Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: European Conference on Computer Vision, pp. 709–720 (1996)
12. Xiao, J., Kanade, T.: Uncalibrated perspective reconstruction of deformable structures. In: IEEE International Conference on Computer Vision, pp. 1075–1082 (2005)
13. Vidal, R., Abretske, D.: Nonrigid shape and motion from multiple perspective views. In: European Conference on Computer Vision, pp. 205–218 (2006)
14. Hartley, R., Schaffalitzky, F.: Reconstruction from projections using Grassmann tensors. In: European Conference on Computer Vision, pp. 363–375 (2004)
15. Oliensis, J., Hartley, R.: Iterative extensions of the Sturm/Triggs algorithm: convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 2217–2233 (2007)
16. Aanaes, H., Kahl, F.: Estimation of deformable structure and motion. In: ECCV Workshop on Vision and Modelling of Dynamic Scenes (2002)
17. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably-convergent iterative methods for projective structure from motion. In: Conference on Computer Vision and Pattern Recognition, vol. I, pp. 1018–1025 (2001)
18. Costeira, J., Kanade, T.: A multibody factorization method for independently moving objects. *International Journal of Computer Vision* 29, 159–179 (1998)