Non-Rigid Structure from Motion with Complementary Rank-3 Spaces

Paulo F.U. Gotardo and Aleix M. Martinez
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH 43210, USA
{gotardop,aleix}@ece.osu.edu

Abstract

Non-rigid structure from motion (NR-SFM) is a difficult, underconstrained problem in computer vision. This paper proposes a new algorithm that revises the standard matrix factorization approach in NR-SFM. We consider two alternative representations for the linear space spanned by a small number K of 3D basis shapes. As compared to the standard approach using general rank-$3K$ matrix factors, we show that improved results are obtained by explicitly modeling K complementary spaces of rank-3. Our new method is positively compared to the state-of-the-art in NR-SFM, providing improved results on high-frequency deformations of both articulated and simpler deformable shapes. We also present an approach for NR-SFM with occlusion.

1. Introduction

The modeling of deformable 3D shapes such as the human hand, face, and body from 2D image data is a fundamental task in computer vision. Important applications are found in surveillance, biometrics, computer graphics, and human-computer interaction (*e.g.*, [5]). The 3D reconstruction task above is the subject of the computer vision problem of non-rigid structure from motion (NR-SFM) [3]. Given a set of corresponding 2D points in a sequence of images depicting a deformable object, the goal in NR-SFM is to recover the 3D object shape and pose (*i.e.*, relative camera position) in each image.

Structure from motion techniques [9] have improved considerably over the past two decades using the assumption of object rigidity. However, in the absence of any prior knowledge on 3D shape deformation, computing NR-SFM is still a difficult, underconstrained problem. For this reason, recent research work has attempted to define new, general constraints for 3D shape deformation [1–3, 7, 11, 12].

The standard matrix factorization approach of [3] constraints all 3D shapes to lie within the linear space spanned by a small number K of 3D basis shapes. For a matrix $\mathbf{W}$ containing the input 2D points, the solution is computed by modeling $\mathbf{W} = \mathbf{MS}$ as a product of a motion factor $\mathbf{M}$ and a shape basis $\mathbf{S}$. With $\mathbf{M}$ and $\mathbf{S}$ constrained to be of a low-rank $3K$, a solution is obtained using singular value decomposition (SVD) [6]. Recently, a dual approach was proposed in [2] that models independent 3D point trajectories instead of 3D shapes. In this case, deformation is assumed to be only gradual (*i.e.*, smooth) from one image to the next. Then, the 3D point trajectories are modeled compactly in the domain of the Discrete Cosine Transform (DCT) basis vectors. This dual approach provides significantly improved results on articulated shapes with more complex deformation. However, it cannot model high-frequency deformation without relaxing the rank-$3K$ constraint above. Thus, its results are often over-smoothed.

In this paper, we propose a novel NR-SFM algorithm that subsumes the two approaches described above. Our approach models a smoothly deforming 3D shape as a single point moving along a smooth time-trajectory within a linear shape space (Fig. 1). We show that our method provides better reconstruction of high-frequency deformation without relaxing the rank-$3K$ constraint. In addition, we model the basis factor $\mathbf{S}$ as an implicitly function of $\mathbf{M}$ and $\mathbf{W}$, with the NR-SFM solution computed in terms of $\mathbf{M}$ only. We consider two alternative definitions of $\mathbf{S}$ and show that a derivation that explicitly models K complementary rank-3 spaces outperforms the method using a general rank-$3K$ derivation. Finally, we also offer an approach for NR-SFM with occlusion, with $\mathbf{W}$ presented with missing data.

This paper is organized as follows: Section 2 reviews the standard matrix factorization model in NR-SFM and its dual representation with independent 3D point trajectories. These two approaches are generalized in our new model described in Section 3. Experimental results are presented in Section 4. The conclusion is presented in Section 5.

2. Matrix Factorization Models in NR-SFM

Consider a NR-SFM problem with T images (cameras) and n input 2D point tracks. Let $[x_{t,j}, y_{t,j}]^T$ be the 2D projection of the j^{th} 3D point at time t (the t^{th} image), $t = 1, 2, \ldots, T$, $j = 1, 2, \ldots, n$. These input 2D points are

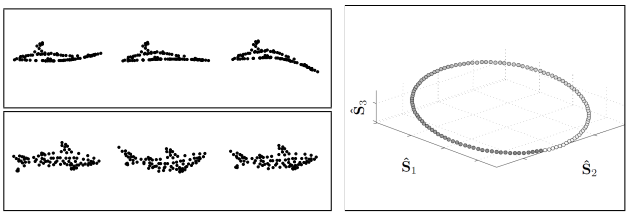

Figure 1. The 3D Shape trajectory model: (a) two views of the 3D basis shapes $\widehat{\mathbf{S}}_1$, $\widehat{\mathbf{S}}_2$, and $\widehat{\mathbf{S}}_3$, computed from the bending shark dataset in [12]; (b) shape space where the gradually deforming 3D shape is represented as a point moving along a smooth trajectory.

given in the observation matrix,

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

For now, assume $\mathbf{W}$ is complete, meaning that no 2D points became occluded during tracking. Considering an orthographic projection model in all cameras, the mean column vector $\mathbf{t} \in \mathbb{R}^{2T}$ of $\mathbf{W}$ gives a stack of 2D translations, one for each image, resulting from the camera translation in a world coordinate system centered on the observed 3D object. Here, we assume the columns of $\mathbf{W}$ are zero-mean (*i.e.*, $\mathbf{t}$ was computed and subtracted from each column).

2.1. Modeling 3D Shapes in a Linear Space

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

$$\mathbf{W} = \underbrace{\mathbf{D}\,(\mathbf{C} \otimes \mathbf{I}_3)}_{\mathbf{M}} \underbrace{\begin{bmatrix} \widehat{\mathbf{S}}_1 \\ \widehat{\mathbf{S}}_2 \\ \vdots \\ \widehat{\mathbf{S}}_K \end{bmatrix}}_{\mathbf{S}}, \quad (2)$$

where $\otimes$ denotes the Kronecker product, $\mathbf{I}_3$ 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

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

For each image t, the unknown 3D shape is modeled as a linear combination of K basis shapes $\widehat{\mathbf{S}}_k \in \mathbb{R}^{3 \times n}$ $(k = 1, 2, ..., K)$ as described by the shape (deformation) coefficients $c_{t,k}$. The camera orientation (*i.e.*, 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.

Factors $\mathbf{M}$ and $\mathbf{S}$ are computed from the singular value decomposition $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = (\mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}})(\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T) = \overline{\mathbf{M}}\overline{\mathbf{S}}$, 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 $\mathbf{Q}$, leading to the solution $\mathbf{W} = (\overline{\mathbf{M}}\mathbf{Q})(\mathbf{Q}^{-1}\overline{\mathbf{S}}) = \mathbf{M}\mathbf{S}$.

