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. Later, a non-rigid 2-D model is computed by modeling the residual (e.g., PCA). 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, independent PA and PCA steps might result in a sub-optimal model, 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. We illustrate the benefits of SPA in two experiments. First, SPA is used to learn 2-D face and body models from 3-D datasets. Experiments on the FaceWarehouse and CMU motion capture (MoCap) datasets show the benefits of our 2-D models against the state-of-the-art PA approaches and conventional 3-D models. Second, SPA learns an unbiased 2-D model from CMU MoCap dataset and it is used to estimate the human pose on the Leeds Sports dataset. Our feature selection by subspace matching formulation show the benefits of our models over state-of-the-art approaches in human pose estimation.

1 Introduction

In computer vision, Procrustes Analysis (PA) has been extensively used to align shapes (e.g., [30, 6]) and appearance (e.g., [32, 18]) 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 [11]. Firstly, the rigid transformations are removed by aligning the training set w.r.t. the mean using

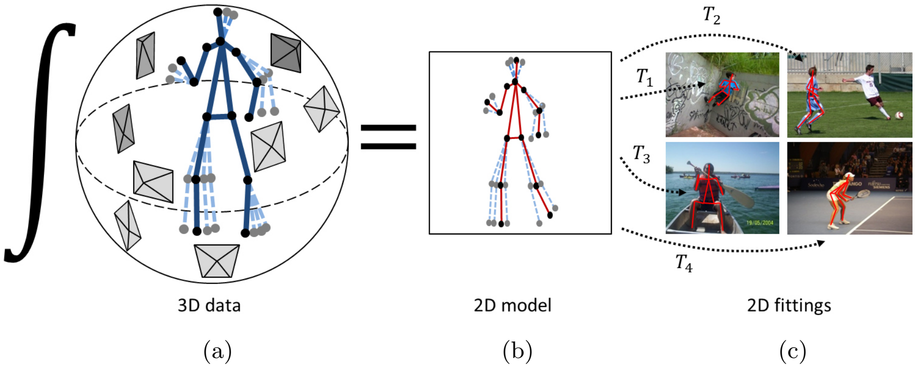


Fig. 1. Illustration of Continuous Subspace Procrustes Analysis (CSPA). CSPA builds an unbiased 2-D model of human joints' variation (b) by integrating over all possible viewpoints of a 3-D motion capture data (a). This 2-D body shape model is used to detect body joints from different viewpoints (c).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

PA; next, the remaining deformations are modeled using Principal Component Analysis (PCA) [26, 8].

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 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) [34, 33, 4]. 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. As we show in the experimental section, the learned 2-D model has the same representational power of a 3-D model but leads to faster fitting algorithms [22]. 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 (MoCap) 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 MoCap 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 human pose estimation (i.e., constrain body joints, see Fig. 1 (c)). Observe that the SPA subspace model is able to reconstruct 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.

In order to estimate the human pose in images, state-of-the-art approaches [25, 36, 28, 29] use discriminative detectors to estimate the likelihood of image pixels to belong to each body part. Then, body configurations are constrained by generative models [2, 25, 36], also trained from labeled images. Although successful, these 2-D models typically require a large amount of training data across views to achieve view-invariance. In a preliminary version of this work [27], we showed that unbiased 2-D models learned from 3-D data outperform those trained from 2-D data, also on human pose estimation datasets. In order to reconstruct body configurations from different viewpoints, this paper reformulates the human pose estimation problem as a subspace matching [31, 21] between image pixels and 2-D deformable models trained on 3-D MoCap data. As we show in the experimental section, our method outperforms state-of-the-art approaches on Leeds Sports dataset [16] (LSP) because it is able to handle large viewpoint variations. In addition, our method is robust to occlusions and outliers, and we efficiently solved the subspace matching problem with linear programming.

The rest of the document is organized as follows, Section 2 reviews previous work in PA and motivates SPA, detailed in Section 3. In Section 4 we describe our feature selection method to use SPA models for human pose estimation. Section 5 reports our experimental results and, finally, Section 6 presents the conclusions and outlines our future work. Additionally, we review the vec-transpose operator in Appendix A, we discuss how to build 2-D models directly from a 3-D model in Appendix B and we provide additional details about derivation steps of CSPA in Appendix C.

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 notation[6]) composed by ℓ landmarks $\mathbf{D}_i \in \mathbb{R}^{d\times\ell}, \forall i = 1,\ldots,m$, PA [9, 12, 11, 14,

[6] 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

3] 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* [14, 11, 3] (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 ($d = 2$), $\mathbf{D}_i = \begin{bmatrix} x_1 \; x_2 \dots x_\ell \\ y_1 \; y_2 \dots y_\ell \end{bmatrix}$. Alternatively, PA can be optimized using the *data-space model* [3] (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 [37]. 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 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) [3]. Benefits of both models are combined in [3]. Recently, Pizarro et al. [30] 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

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.

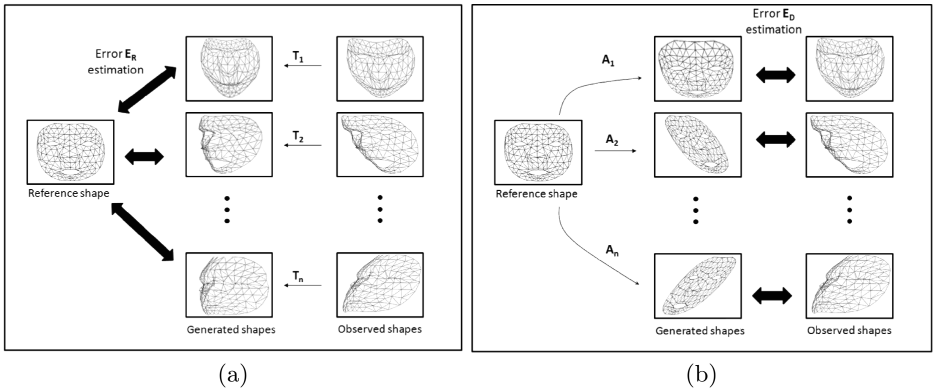


Fig. 2. (*a*): Reference-space model. (*b*): Data-space model. Note that $\mathbf{A}_i = \mathbf{T}_i^{-1}$.

the main difference when applying PA to align appearance and shape. Frey and Jojic [10] 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 [32] proposed parameterized component analysis: a method that learns a subspace of appearance invariant to affine transformations. Miller et al. proposed the congealing method [18], which uses an entropy measure to align images with respect to the distribution of the data. Kookinos and Yuille [17] 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:

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

where $\mathbf{P}_j = \mathbf{PR}(\boldsymbol{\omega}_j)$ is an orthographic projection of a 3-D rotation $\mathbf{R}(\boldsymbol{\omega}_j)$ in a given domain $\boldsymbol{\Omega}$, defined by the rotation angles $\boldsymbol{\omega}_j = \{\phi, \theta, \psi\}$. 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 $SO(3)$ [24]. 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 [14] by formulating PPA within a functional analysis framework. CPA minimizes:

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

where $d\boldsymbol{\omega} = \frac{1}{8\pi^2} \sin(\theta) d\phi d\theta d\psi$ ensures a uniform sampling of the $SO(3)$ space [24] for the rotated 3-D object. 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 $\mathbf{M}$ can be found in closed form:

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

where $\mathbf{Z} = (\mathbf{M}\mathbf{M}^T)^{-1} \mathbf{M} \left(\sum_{i=1}^{n} (\mathbf{D}_i^T \otimes \mathbf{D}_i^T) \operatorname{vec}(\mathbf{X}) \right)^{(\ell)}$, and the definite integral[7] $\mathbf{X} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T \mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \in \mathbb{R}^{3\times 3}$ averages the rotation covariances. Note that $\mathbf{X}$ is not data dependent, and it can be computed off-line.

