A Procrustean Markov Process for Non-Rigid Structure Recovery

Minsik Lee Chong-Ho Choi Songhwai Oh
Graduate School of CST, AICT / Department of ECE, ASRI, Seoul National University, Korea
mlee.paper@gmail.com {chchoi, songhwai}@snu.ac.kr

Abstract

Recovering a non-rigid 3D structure from a series of 2D observations is still a difficult problem to solve accurately. Many constraints have been proposed to facilitate the recovery, and one of the most successful constraints is smoothness due to the fact that most real-world objects change continuously. However, many existing methods require to determine the degree of smoothness beforehand, which is not viable in practical situations. In this paper, we propose a new probabilistic model that incorporates the smoothness constraint without requiring any prior knowledge. Our approach regards the sequence of 3D shapes as a simple stationary Markov process with Procrustes alignment, whose parameters are learned during the fitting process. The Markov process is assumed to be stationary because deformation is finite and recurrent in general, and the 3D shapes are assumed to be Procrustes aligned in order to discriminate deformation from motion. The proposed method outperforms the state-of-the-art methods, even though the computation time is rather moderate compared to the other existing methods.

1. Introduction

Non-rigid structure from motion (NRSfM) [14] is a fundamental problem in computer vision to recover 3D shapes of a deforming object from a set of 2D observations. It is a non-rigid counterpart of structure from motion (SfM) problem [12], which deals with a stationary object. SfM can be efficiently solved using the Tomasi-Kanade factorization method [12], because the rank assumption in this approach is fairly accurate. Extending this method to non-rigid cases [4,9,13,14] were successful for simple deformations but not for stronger deformations, even though the solution was proven to be unique [1]. This is mainly because the fixed rank assumption is not quite adequate for non-rigid cases, which eventually makes the estimation poor.

An alternative is to use a *trajectory-based* approach [2, 5]. This 'dual' approach focuses on the trajectories of points, which is modeled using DCT bases. Trajectory-based approaches basically impose a smoothness constraint on the motion or deformation of objects. Since real objects deform and move continuously, it is a good assumption to make in NRSfM, and indeed these approaches perform better for the tested data sets in [2, 5]. However, these approaches require the number of DCT bases to be predetermined, which is not practical.

Meanwhile, Cho *et al*. [3] proposed an approach that extended generalized Procrustes analysis (GPA) for NRSfM. Later, Lee *et al*. [8] proposed the Procrustean normal distribution (PND) to solve NRSfM without requiring any prior information. The PND, which is a special case of the normal distribution, represents the distribution of Procrustes aligned shapes, *i.e*., the distribution of shape deformations. The PND excludes any changes in 3D coordinates due to similarity transform to separate motion and deformation, which helps to accurately reconstruct a 3D structure. Consequently, EM-PND, an EM algorithm based on the PND, provides better reconstruction results without any additional constraints such as the rank restriction. This method gives *de facto* state-of-the-art performance for most cases, but not when there are drastic changes in shape. In these cases, trajectory-based methods can give better reconstructions thanks to the smoothness assumption, but of course, with an appropriate choice of the number of DCT bases.

Large deformations are common in most video sequences, so the capability to handle large deformation is very important for the practicality of an NRSfM scheme. To handle large shape changes in NRSfM, the smoothness assumption is vital. Large changes in data imply that the shape goes through a large variation as the video sequence progresses, so it is better to give more weights to nearby frames than far-away ones in speculating a specific 3D shape. However, to make the smoothness assumption really useful, it should not require predetermined parameters, such as the number of DCT bases.

Using a Markov-type model and learning the degree of smoothness in the recovery process can be a solution to this. In fact, there was an attempt to apply a Markov model to NRSfM in the literature. Torresani *et al*. [13] proposed EM-LDS that used a first-order linear Markov model to in-

corporate the smoothness assumption, in conjunction with two other schemes. According to our experimental results in Section 4, however, it is difficult to say that EM-LDS is better than their main scheme, EM-PPCA, because EM-LDS often gave much worse results than EM-PPCA. This means that we may need to impose additional constraints to the Markov model.

In this paper, we show that a simple Markov model can still be effective for NRSfM. With appropriate assumptions, we can make the model sufficiently powerful without increasing the model order. To demonstrate this, we propose a simple Markov process, which will be called as the Procrustean Markov processs (PMP) hereafter, that can effectively model shape deformation. The PMP is a first-order stationary Markov process with Procrustes alignment, whose steady-state distribution is a PND. The most important contribution of this paper is to incorporate a stationarity assumption for this Markov model, which was not utilized in EM-LDS. This is an important assumption, because the deformation of an object is usually finite and recurrent. For example, arms do not elongate infinitely and jaws do not drop to the ground, but constantly go back to their original positions in human motion. Hence, restricting the overall deformation using the stationarity constraint can make the model focus only on likely deformations.

Each state of the PMP satisfies the 7-dimensional PND constraint [8] in order for all the individual shapes to be Procrustes aligned. This makes the motion and deformation well separated as in the PND. Hence, the PMP has the same advantages of the PND while satisfying the smoothness constraint. As a result, the PMP has most of the desired characteristics discussed in the literature of NRSfM. The proposed scheme, EM-PMP, is an EM algorithm that learns the parameters of the PMP, including the smoothness parameter of the Markov process, from a sequence of given 2D observations. Experimental results show that EM-PMP not only gives much better result than EM-PND for data with large deformation, but also gives much smaller errors for many of the cases in the experiments.

Another thing to mention is that, the first-order linear model of the PMP is much simpler than that of EM-LDS if we do not take the Procrustes and stationarity constraints into account: It has only one parameter in the model, which makes the optimization tractable with the stationarity constraint. We actually tested several variant Markov models with the Procrustes constraint but without the stationarity constraint, and these models did not give superb performance as EM-PMP. This suggests that the stationarity constraint is essential in solving this problem.

The remainder of this paper is organized as follows: We present a brief review of the PND and the definition of the PMP in Section 2. Based on the PMP, the proposed EM-PMP is introduced in Section 3. The experiments are followed in Section 4, and finally, we conclude the paper in Section 5.

2. Procrustean Markov process

A Markov model for NRSfM must be designed carefully to capture the characteristics of real data. Deformation in the model needs to be well separated from motion in order to correctly infer the dependency between shapes, and unlikely shape changes during deformation should be avoided as much as possible. We propose the Procrustean Markov process (PMP) to meet these criteria. The PMP is a stationary Markov process whose steady-state distribution is a PND. Since the PMP is assumed to be a stationary Markov process, it prevents shapes to undergo potentially infinite deformation, which is unlikely in the real world. Moreover, the change between sequential frames of the PMP is assumed to have a PND-like distribution that satisfies the same PND constraints as the steady-state distribution. In this way, the whole sequence of 3D shapes will be aligned as in the generalized Procrustes analysis (GPA) [6], resulting in an accurate separation of motion and deformation.

