Kernel Non-Rigid Structure from Motion

Paulo F. U. Gotardo, Aleix M. Martinez
Department of Electrical and Computer Engineering
The Ohio State University, Columbus (OH), USA
{gotardop,aleix}@ece.osu.edu

Abstract

Non-rigid structure from motion (NRSFM) is a difficult, underconstrained problem in computer vision. The standard approach in NRSFM constrains 3D shape deformation using a linear combination of K basis shapes; the solution is then obtained as the low-rank factorization of an input observation matrix. An important but overlooked problem with this approach is that non-linear deformations are often observed; these deformations lead to a weakened low-rank constraint due to the need to use additional basis shapes to linearly model points that move along curves.

Here, we demonstrate how the kernel trick can be applied in standard NRSFM. As a result, we model complex, deformable 3D shapes as the outputs of a non-linear mapping whose inputs are points within a low-dimensional shape space. This approach is flexible and can use different kernels to build different non-linear models. Using the kernel trick, our model complements the low-rank constraint by capturing non-linear relationships in the shape coefficients of the linear model. The net effect can be seen as using non-linear dimensionality reduction to further compress the (shape) space of possible solutions.

1. Introduction

The recovery of 3D object shapes from 2D image data is a fundamental task in computer vision. The recovered 3D shapes provide necessary information to applications in object recognition, face perception, biometrics, computer graphics, and human-computer interaction, among many others [2, 3, 6, 7, 10, 11, 13–15]. In many of these scenarios, the 3D object of interest undergoes a series of shape deformations while being observed under a varying pose. The recovery task is then known as the problem of non-rigid structure from motion (NRSFM). Given a set of corresponding 2D points in a sequence of images depicting a deformable object, the goal in NRSFM is to recover the 3D object shape and pose (*i.e.*, relative camera position) in each image. In the absence of any prior knowledge on 3D shape deformation, such as object rigidity, computing NRSFM is still a difficult, underconstrained problem.

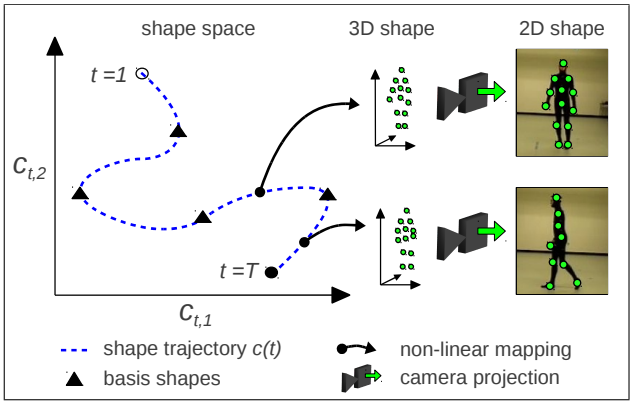

Figure 1. The deformation of a 3D shape (walking person) is modeled as the smooth time-trajectory of a point in a 2-dimensional shape space. The 3D reconstruction of observed 2D shapes are the outputs of a non-linear mapping of points on this *shape trajectory*.

The large majority of works in NRSFM are variants of the standard matrix factorization approach first proposed in [3]. This approach constraints all 3D shapes to lie within the linear space spanned by a small number K of unknown 3D basis shapes. Recent research work has attempted to define additional, general constraints to make this NRSFM formulation more tractable [1, 10, 11, 13, 14].

However, work on NRSFM rarely analyzes some of the implications of modeling shape deformation as linear combinations of basis shapes. As noted in [4], the deformation of some shapes is better represented by moving points along curves. This is the case, for example, of shapes with many uncorrelated articulations, relative rotation, and bending effects. Linear models can only approximate these shape deformations at the cost of requiring multiple basis elements that increase the number of unknowns that need to be estimated. As a result, the 3D reconstruction task becomes less and less constrained and prone to error (*e.g.*, a residual error on the small shape deformation that is difficult to eliminate). This argument also applies to the case where a deformable

3D shape is seen as a single point moving within a space containing the true, low-dimensional shape manifold.

In this paper, we propose a *Kernel NRSFM* approach to model and recover non-linear 3D shape deformation from 2D image streams (Fig. 1). We first demonstrate how the "kernel trick," commonly used for non-linear dimensionality reduction in pattern recognition [12], can be applied to the standard matrix factorization approach in NRSFM. As a result, we model 3D shapes as the outputs of a non-linear mapping whose inputs are points within a low-dimensional shape space. Our model complements the low-rank constraint by capturing non-linear relationships in the shape coefficients of the linear model. The dimensionality h of the new shape space is usually very small ($h = 2$ in our experiments) and h is also independent of the number of basis shapes K used by the kernalized NRSFM algorithm.

In addition, assuming shape deformation is gradual, we solve for the smooth time-trajectory of a single point within the h-dimensional shape space. This trajectory is compactly represented using only the low-frequency coefficients of the Discrete Cosine Transform (DCT) as in [6]. Using this representation, we introduce a novel formulation of the shape basis constraints of [14] and enforce the basis shapes to lie somewhere along the smooth shape trajectory, without the need to correspond to one of the observed shapes. As a result, each basis shape is modeled by a single unknown, a time-parameter, regardless of the dimensionality h. Therefore, the total number of unknowns in our model is reduced considerably, while providing flexibility of representation.

This paper is organized as follows. Section 2 reviews the NRSFM methods that are more closely related to our approach and presents the basic formulation we will need to derive our algorithm. Section 3 introduces the kernel trick into the NRSFM matrix factorization approach, discussing the implications and also problems that are addressed by our new algorithm derived in Section 4. Experimental results are presented in Section 5 and the conclusion in Section 6.

2. Related Work and Basic Formulation

2.1. Modeling 3D Shapes in a Linear Space

We first review the seminal matrix factorization model of [3]. For a NRSFM problem with T images (cameras), the n input 2D point tracks are given in matrix form as

$$\mathbf{W} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ y_{1,1} & y_{1,2} & \dots & y_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T,1} & x_{T,2} & \dots & x_{T,n} \\ y_{T,1} & y_{T,2} & \dots & y_{T,n} \end{bmatrix} \in \mathbb{R}^{2T \times n}, \quad (1)$$

where $[x_{t,j}, y_{t,j}]^T$ is the 2D projection of the j^{th} 3D point at time t (*i.e.*, on the t^{th} image), $t = 1, 2, \dots, T$, $j = 1, 2, \dots, n$.