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

Our work extends [14] 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: human pose estimation in still images, as well as faces 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 n 3-D shapes represented by ℓ landmarks, where $\mathbf{d}_i = \mathrm{vec}(\mathbf{D}_i) \in \mathbb{R}^{3\ell\times 1}$, 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:

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

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

where $\mathbf{P}_j = \mathbf{P}\mathbf{R}(\boldsymbol{\omega}_j)$ is a particular 3-D rotation, $\mathbf{R}(\boldsymbol{\omega}_j)$, that is projected using an orthographic projection into 2-D, $\boldsymbol{\mu} = \mathrm{vec}(\mathbf{M}) \in \mathbb{R}^{2\ell\times 1}$ 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 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 (see Fig. 3 (a), where the first three basis of the subspace capture non-rigid and rigid deformations). 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

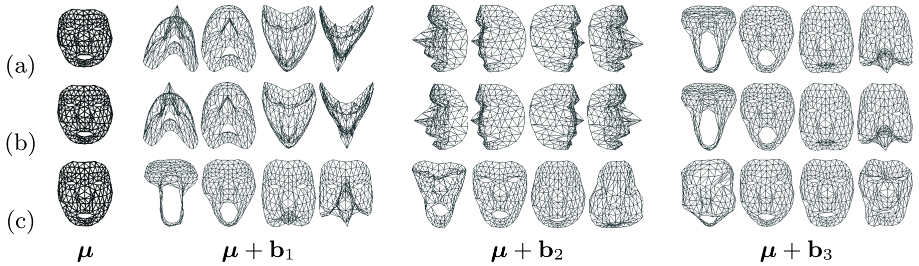

(a)
(b)
(c)
$\boldsymbol{\mu}$
$\boldsymbol{\mu}+\mathbf{b}_1$
$\boldsymbol{\mu}+\mathbf{b}_2$
$\boldsymbol{\mu}+\mathbf{b}_3$

Fig. 3. Illustration of the reference shape ($\boldsymbol{\mu}$) and the first three basis ($\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$) of the 2-D subspace models from (a) *DSPA*, and (b) *CSPA*; as well as a conventional (c) 3-D model (PA + PCA). We sampled each basis 4 times between the standard limits [7] to show their deformation behavior. All models were trained on the FaceWarehouse [5] dataset, with 10 3-D faces from expressions number 0 and 1 (neutral and open mouth, respectively). Pitch an yaw integration limits were set to $\phi, \theta \in [-\pi/2, \pi/2]$ for (b), and 100 projections were generated for each 3-D shape within the same interval to train (a). Note that $\boldsymbol{\mu}$ and $\mathbf{b}_i$ in (c) are 3-D. They are projected frontally for a better comparison.

point. If the jth 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 [4], 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 the 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 $\mathbf{C} \in \mathbb{R}^{k\times nr}$. Finally, 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 introduction, the discrete formulation of PA is not efficient in space nor time, and might suffer from not uniform sampling of the original space. CSPA generalizes DSPA by extending it with a functional 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 C) 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 the eigenvalues [7], which are computed with CSPA. The three principal basis between these limits are illustrated in Fig. 3. We show how the first 2 basis of CSPA (Fig. 3 (b)) and DSPA (Fig. 3 (a)) learn viewpoint changes, as well as the common expression for all the subjects in the training set (mouth opening) is learned as the third basis. Note that the 3-D (PA+PCA) model (Fig. 3 (c)) learns the common facial expression in the first basis (because the 3-D shapes are not rotated to train the 3-D model), and its following basis model inter-person differences. These distinctive person characteristics are also learned by SPA models in their following basis.

4 Subspace Feature Selection for Human Pose Estimation

This section describes how CSPA can be applied to estimate the human pose in images, given the unbiased 2-D model computed in Section 3. Human pose estimation refers to the problem of finding body configuration of humans in images [25, 36, 28, 29]. When body configurations are modeled by means of a 2-D subspace model, we can formulate the human pose estimation challenge as a subspace feature selection [31, 21], between a 2-D deformable model of joints' variation and a pool of features for each body joint. These features or pixel candidates are the result of running state-of-the-art body part detectors.

The goal of subspace feature selection is to choose the subset of ℓ landmarks from n_f candidate image features or landmarks that minimize the distance to a subspace model. It was first introduced in [31] for establishing correspondences between a sparse set of d-dimensional image features $\mathbf{Q} \in \mathbb{R}^{d \times n_f}$ and a previously learned model of frontal faces. Given the candidate features and a model composed of a reference shape $\mathbf{M} \in \mathbb{R}^{d \times \ell}$ and k basis $\mathbf{B} \in \mathbb{R}^{d\ell \times k}$, the problem consists on finding the optimal correspondence $\mathbf{S}$ and the subspace coefficients $\mathbf{c} \in \mathbb{R}^{k \times 1}$ which minimize the following error:

$$\min_{\mathbf{S},\mathbf{c}} \left\| \mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - \boldsymbol{\mu} - \mathbf{B}\mathbf{c} \right\|_2^2, \qquad (18)$$
$$\text{s.t.} \quad \mathbf{S} \in \{0,1\}^{\ell \times n_f}, \mathbf{S}\mathbf{1}_{n_f} = \mathbf{1}_\ell,$$

where $\boldsymbol{\mu} = \mathrm{vec}(\mathbf{M}) \in \mathbb{R}^{d\ell \times 1}$ is the vectorization of the mean and the constraint enforces to select only one candidate for each landmark. To reduce the number of parameters, $\mathbf{c}$ is replaced by its optimal value $\mathbf{c} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - \boldsymbol{\mu})$ and the solution of $\mathbf{S}$ is found by means of Quadratic Programming (QP). Although novel, this formulation has three main drawbacks: (1) QP is computationally expensive and the solution is found by combining the error of two QP problems, one for the shape (location of the pixels in the image, $d = 2$), and another one for the appearance (SIFT description of the image at those locations, $d = 128$); (2) only frontal objects (faces) are modeled; and (3) deformation parameters $\mathbf{c}$ are not restricted to be plausible values [7].

Feature selection has also been studied in the topic of graph matching. In [19], they introduced a matching method based on a locally affine-invariant geometric constraint and Linear Programming (LP) techniques. This work was extended in [38], making the method more robust to non-rigid facial poses contained in the training set, and adding additional constraints to reduce the search space.

In this work, we build on [31] but solving the above mentioned drawbacks: (1) we reformulated the joint shape and appearance minimization as a single LP problem [19] instead of two QP problems, making feasible to handle the large number of candidate features of human pose estimation problems ($n_f \geq 2 \cdot 10^4$); (2) we added an affinity transformation $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ to model non-frontal objects; and (3) we introduced constraints on the subspace parameters to guide the optimization to plausible values of deformation.

Moreover, we borrowed landmark-candidate association formulation and constraints (see Fig. 4) from the graph matching literature [38]. In the rest of the

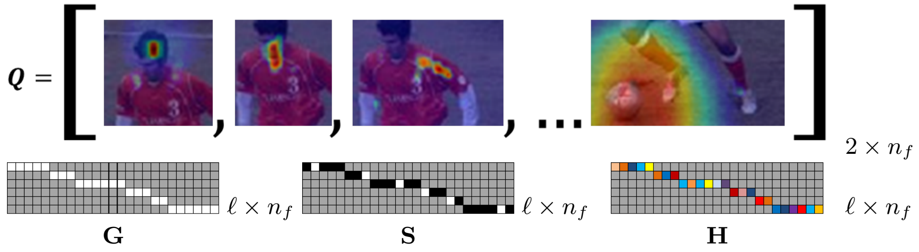


Fig. 4. Illustration of the candidate features matrix $\mathbf{Q}$, as the concatenation of the detector responses for each body joint. More specifically, $\mathbf{Q}$ concatenates those pixel locations $\mathbf{Q}^t$ with high detection score after applying each t^{th} joint's filter. Association matrix $\mathbf{G}$ is illustrated by a sparse matrix, only having ones in those positions of each t^{th} row that correspond with $\mathbf{Q}^t$ candidates. Similarly, $\mathbf{H}$ provides an association cost for each of those selections, obtained from the detection score. $\mathbf{S}$ shows an example of feature selection matrix, satisfying $\mathbf{G}$ restrictions and $\mathbf{H}$ cost.

paper, $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^\ell] \in \mathbb{R}^{2 \times n_f}$ denotes the set of 2-D candidate image pixels, where $\mathbf{Q}^t \in \mathbb{R}^{2 \times n_t}$ is the subset of candidates of the t^{th} landmark and $n_f = \sum_{t=1}^{\ell} = n_t$. Each set t of candidates results from applying the state-of-the-art body part detector [36] for the corresponding joint. Each of the n_f candidates is known to be associated with one of the ℓ landmarks and have an assignation cost, depending on the detector response. The landmark-candidate relation is encoded in the binary matrix $\mathbf{G} \in \{0,1\}^{\ell \times n_f}$, where $g_{ti} = 1$ if the i^{th} candidate belongs to the t^{th} landmark. In the same way, the assignation cost h_{ti} of choosing the i^{th} candidate as the t^{th} landmark is computed as the SVM score by an efficient two-pass dynamic programming inference [25] and encoded in the matrix $\mathbf{H} \in \mathbb{R}^{\ell \times n_f}$.