2.2. Modeling Independent 3D Point Trajectories

Akhter *et al.* [2] propose a dual factorization model in which the deformable 3D shape is seen as a collection of individual 3D point trajectories over time. This point trajectory approach (PTA) assumes shape deformation over time is gradual. Thus, each smooth 3D point trajectory is compactly represented as a linear combination of K low-frequency DCT basis vectors.

Let $\mathbf{\Omega}_K$ denote a $T \times K$ DCT basis matrix whose f^{th} column is the f^{th}-frequency cosine wave with entries

$$\omega_{tf} = \frac{\sigma_f}{\sqrt{T}} \cos\left(\frac{\pi(2t-1)(f-1)}{2T}\right), t = 1, 2, \dots, T, \quad (4)$$

where $\sigma_1 = 1$ and, for $f \geq 2$, $\sigma_f = \sqrt{2}$. In addition, let $\boldsymbol{\omega}_t^T$ denote the t^{th} row of this orthonormal matrix $\mathbf{\Omega}_K$.

The PTA method models $\mathbf{W}$ using the rank-3K factors,

$$\mathbf{W} = \underbrace{(\mathbf{D}\mathbf{\Theta})}_{\mathbf{M}}\mathbf{S}, \quad (5)$$

where $\mathbf{D}$ is as in (3) and

$$\mathbf{\Theta} = \begin{bmatrix} \boldsymbol{\omega}_1^T & & & \\ & \boldsymbol{\omega}_1^T & & \\ & & \boldsymbol{\omega}_1^T & \\ & & \vdots & \\ \boldsymbol{\omega}_T^T & & & \\ & \boldsymbol{\omega}_T^T & & \\ & & \boldsymbol{\omega}_T^T & \end{bmatrix}. \quad (6)$$

The columns of $\mathbf{\Theta}$ are basis vectors of the time-trajectory of a 3D point, $[\, X_j(t)\ Y_j(t)\ Z_j(t)\,]^T$. The j^{th} column of $\mathbf{S}$ is now interpreted as the vector of $3K$ low-frequency DCT coefficients describing the X-, Y-, and Z-coordinates of the j^{th} 3D point trajectory within the space spanned by $\mathbf{\Theta}$ [2].

Factors $\mathbf{M}$ and $\mathbf{S}$ are computed from the SVD of $\mathbf{W}$ as above. In this case, however, the Euclidean upgrade step is significantly simplified because now the only unknowns in $\mathbf{M}$ are the rotations $\widehat{\mathbf{R}}_t$. That is, the known DCT constants w_{tf} introduce additional constraints that make it possible to upgrade the initial solution by computing only three columns of the $(3K \times 3K)$ corrective matrix $\mathbf{Q}$ [1, 2].

3. Computing a Smooth 3D Shape Trajectory with Complementary Rank-3 Spaces

In this section, we generalize the dual approaches summarized above. We model 3D shape deformation as a single, smooth time-trajectory within a linear space spanned by 3D basis shapes. Two alternative algorithms for computing the DCT coefficients in our model are presented, with results compared in Section 4.

For clarity, we present our ideas incrementally, starting with a baseline algorithm [7] that computes rank-$3K$ factors $\mathbf{M}$ and $\mathbf{S}$, the standard approach in NR-SFM. *These derivations will then evolve into a new approach that explicitly models complementary rank-3 spaces* (Section 3.5). Finally, an extension for NR-SFM with occlusion (with $\mathbf{W}$ incomplete) is also described.

3.1. The 3D Shape Trajectory Model

We first reconsider the factorization approach expressed in (2). Note that, for each image t, the associated row $\mathbf{c}_t^T \in \mathbb{R}^K$ of $\mathbf{C}$ describes an observed 3D shape as a single point in the K-dimensional space spanned by the basis shapes $\widehat{\mathbf{S}}_1, \ldots, \widehat{\mathbf{S}}_K$. Assuming smooth 3D shape deformation over time, we consider the smooth time-function $\mathbf{c}_t^T = c(t)$ to describe *a single point in shape space that moves along a single, smooth 3D shape trajectory* (Fig. 1).

Equivalently, each shape coordinate $c_{t,k}$ is assumed to vary smoothly with t. We thus represent each column of $\mathbf{C}$ using a linear combination $\mathbf{x}_k \in \mathbb{R}^d$ of d DCT basis vectors,

$$\mathbf{C} = \mathbf{\Omega}_d \begin{bmatrix} \mathbf{x}_1 & \ldots & \mathbf{x}_K \end{bmatrix} = \mathbf{\Omega}_d \mathbf{X}, \quad \mathbf{X} \in \mathbb{R}^{d \times K}. \tag{7}$$

Here, $\mathbf{\Omega}_d$ is a DCT basis matrix with entries as in (4). The unknown factor $\mathbf{X}$ describes the 3D shape trajectory in DCT domain, considering the K basis shapes in factor $\mathbf{S}$. For simplicity, we assume that the number of DCT components d and the rank parameter K have been fixed *a priori*.

Therefore, our NR-SFM approach models $\mathbf{W} = \mathbf{MS}$, the product of two rank-3K factors in (2), but with an additional smoothness constraint applied on $\mathbf{M}$,

$$\mathbf{M} = \mathbf{D}\left(\mathbf{\Omega}_d \mathbf{X} \otimes \mathbf{I}_3\right). \tag{8}$$

Furthermore, we also note that the factor $\mathbf{S}$ can be considered only implicitly, as a function of $\mathbf{W}$ and $\mathbf{M}$. For instance, a common definition is given by $\mathbf{S} = \mathbf{M}^\dagger \mathbf{W}$, with $\dagger$ denoting the Moore-Penrose pseudo-inverse [6]. Hence, the next sections will formulate the NR-SFM as an iterative optimization problem in terms of $\mathbf{M}$ alone.

To solve a NR-SFM problem, all we need to compute is the rotation matrix $\mathbf{D}$ and DCT coefficient matrix $\mathbf{X}$ in $\mathbf{M}$, as in (8). As described next, $\mathbf{D}$ is computed by an initialization algorithm. We then proceed by solving exclusively for the 3D shape trajectory $\mathbf{X}$ in the DCT domain. Optimization is performed using an iterative, Gauss-Newton algorithm that also requires an initialization of $\mathbf{X}$.