Without loss of generality, assume for now that: (1) $\mathbf{W}$ is complete, meaning that no 2D points became occluded during tracking; and (2) its mean column vector $\mathbf{t} \in \mathbb{R}^{2T}$ has been subtracted from all columns, making them zero-mean. With an orthographic projection model and a world coordinate system centered on the observed 3D object, $\mathbf{t}$ gives the observed 2D camera translations in each image.

The authors of [3] model $\mathbf{W}$ as a product of two matrix factors of low-rank $3K$, $\mathbf{M} \in \mathbb{R}^{2T \times 3K}$ and $\mathbf{S} \in \mathbb{R}^{3K \times n}$,

$$\mathbf{W} = \underbrace{\mathbf{D}\,(\mathbf{C} \otimes \mathbf{I}_3)}_{\mathbf{M}} \underbrace{\begin{bmatrix} \widehat{\mathbf{S}}_1 \\ \vdots \\ \widehat{\mathbf{S}}_K \end{bmatrix}}_{\mathbf{S}}. \quad (2)$$

Here $\otimes$ denotes the Kronecker product and $\mathbf{I}_3$ is the 3×3 identity matrix. The coefficients of factor $\mathbf{M}$ are separated in a block-diagonal rotation matrix $\mathbf{D} \in \mathbb{R}^{2T \times 3T}$ and a shape coefficient matrix $\mathbf{C} \in \mathbb{R}^{T \times K}$ defined as

$$\mathbf{D} = \begin{bmatrix} \widehat{\mathbf{R}}_1 & & & \\ & \widehat{\mathbf{R}}_2 & & \\ & & \ddots & \\ & & & \widehat{\mathbf{R}}_T \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} c_{1,1} & \dots & c_{1,K} \\ c_{2,1} & \dots & c_{2,K} \\ \vdots & \ddots & \vdots \\ c_{T,1} & \dots & c_{T,K} \end{bmatrix}. \quad (3)$$

Let $\mathbf{c}_t^T$ be the t^{th} row of $\mathbf{C}$. The unknown 3D shape of the t^{th} image is modeled as the matrix function

$$S(\mathbf{c}_t^T) = (\mathbf{c}_t^T \otimes \mathbf{I}_3)\mathbf{S} = \sum_{k=1}^{K} c_{t,k} \widehat{\mathbf{S}}_k, \quad (4)$$

that is, a *linear* combination of K basis shapes $\widehat{\mathbf{S}}_k \in \mathbb{R}^{3 \times n}$ as described by the shape coordinates $c_{t,k}$. The camera orientation (object pose) at image t is given by $\widehat{\mathbf{R}}_t \in \mathbb{R}^{2 \times 3}$, a 3D rotation followed by an orthogonal projection to 2D.

The factors $\mathbf{M}$ and $\mathbf{S}$ are computed from the singular value decomposition (SVD) $\mathbf{W} = (\mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}})(\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T) = \overline{\mathbf{M}}\overline{\mathbf{S}}$, with all but the largest $3K$ singular values in $\mathbf{\Sigma}$ set to zero. This non-unique, "implicit" solution is defined only up to a rank-$3K$ ambiguity matrix $\mathbf{Q} \in \mathbb{R}^{3K \times 3K}$. To recover $\mathbf{D}$ and $\mathbf{C}$, an Euclidean upgrade step [1] finds a corrective $\mathbf{Q}$ for the solution $\mathbf{W} = (\overline{\mathbf{M}}\mathbf{Q})(\mathbf{Q}^{-1}\overline{\mathbf{S}}) = \mathbf{M}\mathbf{S}$.

2.2. Smooth Shape Trajectories in a Linear Space

To further constrain the estimation of the model above, many authors assume that the observed 3D shape deformation is only gradual over time $t = 1, \dots, T$ [2, 6, 11, 13]. For instance, the *shape trajectory approach* (STA) in [6] considers $\mathbf{c}_t^T = c(t)$ as a single K-dimensional point describing a smooth time-trajectory within an unknown linear shape space. This means that each shape coordinate $c_{t,k}$

varies smoothly with t. The shape trajectory is then modeled compactly using a small number d of low-frequency DCT coefficients,

$$\mathbf{C} = \mathbf{\Omega}_d \begin{bmatrix} \mathbf{x}_1, & \ldots, & \mathbf{x}_K \end{bmatrix} = \mathbf{\Omega}_d \mathbf{X}, \quad \mathbf{x}_k \in \mathbb{R}^d. \quad (5)$$

Here, $\mathbf{X} \in \mathbb{R}^{d \times K}$ represents $\mathbf{C}$ compactly in the domain of the truncated DCT basis matrix $\mathbf{\Omega}_d \in \mathbb{R}^{T \times d}$. The f^{th} column of $\mathbf{\Omega}_d$ is the f^{th}-frequency cosine wave with entries

$$\omega_{tf} = \frac{\sigma_f}{\sqrt{T}} \cos\left(\frac{\pi(2t-1)(f-1)}{2T}\right), t = 1, 2, \ldots, T, \quad (6)$$

where $\sigma_1 = 1$ and, for $f \geq 2$, $\sigma_f = \sqrt{2}$. Because the DCT bases are known *a priori*, the number of unknowns in $\mathbf{C}$ is significantly reduced with STA.

In [6], the STA model was shown to subsume the factorization of the *point trajectory approach* (PTA) of [2], which models independent 3D point trajectories instead of 3D shapes. The two models are equivalent when $\mathbf{X}$ above is equal to $\mathbf{X}_0 = [\ \mathbf{I}_K\ \mathbf{0}\ ]^T$, the $K \times K$ identity stacked over a block of zeros. Thus, for a factorization of rank-$3K$, the solutions of PTA correspond to smoothed versions of those of STA. In contrast to PTA, STA can model higher-frequency DCT coefficients in $\mathbf{X}$ without relaxing the low-rank constraint. However, the Euclidean update step of PTA is easier to compute because the only unknowns in factor $\mathbf{M}$ are those of the camera matrix $\mathbf{D}$ [2]. For this reason, STA starts with $\mathbf{X} = \mathbf{X}_0$ and computes $\mathbf{D}$ as done by PTA.

