Subspace Procrustes Analysis

Xavier Perez-Sala[1,3,4], Fernando De la Torre[2], Laura Igual[3,5],
Sergio Escalera[3,5], and Cecilio Angulo[4]

[1] Fundació Privada Sant Antoni Abat, Vilanova i la Geltrú, 08800, Spain
[2] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213. USA
[3] Computer Vision Center, Universitat Autònoma de Barcelona, Bellaterra, Spain.
[4] Universitat Politècnica de Catalunya, Vilanova i la Geltrú, 08800, Spain.
[5] Universitat de Barcelona, Barcelona, 08007, Spain.

Abstract. Procrustes Analysis (PA) has been a popular technique to align and build 2-D statistical models of shapes. Given a set of 2-D shapes PA is applied to remove rigid transformations. Then, a non-rigid 2-D model is computed by modeling (e.g., PCA) the residual. Although PA has been widely used, it has several limitations for modeling 2-D shapes: occluded landmarks and missing data can result in local minima solutions, and there is no guarantee that the 2-D shapes provide a uniform sampling of the 3-D space of rotations for the object. To address previous issues, this paper proposes Subspace PA (SPA). Given several instances of a 3-D object, SPA computes the mean and a 2-D subspace that can simultaneously model all rigid and non-rigid deformations of the 3-D object. We propose a discrete (DSPA) and continuous (CSPA) formulation for SPA, assuming that 3-D samples of an object are provided. DSPA extends the traditional PA, and produces unbiased 2-D models by uniformly sampling different views of the 3-D object. CSPA provides a continuous approach to uniformly sample the space of 3-D rotations, being more efficient in space and time. Experiments using SPA to learn 2-D models of bodies from motion capture data illustrate the benefits of our approach.

1 Introduction

In computer vision, Procrustes Analysis (PA) has been used extensively to align shapes (e.g., [19, 4]) and appearance (e.g., [20, 13]) as a pre-processing step to build 2-D models of shape variation. Usually, shape models are learned from a discrete set of 2-D landmarks through a two-step process [8]. Firstly, the rigid transformations are removed by aligning the training set w.r.t. the mean using PA; next, the remaining deformations are modeled using Principal Component Analysis (PCA) [18, 5].

PA has been widely employed despite suffering from several limitations: (1) The 2-D training samples do not necessarily cover a uniform sampling of all 3-D rigid transformations of an object and this can result in a biased model (i.e., some poses are better represented than others). (2) It is computationally expensive to learn a shape model by sampling all possible 3-D rigid transformations of an

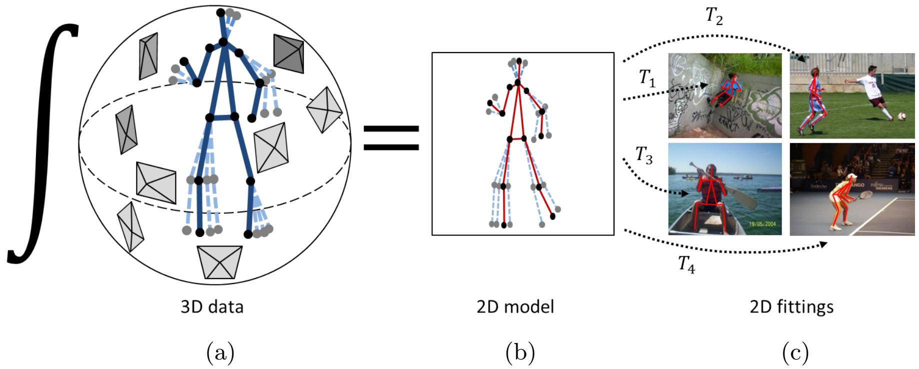


Fig. 1. Illustration of Continuous Subspace Procrustes Analysis (CSPA), which builds an unbiased 2-D model of human joints' variation (b) by integrating over all possible viewpoints of a 3-D motion capture data (a). This 2-D body shape model is used to reconstruct 2-D shapes from different viewpoints (c). Our CSPA model generalizes across poses and camera views because it is learned from a 3-D model.

object. (3) The models that are learned using only 2-D landmarks cannot model missing landmarks due to large pose changes. Moreover, PA methods can lead to local minima problems if there are missing components in the training data. (4) Finally, PA is computationally expensive, it scales linearly with the number of samples and landmarks and quadratically with the dimension of the data.

To address these issues, this paper proposes a discrete and a continuous formulation of Subspace Procrustes Analysis (SPA). SPA is able to efficiently compute the non-rigid subspace of possible 2-D projections given several 3-D samples of a deformable object. Note that our proposed work is the *inverse* problem of Non-Rigid Structure From Motion (NRSFM) [22, 21, 3]. The goal of NRSFM is to recover 3-D shape models from 2-D tracked landmarks, while SPA builds unbiased 2-D models from 3-D data. The learned 2-D model has the same representational power of a 3-D model but leads to faster fitting algorithms [15]. SPA uniformly samples the space of possible 3-D rigid transformations, and it is extremely efficient in space and time. The main idea of SPA is to combine functional data analysis with subspace estimation techniques.

Fig. 1 illustrates the main idea of this work. In Fig. 1 (a), we represent many samples of 3-D motion capture data of humans performing several activities. SPA simultaneously aligns all 3-D samples projections, while computing a 2-D subspace (Fig. 1 (b)) that can represent all possible projections of the 3-D motion capture samples under different camera views. Hence, SPA provides a simple, efficient and effective method to learn a 2-D subspace that accounts for non-rigid and 3-D geometric deformation of 3-D objects. These 2-D subspace models can be used for detection (i.e., constrain body parts, see Fig. 1 (c)), because the subspace models all 3-D rigid projections and non-rigid deformations. As we will show in the experimental validation, the models learned by SPA are able to generalize better than existing PA approaches across view-points (because they are built using 3-D models) and preserve expressive non-rigid deformations. Moreover, computing SPA is extremely efficient in space and time.