3.2. Initialization

The assumed smoothness of 3D shape deformation leads to high-frequency DCT coefficients close to zero. Thus, we propose an initialization of $\mathbf{X}$, denoted $\mathbf{X}_0$, of the form

$$\mathbf{X}_0 = \begin{bmatrix} \widehat{\mathbf{Q}} \\ \mathbf{0} \end{bmatrix}, \quad \widehat{\mathbf{Q}} \in \mathbb{R}^{K \times K}. \tag{9}$$

The goal now is to initialize only a small, rank-K block $\widehat{\mathbf{Q}}$. For any full-rank $\widehat{\mathbf{Q}}$, an equally good initial solution is given by the simple and deterministic initialization

$$\mathbf{X}_0 = \begin{bmatrix} \widehat{\mathbf{Q}} \\ \mathbf{0} \end{bmatrix} \widehat{\mathbf{Q}}^{-1} = \begin{bmatrix} \mathbf{I}_K \\ \mathbf{0} \end{bmatrix}, \tag{10}$$

where $\mathbf{I}_K$ is the $K \times K$ identity matrix. As a proof, note that a NR-SFM solution is non-unique; for a fixed $\mathbf{D}$, factor $\mathbf{M}$ in (8) is defined only up to a full-rank ambiguity matrix $(\widehat{\mathbf{Q}} \otimes \mathbf{I}_3)$, with $\widehat{\mathbf{Q}} \in \mathbb{R}^{K \times K}$. This means that $\mathbf{C}_0 = \mathbf{\Omega}_d \mathbf{X}_0$ is defined only up to an ambiguity matrix $\widehat{\mathbf{Q}}$ as above.

Hence, for a given $\mathbf{D}$, we initialize $\mathbf{M}$ as

$$\mathbf{M}_0 = \mathbf{D}\left(\mathbf{\Omega}_d \mathbf{X_0} \otimes \mathbf{I}_3\right) = \mathbf{D}\left(\mathbf{\Omega}_K \otimes \mathbf{I}_3\right). \tag{11}$$

Note that $K < d$ here and this initial rank-$3K$ solution can only use K low-frequency vectors in the DCT basis.

We now compare the expressions of $\mathbf{M}_0$ in (11) and of $\mathbf{M}$ in (5), which is the solution computed by the PTA method: not only $\mathbf{D}$ is of the same form, but also $(\mathbf{\Omega}_K \otimes \mathbf{I}_3)$ and $\mathbf{\Theta}$ have the same columns, albeit in a different order. Therefore, our coarse initial solution $\mathbf{M}_0$ is equivalent to the final solution $\mathbf{M}$ of PTA (with factor $\mathbf{S}$ presenting a different order of rows). Note that the PTA method cannot consider additional DCT basis vectors without increasing K, leading to a higher rank of $\mathbf{M}$. Our approach, on the other hand, can consider any number $d = K, \ldots, T$ of DCT basis vectors because, in (8), the linear combination represented by $\mathbf{X}$ constrains $\mathbf{M}$ to be of rank-$3K$. That is, our method can better model 3D structure deformation presenting higher-frequency components in the DCT domain, yielding better 3D shape reconstructions.

Empirically, we have found that the coarse solutions computed by PTA contain accurate estimates of the rotation matrices in $\mathbf{D}$ (rigid motion components). We iteratively run PTA with increasing values of $K \in \{1, 2, \ldots, \lfloor \frac{n}{3} \rfloor\}$, obtaining a solution denoted as $\mathbf{D}_K$. Iterations stop automatically when there is no additional improvement in the average camera orthonormality,

$$\varepsilon(\mathbf{D}_K) = \frac{1}{T} \sum_{t=1}^{T} \left\| \mathbf{I}_2 - \widehat{\mathbf{R}}_t \widehat{\mathbf{R}}_t^T \right\|_F^2. \tag{12}$$

Algorithm 1 CSF method for minimizing the error function $f(\mathbf{X})$, with $\mathbf{M}$ defined in terms of $\mathbf{X}$.

$\mathbf{X} \leftarrow$ initial matrix ($\mathbf{X}_0$).
$\delta \leftarrow$ initial damping scalar ($\delta_0 = 10^{-4}$).
repeat
  Compute $\mathbf{M}$ from $\mathbf{X}$.
  Compute gradient ($\mathbf{g}$) and Hessian ($\mathbf{H}$) from the Jacobian and residual terms ($\mathbf{J}_j$ and $\mathbf{r}_j$).
  repeat
    $\delta \leftarrow \delta \times 10$.
    Find $\Delta\mathbf{X}$ from $vec(\Delta\mathbf{X}) \leftarrow (\mathbf{H} + \delta\mathbf{I})^{-1}\mathbf{g}$.
  until $f(\mathbf{X} - \Delta\mathbf{X}) < f(\mathbf{X})$.
  $\mathbf{X} \leftarrow \mathbf{X} - \Delta\mathbf{X}$.
  $\delta \leftarrow \delta \times 10^{-2}$.
until convergence.

With the final solution of PTA identified as a coarse, over-smoothed instance of our shape deformation model, we then proceed to estimate the higher-frequency DCT coefficients in $\mathbf{X}$. This approach is seen as a "coarse-to-fine" optimization, with the NR-SFM solution refined iteratively.

3.3. Optimization

We now describe two Gauss-Newton algorithms to compute $\mathbf{M}$ (*i.e.*, $\mathbf{X}$). Because $\mathbf{M}$ describes a basis for the column space of $\mathbf{W}$, our algorithms used to compute $\mathbf{M}$ are dubbed the Column Space Fitting (CSF) methods.

The two CSF methods presented in the following differ on the definition of factor $\mathbf{S}$ as an implicit function of $\mathbf{W}$ and $\mathbf{M}$. Each definition leads to a different form of the cost function $f(\mathbf{X})$, expressing the overall error of the factorization represented by $\mathbf{X}$ in approximating the entries of $\mathbf{W}$. The error $f(\mathbf{X})$ is then minimized as summarized in Algorithm 1. Starting with $\mathbf{X} = \mathbf{X}_0$, in each iteration we update $\mathbf{X}$ by computing an adjustment matrix $\Delta\mathbf{X}$ in vectorized form, $vec(\Delta\mathbf{X})$, which stacks the columns of $\Delta\mathbf{X}$ in a single vector of unknowns. We solve for $vec(\Delta\mathbf{X})$ using the gradient vector ($\mathbf{g}$) and the Hessian matrix ($\mathbf{H}$) of f as given by matrix differential calculus [10]. The damping parameter δ leads to combined Gauss-Newton and steepest-descent iterations when $\mathbf{H}$ becomes singular.