The final optimization stage of STA considers that $\mathbf{S} = \mathbf{M}^\dagger \mathbf{W}$ is a function of $\mathbf{M}$ and $\mathbf{W}$, with $\dagger$ denoting the Moore-Penrose pseudo-inverse [5]. The goal is then to minimize the reprojection error

$$f(\mathbf{M}) = \|\mathbf{W} - \mathbf{W}^*\|_F^2, \quad \mathbf{W}^* = \mathbf{M}\mathbf{S} = \mathbf{M}\mathbf{M}^\dagger\mathbf{W}, \quad (7)$$

where $\|\cdot\|_F$ is the Frobenius norm. Given $\mathbf{D}$ computed as above, $\mathbf{M}$ is treated as a function of $\mathbf{X}$ only. The higher-frequency DCT coefficients in $\mathbf{X}$ are then estimated using an iterative Gauss-Newton algorithm to minimize (7).

2.3. Locally Linear and Articulated Models

The NRSFM algorithm in [11] relaxes the linearity assumption for the shape manifold by using linear models to represent only small neighborhoods of shapes. Consequently, a number of locally linear models need to be estimated and the total number of parameters is larger than that of the standard NRSFM method [3]. Initialization of these parameters requires an elaborate clustering of images with similar shapes, which can also be a performance issue for the case of long image sequences.

Other specialized, articulated shape models [10,15] comprise a number of linear subspaces and depend on a prior, non-trivial process of segmentation of point tracks and classification of their motion subspaces. Although these methods convey additional information (*e.g.*, positions and orientations of joints), experimental results show that misclassification usually happens near joints and axes.

In the next sections, we propose a compact and general model for shapes with linear and non-linear deformations. Our kernel NRSFM algorithm does not require prior clustering of points or images. In contrast to [11], our method can smoothly interpolate between images and reconstruct partially occluded 3D shapes. The flexibility of our approach is also reflected by the fact that different non-linear models can be built according to the kernel function used.

3. Kernel Non-Rigid Structure from Motion

In this section, we first introduce the kernel trick into the NRSFM matrix factorization approach. We then discuss its implications and some issues that are addressed by our new algorithm presented in Section 4.

3.1. The Kernel Trick

Considering $\mathbf{S} = \mathbf{M}^\dagger\mathbf{W}$, NRSFM by matrix factorization reduces to estimating $\mathbf{M}$ as to provide a rank-$3K$ approximation $\mathbf{W}^* \approx \mathbf{W}$ given by $\mathbf{W}^* = \mathbf{M}\mathbf{M}^\dagger\mathbf{W}$ as in (7). We note that $\mathbf{M}^\dagger = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ can alternatively be expressed as $\mathbf{M}^\dagger = \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^\dagger$. The proof is easily obtained from the SVD form of $\mathbf{M}$. The new rank-$3K$ approximation for $\mathbf{W}$ is then modeled as

$$\mathbf{W}^* = \mathbf{M}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^\dagger\mathbf{W} = \widetilde{\mathbf{M}}\widetilde{\mathbf{M}}^\dagger\mathbf{W}. \quad (8)$$

The rank-$3K$ matrix $\widetilde{\mathbf{M}} = \mathbf{M}\mathbf{M}^T \in \mathbb{R}^{2T \times 2T}$ can be used to replace $\mathbf{M}$ in (7) giving equivalent solutions. Substituting $\mathbf{M} = \mathbf{D}(\mathbf{C} \otimes \mathbf{I}_3)$ into (8), we have

$$\mathbf{W}^* = \underbrace{\mathbf{D}(\mathbf{K}_0 \otimes \mathbf{I}_3)\mathbf{D}^T}_{\widetilde{\mathbf{M}}}\widetilde{\mathbf{M}}^\dagger\mathbf{W}, \quad \mathbf{K}_0 = \mathbf{C}\mathbf{C}^T. \quad (9)$$

To apply the kernel trick to (9), we replace $\mathbf{K}_0$ in $\widetilde{\mathbf{M}}$ by a *kernel matrix* $\mathbf{K} \in \mathbb{R}^{T \times T}$ whose $(t,t')^{th}$ entry is $\kappa(\mathbf{c}_t^T, \mathbf{c}_{t'}^T)$, instead of the standard inner product $\mathbf{c}_t^T\mathbf{c}_{t'}$. The function $\kappa(\cdot,\cdot)$ can be regarded as a generalized inner product and is known as a *Mercer kernel* [12]. Here, $\kappa(\cdot,\cdot)$ measures the similarity between two shape vectors, $\mathbf{c}_t^T$ and $\mathbf{c}_{t'}^T$.

For clarity, in the following we will consider only the popular radial basis function (RBF) kernel,

$$\kappa(\mathbf{c}_t^T, \mathbf{c}_{t'}^T) = e^{-\gamma\|\mathbf{c}_t^T - \mathbf{c}_{t'}^T\|_2^2}, \quad (10)$$

where γ is a scale parameter. However, note that our approach can be easily modified to use a different kernel. If the linear kernel is used, for instance, one goes back to linear NRSFM with $\mathbf{K}_0$ as in (9).

The kernel in (10) can be seen as a non-linear mapping of each low-dimensional shape representation $\mathbf{c}_t^T$ into an infinite-dimensional space of radial basis functions, $\kappa(\mathbf{c}_t^T, \cdot)$, where a linear representation is more suitable. The "trick" is that we do not need to explicitly represent the shapes in this infinite-dimensional space because the algorithm depends only on the generalized inner products (10). Thus, the kernel trick gives a combination of two mappings: (i) a non-linear mapping represented by the kernel function, capturing the non-linearity of the problem; and (ii) a linear mapping represented by the matrix products in (9).

In NRSFM by matrix factorization, the problem with replacing $\mathbf{K}_0$ in (9) is that we cannot guarantee that the new kernel matrix $\mathbf{K}$ will be of low-rank K (with $\mathbf{K} \otimes \mathbf{I}_3$ of rank $3K$). Thus, this version of kernel NRSFM becomes largely underconstrained. Next, we address this problem by reintroducing the low-rank constraint into the derivations above.