In this section, we first briefly review the PND, which is the steady-state distribution of the PMP, and then formally define the PMP.

2.1. A brief review of the PND

The PND [8] is a probability distribution of shape deformation that does not contain any changes due to rigid motion. It only considers variations of GPA-aligned shapes in its distribution to distinguish deformation changes from rigid changes, which makes it very useful in non-rigid structure recovery. The PND is a special case of the normal distribution, whose null space is closely related to the similarity transforms of its mean shape. Let $\mathbf{Y} \in \mathbb{R}^{3\times n_p}$ be a PND shape where n_p is the number of points, then

$$\begin{aligned} \mathrm{vec}(\mathbf{Y}) &= \mathrm{vec}\big(\overline{\mathbf{Y}}\big) + \mathbf{Q}\mathbf{v}, \\ \mathbf{v} &\sim \mathcal{N}(0, \mathbf{\Sigma}), \quad \mathbf{Q} = \mathbf{P}_N\big(\overline{\mathbf{Y}}\big)^{\perp}, \end{aligned} \tag{1}$$

where $\mathrm{vec}(\cdot)$ is the vectorization operator, $\overline{\mathbf{Y}}$ is the mean shape with unit norm, and $\mathbf{v} \in \mathbb{R}^{3n_p-7}$ is a zero-mean Gaussian random vector with covariance $\mathbf{\Sigma}$. $\mathbf{Q}$ is a $3n_p \times (3n_p-7)$ dimensional orthogonal matrix, and $\mathbf{P}_N\big(\overline{\mathbf{Y}}\big)$ is a $3n_p \times 7$ matrix. The rank of $\mathbf{P}_N$ is 7, which corresponds to the degrees of freedom of a 3D similarity transform (1 for scale, 3 for rotation, and 3 for translation). It can be shown that the columns of $\mathbf{P}_N$ form the basis for the shape changes of $\overline{\mathbf{Y}}$ due to infinitesimal variations of similarity transforms. This distribution is denoted as $\mathbf{Y} \sim \mathcal{N}_P\big(\overline{\mathbf{Y}}, \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^T\big)$. $\mathbf{P}_N\big(\overline{\mathbf{Y}}\big)$ in the above equation can be described as [8]

$$\begin{aligned} \mathbf{P}_N &= \begin{bmatrix}\mathrm{vec}\big(\overline{\mathbf{Y}}\big) & \mathbf{K}\big(\overline{\mathbf{Y}}\big) & (\mathbf{1}\otimes\mathbf{I})\end{bmatrix}, \\ \mathbf{K}\big(\overline{\mathbf{Y}}\big) &= \begin{bmatrix}\big[\overline{\mathbf{y}}_1\big]_\times & \cdots & \big[\overline{\mathbf{y}}_{n_p}\big]_\times\end{bmatrix}^T, \end{aligned} \tag{2}$$

where $\overline{\mathbf{y}}_i$ is the ith column vector of $\overline{\mathbf{Y}}$, $\otimes$ is the Kronecker product, and $[\mathbf{y}]_\times \in \mathbb{R}^{3\times 3}$ is the matrix equivalent of the cross product of $\mathbf{y}$ with another vector [8] and is given as

$$[\mathbf{y}]_\times = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix}. \tag{3}$$

Each column of $\mathbf{P}_N$ represents different motion changes by similarity transform, *i.e.*, the first column is related to scale changes, the second to fourth columns to rotation changes, and the others to translation changes. Since $\mathbf{P}_N$ is related to the similarity transform of the mean shape, the no-rigid-motion state of an individual sample in a PND is determined with respect to (w.r.t.) the mean shape. The conceptual diagram of the PND is depicted in Fig. 1.

An important property of the PND is that its instances are aligned shapes obtained by the modified GPA problem [8]. GPA, which is a standard algorithm for aligning shapes, superimposes multiple landmark shapes to a common reference using rigid transforms. Let us consider the following GPA problem with a modified scale constraint [8]:

$$\begin{aligned} \min_{s_i,\mathbf{R}_i,\mathbf{t}_i,\overline{\mathbf{Y}}} \quad & \sum \left\| \mathbf{s}_i\mathbf{R}_i\mathbf{Y}_i + \mathbf{t}_i\mathbf{1}^T - \overline{\mathbf{Y}} \right\|^2 \\ \text{subject to} \quad & \left\|\overline{\mathbf{Y}}\right\| = 1, \quad \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}, \\ & \mathbf{s}_i \operatorname{tr}\left(\mathbf{R}_i\mathbf{Y}_i\overline{\mathbf{Y}}^T\right) = 1. \end{aligned} \tag{4}$$

Here, $\mathbf{s}_i$, $\mathbf{R}_i$, and $\mathbf{t}_i$ are the scale, rotation, and translation of the ith shape $\mathbf{Y}_i$. If $\{\mathbf{Y}_i\}$ are all the possible samples of a PND, then $\mathbf{s}_i = 1$, $\mathbf{R}_i = \mathbf{I}$, and $\mathbf{t}_i = \mathbf{0}$ is very close[1] to a local optimum of this problem, *i.e.*, PND samples no longer needs translation, rotation, and scaling in solving this modified GPA problem. This means that a PND only contains aligned shapes as samples, excluding any rigid motion changes. Due to this property, the PND can accurately model the deformation of a shape, which helps to effectively separate deformation from motion in NRSfM.

The modified scale constraint in this GPA problem (the last constraint in (4)) makes each individual shape lie on a linear subspace which is tangent to the mean shape [8], whereas the original scale constraint in GPA makes the aligned shapes lie on a unit-ball, which requires more effort in modeling a probability distribution. It is this property of the modified scale constraint that makes the PND possible to be defined as a normal distribution. The probability density function (pdf) of a PND is given as follows:

$$\begin{aligned} p(\mathbf{Y}) &\propto \frac{1}{|\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{v}'^T\mathbf{Q}\mathbf{\Sigma}^{-1}\mathbf{Q}^T\mathbf{v}'\right)\delta\left(\mathbf{Q}_N^T\mathbf{v}'\right), \\ \mathbf{v}' &= \operatorname{vec}\left(\mathbf{Y} - \overline{\mathbf{Y}}\right), \end{aligned} \tag{5}$$

[1]This trivial solution is not a local optimum, because the domain of the PND excludes all rigid transformed shapes, but some of reflected ones. However, the probability of the shapes with these reflections are usually very small in a PND.