2 Procrustes Analysis Revisited

This section describes three different formulations of PA with a unified and enlightening matrix formulation.

Procrustes Analysis (PA): Given a set of m centered shapes (see footnote for notation[1]) composed by ℓ landmarks $\mathbf{D}_i \in \mathbb{R}^{d\times\ell}, \forall i = 1,\ldots,m$, PA [6, 9, 8, 10, 2] computes the d-dimensional reference shape $\mathbf{M} \in \mathbb{R}^{d\times\ell}$ and the m transformations $\mathbf{T}_i \in \mathbb{R}^{d\times d}$ (e.g., affine, Euclidean) that minimize the *reference-space model* [10, 8, 2] (see Fig. 2 (a)):

$$E_R(\mathbf{M},\mathbf{T}) = \sum_{i=1}^{m} ||\mathbf{T}_i\mathbf{D}_i - \mathbf{M}||_F^2, \tag{1}$$

where $\mathbf{T} = [\mathbf{T}_1^T, \cdots, \mathbf{T}_m^T]^T \in \mathbb{R}^{dm\times d}$. In the case of two-dimensional shapes ($d = 2$), $\mathbf{D}_i = \begin{bmatrix} x_1 \; x_2 \; \ldots \; x_\ell \\ y_1 \; y_2 \; \cdots \; y_\ell \end{bmatrix}$. Alternatively, PA can be optimized using the *data-space model* [2] (see Fig. 2 (b)):

$$E_D(\mathbf{M},\mathbf{A}) = \sum_{i=1}^{m} ||\mathbf{D}_i - \mathbf{A}_i\mathbf{M}||_F^2, \tag{2}$$

where $\mathbf{A} = [\mathbf{A}_1^T, \cdots, \mathbf{A}_m^T]^T \in \mathbb{R}^{dm\times d}$. $\mathbf{A}_i = \mathbf{T}^{-1} \in \mathbb{R}^{d\times d}$ is the inverse transformation of $\mathbf{T}_i$ and corresponds to the rigid transformation for the reference shape $\mathbf{M}$.

The error function Eq. (1) of the reference-space model minimizes the difference between the reference shape and the registered shape data. In the data-space model, the error function Eq. (2) compares the observed shape points with the transformed reference shape, i.e., shape points predicted by the model and based on the notion of average shape [23]. This difference between the two models leads to different properties. Since the reference-space cost (E_R, Eq. (1)) is a sum of squares and it is convex in the optimization parameters, it can be optimized globally with Alternated Least Squares (ALS) methods. On the other hand, the data-space cost (E_D, Eq. (2)) is a bilinear problem and non-convex. If there is no missing data, the data-space model can be solved using the Singular Value

[1] Bold capital letters denote a matrix $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. $\mathbf{x}_i$ represents the i^{th} column of the matrix $\mathbf{X}$. x_{ij} denotes the scalar in the i^{th} row and j^{th} column of the matrix $\mathbf{X}$. All non-bold letters represent scalars. $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix. $\|\mathbf{x}\|_2 = \sqrt[2]{\sum_i |x_i|^2}$ and $\|\mathbf{X}\|_F = \sqrt{\sum_{ij} x_{ij}^2}$ denote the 2-norm for a vector and the Frobenius norm of a matrix, respectively. $\mathbf{X} \otimes \mathbf{Y}$ is the Kronecker product of matrices and $\mathbf{X}^{(p)}$ is the vec-transpose operator, detailed in Appendix A.

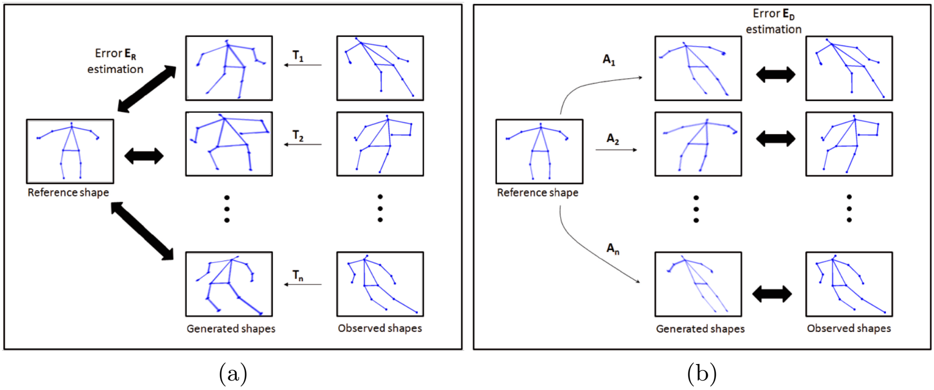


Fig. 2. (a): Reference-space model. (b): Data-space model. Note that $\mathbf{A}_i = \mathbf{T}_i^{-1}$.

Decomposition (SVD). A major advantage of the data-space model is that it is *gauge invariant* (i.e., the cost does not depend on the coordinate frame in which the reference shape and the transformations are expressed) [2]. Benefits of both models are combined in [2]. Recently, Pizarro et al. [19] have proposed a convex approach for PA based on the reference-space model. In their case, the cost function is expressed with a quaternion parametrization which allows conversion to a Sum of Squares Program (SOSP). Finally, the equivalent semi-definite program of a SOSP relaxation is solved using convex optimization.

PA has also been applied to learn appearance models invariant to geometric transformations. When PA is applied to shapes, the geometric transformation (e.g., $\mathbf{T}_i$ or $\mathbf{A}_i$) can be directly applied to the image coordinates. However, to align appearance features the geometric transformations have to be composed with the image coordinates, and the process is a bit more complicated. This is the main difference when applying PA to align appearance and shape. Frey and Jojic [7] proposed a method for learning a factor analysis model that is invariant to geometric transformations. The computational cost of this method grows polynomially with the number of possible spatial transformations and it can be computationally intensive when working with high-dimensional motion models. To improve upon that, De la Torre and Black [20] proposed parameterized component analysis: a method that learns a subspace of appearance invariant to affine transformations. Miller et al. proposed the congealing method [13], which uses an entropy measure to align images with respect to the distribution of the data. Kookinos and Yuille [12] proposed a probabilistic framework and extended previous approaches to deal with articulated objects using a Markov Random Field (MRF) on top of Active Appearance Models (AAMs).