3.2. The Low-Rank Constraint in Kernel NRSFM

To derive a low-rank formulation of kernel NRSFM, we consider a sparse approximation for the kernel matrix $\mathbf{K}$ that has been used to speed up kernel methods [12]. Here, the idea translates into reconstructing the shapes $\mathbf{c}_t^T$, $\forall t$, based only on their similarities to a subset of these shapes, the *active set*, $\{\mathbf{b}_1^T, \mathbf{b}_2^T \ldots, \mathbf{b}_K^T\} \subset \{\mathbf{c}_1^T, \mathbf{c}_2^T \ldots, \mathbf{c}_T^T\}$.

For an arbitrary subset with K shape vectors $\mathbf{b}_k^T$, we obtain a rank-K approximation of the kernel matrix,

$$\mathbf{K} \approx \mathbf{K}_{\mathbf{c},\mathbf{b}} \mathbf{K}_{\mathbf{b},\mathbf{b}}^{-1} \mathbf{K}_{\mathbf{c},\mathbf{b}}^T, \tag{11}$$

where the $(t,k)^{th}$ entry of $\mathbf{K}_{\mathbf{c},\mathbf{b}} \in \mathbb{R}^{T \times K}$ is $\kappa(\mathbf{c}_t^T, \mathbf{b}_k^T)$, and the $(k,k')^{th}$ entry of $\mathbf{K}_{\mathbf{b},\mathbf{b}} \in \mathbb{R}^{K \times K}$ is $\kappa(\mathbf{b}_k^T, \mathbf{b}_{k'}^T)$.

To derive kernel NRSFM algorithms with $\mathbf{K}$ as in (11), we use

$$\overline{\mathbf{M}} = \mathbf{D}(\mathbf{K}_{\mathbf{c},\mathbf{b}} \mathbf{K}_{\mathbf{b},\mathbf{b}}^{-\frac{1}{2}} \otimes \mathbf{I}_3) \in \mathbb{R}^{2T \times 3K} \tag{12}$$

as a replacement for factor $\mathbf{M}$ in (8). Another equivalent solution can be obtained with $\overline{\mathbf{M}}$ replacing factor $\mathbf{M}$ in (7). Because the factorization in (7) is non-unique, there are in fact multiple equivalent solutions of the form $\mathbf{M} = \overline{\mathbf{M}}\mathbf{Q}$, with $\overline{\mathbf{M}}$ as above and $\mathbf{Q} \in \mathbb{R}^{3K \times 3K}$ arbitrary, but full-rank. Now consider the particular solution with $\mathbf{Q} = (\mathbf{K}_{\mathbf{b},\mathbf{b}}^{\frac{1}{2}} \otimes \mathbf{I}_3)$. From (12), our final form of $\mathbf{M}$ is then

$$\mathbf{M} = \mathbf{D}(\mathbf{K}_{\mathbf{c},\mathbf{b}} \otimes \mathbf{I}_3). \tag{13}$$

This new form becomes more similar to that of the original $\mathbf{M}$ in (2). With (13) and $\mathbf{W}^* = \mathbf{M}\mathbf{M}^\dagger\mathbf{W}$, we now express the 3D shape of image t as the non-linear matrix function

$$S(\mathbf{c}_t^T) = \left(\boldsymbol{\kappa}(\mathbf{c}_t^T) \otimes \mathbf{I}_3\right) \mathbf{M}^\dagger \mathbf{W} \in \mathbb{R}^{3 \times n}, \tag{14}$$

where $\boldsymbol{\kappa}(\mathbf{c}_t^T) \in \mathbb{R}^{1 \times K}$ is the t^{th} row of $\mathbf{K}_{\mathbf{c},\mathbf{b}}$,

$$\boldsymbol{\kappa}(\mathbf{c}_t^T) = [\, \kappa(\mathbf{c}_t^T, \mathbf{b}_1^T) \;\ldots\; \kappa(\mathbf{c}_t^T, \mathbf{b}_K^T) \,]. \tag{15}$$

Equation (14) gives the non-linear mapping in Fig. 1.

After reintroducing the low-rank constraint, our kernel NRSFM approach considers the kernel matrix $\mathbf{K}$ in (11) only implicitly. However, our factorization model with (13) still captures the essence of the kernel trick: the *kernel function* in (10) defines a non-linear mapping of shape vectors into a high-dimensional space where a linear representation is computed from inner products only (the entries of $\mathbf{K}_{\mathbf{c},\mathbf{b}}$).

In addition, note that using the above active set in NRSFM is similar to enforcing the shape basis constraints of [14] – *i.e.*, constraining the basis shapes ($\mathbf{b}_k^T$) to be identical to a subset of the observed shapes ($\mathbf{c}_k^T$). The problem with these basis constraints is that, from all possible combinations of K out of T shapes, we do not know the active set that best represents all shapes. To avoid a complex search algorithm for the optimal active set, we elaborate this approach further in the following section.

4. Kernel STA

This section presents our new NRSFM algorithm, referred to as the *Kernel Shape Trajectory Approach* (KSTA). KSTA uses the non-linear mapping of kernel NRSFM to further constrain the STA model (Section 2.2) and reduce the number of dimensions of the shape space. First, we avoid the combinatorial problem of Section 3.2 by removing the basis constraints. Inspired by the STA model, we derive novel basis constraints that generalize the idea in [14].

4.1. New Basis Constraints on the Shape Trajectory

Removing the shape basis constraints simply implies that, while we factorize $\mathbf{W}$ with $\mathbf{M}$ as in (13), we need to estimate the basis shapes $\mathbf{b}_k^T$, $\forall k$, together with the shape vectors $\mathbf{c}_t^T$. Importantly, a basis shape no longer needs to be equal to a shape observed in one of the T images. We now have as additional unknowns the K basis shapes, $\mathbf{b}_k^T$, and the parameters of the kernel function used to compute the entries $\kappa(\mathbf{c}_t^T, \mathbf{b}_k^T)$ of $\mathbf{K}_{\mathbf{c},\mathbf{b}}$.

Before we proceed with the modeling of additional constraints for $\mathbf{c}_t^T$ and $\mathbf{b}_k^T$, we note another important difference introduced by the model in Section 3.2: now the low-rank constraint is defined based solely on the number K of basis shapes $\mathbf{b}_k^T$. This means that all vectors $\mathbf{c}_t^T$ and $\mathbf{b}_k^T$ *can be modeled within a shape space with dimensionality* $h \leq K$ (many kernel functions only require that $\mathbf{c}_t^T$ and $\mathbf{b}_k^T$ be of the same dimension to compute their inner product).