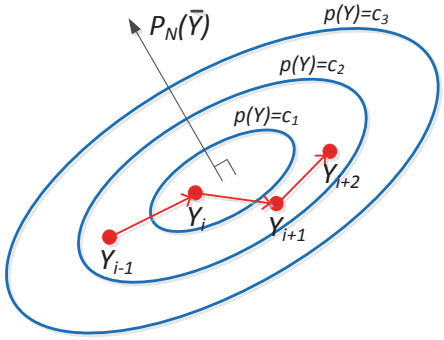


Figure 1. The conceptual diagrams of the PND and PMP. The PND (blue) is orthogonal to the subspace $\mathbf{P}_N(\overline{\mathbf{Y}})$, which is related to similarity transforms of its mean shape $\overline{\mathbf{Y}}$. The PMP (red) is a stationary Markov process whose steady-state distribution is a PND.

where $\mathbf{Q}_N$ is an orthogonal matrix that spans the same space as $\mathbf{P}_N$. Samples Y that satisfy $\mathbf{Q}_N^T\mathbf{v}' \neq 0$ constitute the null space of the PND. Therefore, when $\mathbf{Y}$ is not in the null space, $\delta\left(\mathbf{Q}_N^T\mathbf{v}'\right)$ becomes 1 and $p(\mathbf{Y})$ becomes a Gaussian distribution.

2.2. Definition of the PMP

Although the PND has demonstrated promising results in NRSfM [8], it does not incorporate the smoothness assumption which can be useful for data with large changes. The PMP is a Markov process version of the PND to handle large changes during deformation.

Let us define a first-order linear Markov process as

$$\operatorname{vec}(\mathbf{Y}_i) = \alpha \operatorname{vec}\left(\mathbf{Y}_{i-1} - \overline{\mathbf{Y}}\right) + \operatorname{vec}\left(\overline{\mathbf{Y}}\right) + \boldsymbol{\omega}_i, \tag{6}$$

where α is a smoothness parameter and $\boldsymbol{\omega}_i$ is an i.i.d. Gaussian random vector. The physical meaning of this model is that a shape's deformation from the mean shape is additively changed from that of its previous shape, which is multiplied by the smoothness parameter. Let us also assume that this Markov process is stationary and the steady-state distribution of $\mathbf{Y}_i$ is a PND, *i.e.*,

$$\mathbf{Y}_i \sim \mathcal{N}_P\left(\overline{\mathbf{Y}}, \mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^T\right). \tag{7}$$

Then, in order to make this assumption valid, $\boldsymbol{\omega}_i$ must be of the form,

$$\boldsymbol{\omega}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}\mathbf{H}\mathbf{Q}^T\right), \tag{8}$$

and the following condition can be derived from computing the steady-state covariance matrices in both sides of (6):

$$\mathbf{\Sigma} = \alpha^2\mathbf{\Sigma} + \mathbf{H}. \tag{9}$$

The condition, $\|\alpha\| \leq 1$, follows from the positive definiteness of $\mathbf{\Sigma}$ and $\mathbf{H}$. This is a very simple case of Lyapunov equation [7]. Since this condition ensures the stationarity of the Markov process, we call this as the stationarity constraint of the PMP. The reason we define the transition α

as a scalar rather than a matrix is to make the optimization problem solvable with the stationarity constraint in NRSfM. If a matrix is used instead of α in (6), the Lyapunov equation derived from the steady-state covariance of (6) becomes more complex and the optimization becomes very difficult to solve. As mentioned in Section 1, the stationarity assumption is important to prevent unlikely situations such as infinitely extending arms.

Since $\mathbf{Y}_i - \overline{\mathbf{Y}} = \mathbf{Q}\mathbf{Q}^T\left(\mathbf{Y}_i - \overline{\mathbf{Y}}\right)$ and $\mathbf{Q}^T\overline{\mathbf{Y}} = \mathbf{0}$ from (1) and (2), the Markov process (6) can also be expressed as

$$\mathrm{vec}(\mathbf{Y}_i) = \alpha\mathbf{Q}\mathbf{Q}^T\,\mathrm{vec}(\mathbf{Y}_{i-1}) + \mathrm{vec}\big(\overline{\mathbf{Y}}\big) + \boldsymbol{\omega}_i, \quad (10)$$

and the conditional distribution is given as

$$p(\mathbf{Y}_i|\mathbf{Y}_{i-1}) \propto \exp\left(-\frac{1}{2}\mathbf{v}''^T\mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^T\mathbf{v}''\right)\delta\big(\mathbf{Q}_N^T\mathbf{v}''\big),$$
$$\mathbf{v}'' = \mathrm{vec}\big(\mathbf{Y}_i - \overline{\mathbf{Y}}\big) - \alpha\mathbf{Q}\mathbf{Q}^T\,\mathrm{vec}(\mathbf{Y}_{i-1}). \quad (11)$$

The smoothness parameter α determines the degree of smoothness in a PMP. A PMP is identical to a series of i.i.d. PND random shapes when $\alpha = 0$, and to a (non-deforming) static shape when $\alpha = 1$. The conceptual diagram of the PMP is also shown in Fig. 1.

The probability of a PMP sequence $\{\mathbf{Y}_i\}, 1 \le i \le n_s$, is given as

$$p(\{\mathbf{Y}_i\}) = p(\mathbf{Y}_1)\prod p(\mathbf{Y}_j|\mathbf{Y}_{j-1}). \quad (12)$$

Interestingly, by manipulating this equation, we can find out that a PMP sequence in reverse order has the same probability:

$$p(\mathbf{Y}_0, \ldots, \mathbf{Y}_{n_s}) = p(\mathbf{Y}_{n_s}, \ldots, \mathbf{Y}_0). \quad (13)$$

This means that the PMP is a reversible Markov process. Indeed, the PMP can be shown to satisfy the detailed balance equation, *i.e.*, $p(\mathbf{Y}_i|\mathbf{Y}_{i-1})\,p(\mathbf{Y}_{i-1}) = p(\mathbf{Y}_{i-1}|\mathbf{Y}_i)\,p(\mathbf{Y}_i)$. Because of this property, many solutions of the optimization process in Section 3 are found in symmetric forms. The fact that the PMP is reversible means that any deformation can occur in reverse order. Most deformations in the real world are reversible, which makes the PMP more appropriate for practical shape deformations.