Projected Procrustes Analysis (PPA): Due to advances in 3-D capture systems, nowadays it is common to have access to 3-D shape models for a variety of objects. Given n 3-D shapes $\mathbf{D}_i \in \mathbb{R}^{3\times\ell}$, we can compute r projections $\mathbf{P}_j \in$

$\mathbb{R}^{2\times 3}$ for each of them (after removing translation) and minimize PPA:

$$E_{\mathrm{PPA}}(\mathbf{M}, \mathbf{A}_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{r} \|\mathbf{P}_j\mathbf{D}_i - \mathbf{A}_{ij}\mathbf{M}\|_F^2\,, \tag{3}$$

where $\mathbf{P}_j$ is an orthographic projection of a 3-D rotation $\mathbf{R}(\boldsymbol{\omega})$ in a given domain $\boldsymbol{\Omega}$, defined by the rotation angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. Note that, while data and reference shapes are d-dimensional in Eq. (1) and Eq. (2), data $\mathbf{D}_i$ and reference $\mathbf{M} \in \mathbb{R}^{2\times\ell}$ shapes in Eq. (3) are fixed to be 3-D and 2-D, respectively. Hence, $\mathbf{A}_{ij} \in \mathbb{R}^{2\times 2}$ is a 2-D transformation mapping $\mathbf{M}$ to the 2-D projection of the 3-D data. ALS is a common method to minimize Eq. (2) and (3). ALS alternates between minimizing over $\mathbf{M}$ and $\mathbf{A}_{ij} \in \mathbb{R}^{2\times 2}$ with the following expressions:

$$\mathbf{A}_{ij} = \mathbf{P}_j\mathbf{D}_i\mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1} \qquad \forall i, j, \tag{4}$$

$$\mathbf{M} = (\sum_{i=1}^{n}\sum_{j=1}^{r}\mathbf{A}_{ij}^T\mathbf{A}_{ij})^{-1}(\sum_{i=1}^{n}(\sum_{j=1}^{r}\mathbf{A}_{ij}^T\mathbf{P}_j)\mathbf{D}_i). \tag{5}$$

Note that PPA and its extensions deal with missing data naturally. Since they use the whole 3-D shape of objects, the enhanced 2-D dataset resulting of projecting the data from different viewpoints can be constructed without occluded landmarks.

Continuous Procrustes Analysis (CPA): A major limitation of PPA is the difficulty to generate uniform distributions in the Special Orthogonal group $SO(3)$ [17]. Due to the topology of $SO(3)$, different angles should be sampled following different distributions, which becomes difficult when the rotation matrices must be confined in a specific region $\boldsymbol{\Omega}$ of $SO(3)$, restricted by rotation angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. Moreover, the computational complexity of PPA increases linearly with the number of projections (r) and 3-D objects (n).

In order to deal with these drawbacks, a continuous formulation (CPA) was proposed in [10] by formulating PPA within a functional analysis framework. CPA minimizes:

$$E_{\mathrm{CPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n}\int_{\boldsymbol{\Omega}} \|\mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{A}(\boldsymbol{\omega})_i\mathbf{M}\|_F^2\, d\boldsymbol{\omega}, \tag{6}$$

where $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\theta)d\phi d\theta d\psi$ ensures uniformity in $SO(3)$ [17]. This continuous formulation finds the optimal 2-D reference shape of a 3-D dataset, rotated and projected in a given domain $\boldsymbol{\Omega}$, by integrating over all possible rotations in that domain. The main difference between Eq. (3) and Eq. (6) is that the entries in $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$ and $\mathbf{A}(\boldsymbol{\omega})_i \in \mathbb{R}^{2\times 2}$ are not scalars anymore, but functions of the integration angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. After some linear algebra and functional analysis, it is possible to find an equivalent expression to the discrete approach

(Eq. (3)), where $\mathbf{A}(\boldsymbol{\omega})_i$ and $\mathbf{M}$ have the following expressions:

$$\mathbf{A}(\boldsymbol{\omega})_i = \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1} \qquad \forall i, \tag{7}$$

$$\mathbf{M} = \left(\sum_{i=1}^{n}\int_{\boldsymbol{\Omega}} \mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{A}(\boldsymbol{\omega})_i d\boldsymbol{\omega}\right)^{-1}\left(\sum_{i=1}^{n}\left(\int_{\boldsymbol{\Omega}} \mathbf{A}(\boldsymbol{\omega})_i^T\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega}\right)\mathbf{D}_i\right). \tag{8}$$

It is important to notice that the 2-D projections are not explicitly computed in the continuous formulation. The solution of $\mathbf{M}$ is found using fixed-point iteration in Eq. (6):

$$\mathbf{M} = (\mathbf{Z}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{-1})^{-1}\mathbf{Z}, \tag{9}$$

where $\mathbf{X} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T\mathbf{P}(\boldsymbol{\omega})d\boldsymbol{\omega} \in \mathbb{R}^{3\times 3}$ averages the rotation covariances and[2] $\mathbf{Z} = (\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M}\left(\sum_{i=1}^{n}(\mathbf{D}_i^T \otimes \mathbf{D}_i^T)\operatorname{vec}(\mathbf{X})\right)^{(\ell)}$. Note that the definite integral $\mathbf{X}$ is not data dependent, and it can be computed off-line.