Thus, from now on we will consider the shapes

$$\mathbf{c}_t^T \in \mathbb{R}^h, \forall t, \quad \text{and} \quad \mathbf{b}_k^T \in \mathbb{R}^h, \forall k, \tag{16}$$

as points in an h-dimensional shape space. Because h determines the number of unknowns we need to estimate (as discussed below), h should be small as to yield a compact model. In our experiments, complex non-linear deformations were modeled with a very small $h = 2$.

Assuming that shape deformation is smooth from one image to the next, we adapt the STA model in (5) and consider $\mathbf{c}_t^T = c(t)$ to describe a smooth time-trajectory within the h-dimensional shape space. Thus,

$$\begin{bmatrix} \mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_T^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_1^T \\ \vdots \\ \boldsymbol{\omega}_T^T \end{bmatrix} \mathbf{X} = \boldsymbol{\Omega}_d \mathbf{X}, \quad \mathbf{X} \in \mathbb{R}^{d \times h}, \tag{17}$$

where $\boldsymbol{\omega}_t^T$ is the t^{th} row of the DCT matrix $\boldsymbol{\Omega}_d$ and $\mathbf{X}$ is a compact representation of the shape trajectory.

To model $\mathbf{b}_k^T$, we introduce new basis constraints that represent a compromise between the traditional basis constraints and the unconstrained case. We will therefore only require that the $\mathbf{b}_k^T$ be *similar* (not necessarily equal) to some $\mathbf{c}_t^T$. This new constraint is enforced by modeling $\mathbf{b}_k^T$ somewhere along the continuous shape trajectory $c(t)$. That is, we model the time-samples

$$\mathbf{b}_k^T = b(t_k) = \omega(t_k)^T \mathbf{X}, \quad t_k \in [1, T], \forall k, \tag{18}$$

where $\omega(t_k)^T$ is a row vector of d low-frequency cosine terms (6) at time t_k. Hence, *each basis shape introduces only a single, continuous new variable, t_k, regardless of the dimensionality h of the shape space*.

4.2. Model Analysis

Indeed, linear models can represent a set of 3D shapes showing non-linearly deformations – as the number of basis shapes K approaches the number of observed shapes T, perfect representation is possible. Our claim is that, in linear NRSFM, non-linear deformations reduce the effectiveness of the low-rank constraint due to the need to use additional basis shapes, increasing K.

Using the kernel trick, our model complements the low-rank constraint by capturing the non-linear relationships in the shape coefficients of the linear model – *i.e.*, the new $\mathbf{K}_{\mathbf{c},\mathbf{b}}$ approximates the original $\mathbf{C}$ in (2). The net effect can be seen as using non-linear dimensionality reduction to further compress the (shape) space of possible solutions.

Let's consider the compactness of the KSTA model above in comparison to linear NRSFM. The original method in [3] defines $\mathbf{C}$ using TK parameters, while STA requires dK (with $d \ll T$ for smooth deformations). Through non-linear dimensionality reduction and new basis constraints, KSTA can further compress the model of $\mathbf{K}_{\mathbf{c},\mathbf{b}}$ to $dh+K+1$ unknowns, with $h \leq K \leq d$. This number includes the parameter γ used by the RBF kernel (10).

We note that our model bears some similarity with the non-linear dimensionality reduction approach of [8], where the dimensionality of the latent space (h) and the size of the active set (K) are treated as user supplied parameters. On the other hand, our method is not probabilistic and we further compress the representation in the h-dimensional space using the DCT-based trajectory model (17).

Another observation is that the user-supplied parameters d and h are often easy to set (typically, $d \in \{0.1T, 0.3T\}$ and $h \in \{2, 3\}$). Results vary considerably more according to the choice of K, as observed with other NRSFM methods. Cross-validation methods and additional priors have been used to automatically estimate K and to regularize the reconstruction process when K is over-estimated [13]; these techniques will be used with our model in future experiments. Here, for clarity, we focus on the new ideas introduced by KSTA and assume K is known from prior experience with a particular application scenario.

4.3. Optimization

With $\mathbf{M}$ as in (13), we minimize the reprojection error

$$f(\mathbf{M}) = \|\mathbf{W} - \mathbf{M}\mathbf{M}^\dagger \mathbf{W}\|_F^2, \tag{19}$$

using the Gauss-Newton algorithm in [6]. To initialize $\mathbf{M}$, we use $\mathbf{D}$ and $\mathbf{X}$ as computed by STA with $K = h$. The initial basis shapes $\mathbf{b}_k^T$ are computed from equally-spaced points $t_1, \ldots, t_K$ in the interval $[1, T]$, $t_k = k\frac{(T-1)}{(K+1)}$. Let σ_b be the average Euclidean distance from each $\mathbf{c}_t^T$ to each $\mathbf{b}_k^T$; the initial kernel parameter is then $\gamma = (2\sigma_b^2)^{-1}$.

In this paper, we fix the initial camera matrix $\mathbf{D}$ and only optimize with respect to the parameters $\mathbf{X}$, $t_1, \ldots, t_K$, and γ. However, our algorithm can be modified to perform the optimization of these parameters and also $\mathbf{D}$ in alternation, as in [10, 13]. The derivation of the gradient and Hessian terms is summarized in Appendix A.

4.4. NRSFM with Occlusion

In cases of occlusion, $\mathbf{W}$ is incomplete and we need to modify the cost function (19) as in [6]. Let the complete vector $\hat{\mathbf{w}}_j \in \mathbb{R}^{2T_j}$ ($T_j \leq T$) denote all the observed entries in the j^{th} column of $\mathbf{W}$. Also, define $\boldsymbol{\Pi}_j \in \mathbb{R}^{2T_j \times 2T}$ as a row-amputated identity matrix such that $\mathbf{M}_j = \boldsymbol{\Pi}_j \mathbf{M}$ and $\hat{\mathbf{t}}_j = \boldsymbol{\Pi}_j \mathbf{t}$ have the rows of $\mathbf{M}$ and $\mathbf{t}$ that correspond to the rows of entries in $\hat{\mathbf{w}}_j$. The camera translation vector $\mathbf{t}$ is obtained from the initialization using STA. We then minimize the reprojection error of the available observations,

$$f(\mathbf{M}) = \sum_{j=1}^{n} \left\| \left(\mathbf{I} - \mathbf{M}_j \mathbf{M}_j^\dagger \right) (\hat{\mathbf{w}}_j - \hat{\mathbf{t}}_j) \right\|_2^2. \tag{20}$$

Appendix A discusses the use of the Gauss-Newton algorithm in [6] to minimize this new cost function.