3. The proposed algorithm: EM-PMP

The proposed algorithm, EM-PMP, learns the parameters of the PMP including α from a given 2D observations using EM algorithm. Let $\mathbf{D}_i \in \mathbb{R}^{n_d \times n_p}$ be the input landmark data, observed by an orthographic camera, of the ith frame, and $\mathbf{W}_i \in \mathbb{R}^{n_d \times n_p}$ be the weight matrix of ones and zeros that indicates whether the corresponding elements are observed or missing. In the case of NRSfM, the last row of $\mathbf{W}_i$ is filled with zeros because the z coordinates are unknown. We also assume that the translation component is initially removed from each $\mathbf{D}_i$ as

$$d_{ijk} \leftarrow \begin{cases} d_{ijk} - \frac{\sum_l w_{ijl}d_{ijl}}{\sum_l w_{ijl}} & \text{if } w_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases}, \quad (14)$$

where d_{ijk} and w_{ijk} are the (j,k)th elements of $\mathbf{D}_i$ and $\mathbf{W}_i$, respectively. After this initialization, $\mathbf{D}_i$ satisfies $\mathbf{D}_i\mathbf{1} = \mathbf{0}$. In the following subsections, we will explain the detailed procedure of EM-PMP.

3.1. Objective function

Let us define the parameter set Φ as $\Phi = \big\{\sigma, s_i, \mathbf{R}_i, \overline{\mathbf{Y}}, \boldsymbol{\Sigma}, \mathbf{H}, \alpha\big\}$ where σ is the standard deviation of the Gaussian noise for each element of $\mathbf{D}_i$. Given that $\mathbf{X}_i$ is the set of hidden parameters representing the true 3D shape of the ith frame, the objective function is given as

$$\log(p(\{\mathbf{D}_i, \mathbf{X}_i\}|\Phi)) = \log(p(\{\mathbf{X}_i\}|\Phi)) + \sum\log(p(\mathbf{D}_i|\mathbf{X}_i, \Phi)). \quad (15)$$

The likelihood of $\mathbf{D}_i$ is defined as

$$p(\mathbf{D}_i|\mathbf{X}_i, \Phi) \propto \frac{1}{\sigma^{n_i^W}}\exp\left(-\frac{1}{2\sigma^2}g_i^2\right),$$
$$g_i^2 = \min_{\mathbf{t}_i}\|\mathbf{D}_i - \mathbf{W}_i \odot (\mathbf{X}_i - \mathbf{t}_i)\|^2, \quad (16)$$
$$n_i^W = \sum_j \max\Big\{0, \Big(\sum_k w_{ijk}\Big) - 1\Big\},$$

where $\odot$ is the Hadamard product. This means that $\mathbf{D}_i$ is represented as a combination of Gaussian noise and the remnant of $\mathbf{W}_i \odot \mathbf{X}_i$ after removing the translation component. g_i can be alternatively expressed as

$$g_i^2 = \|\mathrm{vec}(\mathbf{D}_i) - \mathbf{F}_i\,\mathrm{vec}(\mathbf{X}_i)\|^2,$$
$$\mathbf{F}_i = \widehat{\mathbf{W}}_i - \widehat{\mathbf{W}}_i\left(\mathbf{1}\mathbf{1}^T \otimes \mathrm{diag}(\mathbf{c}_i)\right)\widehat{\mathbf{W}}_i, \quad (17)$$
$$\widehat{\mathbf{W}}_i = \mathrm{diag}(\mathrm{vec}(\mathbf{W}_i)),$$

where $\mathbf{c}_i$ is an n_d-dimensional vector whose jth element is $\frac{1}{\sum_l w_{ijl}}$ (0 if the jth row of $\mathbf{W}_i$ is filled with zeros), and $\mathrm{diag}(\mathbf{a})$ denotes a diagonal matrix with the elements of vector $\mathbf{a}$ on the main diagonal. Note that $\mathbf{F}_i$ is a projection matrix, *i.e.*, $\mathbf{F}_i^2 = \mathbf{F}_i$, and multiplying $\mathbf{F}_i$ to $\mathrm{vec}(\mathbf{X}_i)$ removes the translation component in $\mathbf{X}_i$ as in (14).

For the prior distribution of $\mathbf{X}_i$, we assume that the aligned shapes $\{\mathbf{Y}_i\} = \{s_i\mathbf{R}_i\mathbf{X}_i\}$ are a PMP sequence with the probability density function

$$p(\mathbf{Y}_1|\Phi) \sim \mathcal{N}_P\big(\overline{\mathbf{Y}}, \mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T\big),$$
$$p(\mathbf{Y}_i|\mathbf{Y}_{i-1}, \Phi) \sim \mathcal{N}\big(\alpha\left(\mathbf{Y}_{i-1} - \overline{\mathbf{Y}}\right) + \overline{\mathbf{Y}}, \mathbf{Q}\mathbf{H}\mathbf{Q}^T\big). \quad (18)$$

Note that $p(\mathbf{Y}_i|\mathbf{Y}_{i-1}, \Phi)$ is not a Procrustean distribution but a Gaussian distribution as described in Section 2.2.

3.2. E-step

In E-step, we calculate the distribution of $\{\mathbf{X}_i\}$ given $\{\mathbf{D}_i\}$ and the current estimates of Φ. Since $\{\mathbf{X}_i\}$ is a first-order linear Markov process, we can calculate its distribution using Kalman smoothing [11]. Kalman smoothing is composed of forward and backward steps, and the forward step is composed of predict and update steps. For ease of explanation, we will describe the steps w.r.t. the aligned shapes $\{\mathbf{Y}_i\} = \{s_i\mathbf{R}_i\mathbf{X}_i\}$.

Let $\boldsymbol{\mu}_{1|1}$ and $\mathbf{C}_{1|1}$ be the mean and covariance, respectively, of $\mathrm{vec}(\mathbf{Y}_1)$ estimated from the observation $\mathbf{D}_1$ and its prior distribution. If we denote $\mathbf{D}'_i = \frac{1}{s_i}\mathbf{R}_i^T\mathbf{D}_i$, $\mathbf{R}'_i = (\mathbf{I}\otimes\mathbf{R}_i)$, and $\mathbf{F}'_i = \frac{1}{s_i^2}\mathbf{R}'_i\mathbf{F}_i\mathbf{R}_i'^T$, then they are

$$\begin{aligned}\mathbf{C}_{1|1} &= \Big(\frac{1}{\sigma^2}\mathbf{F}'_1 + \mathbf{Q}\boldsymbol{\Sigma}^{-1}\mathbf{Q}^T\Big)^{-1},\\ \boldsymbol{\mu}_{1|1} &= \frac{1}{\sigma^2}\mathbf{C}_{1|1}\,\mathrm{vec}(\mathbf{D}'_1)\,.\end{aligned} \tag{19}$$