Our work builds on [10] but extends it in several ways. First, CPA only computes the reference shape of the dataset. In this paper, we add a subspace that is able to model non-rigid deformations of the object, as well as rigid 3-D transformations that the affine transformation cannot model. As we will describe later, adding a subspace to the PA formulation is not a trivial task. For instance, modeling a subspace following the standard methodology based on CPA would still require to generate r rotations for each 3-D sample. Hence, the CPA efficiency is limited to rigid models while our approach is not. Second, we provide a discrete and continuous formulation in order to provide a better understanding of the problem, and experimentally show that it converges to the same solution when the number of sampled rotations (r) increases. Finally, we evaluate the models in two challenging problems: pose estimation in still images and joints' modeling.

3 Subspace Procrustes Analysis (SPA)

This section proposes Discrete Subspace Procrustes Analysis (DSPA) and Continuous Subspace Procrustes Analysis (CSPA) to learn unbiased 2-D models from 3-D deformable objects.

Discrete Subspace Procrustes Analysis (DSPA): Given a set of r viewpoints $\mathbf{P}_j \in \mathbb{R}^{2\times 3}$ of the n 3-D shapes, where $\mathbf{d}_i = \operatorname{vec}(\mathbf{D}_i) \in \mathbb{R}^{3\ell\times 1}$, DSPA extends PA by considering a subspace $\mathbf{B} \in \mathbb{R}^{2\ell\times k}$ and the weights $\mathbf{c}_{ij} \in \mathbb{R}^{k\times 1}$ which model the non-rigid deformations that the mean $\mathbf{M}$ and the transformation $\mathbf{A}_{ij}$ are not able to reconstruct. DSPA minimizes the following function:

$$E_{\text{DSPA}}(\mathbf{M}, \mathbf{A}_{ij}, \mathbf{B}, \mathbf{c}_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{r}\left\|\mathbf{P}_j\mathbf{D}_i - \mathbf{A}_{ij}\mathbf{M} - (\mathbf{c}_{ij}^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)}\right\|_F^2 = \tag{10}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{r}\left\|(\mathbf{I}_\ell \otimes \mathbf{P}_j)\mathbf{d}_i - (\mathbf{I}_\ell \otimes \mathbf{A}_{ij})\boldsymbol{\mu} - \mathbf{B}\mathbf{c}_{ij}\right\|_2^2, \tag{11}$$

[2] See Appendix A for an explanation of the vec-transpose operator.

where $\mathbf{P}_j$ is a particular 3-D rotation, $\mathbf{R}(\boldsymbol{\omega})$, that is projected using an orthographic projection into 2-D, $\boldsymbol{\mu} = \text{vec}(\mathbf{M}) \in \mathbb{R}^{2\ell \times 1}$ is the vectorized version of the reference shape, $\mathbf{c}_{ij}$ are the k weights of the subspace for each 2-D shape projection, and $\mathbf{B}^{(2)} \in \mathbb{R}^{2k \times \ell}$ is the reshaped subspace. Observe that the only difference with Eq. (3) is that we have added a subspace. This subspace will compensate for the non-rigid components of the 3-D object and the rigid component (3-D rotation and projection to the image plane) that the affine transformation cannot model. Recall that a 3-D rigid object under orthographic projection can be recovered with a three-dimensional subspace (if the mean is removed), but PA cannot recover it because it is only rank two. Also, observe that the coefficient $\mathbf{c}_{ij}$ depends on two indexes, i for the object and j for the geometric projection. Dependency of $\mathbf{c}_{ij}$ on the geometric projection is a key point. If j index is not considered, the subspace would not be able to capture the variations in pose and its usefulness for our purposes would be unclear. Although Eq. (10) and the NRSFM problem follow similar formulation [3], the assumptions are different and variables have opposite meanings. For instance, the NRSFM assumptions about rigid transformations do not apply here, since $\mathbf{A}_{ij}$ are affine transformations in our case.

Given an initialization of $\mathbf{B} = 0$, DSPA is minimized by finding the transformations $\mathbf{A}^*_{ij}$ and reference shape $\mathbf{M}^*$ that minimize Eq. (3), using the same ALS framework as in PA. Then, we substitute $\mathbf{A}^*_{ij}$ and $\mathbf{M}^*$ in Eq. (11) that results in the expression:

$$E_{\text{DSPA}}(\mathbf{B}, \mathbf{c}_{ij}) = \sum_{i=1}^{n} \sum_{j=1}^{r} \left\| \widetilde{\mathbf{D}}_{ij} - (\mathbf{c}_{ij}^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 = \tag{12}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{r} \left\| \widetilde{\mathbf{d}}_{ij} - \mathbf{B}\mathbf{c}_{ij} \right\|_2^2 = \left\| \widetilde{\mathbf{D}} - \mathbf{B}\mathbf{C} \right\|_F^2, \tag{13}$$

where $\widetilde{\mathbf{D}}_{ij} = \mathbf{P}_j \mathbf{D}_i - \mathbf{A}^*_{ij}\mathbf{M}^* \in \mathbb{R}^{2 \times \ell}$, $\widetilde{\mathbf{d}}_{ij} = \text{vec}(\widetilde{\mathbf{D}}_{ij}) \in \mathbb{R}^{2\ell \times 1}$, $\widetilde{\mathbf{D}} = [\widetilde{\mathbf{d}}_1 \ldots \widetilde{\mathbf{d}}_{nr}] \in \mathbb{R}^{2\ell \times nr}$, and $\mathbf{C} \in \mathbb{R}^{k \times nr}$. We can find the global optima of Eq. (13) by Singular Value Decomposition (SVD): $\mathbf{B} = \mathbf{U}$ and $\mathbf{C} = \mathbf{S}\mathbf{V}^T$, where $\widetilde{\mathbf{D}} = \mathbf{U}\mathbf{S}\mathbf{V}^T$.