5. Experimental Results

We compare the performance of KSTA against four state-of-the-art NRSFM methods: the approach using probabilistic principal component analysis (EM-PPCA) to model 3D shapes [13]; the Metric Projections (MP)

method [10]; and the DCT-based PTA [2] and STA [6]. Note that all these four methods make use of linear models.

Our experiments considered the same datasets that where chosen by the authors of the methods above. The number of frames (T) and the number of point tracks (n) are indicated as (T/n) after a dataset's name. We considered the motion capture sequences: *drink* (1102/41), *pick-up* (357/41), *yoga* (307/41), *stretch* (370/41), and *dance* (264/75) of [2]; *face1* (74/37) of [10]; *face2* (316/40), *walking* (260/55), and the synthetic *shark* sequence (240/91) of [13]. We also introduce another full-body motion capture dataset, *capoeira* (250/41), with more complex deformations – the typical sideways swing of this African-Brazilian mixture of dance and martial art (see supplementary video available at *http://www.ece.osu.edu/∼gotardop*).

We followed the same evaluation procedure in [2] and [6]. For each dataset, $\mathbf{W}$ is obtained by applying an orthographic projection on the sequence of 3D shapes. Because the solution of NRSFM methods is defined up to an arbitrary 3×3 rotation, we compute a single rotation that best aligns all reconstructed and original 3D shapes. Let e_{tj} be the reconstruction error (*i.e.,* Euclidean distance) for the j^{th} 3D point of frame t. We then compute a normalized mean 3D error over all points and frames,

$$e_{3D} = \frac{1}{\sigma T n} \sum_{t=1}^{T} \sum_{j=1}^{n} e_{tj}, \quad \sigma = \frac{1}{3T} \sum_{t=1}^{T} (\sigma_{tx} + \sigma_{ty} + \sigma_{tz}), \tag{21}$$

with σ_{tx}, σ_{ty}, and σ_{tz} the standard deviations of the x-, y-, and z-coordinates of the original shape in frame t.

For each algorithm, we report the best result of different runs with $K \in \{2, 3, \ldots, 13\}$. For all datasets, the reconstructions computed with KSTA were modeled within a 2-dimensional shape space ($h = 2$). KSTA had the number of DCT bases set to $d = 0.1T$ (*i.e.*, 10%), except for face1, face2, walking, and capoeira, on which we set $d = 0.3T$ due to the presence of higher frequency deformations. Table 1 compares the performances of the NRSFM methods above in terms of the obtained error e_{3D}. The value of K for the best solutions of PTA, STA, and KSTA are also shown for comparison of these DCT-based methods.

Table 1 shows that the results of KSTA are consistently better than or similar to the best results provided by the other methods on each dataset. Also, note that KSTA models all these deformable shapes using a highly compact 2-dimensional shape space ($h = 2$). Therefore, in comparison to the other linear algorithms, KSTA can better constrain the reconstruction problem while the number of basis shapes K increases (because h is independent of K).

As shown in [2, 6], further decrease in the error e_{3D} for stretch, pick-up, and yoga is prevented by the larger errors in estimating the artificial rotations added to these sequences. Future extensions to our algorithm include the use of a simple technique that refine the estimated rotations and 3D shapes in alternation, as in [13] and [10].

Table 1. Average 3D reconstruction error (e_{3D}) of NRSFM methods on the complete synthetic and motion capture datasets. For the DCT-based PTA, STA, and KSTA methods, factorization rank is also indicated by the value of K in parenthesis.

Dataset	EM-PPCA	MP	PTA	STA	KSTA
Shark	0.0501	0.1571	0.1796 (9)	**0.0081** (3)	**0.0160** (3)
Face1	**0.0434**	0.0734	0.1247 (3)	0.0637 (5)	0.0618 (8)
Face2	**0.0329**	**0.0357**	0.0444 (5)	**0.0363** (3)	**0.0339** (4)
Drink	0.3393	0.4604	**0.0250** (13)	**0.0223** (6)	**0.0156** (12)
Stretch	1.1111	0.8549	0.1088 (12)	**0.0710** (8)	**0.0674** (12)
Pick-up	0.5822	0.4332	**0.2369** (12)	**0.2301** (6)	**0.2322** (6)
Yoga	0.8097	0.8039	0.1625 (11)	**0.1467** (7)	**0.1476** (7)
Dance	0.9839	0.2639	0.2958 (5)	0.2705 (2)	**0.2504** (4)
Walking	0.4917	0.5607	0.3954 (2)	0.1601 (4)	**0.1029** (5)
Capoeira	0.8934	0.3597	0.5065 (6)	0.3405 (4)	**0.2376** (7)

An important observation is that, in comparison to STA, the non-linear mapping of KSTA makes it more sensitive to the initialization of $\mathbf{X}$. By varying the value of K in the initial STA step (but still using the first h-columns of $\mathbf{X}$ as computed with STA), we have run KSTA with different initializations and have obtained e_{3D} error values as low as 0.0858 for the walking dataset. Future experiments will investigate alternative initialization procedures to consistently improve performance on all the datasets above.

Note that the walking dataset has recently been considered [13] as beyond the scope of NRSFM methods without an specialized, articulated model. With KSTA, the reconstruction error on this sequence has becomes of the same order of magnitude as that of the deformable facial shapes. Fig. 2(a) shows examples of 3D reconstructions for this sequence in comparison to the original 3D data (we show results for the same frames appearing in [13]). The smooth and approximately periodic shape trajectory estimated by KSTA is plotted on Fig. 2(b).