3.4. Baseline method: CSF1

Here, we consider a rank-$3K$ $\mathbf{S} = \mathbf{M}^{\dagger}\mathbf{W}$. Let $\mathbf{w}_j \in \mathbb{R}^{2T}$ and $\mathbf{s}_j \in \mathbb{R}^{3K}$ denote, respectively, the j^{th} column of $\mathbf{W}$ and $\mathbf{S}$ ($j = 1, \ldots, n$). Also, define the residual (error) vector $\mathbf{r}_j = \mathbf{w}_j - \mathbf{M}\mathbf{s}_j$, where $\mathbf{s}_j = \mathbf{M}^{\dagger}\mathbf{w}_j$.

With $\mathbf{M}$ defined by $\mathbf{X}$ as in (8), the goal is to minimize

$$f(\mathbf{X}) = \frac{1}{2}\sum_{j=1}^{n} \mathbf{r}_j^T\mathbf{r}_j \;, \quad \mathbf{r}_j = \underbrace{\left(\mathbf{I} - \mathbf{M}\mathbf{M}^{\dagger}\right)}_{\mathbf{P}^{\perp}}\mathbf{w}_j \;, \qquad (13)$$

where $\mathbf{P}^{\perp} \in \mathbb{R}^{2T\times 2T}$ is the projection onto the orthogonal space of $\mathbf{M}$. Therefore, we minimize the sum of squared Euclidean distances from each column of $\mathbf{W}$ to the space spanned by the columns of $\mathbf{M}$.

To compute $\mathbf{g}$ and $\mathbf{H}$, we follow a Gauss-Newton derivation in terms of Jacobian matrices, $\mathbf{J}_j$, of the residual $\mathbf{r}_j$,

$$\mathbf{g} = -\sum_j \mathbf{J}_j^T\mathbf{r}_j \quad \text{and} \quad \mathbf{H} = \sum_j \mathbf{J}_j^T\mathbf{J}_j \;. \qquad (14)$$

To derive the Jacobian terms $\mathbf{J}_j$, we express the differential $d\mathbf{M}$ in terms of $d\mathbf{X}$ and $d\mathbf{r}_j = -\mathbf{J}_j vec(d\mathbf{X})$. Note that,

$$d\mathbf{M} = \underbrace{\mathbf{D}\left(\mathbf{\Omega}_d \otimes \mathbf{I}_3\right)}_{\mathbf{B}_{nr}}\left(d\mathbf{X} \otimes \mathbf{I}_3\right) = \mathbf{B}_{nr}\left(d\mathbf{X} \otimes \mathbf{I}_3\right), \qquad (15)$$

with $\mathbf{B}_{nr} \in \mathbb{R}^{2T\times 3d}$ interpreted as a basis for the column space of $\mathbf{M}$ in our NR-SFM method. Then, the Jacobian terms $\mathbf{J}_j$ are of the form (derivations are in Appendix A),

$$\mathbf{J}_j = \left(\mathbf{s}_j^T \otimes \mathbf{P}^{\perp}\mathbf{B}_{nr}\right)\mathbf{V}, \qquad (16)$$

with $\mathbf{s}_j = \mathbf{M}^{\dagger}\mathbf{w}_j$ denoting the current (and implicit) estimate of the j^{th} column of factor $\mathbf{S}$. Above, $\mathbf{V} \in \mathbb{R}^{9dK\times dK}$ is a binary map satisfying $vec(d\mathbf{X} \otimes \mathbf{I}_3) = \mathbf{V}vec(d\mathbf{X})$. For $\mathbf{X}$ with dimensions $d \times K$, the constant and sparse matrix $\mathbf{V}$ is defined as [10],

$$\mathbf{V} = \mathbf{I}_K \otimes \left[\left(\mathbf{K}_{3d} \otimes \mathbf{I}_3\right)\left(\mathbf{I}_d \otimes vec(\mathbf{I}_3)\right)\right], \qquad (17)$$

where the permutation matrix $\mathbf{K}_{3d} \in \mathbb{R}^{3d\times 3d}$ satisfies $vec(\mathbf{A}^T) = \mathbf{K}_{3d}vec(\mathbf{A})$, for any $\mathbf{A} \in \mathbb{R}^{3\times d}$.

3.5. Defining complementary rank-3 spaces: CSF2

This section proposes an alternative method for computing factor $\mathbf{S}$ as a function of $\mathbf{M}$ and $\mathbf{W}$. To motivate the new approach, we first note that each column $\mathbf{s}_j$ of $\mathbf{S}$ defines a basis for the deformation of a single 3D point of the observed object. The number of degrees of freedom ($3K$) in $\mathbf{s}_j, \forall j$, increases with the rank parameter K. Also note that, in practice, not all object points deform equally. In human faces, for instance, points on the nose are expected to deform less than those on the mouth; other objects may present a subset of points that remain in a rigid setting. Thus, *as K is increased to better model deformation, some object points are modeled with too many degrees of freedom*. As a result, large spurious deformations can be introduced in the recovered 3D shapes. This is the case when the data in $\mathbf{W}$ present weaker constraints due to occlusion (fewer observations), noise, and slow camera motion [2].

To address this problem, we introduce new constraints on the magnitude of the recovered modes of shape deformation, *i.e.*, $\|\widehat{\mathbf{S}}_k\|_F$. Instead of considering a rank-$3K$ factor $\mathbf{S}$, we now explicitly estimate $\widehat{\mathbf{S}}_k$. This is done by explicitly defining $\mathbf{M}$ as a sequence of K complementary 3-dimensional column spaces for $\mathbf{W}$.

Let $\mathbf{M}_k \in \mathbb{R}^{2T\times 3}$ $(k = 1,\dots,K)$ denote each triplet of columns of $\mathbf{M} = [\,\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_K\,]$. The triplets of rows in $\mathbf{S}$ are the basis shapes $\widehat{\mathbf{S}}_k$, as before. Then, note that $\mathbf{W} = \mathbf{MS} = \sum_k \mathbf{M}_k\widehat{\mathbf{S}}_k$. We now define:

$$\begin{aligned}\widehat{\mathbf{S}}_k &= \mathbf{M}_k^{\dagger}(\mathbf{W} - \sum_{k_o=1}^{k-1}\mathbf{M}_{k_o}\widehat{\mathbf{S}}_{k_o}) \\ &= \mathbf{M}_k^{\dagger}\mathbf{P}_{k-1}^{\perp}\dots\mathbf{P}_2^{\perp}\mathbf{P}_1^{\perp}\mathbf{W},\end{aligned} \tag{18}$$

where $\mathbf{P}_k^{\perp} = \mathbf{I} - \mathbf{M}_k\mathbf{M}_k^{\dagger}$ is the projection onto the orthogonal space of $\mathbf{M}_k$. Thus, factor $\mathbf{S}$ is now given by

$$\mathbf{S} = \begin{bmatrix}\widehat{\mathbf{S}}_1\\ \widehat{\mathbf{S}}_2\\ \widehat{\mathbf{S}}_3\\ \vdots\end{bmatrix} = \begin{bmatrix}\mathbf{M}_1^{\dagger}\\ \mathbf{M}_2^{\dagger}\mathbf{P}_1^{\perp}\\ \mathbf{M}_3^{\dagger}\mathbf{P}_2^{\perp}\mathbf{P}_1^{\perp}\\ \vdots\end{bmatrix}\mathbf{W}. \tag{19}$$