Continuous Subspace Procrustes Analysis (CSPA): As it was discussed in the previous section, the discrete formulation is not efficient in space nor time, and might suffer from not uniform sampling of the original space. CSPA generalizes DSPA by re-writting it in a continuous formulation. CSPA minimizes the following functional:

$$E_{\text{CSPA}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i, \mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) =$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{A}(\boldsymbol{\omega})_i \mathbf{M} - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} = \tag{14}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \|(\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\mathbf{d}_i - (\mathbf{I}_\ell \otimes \mathbf{A}(\boldsymbol{\omega})_i)\boldsymbol{\mu} - \mathbf{B}\mathbf{c}(\boldsymbol{\omega})_i\|_2^2 \, d\boldsymbol{\omega}, \tag{15}$$

where $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\theta)d\phi d\theta d\psi$. The main difference between Eq. (15) and Eq. (11) is that the entries in $\mathbf{c}(\boldsymbol{\omega})_i \in \mathbb{R}^{k\times 1}$, $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$ and $\mathbf{A}(\boldsymbol{\omega})_i \in \mathbb{R}^{2\times 2}$ are not scalars anymore, but functions of integration angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$.

Given an initialization of $\mathbf{B} = 0$, and similarly to the DSPA model, CSPA is minimized by finding the optimal reference shape $\mathbf{M}^*$ that minimizes Eq. (6). We used the same fixed-point framework as CPA. Given the value of $\mathbf{M}^*$ and the expression of $\mathbf{A}(\boldsymbol{\omega})_i^*$ from Eq. (7), we substitute them in Eq. (15) resulting in:

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\bar{\mathbf{D}}_i - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} = \tag{16}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\bar{\mathbf{d}}_i - \mathbf{B}\mathbf{c}(\boldsymbol{\omega})_i \right\|_2^2 d\boldsymbol{\omega}, \tag{17}$$

where $\bar{\mathbf{D}}_i = \mathbf{D}_i(\mathbf{I}_\ell - (\mathbf{M}^{*T}(\mathbf{M}^*\mathbf{M}^{*T})^{-1}\mathbf{M}^*))$ and $\bar{\mathbf{d}}_i = \text{vec}(\bar{\mathbf{D}}_i)$. We can find the global optima of Eq. (17) by solving the eigenvalue problem, $\boldsymbol{\Sigma}\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ are the eigenvalues corresponding to columns of $\mathbf{B}$.

After some algebra (see Appendix B) we show that the covariance matrix $\boldsymbol{\Sigma} = ((\mathbf{I}_\ell \otimes \mathbf{Y})\,\text{vec}(\sum_{i=1}^{n}\sum_{j=1}^{r} \bar{\mathbf{d}}_{ij}\bar{\mathbf{d}}_{ij}^T))^{(2\ell)}$ where the definite integral $\mathbf{Y} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega}) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))d\boldsymbol{\omega} \in \mathbb{R}^{2\ell\times 2\ell}$ can be computed off-line, leading to an efficient optimization in space and time. Though the number of elements in matrix $\mathbf{Y}$ increase quadratically with the number of landmarks ℓ, note that the integration time is constant since $\mathbf{Y}$ has a sparse structure with only 36 different non-zero values (recall that $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$).

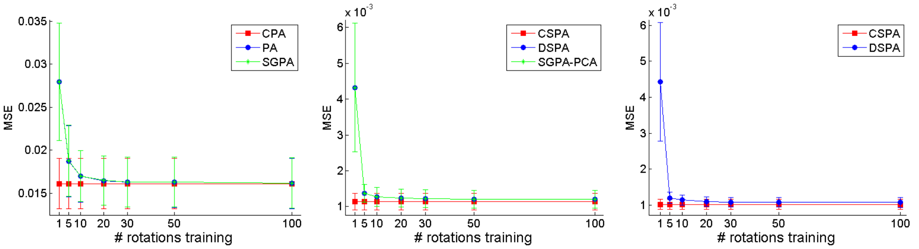
Although $\mathbf{A}(\boldsymbol{\omega})_i$ and $\mathbf{c}(\boldsymbol{\omega})_i$ are not explicitly computed during training, this is not a limitation compared to DSPA. During testing time, training values of $\mathbf{c}(\boldsymbol{\omega})_i$ are not needed. Only the deformation limits in each principal direction of $\mathbf{B}$ are required. These limits also depend on eigenvalues [4], which are computed with CSPA.

4 Experiments & Results

This section illustrates the benefits of DSPA and CSPA, and compares them with state-of-the-art PA methods to build 2-D shape models of human skeletons. First, we compare the performance of PA+PCA and SPA to build a 2-D shape model of Motion Capture (MoCap) bodies using the Carnegie Mellon University MoCap dataset [1]. Next, we compare our discrete and continuous approaches in a large scale experiment. Finally, we illustrate the generalization of our 2-D body model in the problem of human pose estimation using the Leeds Sport Dataset [11]. For all experiments, we report the Mean Squared Error (MSE) relative to the torso size.

4.1 Learning 2-D Joints Models

The aim of this experiment is to build a generic 2-D body model that can reconstruct non-rigid deformations under a large range of 3-D rotations. For



(a) (b) (c)

Fig. 3. Comparisons as a function of the number of training viewpoint projections. (*a*) Rigid and (*b*) Deformable models (using a subspace of 9 basis) from Experiment 1, respectively; (*c*) CSPA and DSPA deformable models (using a subspace of 12 basis) from Experiment 2.

training and testing, we used the Carnegie Mellon University MoCap dataset that is composed of 2605 sequences performed by 109 subjects. The sequences cover a wide variety of daily human activities and sports. Skeletons with 31 joints are provided, as well as RGB video recordings for several sequences. We trained our models using the set of 14 landmarks as is common across several databases for human pose estimation.