In the predict step, the distribution of a frame is predicted based on that of its previous frame as

$$\begin{aligned}\boldsymbol{\mu}_{i|i-1} &= \alpha s_{i-1}\mathbf{Q}\mathbf{Q}^T\mathbf{R}'_{i-1}\boldsymbol{\mu}_{i-1|i-1} + \mathrm{vec}\big(\overline{\mathbf{Y}}\big)\,,\\ \mathbf{C}_{i|i-1} &= \mathbf{Q}\left(\alpha^2\mathbf{Q}^T\mathbf{C}_{i|i-1}\mathbf{Q} + \mathbf{H}\right)\mathbf{Q}^T.\end{aligned} \tag{20}$$

In the update step, these estimates are updated based on the observation $\mathbf{D}_i$:

$$\begin{aligned}\mathbf{C}_{i|i} &= \Big(\frac{1}{\sigma^2}\mathbf{F}'_i + \mathbf{Q}\left(\mathbf{Q}^T\mathbf{C}_{i|i-1}\mathbf{Q}\right)^{-1}\mathbf{Q}^T\Big)^{-1},\\ \boldsymbol{\mu}_{i|i} &= \boldsymbol{\mu}_{i|i-1} + \frac{1}{\sigma^2}\mathbf{C}_{i|i}\Big(\mathrm{vec}(\mathbf{D}'_i) - \mathbf{F}'_i\boldsymbol{\mu}_{i|i-1}\Big)\,.\end{aligned} \tag{21}$$

In the forward step, these predict and update steps are alternated for all n_s frames. The result of this forward step is the distributions $p(\mathbf{Y}_i|\mathbf{D}_1,\ldots,\mathbf{D}_i)$. We have not mentioned that we have ignored the Dirac-delta term in (11) for deriving these update steps. It is because the Dirac-delta term makes the updates of the parameters too small in M-step [8], so we have ignored it to speed up the process.

In the backward step, we calculate the mean and covariance of $p(\mathbf{Y}_i|\mathbf{D}_1,\ldots,\mathbf{D}_{n_s})$ from $p(\mathbf{Y}_i|\mathbf{D}_1,\ldots,\mathbf{D}_i)$ and $p(\mathbf{Y}_{i+1}|\mathbf{D}_1,\ldots,\mathbf{D}_{n_s})$ as

$$\begin{aligned}\boldsymbol{\mu}_{i|n_s} &= \boldsymbol{\mu}_{i|i} + \mathbf{L}_i\left(\boldsymbol{\mu}_{i+1|n_s} - \boldsymbol{\mu}_{i+1|i}\right),\\ \mathbf{C}_{i|n_s} &= \mathbf{C}_{i|i} + \mathbf{L}_i\left(\mathbf{C}_{i+1|n_s} - \mathbf{C}_{i+1|i}\right)\mathbf{L}_i^T,\end{aligned} \tag{22}$$

where

$$\mathbf{L}_i = \alpha\mathbf{C}_{i|i}\mathbf{Q}\left(\mathbf{Q}^T\mathbf{C}_{i+1|i}\mathbf{Q}\right)^{-1}\mathbf{Q}^T. \tag{23}$$

For the M-step of EM-PMP, the cross-covariance of $\mathrm{vec}(\mathbf{Y}_i)$ and $\mathrm{vec}(\mathbf{Y}_{i+1})$ is also required, which can be easily calculated as

$$\mathbf{C}_{i,i+1|n_s} = \mathbf{L}_i\mathbf{C}_{i+1|n_s}. \tag{24}$$

To simplify the notations, we denote $\boldsymbol{\mu}_i = \boldsymbol{\mu}_{i|n_s}$, $\mathbf{C}_i = \mathbf{C}_{i|n_s}$, and $\mathbf{C}_{i,i+1} = \mathbf{C}_{i,i+1|n_s}$ if no confusion arises.

After finding the distribution of sequence $\{\mathbf{Y}_i\}$, we can express the distributions of sequence $\{\mathbf{X}_i\}$ by

$$\begin{aligned}\boldsymbol{\mu}'_i &= \frac{1}{s_i}\mathbf{R}_i'^T\boldsymbol{\mu}_i, \quad \mathbf{C}'_i = \frac{1}{s_i^2}\mathbf{R}_i'^T\mathbf{C}_i\mathbf{R}'_i,\\ \mathbf{C}'_{i,i+1} &= \frac{1}{s_i s_{i+1}}\mathbf{R}_i'^T\mathbf{C}_{i,i+1}\mathbf{R}'_{i+1},\end{aligned} \tag{25}$$

where $\boldsymbol{\mu}'_i$ and $\mathbf{C}'_i$ are the mean and covariance of $\mathrm{vec}(\mathbf{X}_i)$, respectively, and $\mathbf{C}'_{i,i+1}$ is the cross-covariance of $\mathrm{vec}(\mathbf{X}_i)$ and $\mathrm{vec}(\mathbf{X}_{i+1})$.

3.3. M-step

In M-step, parameter set Φ is updated so that it maximizes the expectation of (15) w.r.t. $p(\{\mathbf{X}_i\})$ estimated in E-step. The cost function of M-step is given as follows:

$$\begin{aligned}J(\Phi) &= J_{\mathbf{D}} + J_{\mathbf{X}},\\ J_{\mathbf{D}} &= \sum -n_i^W\log(\sigma) - \frac{1}{2\sigma^2}\Big(\left\|\mathrm{vec}(\mathbf{D}_i) - \mathbf{F}_i\boldsymbol{\mu}'_i\right\|^2\\ &\quad + \mathrm{tr}(\mathbf{F}_i\mathbf{C}'_i)\Big),\\ J_{\mathbf{X}} &= \Big(-\frac{n_s}{2}\log|\mathbf{H}| + (3n_p-7)\sum\log(s_i)\Big) + \frac{3n_p-7}{2}\\ &\quad \log\big(1-\alpha^2\big) - \frac{1-\alpha^2}{2}\,\mathrm{tr}\big(\mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^T\left(\mathbf{h}_1\mathbf{h}_1^T + \mathbf{C}_i\right)\big)\\ &\quad - \frac{1}{2}\sum\mathrm{tr}\left(\mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^T\big(\mathbf{h}'_i\mathbf{h}_i'^T + \mathbf{C}_i + \alpha^2\mathbf{C}_{i-1}\right.\\ &\quad \left.- 2\alpha\mathbf{C}_{i,i-1}\big)\right),\end{aligned} \tag{26}$$

where

$$\begin{aligned}\mathbf{h}_i &= \boldsymbol{\mu}_i - \mathrm{vec}\big(\overline{\mathbf{Y}}\big)\,,\\ \mathbf{h}'_i &= \boldsymbol{\mu}_i - \mathrm{vec}\big(\overline{\mathbf{Y}}\big) - \alpha\mathbf{Q}\mathbf{Q}^T\boldsymbol{\mu}_{i-1},\ \text{for}\ i>1.\end{aligned} \tag{27}$$