As a consequence, the basis shape $\widehat{\mathbf{S}}_1$ can be seen as a main shape component while $\widehat{\mathbf{S}}_k$, for $k \geq 2$, is interpreted as a mode of a 3D shape deformation. Note that, as k increases, $\widehat{\mathbf{S}}_k$ and $\mathbf{M}_k$ model (residual) components of $\mathbf{W}$ with decreasing magnitude and not modeled by the previous $\widehat{\mathbf{S}}_{k_o}$ and $\mathbf{M}_{k_o}, \forall k_o < k$. Thus, the model in (19) is in principle similar to 3D shape models based on Principal Component Analysis (PCA), *e.g.*, [12]. Our new method for estimating $\widehat{\mathbf{S}}_k$ prevents the modeling of large, spurious 3D shape deformations as k increases.

Let $\mathbf{P}_k = \mathbf{M}_k\mathbf{M}_k^{\dagger}$ be the projection on the range space of $\mathbf{M}_k$ and $\mathbf{s}_{kj} \in \mathbb{R}^3$ be the j^{th} column of $\widehat{\mathbf{S}}_k$ as in (19). The new residual vector $\mathbf{r}_j$ is now defined as

$$\begin{aligned}\mathbf{r}_j &= \mathbf{w}_j - \mathbf{M}_1\mathbf{s}_{1j} - \mathbf{M}_2\mathbf{s}_{2j} - \mathbf{M}_3\mathbf{s}_{3j} - \dots \\ &= \left((\mathbf{I} - \mathbf{P}_1) - \mathbf{P}_2\mathbf{P}_1^{\perp} - \mathbf{P}_3\mathbf{P}_2^{\perp}\mathbf{P}_1^{\perp} - \dots\right)\mathbf{w}_j \\ &= \left((\mathbf{I} - \mathbf{P}_2) - \mathbf{P}_3\mathbf{P}_2^{\perp} - \dots\right)\mathbf{P}_1^{\perp}\mathbf{w}_j \\ &= \left((\mathbf{I} - \mathbf{P}_3) - \dots\right)\mathbf{P}_2^{\perp}\mathbf{P}_1^{\perp}\mathbf{w}_j \\ &= \mathbf{P}_K^{\perp}\mathbf{P}_{K-1}^{\perp}\dots\mathbf{P}_2^{\perp}\mathbf{P}_1^{\perp}\mathbf{w}_j\,.\end{aligned} \tag{20}$$

That is, $\mathbf{M}_k$ models a region of the column space of $\mathbf{W}$ that is orthogonal to the region modeled by the other $\mathbf{M}_{k_o}, \forall k_o < k$. With $\mathbf{r}_j$ defined as above, we minimize

$$f(\mathbf{X}) = \frac{1}{2}\sum_{j=1}^{n}\mathbf{r}_j^T\mathbf{r}_j, \quad \mathbf{M}_k = \mathbf{D}\left(\mathbf{\Omega}_d\mathbf{x}_k \otimes \mathbf{I}_3\right), \tag{21}$$

with $\mathbf{M}_k$ defined by $\mathbf{x}_k$, the k^{th} column of $\mathbf{X}$ as in (8).

To compute $\mathbf{X}$, we use Algorithm 1 with Jacobian terms

$$\mathbf{J}_j = \left[\mathbf{s}_{1j}^T \otimes \mathbb{P}_1^{\perp}\mathbf{B}_{nr} \quad \dots \quad \mathbf{s}_{Kj}^T \otimes \mathbb{P}_K^{\perp}\mathbf{B}_{nr}\right]\mathbf{V}, \tag{22}$$

$$\text{with} \qquad \mathbb{P}_k^{\perp} = \mathbf{P}_K^{\perp}\mathbf{P}_{K-1}^{\perp}\dots\mathbf{P}_k^{\perp}. \tag{23}$$

The complete derivation of $\mathbf{J}_j$ is given in Appendix B.

3.6. NR-SFM with Occlusion

The two algorithms derived above assume $\mathbf{W}$ is complete, which is also a requirement in the initial computation of $\mathbf{t}$ and $\mathbf{D}$ (as in the PTA method). For the problem of NR-SFM with occlusion, with $\mathbf{W}$ presented with missing data, this section derives a method that can be used to recover a complete, low-rank $\mathbf{W}$ prior to using the methods above.

Consider $\mathbf{W} = \mathbf{MS}$ with factors $\mathbf{M} \in \mathbb{R}^{2T\times r}$ and $\mathbf{S} \in \mathbb{R}^{r\times n}$ of a predefined rank r. 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 $\mathbf{\Pi}_j \in \mathbb{R}^{2T_j\times 2T}$ as a row-amputated identity matrix such that $\mathbf{M}_j = \mathbf{\Pi}_j\mathbf{M}$ has the rows in $\mathbf{M}$ that correspond to the rows of entries in $\hat{\mathbf{w}}_j$. Then, consider the residual vectors $\mathbf{r}_j = \hat{\mathbf{w}}_j - \mathbf{M}_j\mathbf{s}_j$, where the complete column vector $\mathbf{s}_j = \mathbf{M}_j^{\dagger}\hat{\mathbf{w}}_j$.

As in (13), we define the projection $\mathbf{P}_j^{\perp} = (\mathbf{I} - \mathbf{M}_j\mathbf{M}_j^{\dagger})$ and minimize the sum of squared Euclidean distances from $\hat{\mathbf{w}}_j$ to the subspace spanned by $\mathbf{M}_j$, the corresponding rows in $\mathbf{M}$. This approach uses only the available entries (constraints) of $\mathbf{W}$ and is common in methods for matrix factorization with missing data [4, 7, 8].

Here, we will assume that both 3D shape deformation and camera motion are smooth over time (t). This is a common scenario for NR-SFM using monocular video. As a result, the columns of $\mathbf{W}$ describe smooth 2D point trajectories with a compact representation in the DCT domain. We then model the r-dimensional column space of $\mathbf{W}$ as

$$\mathbf{M} = \underbrace{(\mathbf{\Omega}_d \otimes \mathbf{I}_2)}_{\mathbf{B}_{2D}}\mathbf{X}, \quad \mathbf{X} \in \mathbb{R}^{2d\times r}, \tag{24}$$

where $\mathbf{B}_{2D}$ duplicates the DCT basis $\mathbf{\Omega}_d$ for the alternated x- and y-coordinates in the rows of $\mathbf{W}$.