Experiment 1: Comparison with State-of-the-Art PA Methods This section compares DSPA, CSPA methods with the state-of-the-art Stratified Generalized Procrustes Analysis (SGPA) [2][3]. For training we randomly selected 3 sequences with 30 frames per sequence from the set of 11 running sequences of the user number 9 (this is due to the memory limitations of SGPA). For testing we randomly selected 2 sequences with 30 frames from the same set. We rotated the 3-D models in the yaw and pitch angles, within the ranges of $\phi, \theta \in [-\pi/2, \pi/2]$. The angles were uniformly selected and we report results varying the number of considered angles (i.e., rotations) between 1 $\sim$ 100 angles in training, and fixed 300 angles for testing.

There are several versions of SGPA. We selected the "Affine-factorization" with the data-space model to make a fair comparison with our method. Recall that under our assumption of non-missing data "Affine-All" and "Affine-factorization" achieve the same solution, with "Affine-factorization" being faster.

Fig. 3 shows the mean reconstruction error and 0.5 of the standard deviation for 100 realizations. Fig. 3 (a) reports the results comparing PA, CPA and SGPA. As expected, PA and SGPA converge to CPA as the number of training rotations increased. However, observe that CPA achieves the same performance, but it is much more efficient. Fig. 3 (b) compares DSPA, CSPA, and SGPA followed by PCA (we will refer to this method SGPA+PCA). From the figure we can

[3] The code was downloaded from author's website (http://isit.u-clermont1.fr/~ab).

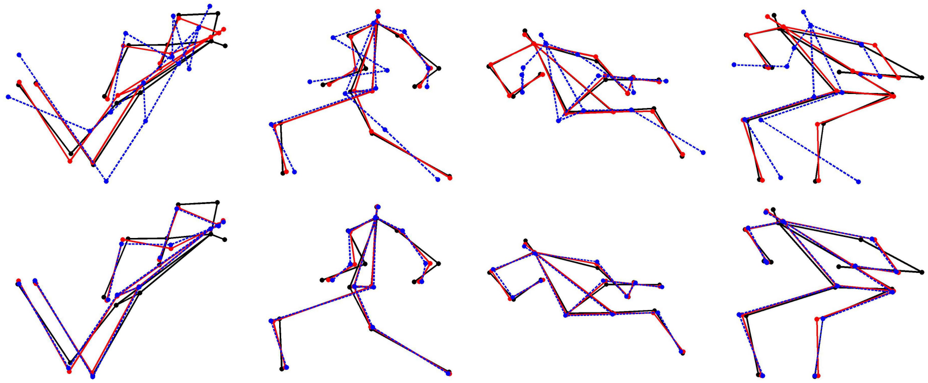


Fig. 4. Experiment 2 results with 1 (*top*), and 30 (*bottom*) rotations. Examples show skeleton reconstructions from continuous (*CSPA* in *solid red lines*) and discrete (*SPA* in *dashed blue lines*) models over ground truth (*solid black lines*).

observe that the mean error in the test for DSPA and SGPA+PCA decrease with the number of rotations in the training, and it converges to CSPA. CSPA provides a bound on the lower error. Observe, that we used 90 3-D bodies (3 sequences with 30 frames) within rotating angles $\phi, \theta \in [-\pi/2, \pi/2]$, and DSPA and SGPA+PCA needed about 30 angles to achieve similar result to CSPA. So, in this case, discrete methods need 30 times more space than the continuous one. The execution times with 30 rotations, on a 2.2GHz computer with 8Gb of RAM, were 1.44 sec. (DSPA), 0.03 sec. (CSPA) and 3.54 sec. (SGPA+PCA).

Experiment 2: Comparison between CSPA and DSPA This experiment compares DSPA and CSPA in a large-scale problem as a function of the number of rotations. For training we randomly selected 20 sequences with 30 frames per sequence. For testing we randomly selected 5 sequences with 30 frames. We rotated the 3-D models in the yaw and pitch angles, within the ranges of $\phi, \theta \in [-\pi/2, \pi/2]$. The angles were uniformly selected and we report results varying the number of angles (i.e., rotations) between $1 \sim 100$ angles in training, and 300 angles for testing.

Fig. 3 (c) shows the mean reconstruction error and 0.5 of the standard deviation for 100 realizations, comparing DSPA and CSPA. As expected, DSPA converges to CSPA as the number of training rotations increases. However, observe that CSPA achieves the same performance, but it is much more efficient. In this experiment, with 6000 3-D training bodies (20 sequences with 30 frames) and domain: $\phi, \theta \in [-\pi/2, \pi/2]$ discrete method required, again, around 30 2-D viewpoint projections to achieve similar results to CSPA. Thus, discrete model DSPA needs 30 times more storage space than CSPA. The execution times with 30 rotations, on a 2.2GHz computer with 8Gb of RAM, were 14.75 sec. (DSPA) and 0.04 sec. (CSPA).

Qualitative results from CSPA and DSPA models trained with different number of rotations are shown in Fig. 4. Note that training DSPA model with 1 rotation (*top*) results in poor reconstruction. However, training it with 30 rotations (*bottom*) leads to reconstructions almost as accurate as made by CSPA.

4.2 Experiment 3. Leeds Sport Dataset

This section illustrates how to use the 2-D body models learned with CSPA to detect body configurations from images. We used the Leeds Sport Dataset (LSP) that contains 2000 images of people performing different sports, some of them including extreme poses or viewpoints (e.g., parkour images). The first 1000 images of the dataset are considered for training and the second set of 1000 images for testing. One skeleton manually labeled with 14 joints is provided for each training and test image.

We trained our 2-D CSPA model in the CMU MoCap dataset [1] using 1000 frames. From the 2605 sequences of the motion capture data, we randomly selected 1000 and the frame in the middle of sequence is selected as representative frame. Using this training data, we built the 2-D CSPA model using the following ranges for the pitch, roll and yaw angles: $\phi, \theta, \psi \in [-3/4\pi, 3/4\pi]$. We will refer to this model as CSPA-MoCap. For comparison, we used the 1000 2-D training skeletons provided by the LSP dataset and run SGPA+PCA to build an alternative 2-D model. We will refer to this model as SGPA+PCA-LSP. Observe, that this model was trained on similar data as the test set.