Given the candidate features, association constraints and cost ($\mathbf{Q}$, $\mathbf{G}$, $\mathbf{H}$), and the shape model (mean $\mathbf{M} \in \mathbb{R}^{2 \times \ell}$, $\mathbf{B} \in \mathbb{R}^{2\ell \times k}$), the problem consists on finding the optimal correspondence $\mathbf{S}$, the affinity transformation $\mathbf{A}$, the translation $\mathbf{t}$, and the deformation weights $\mathbf{c}$ that minimize the following error:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{A},\mathbf{c}} \quad & \eta \operatorname{tr}(\mathbf{H}\mathbf{S}^T) + \left\| \operatorname{vec}(\mathbf{Q}\mathbf{S}^T) - (\mathbf{I}_\ell \otimes \mathbf{A})\boldsymbol{\mu} - \mathbf{B}\mathbf{c} - (\mathbf{1}_\ell \otimes \mathbf{t}) \right\|_1, \qquad (19) \\ \text{s.t.} \quad & \mathbf{S} \in \{0,1\}^{\ell \times n_f}, \mathbf{S}\mathbf{1}_{n_f} = \mathbf{1}_\ell, \\ & s_{ti} = 0, [t,i] \in \{[t,i] | g_{ti} = 0\}, \\ & -3\sqrt{\boldsymbol{\lambda}_j} \le \mathbf{c}_j \le 3\sqrt{\boldsymbol{\lambda}_j}, \forall j = 1..k, \end{aligned}$$

where the first term in the objective function measures the assignation cost, and the second one the self reconstruction error. η is a parameter to trade off between the two terms. In the experiments, we always set the value to $\eta = 100$ and we found the final result was not sensitive to small change of this weight. Note that, instead of using l_2 norm, the reconstruction error is defined in l_1 norm because of its efficiency and robustness. Similarly to Eq. (18), the first constraint enforces $\mathbf{S}$ to select only one candidate for each landmark. However, the second constraint only allows $\mathbf{S}$ to select candidates for the t^{th} landmark from the corresponding set of candidates $\mathbf{Q}^t$ defined by $\mathbf{G}$. Finally, the third constraint imposes the subspace parameters to be plausible deformation values, where $\boldsymbol{\lambda} \in \mathbb{R}^{k\times 1}$ is a column vector containing the first k eigenvalues of the covariance matrix, of the training data.

However, optimizing Eq. (19) is NP-hard because of the integer constraints on $\mathbf{S}$. As in [19, 38], we approximate the problem with a continuous constraint, $\mathbf{S} \in [0,1]^{\ell\times n_f}$, and reformulate the problem in order to apply LP:

$$\min_{\mathbf{S},\mathbf{A},\mathbf{u},\mathbf{v}} \quad \eta\,\mathrm{tr}(\mathbf{H}\mathbf{S}^T) + \mathbf{1}_{2\ell}^T(\mathbf{u}+\mathbf{v}), \tag{20}$$

$$\begin{aligned} \text{s.t.} \quad & \mathrm{vec}(\mathbf{Q}\mathbf{S}^T) - (\mathbf{I}_\ell \otimes \mathbf{A})\boldsymbol{\mu} - \mathbf{B}\mathbf{c} = \mathbf{u} - \mathbf{v}, \mathbf{u} \geq \mathbf{0}_{2\ell}, \mathbf{v} \geq \mathbf{0}_{2\ell} \qquad (21) \\ & \mathbf{S} \in [0,1]^{\ell\times n_f}, s_{ti} = 0, [t,i] \in \{[t,i] | g_{ti} = 0\}, \\ & -3\sqrt{\boldsymbol{\lambda}_j} \leq \mathbf{c}_j \leq 3\sqrt{\boldsymbol{\lambda}_j}, \forall j = 1..k, \end{aligned}$$

where the two auxiliary variables $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{2\ell}$ replace the l_1 norm with a smooth term, and the linear constraint defined in Eq. (21). Finally, we gradually discretize $\mathbf{S}$, after solving the LP, by taking successive refinements based on trust-region shrinking [15]. Note that several elements in $\mathbf{S}$ will be 0 during the optimization process (illustrated in gray in Fig. 4). We simplify the optimization task by removing those elements (i.e. $[t,i] \in \{[t,i] | g_{ti} = 0\}$), reducing the number of variables and the LP cost from $O(\ell n_f)$ to $O(n_f)$.

5 Experiments & Results

This section illustrates the benefits of DSPA and CSPA, and compares them with state-of-the-art PA methods to build shape models of faces and human body joints' variation. First, we compare the performance of PA+PCA and SPA to build a 2-D shape model of faces and Motion Capture (MoCap) using 3-D datasets. For these experiments we use FaceWarehouse [5] and Carnegie Mellon University (CMU) MoCap [1] datasets, respectively. Finally, we illustrate the generalization of our 2-D body model in the problem of human pose estimation, in synthetic experiments on the CMU MoCap dataset, and real experiments on the Leeds Sports (LSP) [16] dataset.

5.1 Learning 2-D Face and Human Body Joints' Models

This section illustrates the benefits of DSPA and CSPA, and compares them with state-of-the-art PA methods to represent 2-D shape models of human skeletons and faces. First, we compare the performance of PA+PCA and SPA to represent a 2-D shape model of faces from FaceWarehouse dataset (Experiment 1). Next, we compare our discrete and continuous approaches in a large scale experiment (Experiment 2). Afterwards, we learn a model to represent 3-D joints of humans from the Carnegie Mellon University MoCap dataset [1]. We compare its generalization with the state-of-the-art PA methods (Experiment 3) and in a large scale experiment (Experiment 4). Finally, we show the benefits of our continuous 2-D model (CSPA) over 3-D models (Experiment 5) in the same datasets.

The aim of Experiments 1 and 2 is to build a generic 2-D face model that can reconstruct non-rigid facial deformation under a large range of 3-D rotations. For training and testing, we used the FaceWarehouse dataset that is composed of 150 subjects, each one with 20 different facial expressions. For all the subjects, dense point meshes are available, as well as RGB data generated from RGBD scans. The original model has 11510 points, and we sub-sampled the mesh to 49 and 162 landmarks, depending on the experiment. We rotated the 3-D faces 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 for 300 angles for testing, while varying the number of angles (i.e., rotations) in training. We report the Mean Squared Error (MSE) relative to the intra-eye size.

Similarly, the aim of Experiments 3 and 4 is to build a generic 2-D skeleton model from 3-D Motion Capture (MoCap). For 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, and we rotated the shapes in the same way as the experiments 1 and 2. We report the MSE relative to the torso size.

Experiment 1: Comparison with State-of-the-Art PA Methods on Faces This section compares DSPA, CSPA methods with the state-of-the-art Stratified Generalized Procrustes Analysis (SGPA)[8] [3]. For training we randomly selected 20 subjects, three expressions per subject and 49 landmarks (this is due to the memory limitations of SGPA). For testing we randomly selected 10 different subjects with the same three expressions as training. We report results varying the number of training rotations between $1 \sim 100$.

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 global optimum, being "Affine-factorization" faster.

Fig. 5 shows the mean reconstruction error and 0.5 of the standard deviation for 100 realizations. Fig. 5 (a) reports the results comparing PA, CPA and SGPA. As expected, PA and SGPA converge to CPA as the number of training rotations increases. However, observe that CPA achieves the same performance, but it is much more efficient. Fig. 5 (b) compares DSPA, CSPA, and SGPA followed by PCA (we will refer to this method SGPA+PCA). From the figure one can observe that error in the test for DSPA and SGPA+PCA decreases with the number of rotations in the training, and it converges to CSPA, which provides a bound on the lower error. Observe, that we used 60 3-D faces (20 subjects

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

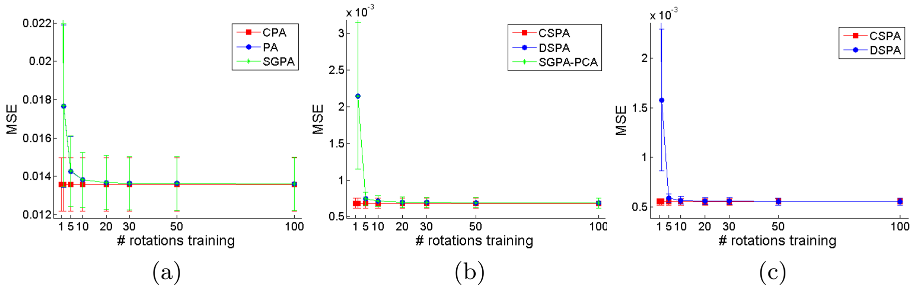


Fig. 5. Comparisons as a function of the number of training viewpoint projections, using a subspace of 25 basis for all deformable models. (*a*) Rigid and (*b*) Deformable models from Experiment 1, respectively; (*c*) CSPA and DSPA deformable models from Experiment 2.