Note that the stationarity constraint (9) is substituted to this cost function to eliminate the term $\boldsymbol{\Sigma}$. Based on this cost function, the optimization problem is described as

$$\begin{aligned}&\max_{\Phi}\quad J(\Phi)\\ &\text{subject to}\quad \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}, \qquad \left\|\overline{\mathbf{Y}}\right\|^2 = 1,\\ &\qquad s_i\,\mathrm{tr}\Big(\mathbf{R}_i\mathbf{M}_i\overline{\mathbf{Y}}^T\Big) = 1, \quad \mathbf{R}_i\mathbf{M}_i\overline{\mathbf{Y}}^T \in \mathbf{S}_+^{n_d},\end{aligned} \tag{28}$$

where $\mathrm{vec}(\mathbf{M}_i) = \boldsymbol{\mu}'_i$. All the constraints in this problem are the PND constraints except that $\mathbf{X}_i$ is replaced with its expectation $\mathbf{M}_i$.

This is a difficult problem to solve because $\mathbf{Q}$ is a complicated function of $\overline{\mathbf{Y}}$. Therefore, we regard $\mathbf{Q}$ and $\overline{\mathbf{Y}}$ as independent variables, and solve this problem alternatingly

for each parameter. By solving $\partial J/\partial \overline{\mathbf{Y}} = 0$ and normalizing the solution, we obtain the update equations for $\overline{\mathbf{Y}}$:

$$\begin{aligned} \overline{\mathbf{Y}}' &= \Big(\sum_{i=1}^{n_s} \boldsymbol{\mu}_i\Big) - \alpha \mathbf{Q}\mathbf{Q}^T \Big(\sum_{i=2}^{n_s-1} \boldsymbol{\mu}_i\Big), \\ \mathrm{vec}\big(\overline{\mathbf{Y}}\big) &= \overline{\mathbf{Y}}' / \left\|\overline{\mathbf{Y}}'\right\|. \end{aligned} \tag{29}$$

$\mathbf{Q}$ is updated based on this new $\overline{\mathbf{Y}}$ as follows: We first calculate $\mathbf{P}_N$ using (2) and then an orthogonal matrix perpendicular to $\mathbf{P}_N$ is computed using QR decomposition to find $\mathbf{Q}$. The scales and rotations are updated based on the PND constraints because the feasible solutions are unique, given that the frames are non-degenerate. Then the update equations become

$$\begin{aligned} \mathbf{M}_i \overline{\mathbf{Y}}^T &= \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{V}_i^T, \\ \mathbf{R}_i &= \mathbf{V}_i \mathbf{U}_i^T, \quad s_i = 1/\operatorname{tr}(\mathbf{\Lambda}_i), \end{aligned} \tag{30}$$

where the first equation is calculated using SVD.

The equation $\partial J/\partial \alpha = 0$ can be rearranged as a cubic equation, *i.e.*,

$$b\alpha^3 - c\alpha^2 - (b + 3n_p - 7)\alpha + c = 0. \tag{31}$$

where

$$\begin{aligned} b &= \sum_{i=2}^{n_s-1} \operatorname{tr}\big(\mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^T \left(\mathbf{h}_i \mathbf{h}_i^T + \mathbf{C}_i\right)\big), \\ c &= \sum_{i=2}^{n_s} \operatorname{tr}\big(\mathbf{Q}\mathbf{H}^{-1}\mathbf{Q}^T \left(\mathbf{h}_{i-1} \mathbf{h}_i^T + \mathbf{C}_{i-1,i}\right)\big). \end{aligned} \tag{32}$$

We can show that this equation always has one real solution in the range of $[-1, 1]$. (see the supplementary material.) Therefore, we can always find a unique α, which can be easily found by solving this cubic equation.

$\mathbf{H}$ can be found by solving $\partial J/\partial \mathbf{H} = 0$, *i.e.*,

$$\begin{aligned} \mathbf{H} = \frac{1}{n_s}\mathbf{Q}^T \Big((1-\alpha^2)\mathbf{h}_1\mathbf{h}_1^T + (1-\alpha^2)\mathbf{C}_1 + \sum_{i=2}^{n_s} \mathbf{h}_i'\mathbf{h}_i'^T \\ + \mathbf{C}_i + \alpha^2 \mathbf{C}_{i-1} - \alpha \mathbf{C}_{i-1,i} - \alpha \mathbf{C}_{i-1,i}^T\Big)\mathbf{Q}. \end{aligned} \tag{33}$$

Accordingly, $\mathbf{\Sigma}$ is calculated from (9) as $\mathbf{\Sigma} = \frac{1}{1-\alpha^2}\mathbf{H}$. σ can also be similarly calculated, however, we have empirically found out that σ decreases too fast to yield a good solution in the EM iterations. This is because it is much easier to decrease σ than to adjust the other parameters in increasing J. Therefore, we introduce a constant β (> 1) to compensate this behavior:

$$\sigma^2 = \frac{\beta}{\sum n_i^W} \sum \left\|\mathrm{vec}(\mathbf{D}_i) - \mathbf{F}_i \mathbf{m}_i\right\|^2 + \operatorname{tr}(\mathbf{F}_i \mathbf{C}_i). \tag{34}$$

For the experiments, we used $\beta = 2$.

This update procedure should be repeated until convergence in each M-step, but we have found empirically that a single iteration for each step is sufficient.