To reconstruct an incomplete $\mathbf{W}$, we compute $\mathbf{X}$ in (24) using Algorithm 1 with Jacobian terms

$$\mathbf{J}_j = \mathbf{s}^T \otimes \mathbf{P}_j^{\perp}\mathbf{\Pi}_j\mathbf{B}_{2D}, \tag{25}$$

derived as in Appendix A, with $d\mathbf{M}_j = \mathbf{\Pi}_j\mathbf{B}_{2D}d\mathbf{X}$.

Because $\mathbf{M}$ is a basis for smooth 2D point trajectories in the columns of $\mathbf{W}$, the high-frequency DCT coefficients of $\mathbf{X}$ are expected to be close to zero. Additionally, noting that the factorization $\mathbf{W} = \mathbf{MS}$ is defined only up to an ambiguity matrix $\mathbf{Q} \in \mathbb{R}^{r\times r}$, we initialize $\mathbf{X}$ using the deterministic initialization $\mathbf{X}_0 = [\,\mathbf{I}_r\ \mathbf{0}\,]^T$ as above.

4. Experimental Results

We compare the performance of our algorithms, CSF1 and CSF2, against three state-of-the-art, non-rigid SFM methods: (i) the algorithm modeling 3D shape using probabilistic principal component analysis (EM-PPCA) [12]; (ii) the Metric Projections (MP) method [11]; and (iii) the DCT-based 3D point trajectory approach (PTA) [2].

Our experiments considered the same datasets that were 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 start with the motion capture sequences: *drink* (1102/41), *pick-up* (357/41), *yoga* (307/41), *stretch* (370/41), and *dance* (264/75) used in [2]; *face1* (74/37) of [11]; *face2* (316/40) and *walking* (260/55) of [12]. We also consider the different, synthetic sequences *shark1* (240/91) of [12] and *shark2* (240/91) of [2].

To allow for comparison against the results reported in [2], our experiments follow the same procedure therein. For each dataset, $\mathbf{W}$ is obtained by applying an orthographic projection on the sequence of 3D shapes. Because the solution of non-rigid SFM 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,

$$e_{3D} = \frac{1}{\sigma T n} \sum_{t=1}^{T} \sum_{j=1}^{n} e_{tj}, \quad \sigma = \frac{1}{3T} \sum_{t=1}^{T} (\sigma_{tx} + \sigma_{ty} + \sigma_{tz}), \tag{26}$$

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.

Following the methodology in [2], we ran the algorithms with different values of $K \in \{2, 3, \ldots, 13\}$, reporting the best result. In all runs, CSF1 and CSF2 had the number of DCT basis set to $d = 0.1T$ (*i.e.*, 10%), except for the two face datasets on which we set $d = \frac{T}{3}$ (*i.e.*, 33%) due to the presence of higher frequency deformations. Table 1 compares the performances of the NR-SFM methods above in terms of the obtained error e_{3D}. The value of K for the best solutions obtained with PTA, CSF1 and CSF2 are also shown for comparison of these closely related methods.

Table 1 shows that the results of CSF1 and CSF2 are consistently similar or better than the best results provided by the other methods on each dataset. As compared to PTA, CSF1 and CSF2 compute better solutions and at a lower rank $3K$ by more efficiently using higher DCT frequency components to model deformation. In addition, the basis shape model of CSF2, defining complementary 3-dimensional spaces, provides significant improvements over CSF1 on dance and walking, while also modeling better some small magnitude deformations on the other datasets. The 3D shapes recovered by CSF1 and CSF2 for the walking sequence are shown in Fig. 2.

The motion capture sequences containing highly articulated bodies highlight the superiority of PTA, CSF1, and CSF2 compared to EM-PPCA, and MP. On the sequences drink and stretch, the improvement offered by CSF1 and CSF2 over PTA is small because the reconstruction of PTA already provides small errors on these very smooth shape deformations. On pick-up and yoga, the marginal improvement offered by CSF1, and CSF2 is due to large errors in the rotation matrix $\mathbf{D}$ recovered using PTA, as shown in [2]. However, the results on the difficult walking and dance datasets show the advantages of using the 3D shape trajectory model of CSF1 and CSF2.

Table 1. Average 3D reconstruction error (e_{3D}) of non-rigid SFM methods on the complete synthetic and motion capture datasets. For the related PTA and CSF methods, factorization rank is also indicated by the value of K in parenthesis.

Dataset	EM-PPCA	MP	PTA	CSF1	CSF2
Drink	0.3393	0.4604	**0.0250** (13)	**0.0223** (6)	**0.0223** (6)
Stretch	1.1111	0.8549	0.1088 (12)	**0.0710** (8)	**0.0684** (8)
Pick-up	0.5822	0.4332	**0.2369** (12)	**0.2301** (6)	**0.2277** (3)
Yoga	0.8097	0.8039	0.1625 (11)	**0.1467** (7)	**0.1465** (7)
Dance	0.9839	0.2639	0.2958 (5)	0.2705 (2)	**0.1942** (7)
Walking	0.4917	0.5607	0.3954 (2)	0.1863 (2)	**0.1041** (5)
Face1	**0.0434**	0.0734	0.1247 (3)	0.0637 (5)	**0.0526** (5)
Face2	**0.0329**	**0.0357**	0.0444 (5)	**0.0363** (3)	**0.0312** (5)
Shark1	0.0501	0.1571	0.1796 (9)	**0.0081** (3)	0.0437 (5)
Shark2	0.0529	0.1346	0.3120 (9)	0.2538 (2)	**0.0052** (3)

On face1 and face2, PTA is not capable of modeling the high-frequency deformation of the facial structures (*e.g.*, lips and chin), recovering mostly rigid shapes. CSF1 and CSF2 provide small reconstruction errors even while computing only **33**% of all DCT components. Considering 75% of DCT components ($d = 0.75T$), the errors of CSF1 and CSF2 on face1 decrease to 0.0625 and 0.0476, respectively. On face2, the new errors are 0.0328 and 0.0310.

With a full DCT basis, CSF1 recovers the 3D shape deformation of shark1 perfectly ($e_{3D} = 0.00004$ is negligible). However, CSF1 does not perform as well on shark2, which has a different motion and also small differences in the ground truth 3D shapes, as compared to shark1. Because these differences are non-trivial, future work will further investigate these results. As for CSF2, a small e_{3D} is obtained on shark1, with perfect reconstruction observed on shark2.