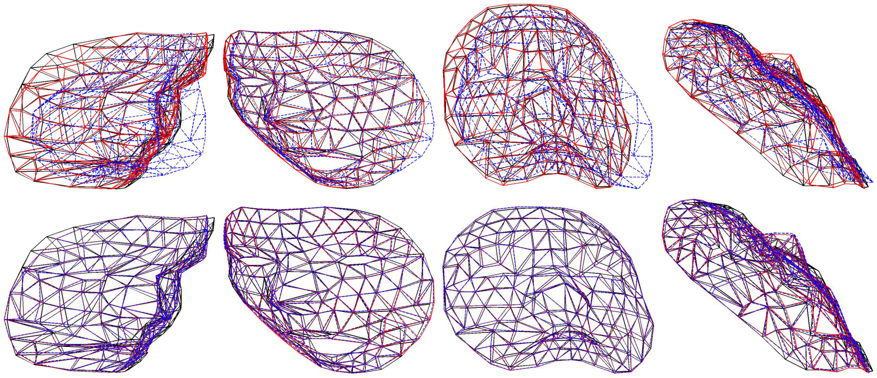
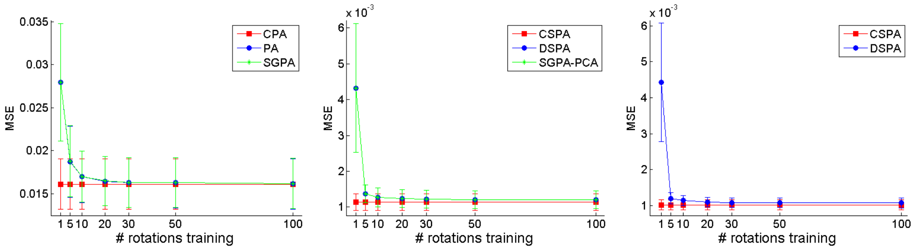


Fig. 6. Experiment 2 results with 1 (*top*) and 20 (*bottom*) rotations. *CSPA* (*solid red lines*) and *DSPA* (*dashed blue lines*) face reconstructions over ground truth (*solid black lines*).

and 3 expressions) within rotating angles $\phi, \theta \in [-\pi/2, \pi/2]$, and DSPA and SGPA+PCA needed about 20 angles to achieve similar result to CSPA. In this case, discrete methods need 20 times more space than the continuous. The execution times for each iteration with 20 rotations, on a 2.2GHz computer with 8Gb of RAM, were 0.62 sec. (DSPA), 0.18 sec. (CSPA) and 1.47 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 between $1 \sim 100$. For training we randomly selected 120 subjects, five expressions per subject and 162 landmarks. For testing we randomly selected 30 different subjects with the same five expressions as training.

Fig. 5 (c) shows the mean reconstruction error and 0.5 of the standard deviation for the 100 realizations, comparing DSPA and CSPA. As expected, DSPA converges to CSPA as the number of training rotations increases. However, ob-



(a) (b) (c)

Fig. 7. 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.

serve that CSPA achieves the same performance, but it is much more efficient. In this experiment, with 6000 3-D training faces (120 subjects and 5 expressions) and domain: $\phi, \theta \in [-\pi/2, \pi/2]$ discrete method required, again, around 20 2-D viewpoint projections to achieve similar results to CSPA. Thus, discrete model DSPA needs 20 times more storage space than CSPA. The execution times for each iteration with 20 rotations, on a 2.2GHz computer with 8Gb of RAM, were 6.17 sec. (DSPA) and 0.52 sec. (CSPA).

Qualitative results from CSPA and DSPA models trained with different number of rotations are shown in Fig. 6. Note that training DSPA model with 1 rotation (*top*) results in not properly reconstructed faces. However, training it with 20 rotations (*bottom*) leads to reconstructions almost as accurate as made by CSPA.

Experiment 3: Comparison with State-of-the-Art PA Methods Similarly to Experiment 1, this section compares DSPA, CSPA methods with the state-of-the-art Stratified Generalized Procrustes Analysis (SGPA) [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. 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.

Fig. 7 shows the mean reconstruction error and 0.5 of the standard deviation for the 100 realizations. Fig. 7 (a) reports the results comparing PA, CPA and SGPA. As expected, PA and SGPA converge to CPA as the number of training rotations increases. However, observe that CPA achieves the same performance, but it is much more efficient. Fig. 7 (b) compares DSPA, CSPA, and SGPA followed by PCA (we will refer to this method SGPA+PCA). From the figure one can 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

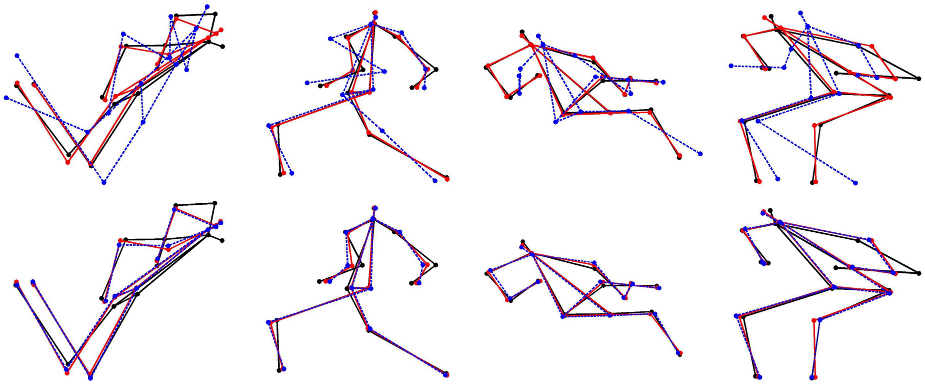


Fig. 8. Experiment 2 results with 1 (*top*), and 30 (*bottom*) rotations. Examples show skeleton reconstructions from continuous (*CSPA* in *solid red lines*) and discrete (*SPA* in *dashed blue lines*) models over ground truth (*solid black lines*).

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 4: 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. 7 (c) shows the mean reconstruction error and 0.5 of the standard deviation for the 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. 8. 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.

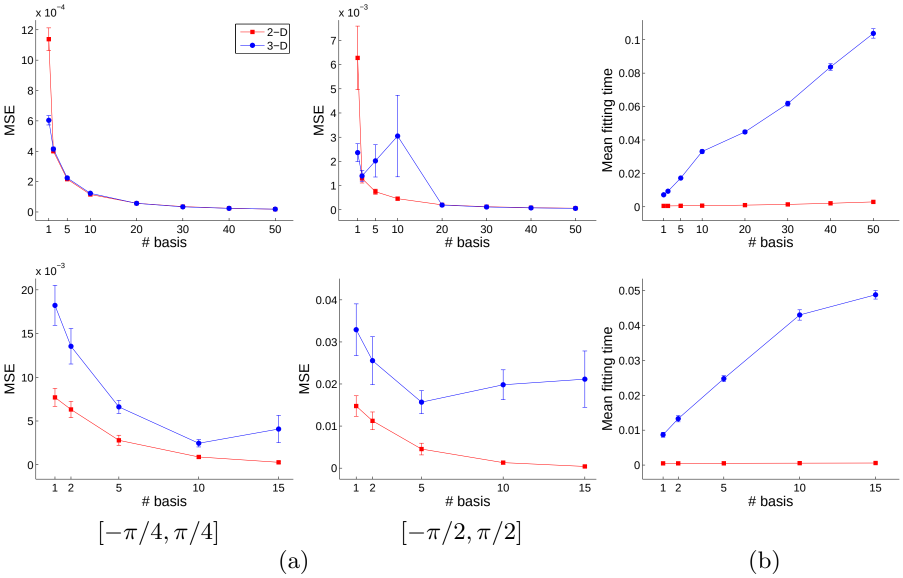
Experiment 5: 2-D vs 3-D Models In previous experiments we have shown that learning 2-D models with CSPA overcomes typical 2-D models learned with DSPA or PCA. This is because the use of 3-D data allows us to build unbiased models, able to generalize among different viewpoints. The question that strikes at this point is: Why do not use a 3-D models directly in test time rather than using the 3-D data to learn a 2-D model? From the comparison between 2-D and 3-D face models performed in [22], one concludes that both models have the same representation power, with 2-D models being faster in real time fitting. Following [22] we perform a comparison between 2-D and 3-D models, in the task of generalization to unseen samples.