3.4. Initialization and 3D reconstruction

For the initialization of s_i, $\mathbf{R}_i$, and $\overline{\mathbf{Y}}$, the initialization scheme in [5] is used. We also use the pre-iteration stage of EM-PND to speed up the overall process [8]. α is initialized by solving the following problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{1-\alpha^2} \sum \left\|\mathbf{Y}_i' - \alpha \mathbf{Y}_{i-1}'\right\|^2 \\ \text{subject to} \quad & |\alpha| \le 1, \end{aligned} \tag{35}$$

where $\mathbf{Y}_i' = s_i \mathbf{R}_i \mathbf{X}_i - \overline{\mathbf{Y}}$. This is equivalent to minimizing the trace of the sample estimate of $\mathbf{\Sigma}$. Here, we use the results of the pre-iteration stage for the missing entries of $\mathbf{X}_i$. The solution to this problem can be easily found as

$$\begin{aligned} \alpha &= \kappa - \sqrt{\kappa^2 - 1}, \\ \kappa &= \frac{\left\|\mathbf{Y}_1'\right\|^2 + \left\|\mathbf{Y}_{n_s}'\right\|^2 + 2\sum_{i=2}^{n_s-1} \left\|\mathbf{Y}_i'\right\|^2}{2\sum_{i=2}^{n_s} \operatorname{tr}\big(\mathbf{Y}_{i-1}'^T \mathbf{Y}_i'\big)}. \end{aligned} \tag{36}$$

$\mathbf{\Sigma}$, $\mathbf{H}$, and σ were initialized as $10^{-3}\mathbf{I}$, $\frac{10^{-3}}{1-\alpha^2}\mathbf{I}$, and 10^{-2}, respectively, in the experiments of Section 4. The EM procedure was terminated when the change of $\frac{1}{n_s(3n_p-7)}J$ became less than 0.01. After performing the EM algorithm until convergence, the final sequence $\{\mathbf{M}_i\}$ become the reconstructed 3D shapes.

4. Experimental Results

We performed NRSfM experiments using EM-PMP for various data sets. We used the well-known motion capture data sets [2, 13] and also the manually annotated data of the Face Recognition Grand Challenge (FRGC) 2.0 Database [10] used in [8]. The annotated data, having no temporal dependence, was tested in order to see how EM-PMP performs in such extreme cases.

We conducted experiments with/without noise and/or missing points. The standard deviation of the Gaussian noise was set as $\sigma_{\text{noise}} = 0.02 \max_{i,j,k}\{|d_{ijk}|\}$. For the missing data, we randomly set 30 percents of the landmarks as missing. We compared the propose method with other state-of-the-art schemes, which were EM-PPCA, EM-LDS [13], CSF2 [5], SPM [4], and EM-PND [8]. The parameters of these methods were set in accordance with their original paper, and we used the block matrix method for SPM, which is known to be the best algorithm according to [4]. The performance was evaluated in terms of normalized reconstruction error, *i.e.*,

$$e_i = \|\widehat{\mathbf{X}}_i - \mathbf{X}_i^*\| / \|\mathbf{X}_i^*\|, \tag{37}$$

Table 1. Average errors w/o noise and missing data

data	PPCA	LDS	CSF2	SPM	PND	PMP
dance	0.2325	0.3112	0.1349	0.1454	0.1834	0.1278
drink	0.1292	0.1463	0.0123	0.0216	0.0037	0.0018
pickup	0.5149	0.4422	0.0607	0.0356	0.0372	0.0127
stretch	0.5393	0.3948	0.0219	0.0288	0.0156	0.0124
yoga	0.6100	0.4643	0.0226	0.0224	0.0140	0.0128
face	0.0208	0.0331	0.0209	0.0233	0.0165	0.0166
shark	0.0688	0.1109	0.0551	0.5475	0.0134	0.0099
walking	0.1485	0.3318	0.0708	0.0861	0.0465	0.0424
FRGC	0.1469	0.1345	0.1926	0.1094	0.0727	0.0727
av. err.	0.2679	0.2632	0.0658	0.1133	0.0448	0.0343
rel. err.	15.5310	14.5405	2.0289	6.2956	1.0000	0.7656

Table 2. Average errors w/ noise and w/o missing data

data	PPCA	LDS	CSF2	SPM	PND	PMP
dance	0.2229	0.3185	0.1544	0.1510	0.1806	0.1415
drink	0.1764	0.1507	0.0365	0.0407	0.0339	0.0244
pickup	0.5037	0.3979	0.0705	0.0581	0.0409	0.0304
stretch	0.5479	0.3872	0.0543	0.0652	0.0444	0.0341
yoga	0.5287	0.5162	0.0529	0.0822	0.0409	0.0306
face	0.0464	0.0598	0.0543	0.1054	0.0403	0.0405
shark	0.0486	0.0827	0.1043	0.1784	0.0600	0.0610
walking	0.1364	0.2474	0.0966	–	0.0770	0.0929
FRGC	0.1980	0.1228	0.2061	0.1840	0.0889	0.0887
av. err.	0.2677	0.2537	0.0922	0.1081	0.0674	0.0605
rel. err.	5.5533	4.9707	1.4256	1.8242	1.0000	0.8876

Table 3. Average errors w/ missing data and w/o noise

data	PPCA	LDS	CSF2	PND	PMP
dance	0.2632	1.1464	0.1415	0.1766	0.1410
drink	0.1692	0.0790	0.0357	0.0055	0.0018
pickup	0.4969	0.2954	0.0933	0.0149	0.0151
stretch	0.6735	0.3899	0.0597	0.0150	0.0123
yoga	0.1463	0.4051	0.0854	0.0181	0.0166
face	0.2978	0.6386	0.0412	0.0177	0.0174
shark	0.1374	63.3659	0.0653	0.0166	0.0116
walking	0.1361	12.0950	0.1033	0.0469	0.0507
FRGC	0.1764	0.1122	0.4505	0.0805	0.0802
av. err.	0.2774	8.7253	0.1195	0.0435	0.0385
rel. err.	16.5312	466.8489	4.0347	1.0000	0.8484

Table 4. Average errors w/ noise and missing data

data	PPCA	LDS	CSF2	PND	PMP
dance	0.2732	0.4292	0.1501	0.1601	0.1445
drink	0.1868	13.3213	0.0428	0.0408	0.0299
pickup	0.4931	0.3310	0.0948	0.0486	0.0362
stretch	0.6881	0.3379	0.0707	0.0535	0.0405
yoga	0.1401	0.4084	0.1583	0.0488	0.0357
face	0.2273	0.5889	0.0583	0.0464	0.0460
shark	0.1323	2.0375	0.0872	0.0672	0.0669
walking	0.1541	3.4281	0.1095	0.0842	0.0996
FRGC	0.2196	0.1171	0.4574	0.0968	0.0966
av. err.	0.2794	2.3333	0.1366	0.0718	0.0662
rel. err.	4.7922	48.4016	1.8980	1.0000	0.8930

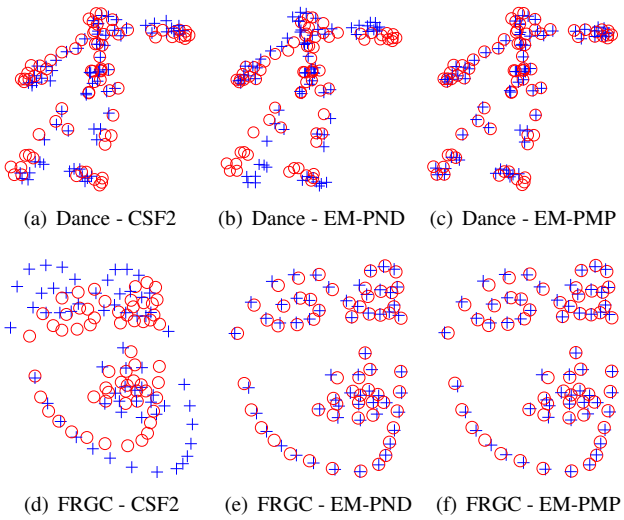
where $\mathbf{X}_i^*$ and $\widehat{\mathbf{X}}_i$ is the ith ground truth and reconstructed shapes, respectively. Since a reconstructed shape have reflection ambiguity, we also measured the error for the inverted shape and picked the smaller error. All the experiments were repeated ten times and the results were averaged.