The flexible, kernel-based model of KSTA represents a promising contribution towards the development of new NRSFM algorithms that can reconstruct 3D shapes with more complex deformations. This argument is also supported by the large performance improvement obtained on the capoeira dataset. Fig. 3(a) shows example reconstructions obtained with KSTA on this sequence. The sharp changes in the shape trajectory of Fig. 3(b) captures the time instant when the sideways motion is reversed. In this sequence, sudden changes in the motion of hands and feet correspond to high-frequency deformation and introduce localized reconstruction errors; see result for frame 120 in Fig. 3(a). In general, deformation smoothness is also more weakly defined at the beginning and end of a sequence. Future work will address detection and correction of these issues, affecting the results of NRSFM algorithms that ex-

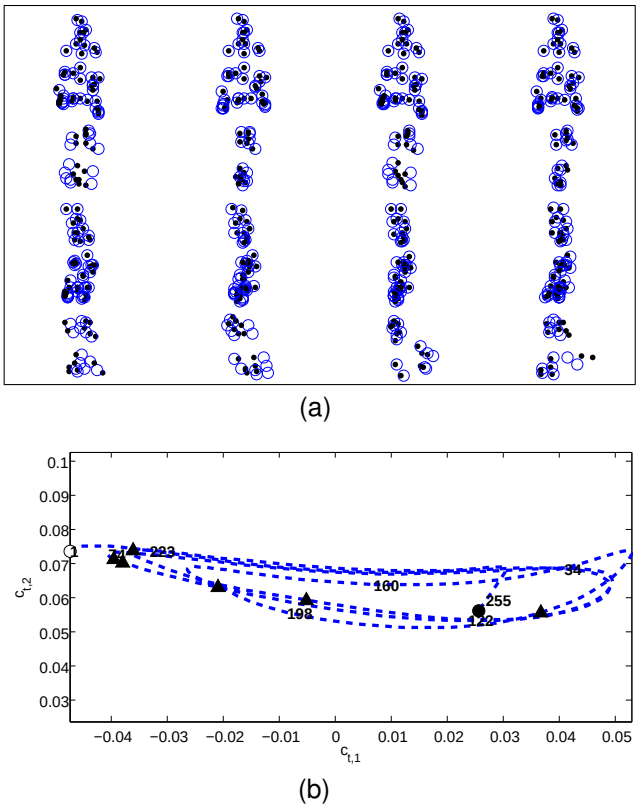

Figure 2. Results of KSTA on the walking sequence. (a) Ground-truth (dots) and recovered 3D shapes (circles) for frames 1, 34, 74, 122 (top), 160, 198, 223, and 255 (bottom). (b) Smooth trajectory in 2-dimensional shape space (triangles indicate the bases shapes).

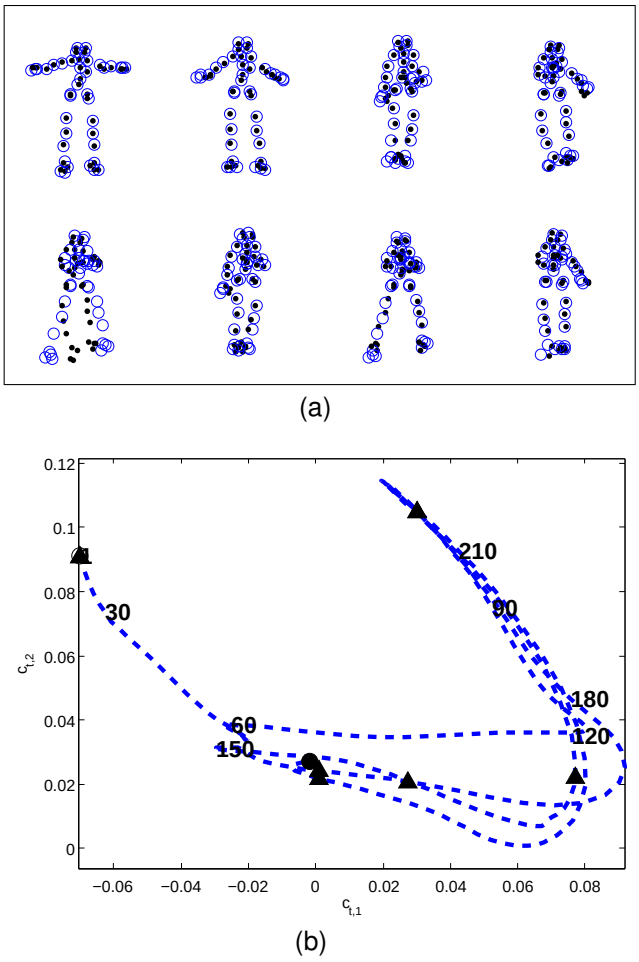

Figure 3. Results of KSTA on the capoeira sequence. (a) Ground-truth (dots) and recovered 3D shapes (circles) for frames 1, 30, 60, 90 (top), 120, 150, 180, and 210 (bottom). (b) Smooth trajectory in 2-dimensional shape space (triangles indicate the basis shapes).

plore the assumption of smoothness of deformation.

To assess the performance of KSTA in NRSFM with occlusion, we reproduce the experiment in [6, 10] and apply the algorithms above to the face2 dataset with $\rho\%$ of its 2D entries randomly discarded. KSTA was run to minimize the objective (20) with the same parameters described above for the complete dataset. Note that PTA does not handle occlusion and, in [10], the performance EM-PPCA was shown to be inferior to that of MP. Results of MP, STA, and KSTA, averaged over 100 trials, are shown in Fig. 4. While the average e_{3D} of MP increases with ρ above 30%, the performance of STA and KSTA show little variation over all the tested levels of random occlusion.

The runtimes of KSTA on the complete datasets above were usually very short, around 5 minutes on a laptop PC with a 3 GHz dual-core processor. KSTA was implemented in Mathwork's Matlab^{TM}and the source code is available at *http://www.ece.osu.edu/∼gotardop/*.

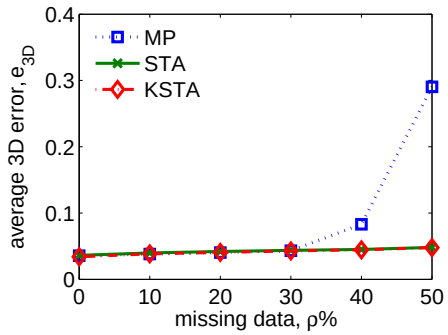

Figure 4. Performance on the face2 sequence with missing data.

6. Conclusion

This paper addressed the problem of modeling complex non-linear shape deformations that weaken a main pillar of NRSFM, the low-rank constraint. Using the kernel trick, we proposed a new approach that generalizes the standard matrix factorization in NRSFM. The derived kernel-based model complements the low-rank constraint by capturing the non-linear structure in the shape coefficients of the lin-

ear model. Experimental results show that some complex articulated deformations can be modeled with a reconstruction error of the same order of magnitude as that of simpler cases with facial shape deformations. These results are obtained without the use of specialized articulation models.