This section compares unbiased 2-D (CSPA) to 3-D models in the task of faces and skeletons modeling. In this comparison the 2-D model will be learned using CSPA from Eq. (14). On the other hand, we will train the 3-D model optimizing Eq. (2) with the number of dimensions $d = 3$, and $\mathbf{A} \in \mathbb{R}^{3\times 3}$ being a rotation matrix. For the 2-D fitting of the 3-D model, we will use the standard algorithm from [13, 35], where the deformation parameters $\mathbf{c}_{3\mathrm{D}} \in \mathbb{R}^{k_{3\mathrm{D}}\times 1}$ of the 3-D model $\mathbf{M}_{3\mathrm{D}} + (\mathbf{B}_{3\mathrm{D}}\mathbf{c}_{3\mathrm{D}})^{(3)}$, as well as the rotation and scaling of the projection matrix $\mathbf{P} \in \mathbb{R}^{2\times 3}$, are estimated until convergence in a 2-step iterative algorithm[9]. For a fair comparison between models, the intrinsic camera matrix in $\mathbf{P}$ is fixed to be a scaled orthographic projection.

We compared 2-D and 3-D methods on FaceWarehouse and CMU MoCap datasets for faces and body joints' modeling, respectively. For both datasets, we performed the comparison with different angle domains ($\phi, \theta \in [-\pi/4, \pi/4]$ and $\phi, \theta \in [-\pi/2, \pi/2]$) for train and test, and we report results varying the number of subspace basis for both 2-D and 3-D models. For training the models on the FaceWarehouse dataset we randomly selected 120 subjects, 20 expressions per subject and 162 landmarks. For testing, we randomly selected 30 different subjects performing 20 different expressions (all expressions of the dataset). For training the models on the CMU MoCap dataset, we randomly selected 80 sequences with 30 frames per sequence and 14 landmarks. For testing we randomly selected 20 different sequences with 30 frames. Recall that all models in this experiment are trained with 3-D data. For testing, we rotated and projected 30 times each test shape.

Fig. 9 (a) shows the mean reconstruction error and 0.5 of the standard deviation for 100 realizations, incrementing the number of basis of the subspace models. We show the MSE for both experiments performed in $[-\pi/4, \pi/4]$ and $[-\pi/2, \pi/2]$ angle domains. Fig. 9 (b) reports the mean fitting time. Since experiments in both angle domains have similar test times, we only provide the time of one of them ($[-\pi/2, \pi/2]$) to avoid redundancy.

[9] The code was downloaded from author's website (http://www.research.rutgers.edu/~feiyang/web2/face_morphing.htm).



$[-\pi/4, \pi/4]$ $[-\pi/2, \pi/2]$

(a) (b)

Fig. 9. Experiment 5 results on FaceWarehouse (*top*) and CMU MoCap (*bottom*) datasets in $[-\pi/4, \pi/4]$ and $[-\pi/2, \pi/2]$ angle domains. Comparisons between 2-D and 3-D models as a function of the number of subspace basis, in terms of (*a*) mean reconstruction error and (*b*) mean fitting time.

Fig. 9 (*top*) reports the comparison on FaceWarehouse dataset. For narrow angle domains ($[-\pi/4, \pi/4]$), both 3-D and 2-D face models have similar performance, but 2-D models being faster (Fig. 9 (b)). However, 2-D models are more stable than 3-D models in the experiment with a wider test domain ($[-\pi/2, \pi/2]$). The fitting algorithm between the 3-D model and the 2-D test shape fails to estimate the projection matrix under extreme viewpoints, leading to a poor convergence. Note that the 3-D subspace will compensate the poorly estimated projection matrices, with enough number of basis.

The same effect occurs with models of body joints' variation in Fig. 9 (*bottom*), however, 2-D models outperformed 3-D for any number of basis on CMU MoCap dataset. Although the performance deteriorates on both datasets under large rotations, this is more evident on CMU MoCap dataset due to the high variability non-rigid deformations of the human body (see Fig. 10).

Note that in those situations where 2-D models obtain similar reconstruction error than 3-D models, increasing the number of basis of the 2-D model would lead to more accurate reconstructions than 3-D models, still benefiting from the fast 2-D model fitting.

5.2 Human pose estimation

This section compares our unbiased 2-D models and the subspace matching method against state-of-the-art algorithms, in the problem of human pose esti-

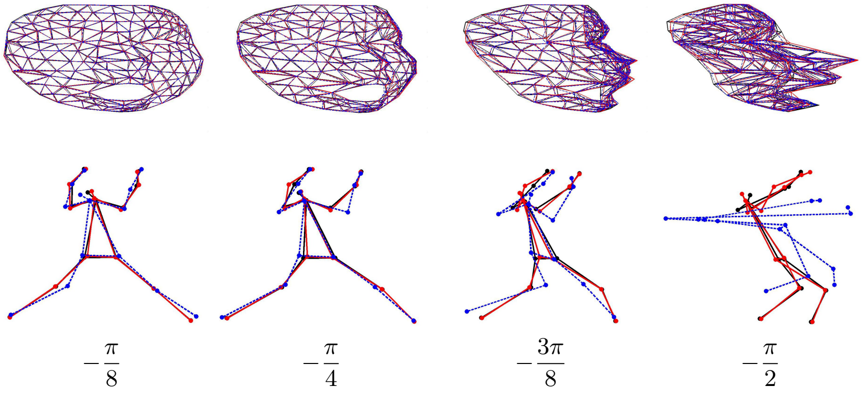
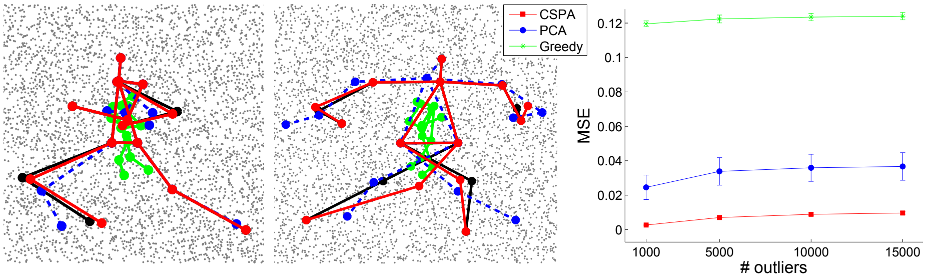


Fig. 10. Qualitative results from Experiment 5, rotating the test shapes in yaw on FaceWarehouse (*top*) and CMU MoCap (*bottom*) datasets. 2-D model (*solid red lines*) and 3-D model (*dashed blue lines*) reconstructions over ground truth (*solid black lines*). For both models, the number of basis was $k = 14$ on CMU MoCap dataset, and $k = 25$ on FaceWarehouse dataset.

mation. We performed synthetic experiments on the CMU MoCap dataset, and real experiments on the Leeds Sports (LSP) [16] dataset. For all experiments in this section we used the continuous version of our 2-D models, CSPA, trained with a set of 14 body joints.

Experiment 6: CMU MoCap dataset The aim of this experiment is to show the performance of our subspace feature selection method in the problem of human pose estimation, as a function of the number of outliers in the image. This synthetic experiment compares our method against two baselines on the CMU MoCap dataset: a greedy method for feature selection, and a method restricting the shape as [31]. Since this model is composed by a mean and a PCA of the data, we refer to this model as *PCA*. Recall that we introduced an affinity transformation to the feature selection formulation, which allows us to use a CSPA model in our approach. We refer to our method as *CSPA*. Also note that we are using our own implementation of [31] optimized in l_1 norm, since it was infeasible to perform this experiment with the original implementation (we add 100 times more features candidates and double of the number of landmarks in our experiment).