To simulate missing data in the shark2 and walking datasets, we randomly discard ρ% of the 2D entries in $\mathbf{W}$. Note that these sequences present smooth camera motion and smooth 3D shape deformation. Thus, we use the method of Section 3.6, referred to as CSF0, to reconstruct the complete 2D point trajectories in $\mathbf{W}$. This is done before applying CSF1 and CSF2 (with K as in Table 1). Here, CSF0 was run with $d = 0.25T$ and rank $r = 7$. Let $\mathbf{W}_0$ be the complete matrix, we normalize the 2D reconstruction error for the incomplete $\mathbf{W}$ by the average of the standard deviations of x- and y-coordinates (σ_{tx} and σ_{ty}, $\forall t$), similar to the normalization done for e_{3D} above.

On both shark2 and walking, and up to 75% random occlusion, the recovered 3D shapes are visually similar to

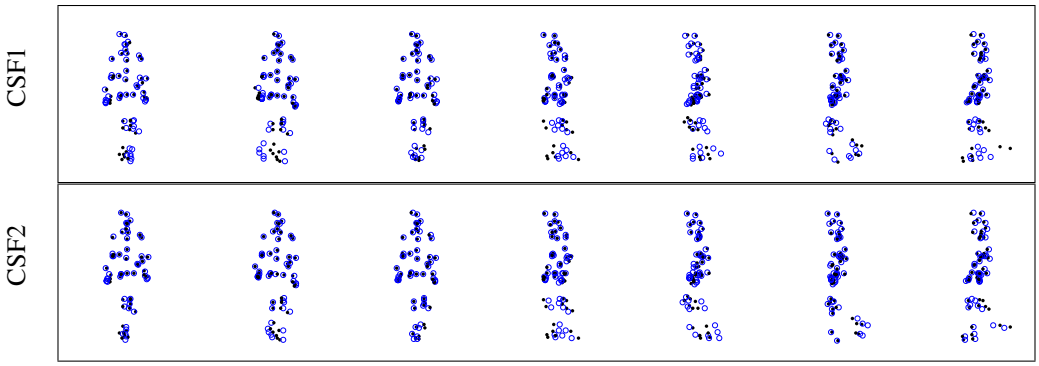

Figure 2. Results of CSF1 and CSF2 on the walking sequence of [12]. Reconstructed 3D shapes (blue circles) are shown against the original 3D data (dark dots). Frames 34, 74, 122, 160, 198, 223, and 255 are displayed above.

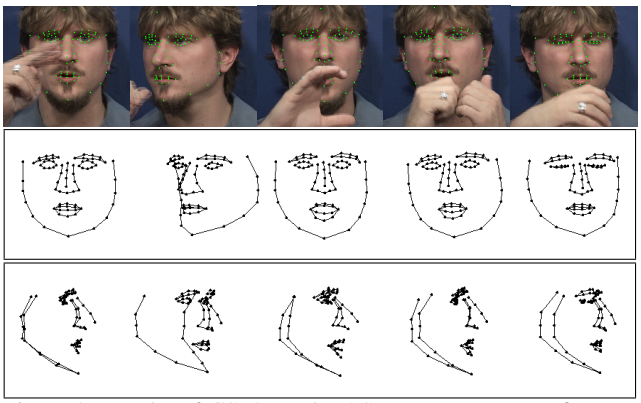

Figure 3. Results of CSF2 on the ASL sequence: (*top*) five out of 114 images with annotated facial landmarks in green; (*middle,bottom*) two orthogonal views of the recovered 3D shapes.

those obtained from the original, complete $\mathbf{W}$ (*e.g.*, Fig. 2). Of 10 runs with $\rho = 75\%$, the average (maximum) 2D reconstruction error for $\mathbf{W}$ was 0.0066 (0.0211). The average (max.) 3D error of CSF1 was 0.2575 (0.2781) and that of CSF2 was 0.0188 (0.0376). After 10 runs on the walking sequence, the average (max.) 2D error for the reconstructed $\mathbf{W}$ was 0.0475 (0.0527). The 3D error of CSF1 was then 0.2103 (0.3543) and that of CSF2 was 0.1772 (0.2342). Except for CSF1 on shark2, these results on incomplete data are still better than those of EM-PPCA, MP, and PTA on the complete shark2 and walking datasets. Despite the large amount of missing data, CSF0 was able to reconstruct $\mathbf{W}$ such that only a small penalty was observed on the 3D shapes recovered by CSF1 and CSF2.

Finally, we now consider a 114-image (4 seconds long) face close-up video of an American Sign Language (ASL) sentence [5]. In this case, head rotation and hand gesticulation often cause the occlusion of facial features. Facial landmarks were manually annotated in each image when visible. The resulting $\mathbf{W} \in \mathbb{R}^{228\times 75}$ is missing 11.5% of its data and has small magnitude annotation errors (noise) due to motion blur in the images. Fig. 3 shows example 3D face shapes recovered using CSF2 ($K = 6$ and $d = 0.4T$), after the initial CSF0 step ($r = 5$, $d = 0.4T$). Note that even when a hand occludes the mouth, our 3D shape trajectory model provides a correct estimate for the occluded 3D shape by enforcing smoothness of deformation while modeling the visible shapes in adjacent images. These results of CSF2 indicate correct recovery of pose and deformation of mouth and eyes despite the occurrence of occlusion.

5. Conclusion

This paper presents a novel matrix factorization method for NR-SFM. Our method models smooth 3D shape deformation, compactly, as the time-trajectory of a single point within a linear shape space. We discuss the implicit representation of the 3D basis shapes in $\mathbf{S}$ as a function of the motion factor, $\mathbf{M}$, and the input data, $\mathbf{W}$. As a result, we show that our approach considering complementary rank-3 column spaces of $\mathbf{W}$ outperforms the general rank-3K constraint of the standard factorization model. Improved 3D reconstruction is obtained for both articulated and simpler deformable shapes, as compared to state-of-the-art NR-SFM algorithms. In addition, we present a simple and effective approach for NR-SFM with occlusion. Future work will consider the automatic selection of the number of elements in the DCT and shape bases (d and K). The use of non-linear shape models will also be investigated.