Tables 1-4 show the reconstruction errors under various conditions. Here, we dropped the prefix 'EM-' in the names of the methods due to the space limitations. There are no experimental results for missing data using SPM because SPM does not provide a method to handle missing data. When an algorithm does not converge, its result is denoted as "–". "av. err." means the average error over all data set and "rel. err." means the average relative error compared to EM-PND. Here, EM-PMP gives the best performance or at least very close to the best performance except four cases. The performance gain of EM-PMP w.r.t. the second best method, EM-PND, is over 20% for about half of the cases and is up to 67%, which indicates the effectiveness of the PMP model in NRSfM. For the case of the shark sequence with noise, EM-PPCA gives better performance than EM-PMP. This is because the sequence was artificially generated by superposing two shape bases [13]. Thus this data satisfies the fixed-rank assumptions of EM-PPCA, giving the best performance. However, the rank of shape bases in not known in real cases and EM-PPCA shows poor performance for the other cases. For the cases of the walking sequence with noise and/or missing data, EM-PND gives better performance than EM-PMP. This seems to be due to the characteristics of the data where the view points in the walking sequence is static for the most of the time. Because of the Markovian nature of the PMP, the z-coordinates tend to converge to the mean shape, if there are not enough frames of different view points. In these cases, EM-PND can be a better choice. However, even in these cases, the performance of EM-PMP is better than the other methods except EM-PND.

Note that the performances of EM-PND and CSF2 depend on the characteristics of data. CSF2 gives very good performance for the dance sequence which has large deformations, because it assumes temporal dependence between frames. On the contrary, it performs poorly for the FRGC sequence because it is an extreme case that does not have any temporal dependence. EM-PND shows exactly the opposite trend compared to CSF2. On the other hand, EM-PMP gives very good performance for both cases, because the smoothness parameter is learned during the fitting process. It is remarkable that EM-PMP performs as good as EM-PND when there is no temporal dependence, which means that EM-PMP can perform like EM-PND in such cases. Hence, EM-PMP can be considered as a better solution for most of the cases.

Another thing to note is the performance of EM-LDS. EM-LDS also uses a Markov model as EM-PMP, but its performance gain is not significant compared to its time-independent version, EM-PPCA. As mentioned in Section 1, this is due to the lack of appropriate assumptions for shape deformation. We used the stationarity assumption in our model and achieved much better results, which suggest that our assumption is very effective for NRSfM.

EM-PMP has no parameters to be determines by a user, which makes it easy to use in practical applications. EM-PMP takes around five minutes to reconstruct one of these sequences, which is moderate compared to the other state-of-the-art schemes (a few hours for SPM, around 40 seconds for CSF2, and around one minute for EM-PND). Figure 2 compares the reconstructed points and the corresponding ground truth for EM-PMP, EM-PND, and CSF2. Here, we can see that EM-PMP gives better fits. The videos of reconstructed shapes are also provided in the supplementary material to confirm the performance of EM-PMP.



(a) Dance - CSF2 (b) Dance - EM-PND (c) Dance - EM-PMP

(d) FRGC - CSF2 (e) FRGC - EM-PND (f) FRGC - EM-PMP

Figure 2. Reconstructed results of EM-PMP, EM-PND, and CSF2 ($\circ$: ground truth, $+$: reconstructed points).

5. Conclusion

In this paper, we proposed EM-PMP to solve NRSfM problems incorporating the smoothness constraint, without requiring any prior information. The PMP is an extended version of the PND to a first-order Markov process, which is assumed to be stationary in order to avoid unlikely deformation. The additive changes in the PMP is constrained to satisfy the PND constraint, which forces a sequence of PMP shapes to be constituted by only shape deformations. Experimental results show that EM-PMP outperforms all the other schemes, which demonstrates that the PMP is a very effective model for NRSfM. Even when there is no temporal dependence in data, EM-PMP shows nearly the same performance as EM-PND because EM-PMP automatically learns the degree of smooth during the fitting process. One of future work is to extend the Procrustean shape model to dense trajectory cases to increase the resolution of 3D reconstruction. Another is to propose an incremental learning method of the shape model so that it can be applied in real-time for non-rigid 3D reconstruction.

References

[1] I. Akhter, Y. Seikh, and S. Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *Proc. Computer Vision and Pattern Recognition*, June 2009.

[2] I. Akhter, Y. Seikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, July 2011.

[3] J. Cho, M. Lee, C.-H. Choi, and S. Oh. EM-GPA: Generalized procrustes analysis with hidden variables for a 3D shape model. *Computer Vision and Image Understanding*, 117(11):1549–1559, Nov 2013.

[4] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. In *Proc. Computer Vision and Pattern Recognition*, June 2012.

[5] P. F. U. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *Proc. Computer Vision and Pattern Recognition*, June 2011.

[6] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, March 1975.

[7] G. Kitagawa. An algorithm for solving the matrix equation $X = FXF^T + S$. *Int'l J. Control*, 25(5):745–753, 1977.

[8] M. Lee, J. Cho, C.-H. Choi, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. In *Proc. Computer Vision and Pattern Recognition*, June 2013.

[9] M. Paladini, A. Del Bue, M. Stošić, M. Dodig, J. ao Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. In *Proc. Computer Vision and Pattern Recognition*, June 2009.

[10] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Proc. Computer Vision and Pattern Recognition*, June 2005.

[11] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the EM algorithm. *J. Time Series Analysis*, 3(4):253–264, July 1982.

[12] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int'l J. Computer Vision*, 9(2):137–154, November 1992.

[13] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(5):878–892, May 2008.

[14] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *Int'l J. Computer Vision*, 67(2):233–246, April 2006.