For training we randomly selected 3 sequences, each one with 30 frames, from the set of 11 running sequences of the user number 9. For testing we randomly selected 2 sequences with 30 frames from the same set, and we rotated 30 times each 3-D shape in the yaw and pitch angles, within the ranges of $\phi, \theta \in [-\pi/2, \pi/2]$, as the training domain. For each projected 2-D skeleton we synthetically added $1 \sim 15000$ random outliers in the frame of the image, uniformly distributed per each joint. See Fig. 11 (a) for examples of random feature candidates.



(a) (b)

Fig. 11. Experiment 6 results on CMU MoCap dataset. (*a*) *CSPA* model (*solid red lines*), *PCA* model (*dashed blue lines*), and *Greedy* (*green solid lines*) reconstructions over ground truth (*solid black lines*) and 5000 outliers (*grey dots*); and (*b*) MSE for each method as a function of the number of outliers.

We built the candidates matrix $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^\ell] \in \mathbb{R}^{2\times n_f}$ by concatenating the pixel locations $\mathbf{Q}^t \in \mathbb{R}^{2\times n_t}$ of the candidates features of each t^{th} landmark. The association cost of each candidate in matrix $\mathbf{H}$ is the euclidean distance between the candidate feature and the ground truth landmark location plus a random noise. We report the MSE relative to the torso size, varying the number of candidates for three methods.

Fig. 11 (b) shows the mean reconstruction error and the standard deviation for the 100 realizations. As expected, methods restricting the search with a shape model have better performance than the greedy approach. Moreover, observe that our approach using the *CSPA* model outperforms the one using just a *PCA* model. This is due to the addition to the affinity transformation, as well as the limits on the deformation parameters, in the feature selection formulation. Fig. 11 (a) shows two examples of the user number 9 of CMU MoCap dataset from two different viewpoints. Qualitative results also show that our method achieves a better fitting by means of a selection method robust to outliers. The execution times with 15000 outliers, on a 2.2GHz computer with 8Gb of RAM, were 0.72 sec. (PCA) and 0.68 sec. (CSPA) per image.

Experiment 7: Leeds Sport Dataset In this experiment, we tested the performance of CSPA models, in combination to the proposed subspace matching method, to detect humans on Leeds Sports (LSP) dataset. LSP contains 2000 images of people performing different sports, some of them including extreme viewpoints. We performed the comparison in the test set of 1000 images. 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 built the candidates matrix $\mathbf{Q} = [\mathbf{Q}^1, \ldots, \mathbf{Q}^\ell] \in \mathbb{R}^{2\times n_f}$ by concatenating the pixel locations

$\mathbf{Q}^t \in \mathbb{R}^{2\times 1000}$ of the 1000 candidates pixels with higher response of each t^{th} joint. Where the association cost of each candidate in matrix $\mathbf{H}$ is the normalized response for each pixel, obtained the SVM detector score [36]. We will refer to this model as *CSPA*. To evaluate the performance, we compared our approach with the state-of-the-art pose estimation method proposed by Yang and Ramanan[10] [36]. The error for each method is computed as the pixel distance between the estimated and ground-truth part locations.

Table 1 compares the error for each body joint of our method against [36], and a greedy approach. Our method improves the accuracy of all estimated joints, compared to the baselines, and only the Neck estimation of the greedy approach is better. Part of this is due to different anatomical labeling between LSP dataset and the training set of our *CSPA* model, CMU MoCap dataset. Qualitative results in Fig. 12 show that our approach has similar results to the state-of-the-art, but being more accurate in the estimation of the limb lengths.

Table 1. Comparison of human pose estimation approaches on LSP dataset. Errors in pixels are provided for each body joint (left and right joints are averaged), as well as the mean estimated error for the 14 joints.

Method	Head	Neck	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
YR [36]	21.75	18.97	20.54	31.27	49.03	22.78	27.24	38.42	29.95
Greedy	22.48	**18.41**	20.73	32.81	48.58	23.41	27.36	40.04	30.48
CSPA	**21.58**	18.48	**19.83**	**29.39**	**43.69**	**21.97**	**26.28**	**37.02**	**28.32**

The execution time per image of our feature selection method, on a 2.2GHz computer with 8Gb of RAM, was 6.84 sec. The most computationally intensive part of the method is calculating the response for each image using [36], which is shared with all compared methods.

6 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) 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. CSPA has two advantages over traditional PA and PPA: (1) it generates unbiased models because it uniformly covers the space of projections, and (2) it is more efficient in space and time. Experiments comparing 2-D SPA models of faces and bodies show improvements w.r.t. state-of-the-art PA methods. Additionally, we show here that CSPA generates 2-D models that generalize as well as 3-D models, but are faster to fit in test time. We reformulated the human pose estimation

[10] The code was downloaded from author's website (http://www.ics.uci.edu/~dramanan/).

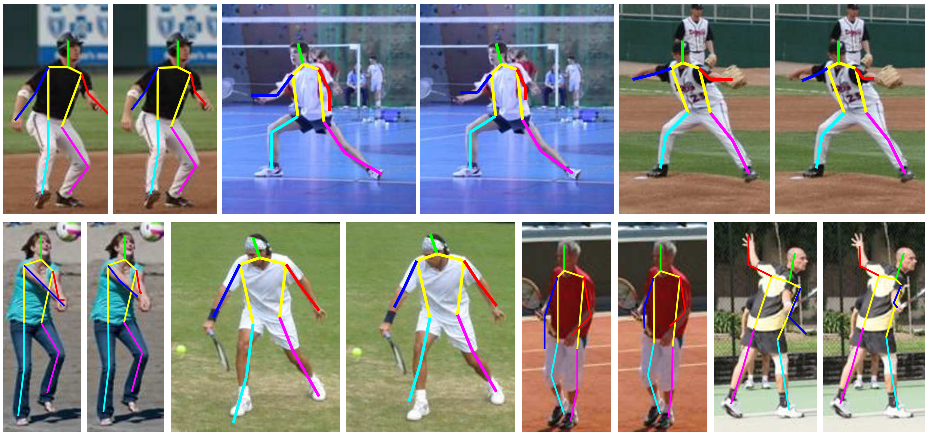


Fig. 12. Qualitative results for the LSP dataset. Left image from each pair of images shows the result from [36], and the right image shows our full approach using the *CSPA* model. Note how the *CSPA* leads to a more precise fitting of the body joints and more accurate limb lengths from different viewpoints.

task as a subspace matching problem, and we proposed a feature selection approach to robust to occlusions and large amount of outliers. In particular, CSPA models trained with motion capture data, combined with our subspace matching method, outperformed human pose estimation state-of-the-art approaches on the LSP dataset, since our unbiased 2-D models can successfully reconstruct different viewpoints, and the proposed feature matching method is able to handle occlusions and outliers. In future work, we plan to provide an in depth validation of 2-D models directly built from 3-D models.

7 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, SUR, Departament d'Economia i Coneixement, and 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 [20, 23]. 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 $\mathbf{A}^{(p)} \in \mathbb{R}^{pn \times \frac{m}{p}}$,

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

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

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: How to build a 2-D model from a 3-D model

We argued that unbiased 2-D and 3-D models have the same reconstruction power, being 2-D models faster, as well as we detailed how to build multi-view 2-D models from 3-D data. However, we might be interested in building an unbiased 2-D model even though we do not have access to the 3-D training data (e.g. NRSFM model built from 2-D data). Here we discuss how to build a 2-D model directly from a 3-D model, integrating over all possible viewpoints but also along the deformation parameters. Clearly, building a model from a previous learned model will lead to a loose of information, but benefits in some applications (e.g. real time fitting, enlarged pose variation models) can outweigh the information loss.

A method to *downgrade* a 3-D model to its homologous in 2-D was presented in [22]. They generate a 2-D dataset by a systematic sampling of the deformation and rotation parameters of the 3-D model. Then, they built a 2-D model from this enhanced 2-D dataset in a conventional manner. However, a uniform sampling of the rotation angles does not lead to a uniform sampling of the rotation space $SO(3)$. In addition, it is not clear how much sub-sampling is needed in the deformation parameters in order to generate a synthetic dataset with similar variance to the original training data. Just to give some example values, imagine that our model has only $k = 10$ basis, and we need $r = 20$ rotations to cover the domain of viewpoints that we are modeling. If we sample 4 times each axis of variance, we will need over $2 \cdot 10^7$ 2-D samples to train the 2-D model. Note that handle this dataset would be a large scale problem, even though we did not take extreme values for the example.