A. Derivation of Jacobian Terms in CSF1

Following a Gauss-Newton approach, the first and second differentials of f in (13) are

$$df = \frac{1}{2}\sum_j \left(d\mathbf{r}_j^T \mathbf{r}_j + \mathbf{r}_j^T d\mathbf{r}_j\right) = \sum_j \mathbf{r}_j^T d\mathbf{r}_j, \tag{27}$$

$$d^2 f \approx \sum_j d\mathbf{r}_j^T d\mathbf{r}_j, \tag{28}$$

with second order terms $d^2\mathbf{r}_j$ neglected in $d^2 f$ and

$$d\mathbf{r}_j = -d\left(\mathbf{M}\mathbf{M}^\dagger\right)\mathbf{w}_j. \tag{29}$$

From [10], we have

$$d\left(\mathbf{M}\mathbf{M}^\dagger\right) = \mathbf{P}^\perp d\mathbf{M}\mathbf{M}^\dagger + \left(\mathbf{P}^\perp d\mathbf{M}\mathbf{M}^\dagger\right)^T. \tag{30}$$

with $\mathbf{P}^\perp$ as in (13). Notice that the right-most term in (30) vanishes when multiplied by $\mathbf{r}_j^T = (\mathbf{P}^\perp\mathbf{w}_j)^T$ on the left, because $\mathbf{r}_j$ is in the null-space of $\mathbf{M}$:

$$\mathbf{r}_j^T\left(\mathbf{P}^\perp d\mathbf{M}\mathbf{M}^\dagger\right)^T = \left(\mathbf{P}^\perp d\mathbf{M}(\mathbf{M}^\dagger\mathbf{P}^\perp)\mathbf{w}_j\right)^T = \mathbf{0}, \tag{31}$$

where the property $\mathbf{M}^\dagger = \mathbf{M}^\dagger\mathbf{M}\mathbf{M}^\dagger$ implies $\mathbf{M}^\dagger\mathbf{P}^\perp = \mathbf{0}$.

We therefore approximate $d\mathbf{r}_j$ as

$$d\mathbf{r}_j \approx -\mathbf{P}^\perp d\mathbf{M}\mathbf{s}_j. \tag{32}$$

To derive the Jacobian terms $\mathbf{J}_j$ from (32), we first substitute $d\mathbf{M}$ with (15) to express $d\mathbf{r}_j$ in terms of $d\mathbf{X}$. We then vectorize both sides of (32) to obtain $d\mathbf{r}_j = -\mathbf{J}_j vec(d\mathbf{X})$. This final step leads to the form of $\mathbf{J}_j$ in (16) by considering the equality $vec(\mathbf{BXA}) = (\mathbf{A}^T \otimes \mathbf{B})vec(\mathbf{X})$, where the operator $\otimes$ is the Kronecker product.

The Jacobian terms obtained from (32) neglect the right-most term in (30). While there is no difference in the gradient vector, it can be shown that the Hessian matrix in (14) neglects terms of the form $\mathbf{r}_j\mathbf{r}_j^T \otimes \mathbf{M}^\dagger\mathbf{M}^{\dagger T}$. Nevertheless, our Hessian approximation, combined with the damping parameter δ in Algorithm 1, is efficient in providing adequate solution updates, $vec(d\mathbf{M})$, despite the simpler form. In our experiments, we have found no difference between results computed as above and using the full Gauss-Newton derivation (which is itself an approximation to the Hessian).

B. Derivation of Jacobian Terms in CSF2

We now follow a Gauss-Newton derivation similar to that in Appendix A, but considering the new form of residual vectors in (20). Using the product rule for differentials,

$$d\mathbf{r}_j = \sum_{k=1}^{K} \mathbb{P}_{k+1}^\perp \left(d\mathbf{P}_k^\perp\right)\mathbf{P}_{k-1}^\perp \ldots \mathbf{w}_j\ , \tag{33}$$

with $\mathbb{P}_{k+1}^\perp$ as in (23). We define $\mathbb{P}_{K+1}^\perp = \mathbf{P}_0^\perp = \mathbf{I}$.

To find $\left(d\mathbf{P}_k^\perp\right)$, we once again use the result in (30). Then, note that $\mathbf{r}_j^T\mathbb{P}_k^\perp = \mathbf{r}_j^T$ and $\mathbf{r}_j^T\mathbf{M}_k^{\dagger T} = \mathbf{0}$ because $\mathbf{r}_j$ is in the null-space of $\mathbf{M}_k, \forall k$. Thus, the right-most term in (30) vanishes when multiplied by $\mathbf{r}_j^T$ on the left (as in Appendix A).

Using the results in the paragraph above, (33) yields

$$d\mathbf{r}_j \approx -\sum_{k=1}^{K} \mathbb{P}_k^\perp\left(d\mathbf{M}_k\right)\mathbf{s}_{kj} = -\sum_{k=1}^{K} vec\left(\mathbb{P}_k^\perp\left(d\mathbf{M}_k\right)\mathbf{s}_{kj}\right). \tag{34}$$

Considering $d\mathbf{M}_k = \mathbf{B}_{nr}\left(d\mathbf{x}_k \otimes \mathbf{I}_3\right)$, as in (15), each $vec(\cdot)$ term in (34) is rewritten as a Kronecker product and the summation is expressed as a product of two matrices,

$$d\mathbf{r}_j \approx -\mathbf{J}_j \begin{bmatrix} vec(d\mathbf{x}_1) \\ \vdots \\ vec(d\mathbf{x}_K) \end{bmatrix} = -\mathbf{J}_j vec(d\mathbf{X}), \tag{35}$$

revealing the form of $\mathbf{J}_j$ in (22).

Acknowledgments

This research was supported by the National Science Foundation, grant RI 0713055, and 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 *CVPR*, pages 1534–1541, 2009. 3065, 3066

[2] I. Akhter, Y. A. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*, December 2008. 3065, 3066, 3068, 3069, 3070

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, volume 2, pages 690–696, 2000. 3065, 3066

[4] P. Chen. Optimization algorithms on subspaces: Revisiting missing data problem in low-rank matrix. *IJCV*, 80(1):125–142, 2008. 3069

[5] L. Ding and A. M. Martinez. Modelling and recognition of the linguistic components in american sign language. *Image and Vision Computing*, 27(12):1826–1844, 2009. 3065, 3071

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in Mathematical Sciences. The Johns Hopkins University Press, 1996. 3065, 3067

[7] P. F. U. Gotardo and A. M. Martinez. Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. *PAMI*, 2011. 3065, 3067, 3069

[8] R. Hartley and F. Schaffalizky. Powerfactorization: 3d reconstruction with missing or uncertain data. In *Proc. Australia-Japan Advanced Workshop on Computer Vision*, 2003. 3069

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 3065

[10] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, 2nd edition, 1999. 3068, 3072

[11] 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 *CVPR*, pages 2898–2905, 2009. 3065, 3069, 3070

[12] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30(5):878–892, 2008. 3065, 3066, 3069, 3070, 3071