Table 1. Experiment 3 results. MSE of our continuous model (*CSPA-MoCap*) trained with 3-D MoCap data, the discrete model trained in the LSP dataset (*SGPA+PCA-LSP*), and both rigid models (*CPA-MoCap*, *SGPA-LSP*).

Model	CPA-MoCap	SGPA-LSP	CSPA-MoCap	SGPA+PCA-LSP
MSE	0.16405	0.16231	**0.01046**	0.01366

Table 1 reports MSE of reconstructing the test skeletons with rigid and deformable models. A subspace of 12 basis is used for both deformable models. Results show that CSPA-MoCap has less reconstruction error than the standard method SGPA+PCA-LSP, even trained in a different dataset (CMU MoCap) than the test. Qualitative results from CSPA-MoCap and SGPA+PCA-LSP models are shown in Fig. 5. Note that CSPA-MoCap provides more accurate reconstructions than SGPA+PCA-LSP because it is able to generalize to all possible 3-D rotations in the given interval.

5 Conclusions

This paper proposes an extension of PA to learn a 2-D subspace of rigid and non-rigid deformations of 3-D objects. We propose two models, one discrete (DSPA)

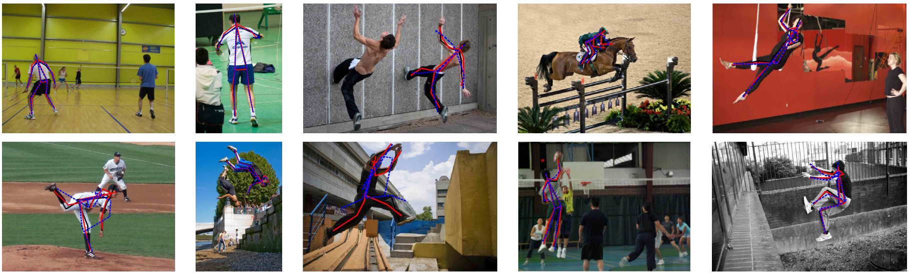


Fig. 5. Experiment 3 examples, reconstructing ground truth skeletons of LSP dataset with *CSPA-MoCap* (*solid red lines*) and *SGPA+PCA-LSP* (*dashed blue lines*) models.

that samples the 3-D rotation space, and one continuous (CSPA) that integrates over $SO(3)$. As the number of projections increases DPSA converges to CSPA. SPA has two advantages over traditional PA, PPA: (1) it generates unbiased models because it uniformly covers the space of projections, and (2) CSPA is much more efficient in space and time. Experiments comparing 2-D SPA models of bodies show improvements w.r.t. state-of-the-art PA methods. In particular, CSPA models trained with motion capture data outperformed 2-D models trained on the same database under the same conditions in the LSP database, showing how our 2-D models from 3-D data can generalize better to different viewpoints. In future work, we plan to explore other models that decouple the rigid and non-rigid deformation by providing two independent basis in the subspace.

6 Acknowledgments

This work is partly supported by the Spanish Ministry of Science and Innovation (projects TIN2012-38416-C03-01, TIN2012-38187-C03-01, TIN2013-43478-P), project 2014 SGR 1219, and the Comissionat per a Universitats i Recerca del Departament d'Innovació, Universitats i Empresa de la Generalitat de Catalunya.

A Appendix. Vec-transpose

Vec-transpose $\mathbf{A}^{(p)}$ is a linear operator that generalizes vectorization and transposition operators [14, 16]. It reshapes matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ by vectorizing each i^{th} block of p rows, and rearranging it as the i^{th} column of the reshaped matrix, such that $\mathbf{A}^{(p)} \in \mathbb{R}^{pn \times \frac{m}{p}}$,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \\ a_{61} & a_{62} & a_{63} \end{bmatrix}^{(3)} = \begin{bmatrix} a_{11} & a_{41} \\ a_{21} & a_{51} \\ a_{31} & a_{61} \\ a_{12} & a_{42} \\ a_{22} & a_{52} \\ a_{32} & a_{62} \\ a_{13} & a_{43} \\ a_{23} & a_{53} \\ a_{33} & a_{63} \end{bmatrix},$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \\ a_{51} & a_{52} & a_{53} \\ a_{61} & a_{62} & a_{63} \end{bmatrix}^{(2)} = \begin{bmatrix} a_{11} & a_{31} & a_{51} \\ a_{21} & a_{41} & a_{61} \\ a_{12} & a_{32} & a_{52} \\ a_{22} & a_{42} & a_{62} \\ a_{13} & a_{33} & a_{53} \\ a_{23} & a_{43} & a_{63} \end{bmatrix}.$$

Note that $(\mathbf{A}^{(p)})^{(p)} = \mathbf{A}$ and $\mathbf{A}^{(m)} = \text{vec}(\mathbf{A})$. A useful rule for pulling a matrix out of nested Kronecker products is, $((\mathbf{BA})^{(p)}\mathbf{C})^{(p)} = (\mathbf{C}^T \otimes \mathbf{I}_p)\mathbf{BA} = (\mathbf{B}^{(p)}\mathbf{C})^{(p)}\mathbf{A}$, which leads to $(\mathbf{C}^T \otimes \mathbf{I}_2)\mathbf{B} = (\mathbf{B}^{(2)}\mathbf{C})^{(2)}$.

B Appendix. CSPA formulation

In this Appendix, we detail the steps from Eq. (14) to Eq. (17), as well as the definition of the covariance matrix, introduced in Section 3.