We discuss here how to build a 2-D model directly from a 3-D model, ensuring a uniform coverage of the rotation space, without the need of generating a huge synthetic 2-D dataset. Given a 3-D model composed by a mean $\mathbf{M}_{\mathrm{3D}} \in \mathbb{R}^{3\times\ell}$, the k_{3D} basis $\mathbf{B}_{\mathrm{3D}} \in \mathbb{R}^{3\ell\times k_{\mathrm{3D}}}$, and their corresponding eigenvalues $\boldsymbol{\lambda}_{\mathrm{3D}} \in \mathbb{R}^{k_{\mathrm{3D}}\times 1}$, we build a 2-D model ($\mathbf{M} \in \mathbb{R}^{2\times\ell}$, $\mathbf{B} \in \mathbb{R}^{2\ell\times k}$) by integrating along the axis of variance $\mathbf{B}_{\mathrm{3D}}$ within a domain $\boldsymbol{\mathit{\Gamma}}$, depending on the eigenvalues $\boldsymbol{\lambda}_{\mathrm{3D}}$, as we will discuss afterwards. Moreover, we rotate and project the 3-D model to the image plane using $\mathbf{P}(\boldsymbol{\omega}) \in \mathbb{R}^{2\times 3}$. Note that we ensure uniformity [24] in $SO(3)$ by means of the definite integral on the rotation domain $\boldsymbol{\mathit{\Omega}}$ and $d\boldsymbol{\omega} = \frac{1}{8\pi^2}\sin(\theta)d\phi d\theta d\psi$.

Given the 3-D model and the rotation domain $\boldsymbol{\mathit{\Omega}}$, we find its homologous 2-D model by minimizing the following error[11]:

$$E_{\text{2D-3D}}(\mathbf{M}, \mathbf{A}(\boldsymbol{\omega})_i, \mathbf{B}, \mathbf{c}(\boldsymbol{\omega})) =$$
$$\int_{\boldsymbol{\mathit{\Gamma}}}\int_{\boldsymbol{\mathit{\Omega}}} \left\| \mathbf{P}(\boldsymbol{\omega}) \left[\mathbf{M}_{\mathrm{3D}} + (\mathbf{c}_{\mathrm{3D}}(\boldsymbol{\gamma})^T \otimes \mathbf{I}_3)\mathbf{B}_{\mathrm{3D}}^{(3)} \right] - \mathbf{A}(\boldsymbol{\omega},\boldsymbol{\gamma})\mathbf{M} - (\mathbf{c}(\boldsymbol{\omega},\boldsymbol{\gamma})^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} d\boldsymbol{\gamma} \tag{22}$$

where $\mathbf{P}(\boldsymbol{\omega})$ is an orthographic projection of a 3-D rotation $\mathbf{R}(\boldsymbol{\omega})$ in the given domain $\boldsymbol{\mathit{\Omega}}$, defined by the rotation angles $\boldsymbol{\omega} = \{\phi, \theta, \psi\}$. The main difference between Eq. (14) and Eq. (22) is that instead of learning a 2-D model from 3-D shapes, our input now is a 3-D model. Hence, entries in the affinity transformation $\mathbf{A}(\boldsymbol{\omega},\boldsymbol{\gamma}) \in \mathbb{R}^{2\times 2}$ and the subspace weights $\mathbf{c}(\boldsymbol{\omega},\boldsymbol{\gamma}) \in \mathbb{R}^{k\times 1}$, $\mathbf{c}_{\mathrm{3D}}(\boldsymbol{\gamma}) \in \mathbb{R}^{k_{\mathrm{3D}}\times 1}$ are not only functions of the integration angles $\boldsymbol{\omega}$, but also functions of the deformation parameters $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_{k_{\mathrm{3D}}}\}$.

In addition, 2-D modeling from a 3-D model would be efficient, since the diagonal matrix $\mathbf{W} = \int_{\boldsymbol{\mathit{\Gamma}}} \mathbf{c}_{\mathrm{3D}}(\boldsymbol{\gamma})^T \mathbf{c}_{\mathrm{3D}}(\boldsymbol{\gamma}) d\boldsymbol{\gamma}$ encoding the deformations does not requires the explicit computation of the definite integral. This statement comes from solving:

$$E_{\mathrm{CPACA}}(\mathbf{B}, \mathbf{c}_i) = \int_{\boldsymbol{\mathit{\Gamma}}} \| [\boldsymbol{\mu}_1 + \mathbf{B}_1\mathbf{c}_1(\boldsymbol{\gamma})] - \mathbf{B}_2\mathbf{c}_2(\boldsymbol{\gamma}) \|_2^2 \, d\boldsymbol{\gamma}, \tag{23}$$

where, assuming zero mean $\boldsymbol{\mu}_1 = \mathrm{vec}(\mathbf{M}_1)$, we find that $\boldsymbol{\mathit{\Sigma}}_2 = \mathbf{B}_1\mathbf{W}\mathbf{B}_1^T$. Since $\boldsymbol{\mathit{\Sigma}}_1 = \mathbf{B}_1\boldsymbol{\mathit{\Lambda}}_1\mathbf{B}_1^T$, and $\boldsymbol{\mathit{\Sigma}}_1 = \boldsymbol{\mathit{\Sigma}}_2$, we find that the optimal value for the matrix encoding the deformations is the diagonal matrix containing the eigenvalues $\mathbf{W} = \mathrm{diag}(\boldsymbol{\lambda}_{\mathrm{3D}})$.

Similarly to CPA model (see Section 2) we find $\mathbf{M}$ by minimizing Eq. (22) using fixed point minimization (i.e. Eq. 9), where:

$$\mathbf{Z} = (\mathbf{M}\mathbf{M}^T)^{-1}\mathbf{M}\left(\mathbf{M}_{\mathrm{3D}}^T\mathbf{X}\mathbf{M}_{\mathrm{3D}} + \mathbf{B}_{\mathrm{3D}}^T((\mathbf{N}\otimes\mathbf{I}_3)\,\mathrm{vec}(\mathbf{X}))^{(3\mathbf{k}_{3D})}\mathbf{B}_{\mathrm{3D}}\right). \tag{24}$$

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

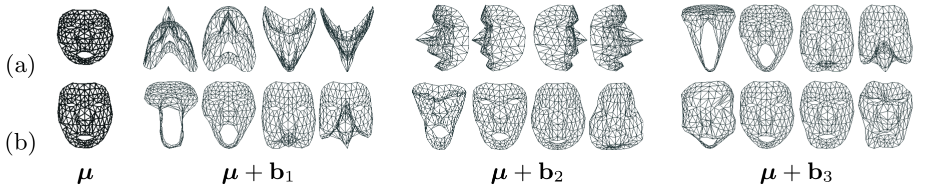

(a)
(b)
$\boldsymbol{\mu}$
$\boldsymbol{\mu} + \mathbf{b}_1$
$\boldsymbol{\mu} + \mathbf{b}_2$
$\boldsymbol{\mu} + \mathbf{b}_3$

Fig. 13. Illustration of the reference shape ($\boldsymbol{\mu}$) and the first three basis ($\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$) of the 2-D subspace model (a) directly build from 3-D model (b). We sampled each basis 4 times between the standard limits [7] to show their deformation behavior. All models were trained on FaceWarehouse [5] dataset, with 10 3-D faces from expressions number 0 and 1 (neutral and open mouth, respectively). Pitch an yaw integration limits were set to $\phi, \theta \in [-\pi/2, \pi/2]$ to train (a). Note that $\boldsymbol{\mu}$ and $\mathbf{b}_i$ in (b) are 3-D. They are projected frontally for a better comparison.

Matrix $\mathbf{N} = (\mathbf{c}_{3\text{D}}(\boldsymbol{\gamma}) \otimes \mathbf{I}_3 \otimes \mathbf{c}_{3\text{D}}(\boldsymbol{\gamma}))$ is a sparse matrix, with the nonzero elements being the eigenvalues in $\mathbf{W}$ and $\mathbf{X} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega})^T \mathbf{P}(\boldsymbol{\omega}) d\boldsymbol{\omega} \in \mathbb{R}^{3\times 3}$ averages the rotation covariances.