Our kernel-based model is flexible and represents a promising contribution towards the development of new NRSFM algorithms that can reconstruct 3D shapes with more complex deformations. Although the results reported here have only considered the RBF kernel, the use of our approach with other kernels is straightforward and will be investigated in the future. Despite our aim at a general method, future investigation is needed to define kernels that better suit the modeling of specific deformable 3D objects from 2D image streams.

A. Gauss-Newton Optimization for KSTA

To use the Gauss-Newton algorithm in [6], we need to express the differential of $\mathbf{M}$ in vectorized form, $vec(d\mathbf{M})$, and in terms of the differential of the vector of unknowns,

$$d\mathbf{u} = [\, vec(d\mathbf{X})^T \; dt_1 \; dt_2 \; \ldots \; dt_K \; d\gamma \,]^T. \tag{22}$$

From (13), using matrix differential calculus [9],

$$\begin{aligned} vec\,(d\mathbf{M}) &= vec\,(\mathbf{D}(d\mathbf{K}_{\mathbf{c},\mathbf{b}} \otimes \mathbf{I}_3)) \\ &= (\mathbf{I}_K \otimes \mathbf{D})\mathbf{V} vec(d\mathbf{K}_{\mathbf{c},\mathbf{b}}), \end{aligned} \tag{23}$$

where $\mathbf{V}$ is a constant and sparse binary mapping that satisfies $vec(d\mathbf{K}_{\mathbf{c},\mathbf{b}} \otimes \mathbf{I}_3) = \mathbf{V} vec(d\mathbf{K}_{\mathbf{c},\mathbf{b}})$, as in [6]. Moreover,

$$\begin{aligned} vec\,(d\mathbf{K}_{\mathbf{c},\mathbf{b}}) &= vec\left(\frac{\partial \mathbf{K}_{\mathbf{c},\mathbf{b}}}{\partial \mathbf{X}} d\mathbf{X}\right) + vec\left(\frac{\partial \mathbf{K}_{\mathbf{c},\mathbf{b}}}{\partial t_1} dt_1\right) + \ldots \\ &\quad + vec\left(\frac{\partial \mathbf{K}_{\mathbf{c},\mathbf{b}}}{\partial t_K} dt_K\right) + vec\left(\frac{\partial \mathbf{K}_{\mathbf{c},\mathbf{b}}}{\partial \gamma} d\gamma\right) \\ &= [\, \mathbf{P}_{\mathbf{X}} \ldots \mathbf{P}_{t_k} \ldots \mathbf{P}_{\gamma} \,]\, d\mathbf{u}. \end{aligned} \tag{24}$$

For the kernel (10) and $\kappa_{t,k}$ the $(t,k)^{th}$ entry of $\mathbf{K}_{\mathbf{c},\mathbf{b}}$, the vectorized partial derivative matrices are stacks of the form

$$\mathbf{P}_{\mathbf{X}} = \begin{bmatrix} \vdots \\ -2\gamma\kappa_{t,k}\left(\mathbf{c}_t^T - \mathbf{b}_k^T\right) \otimes \left(\boldsymbol{\omega}_t^T - \omega(t_k)^T\right) \\ \vdots \end{bmatrix}, \tag{25}$$

$$\mathbf{P}_{t_k} = \begin{bmatrix} \ldots & 2\gamma\kappa_{t,k}\dfrac{\partial \omega(t_k)^T}{\partial t_k}\mathbf{X}\left(\mathbf{c}_t^T - \mathbf{b}_k^T\right)^T \mathbf{I}_K^{(k)} & \ldots \end{bmatrix}^T, \tag{26}$$

$$\mathbf{P}_{\gamma} = \begin{bmatrix} \ldots & \kappa_{t,k}\log_e(\kappa_{t,k})\dfrac{1}{\gamma} & \ldots \end{bmatrix}^T, \tag{27}$$

where $\mathbf{I}_K^{(k)}$ is the k^{th} column of the $K \times K$ identity matrix.

Gradient and Hessian terms are computed as in [6] but using the new form of $vec(d\mathbf{M})$ or, when $\mathbf{W}$ is incomplete,

$$vec(d\mathbf{M}_j) = vec(\mathbf{\Pi}_j d\mathbf{M}) = (\mathbf{I}_K \otimes \mathbf{\Pi}_j) vec(d\mathbf{M}). \tag{28}$$

Acknowledgments

This research was supported by the National Institutes of Health, grants R01 EY 020834 and R21 DC 011081.

References

[1] I. Akhter, Y. Sheikh, and S. Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *Proc. IEEE CVPR*, pages 1534–1541, 2009. 1, 2

[2] I. Akhter, Y. A. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *Proc. NIPS*, 2008. 1, 2, 3, 6

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Proc. IEEE CVPR*, volume 2, pages 690–696, 2000. 1, 2, 3, 5

[4] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models: Their training and application. *CVIU*, 61(1):38–59, 1995. 1

[5] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, 1996. 3

[6] P. F. U. Gotardo and A. M. Martinez. Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Trans. PAMI*, 2011. 1, 2, 3, 5, 6, 7, 8

[7] P. F. U. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *Proc. IEEE CVPR*, pages 3065–3072, 2011. 1

[8] N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Machine Learning Research*, (6):1783–1816, 2005. 5

[9] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, 2 edition, 1999. 8

[10] M. Paladini, A. Del Bue, M. Stošić, M. Dodig, J. Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. In *Proc. IEEE CVPR*, pages 2898–2905, 2009. 1, 3, 5, 6, 7

[11] V. Rabaud and S. Belongie. Rethinking nonrigid structure from motion. In *Proc. IEEE CVPR*, volume 1, pages 1–8, 2008. 1, 2, 3

[12] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002. 2, 3, 4

[13] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. PAMI*, 30(5):878–892, 2008. 1, 2, 5, 6

[14] J. Xiao, J. Chai, and T. Kanade. A closed form solution to non-rigid shape and motion recovery. *IJCV*, 67(2):233–246, 2006. 1, 2, 4

[15] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Trans. PAMI*, 30(5):865–877, 2008. 1, 3