Given the value of $\mathbf{M}^*$ and the optimal expression of $\mathbf{A}(\boldsymbol{\omega})_i^*$ from Eq. (7), we substitute them in Eq. (14) resulting in:

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{H} - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega}, \quad (18)$$

where $\mathbf{H} = \mathbf{M}^{*T}(\mathbf{M}^*\mathbf{M}^{*T})^{-1}\mathbf{M}^*$ and $\mathbf{D}_i \in \mathbb{R}^{3 \times \ell}$. Then,

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i(\mathbf{I}_\ell - \mathbf{H}) - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} \quad (19)$$

leads us to Eq. (16) and Eq. (17), where $\bar{\mathbf{D}}_i = \mathbf{D}_i(\mathbf{I}_\ell - \mathbf{H})$ and $\bar{\mathbf{d}}_i = \text{vec}(\bar{\mathbf{D}}_i)$. From Eq. (17), solving $\frac{\partial E_{\text{CSPA}}}{\partial \mathbf{c}(\boldsymbol{\omega})_i} = 0$ we find:

$$\mathbf{c}(\boldsymbol{\omega})_i^* = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i. \quad (20)$$

The substitution of $\mathbf{c}(\boldsymbol{\omega})_i^*$ in Eq. (17) results in:

$$E_{\text{CSPA}}(\mathbf{B}) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\bar{\mathbf{d}}_i - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \right\|_2^2 d\boldsymbol{\omega} = \tag{21}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \right\|_2^2 d\boldsymbol{\omega} = \tag{22}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \operatorname{tr}\left[\left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \left((\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i\right)^T \right] d\boldsymbol{\omega} = \tag{23}$$

$$\operatorname{tr}\left[\left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right) \boldsymbol{\Sigma} \right], \tag{24}$$

where:

$$\boldsymbol{\Sigma} = \int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \left(\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T \right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))^T d\boldsymbol{\omega}. \tag{25}$$

We can find the global optima of Eq. (24) by solving the eigenvalue problem, $\boldsymbol{\Sigma}\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$, where $\boldsymbol{\Sigma}$ is the covariance matrix and $\boldsymbol{\Lambda}$ are the eigenvalues corresponding to columns of $\mathbf{B}$. However, the definite integral in $\boldsymbol{\Sigma}$ is data dependent. To be able to compute the integral off-line, we need to rearrange the elements in $\boldsymbol{\Sigma}$. Using vectorization and vec-transpose operator[4]:

$$\boldsymbol{\Sigma} = (\operatorname{vec}[\boldsymbol{\Sigma}])^{(2\ell)} = \tag{26}$$

$$\left(\operatorname{vec}\left[\int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \left(\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T \right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))^T d\boldsymbol{\omega} \right] \right)^{(2\ell)} = \tag{27}$$

$$\left(\left(\int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) d\boldsymbol{\omega} \right) \operatorname{vec}\left[\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T \right] \right)^{(2\ell)}, \tag{28}$$

which finally leads to:

$$\boldsymbol{\Sigma} = \left((\mathbf{I}_\ell \otimes \mathbf{Y}) \operatorname{vec}\left[\sum_{i=1}^{n} \bar{\mathbf{d}}_{ij} \bar{\mathbf{d}}_{ij}^T \right] \right)^{(2\ell)}, \tag{29}$$

where the definite integral $\mathbf{Y} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega}) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega})) d\boldsymbol{\omega} \in \mathbb{R}^{4\ell \times 9\ell}$ can be computed off-line.

References

1. Carnegie mellon motion capture database. http://mocap.cs.cmu.edu
2. Bartoli, A., Pizarro, D., Loog, M.: Stratified generalized procrustes analysis. IJCV 101(2), 227–253 (2013)
3. Brand, M.: Morphable 3d models from video. In: CVPR. vol. 2, pp. II–456. IEEE (2001)
4. Cootes, T.F., Edwards, G.J., Taylor, C.J., et al.: Active appearance models. PAMI 23(6), 681–685 (2001)

[4] See Appendix A for the vec-transpose operator.

5. De la Torre, F.: A least-squares framework for component analysis. PAMI 34(6), 1041–1055 (2012)
6. Dryden, I.L., Mardia, K.V.: Statistical shape analysis, vol. 4. John Wiley & Sons New York (1998)
7. Frey, B.J., Jojic, N.: Transformation-invariant clustering using the em algorithm. PAMI 25(1), 1–17 (2003)
8. Goodall, C.: Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society. Series B (Methodological) pp. 285–339 (1991)
9. Gower, J.C., Dijksterhuis, G.B.: Procrustes problems, vol. 3. Oxford University Press Oxford (2004)
10. Igual, L., Perez-Sala, X., Escalera, S., Angulo, C., De la Torre, F.: Continuous generalized procrustes analysis. PR 47(2), 659–671 (2014)
11. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: Proceedings of the British Machine Vision Conference (2010), doi:10.5244/C.24.12
12. Kokkinos, I., Yuille, A.: Unsupervised learning of object deformation models. In: ICCV. pp. 1–8. IEEE (2007)
13. Learned-Miller, E.G.: Data driven image models through continuous joint alignment. PAMI 28(2), 236–250 (2006)
14. Marimont, D.H., Wandell, B.A.: Linear models of surface and illuminant spectra. JOSA A 9(11), 1905–1913 (1992)
15. Matthews, I., Xiao, J., Baker, S.: 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. IJCV 75(1), 93–113 (2007)
16. Minka, T.P.: Old and new matrix algebra useful for statistics. http://research.microsoft.com/en-us/um/people/minka/papers/matrix/, 2000
17. Naimark, M.A.: Linear representatives of the Lorentz group (translated from Russian). New York, Macmillan (1964)
18. Pearson, K.: On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2(11), 559–572 (1901)
19. Pizarro, D., Bartoli, A.: Global optimization for optimal generalized procrustes analysis. In: CVPR. pp. 2409–2415. IEEE (2011)
20. De la Torre, F., Black, M.J.: Robust parameterized component analysis: theory and applications to 2d facial appearance models. CVIU 91(1), 53–71 (2003)
21. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. PAMI 30(5), 878–892 (2008)
22. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. IJCV 67(2), 233–246 (2006)
23. Yezzi, A.J., Soatto, S.: Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. IJCV 53(2), 153–167 (2003)