Similarly to CSPA model (see Section 3), substituting the optimal $\mathbf{M}^*$ and the expression $\mathbf{A}(\boldsymbol{\omega}, \boldsymbol{\gamma})$ in Eq. (22), allows us to find the optimal $\mathbf{B}$ by solving the eigenvalue problem, $\boldsymbol{\Sigma}\mathbf{B} = \mathbf{B}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ are the eigenvalues corresponding to columns of $\mathbf{B}$, and the covariance matrix $\boldsymbol{\Sigma} = ((\mathbf{I}_\ell \otimes \mathbf{Y}) \operatorname{vec}[\mathbf{L}])^{(2\ell)}$, being $\mathbf{Y} = \int_{\boldsymbol{\Omega}} \mathbf{P}(\boldsymbol{\omega}) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega})) d\boldsymbol{\omega} \in \mathbb{R}^{2\ell \times 2\ell}$ and $\mathbf{L} = \boldsymbol{\mu}_{3D}\boldsymbol{\mu}_{3D}^T + \mathbf{B}_{3\text{D}}\mathbf{W}\mathbf{B}_{3\text{D}}^T \in \mathbb{R}^{2\ell \times 2\ell}$.

As we illustrate in Figure 3 and Figure 13, our 2-D model Figure 13 (a) built directly from a 3-D model Figure 13 (b) have the same behavior that those models learned from the original 3-D data, Figure 3 (a-b), rotated and projected to 2-D.

C 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:

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{H} - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega}, \tag{25}$$

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,

$$E_{\text{CSPA}}(\mathbf{B}, \mathbf{c}(\boldsymbol{\omega})_i) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \mathbf{P}(\boldsymbol{\omega})\mathbf{D}_i(\mathbf{I}_\ell - \mathbf{H}) - (\mathbf{c}(\boldsymbol{\omega})_i^T \otimes \mathbf{I}_2)\mathbf{B}^{(2)} \right\|_F^2 d\boldsymbol{\omega} \tag{26}$$

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

$$\mathbf{c}(\boldsymbol{\omega})_i^* = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\bar{\mathbf{d}}_i. \tag{27}$$

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

$$E_{\text{CSPA}}(\mathbf{B}) = \sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| (\mathbf{I}_\ell \otimes \mathbf{P}(\boldsymbol{\omega}))\bar{\mathbf{d}}_i - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \right\|_2^2 d\boldsymbol{\omega} = \tag{28}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \left\| \left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \right\|_2^2 d\boldsymbol{\omega} = \tag{29}$$

$$\sum_{i=1}^{n} \int_{\boldsymbol{\Omega}} \operatorname{tr}\left[\left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)(\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i \left((\mathbf{I}_\ell \otimes \mathbf{P}(\omega))\bar{\mathbf{d}}_i\right)^T\right] d\boldsymbol{\omega} = \tag{30}$$

$$\operatorname{tr}\left[\left(\mathbf{I} - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\right)\boldsymbol{\Sigma}\right], \tag{31}$$

where:

$$\boldsymbol{\Sigma} = \int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \left(\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T\right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))^T d\boldsymbol{\omega}. \tag{32}$$

We can find the global optima of Eq. (31) 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[12]:

$$\boldsymbol{\Sigma} = (\operatorname{vec}[\boldsymbol{\Sigma}])^{(2\ell)} = \tag{33}$$

$$\left(\operatorname{vec}\left[\int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \left(\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T\right) (\mathbf{I}_\ell \otimes \mathbf{P}(\omega))^T d\boldsymbol{\omega}\right]\right)^{(2\ell)} = \tag{34}$$

$$\left(\left(\int_{\boldsymbol{\Omega}} (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) \otimes (\mathbf{I}_\ell \otimes \mathbf{P}(\omega)) d\boldsymbol{\omega}\right) \operatorname{vec}\left[\sum_{i=1}^{n} \bar{\mathbf{d}}_i \bar{\mathbf{d}}_i^T\right]\right)^{(2\ell)}, \tag{35}$$

which finally leads to:

$$\boldsymbol{\Sigma} = \left((\mathbf{I}_\ell \otimes \mathbf{Y}) \operatorname{vec}\left[\sum_{i=1}^{n} \bar{\mathbf{d}}_{ij} \bar{\mathbf{d}}_{ij}^T\right]\right)^{(2\ell)}, \tag{36}$$

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. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR. pp. 1014–1021. IEEE (2009)
3. Bartoli, A., Pizarro, D., Loog, M.: Stratified generalized procrustes analysis. IJCV 101(2), 227–253 (2013)
4. Brand, M.: Morphable 3d models from video. In: CVPR. vol. 2, pp. II–456. IEEE (2001)

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

5. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: a 3d facial expression database for visual computing. Visualization and Computer Graphics, IEEE Transactions on PP(99), 1–1 (2013)
6. Cootes, T.F., Edwards, G.J., Taylor, C.J., et al.: Active appearance models. PAMI 23(6), 681–685 (2001)
7. Cootes, T.F., Taylor, C.J.: Statistical models of appearance for computer vision (2004)
8. De la Torre, F.: A least-squares framework for component analysis. PAMI 34(6), 1041–1055 (2012)
9. Dryden, I.L., Mardia, K.V.: Statistical shape analysis, vol. 4. John Wiley & Sons New York (1998)
10. Frey, B.J., Jojic, N.: Transformation-invariant clustering using the em algorithm. PAMI 25(1), 1–17 (2003)
11. Goodall, C.: Procrustes methods in the statistical analysis of shape. Journal of the Royal Statistical Society. Series B (Methodological) pp. 285–339 (1991)
12. Gower, J.C., Dijksterhuis, G.B.: Procrustes problems, vol. 3. Oxford University Press Oxford (2004)
13. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
14. Igual, L., Perez-Sala, X., Escalera, S., Angulo, C., De la Torre, F.: Continuous generalized procrustes analysis. PR 47(2), 659–671 (2014)
15. Jiang, H., Drew, M.S., Li, Z.N.: Matching by linear programming and successive convexification. PAMI 29(6), 959–975 (2007)
16. 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
17. Kokkinos, I., Yuille, A.: Unsupervised learning of object deformation models. In: ICCV. pp. 1–8. IEEE (2007)
18. Learned-Miller, E.G.: Data driven image models through continuous joint alignment. PAMI 28(2), 236–250 (2006)
19. Li, H., Huang, X., He, L.: Object matching using a locally affine invariant and linear programming techniques. PAMI 35(2), 411–424 (2013)
20. Marimont, D.H., Wandell, B.A.: Linear models of surface and illuminant spectra. JOSA A 9(11), 1905–1913 (1992)
21. Marques, M., Stosić, M., Costeira, J.: Subspace matching: Unique solution to point matching with geometric constraints. In: ICCV. pp. 1288–1294. IEEE (2009)
22. 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)
23. Minka, T.P.: Old and new matrix algebra useful for statistics. http://research.microsoft.com/ minka/papers/matrix/, 2000
24. Naimark, M.A.: Linear representatives of the Lorentz group (translated from Russian). New York, Macmillan (1964)
25. Park, D., Ramanan, D.: N-best maximal decoders for part models. In: ICCV. pp. 2627–2634. IEEE (2011)
26. 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)
27. Perez-Sala, X., De la Torre, F., Igual, L., Escalera, S., Angulo, C.: Subspace procrustes analysis. In: ECCV Workshop on ChaLearn Looking at People (2014)
28. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: CVPR. pp. 588–595. IEEE (2013)

29. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Strong appearance and expressive spatial models for human pose estimation. In: ICCV. pp. 3487–3494. IEEE (2013)
30. Pizarro, D., Bartoli, A.: Global optimization for optimal generalized procrustes analysis. In: CVPR. pp. 2409–2415. IEEE (2011)
31. Roig, G., Boix, X., De la Torre, F.: Optimal feature selection for subspace image matching. In: ICCV Workshops. pp. 200–205. IEEE (2009)
32. 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)
33. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. PAMI 30(5), 878–892 (2008)
34. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. IJCV 67(2), 233–246 (2006)
35. Yang, F., Shechtman, E., Wang, J., Bourdev, L., Metaxas, D.: Face morphing using 3d-aware appearance optimization. In: Graphics Interface. pp. 93–99. Canadian Information Processing Society (2012)
36. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. PAMI 35(12), 2878–2890 (2013)
37. 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)
38. Zhou, F., Brandt, J., Lin, Z.: Exemplar-based graph matching for robust facial landmark localization. In: ICCV. pp. 1025–1032. IEEE (2013)