Continuous Procrustes Analysis to Learn 2D Shape Models from 3D Objects

Laura Igual
Universitat de Barcelona
ligual@ub.edu

Fernando De la Torre
Carnegie Mellon University
ftorre@cs.cmu.edu

Abstract

Two dimensional shape models have been successfully applied to solve many problems in computer vision such as object tracking, recognition, and segmentation. Typically, 2D shape models (e.g. Point Distribution Models, Active Shape Models) are learned from a discrete set of image landmarks once the rigid transformations are removed by applying Procrustes Analysis (PA). However, the standard PA process suffers from two main limitations: (i) the 2D training samples do not necessarily cover a uniform sampling of all 3D transformations of an object. This can bias the estimate of the shape model; (ii) it can be computationally expensive to learn the shape model by sampling 3D transformations; To solve these problems, we propose Continuous Procrustes Analysis *(CPA). CPA uses a continuous formulation that avoids the need to generate 2D projections from all 3D rigid transformations. Furthermore, it builds an efficient (space and time) non-biased 2D shape model from a 3D model of an object. Preliminary experimental results to build 2D shape models of objects and faces show the benefits of CPA over PA.*

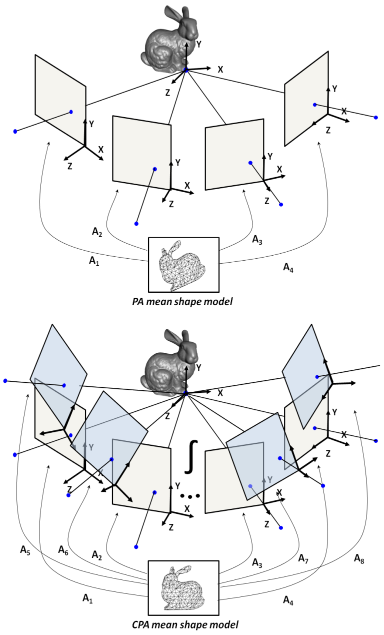

Figure 1. Illustration of Procrustes Analysis (PA) (top) and Continuous Procrustes Analysis (CPA) (bottom) to construct 2D shape models.

1. Introduction

Shape models have been successfully applied to solve many computer vision problems such as: object recognition [22, 13], face tracking [1, 7, 2, 3, 9], and image segmentation [19, 17]. The theoretical validity for the use of 2D shape models was proved by Ullman *et al.* [22] and Tomasi/Kanade [21], who showed how different 2D views of a 3D object can be recovered with a three dimensional subspace (under orthographic projection). This fact has been used in many computer vision algorithms to learn shape models by labeling the projections of 3D objects. Point Distribution Models (PDMs) and Active Shape Models (ASMs) [4] are among the most popular techniques that use 2D shape models. PDMs and ASMs build a shape model from a two dimensional training set of landmarks (projections from a finite set of 3D points on a shape surface). Procrustes Analysis (PA) [10] is used to remove rigid transformations and Principal Component Analysis (PCA) is applied to construct a subspace that models the variation of the normalized shapes [4]. Figure 1 (top) illustrates the PA process: given a 3D object PA computes the 2D mean shape that after rigid transformations (e.g. Euclidean or affine) can better reconstruct (in the least-squares sense) the projections of the object from different views. Although PA has been extensively used, it suffers from several limitations: (i) the 2D training samples do not necessarily cover a uniform sampling of all 3D transformations of an object. This can bias the estimate of the 2D models towards some particular configurations; (ii) it is computationally expensive to compute 2D projections from all possible 3D transformations of an object; (iii) PA aligns the samples w.r.t. the mean to remove rigid transformations. The residual variation of the shape (difference w.r.t the mean) is modeled with unsupervised algorithms (e.g. PCA). Independently estimating the registration parameters and the model shape parameters might result in a loss of relevant information.

To solve some of the limitations of PA (i and ii), this paper proposes *Continuous Procrustes Analysis* (CPA), a continuous formulation of PA. CPA builds a mean model of the object by integrating all possible rotations of a 3D mesh model of an object. Unlike PA, CPA needs a 3D model. Observe that CPA also solves a relatively unexplored problem in computer vision: to build 2D models that can recover possible 3D rigid and non-rigid configurations of the object. CPA generalizes PA using a *continuous* formulation that avoids the need to generate 2D projections from 3D configurations. There are several additional important advantages of the functional approximation of CPA: (1) there is no need to synthesize all possible 3D object configurations under rigid transformations, which greatly improves efficiency in space and time, (2) it is not biased to a non-uniform sampling of the rigid space. Figure 1 (bottom) illustrates the contribution behind CPA. CPA uses a continuous formulation that covers the space of all 3D transformations.

The rest of the paper is organized as follows: Section 2 reviews previous work, Section 3 motivates and derives CPA. Section 4 reports experimental results, Section 5 finalizes the paper with the conclusions and future work.

2. Previous Work

In many computer vision applications, it is important to learn a shape and/or appearance model invariant to geometric transformations (e.g. rotation, scale, non-rigid motion) [8, 10, 6, 15, 5, 1, 14].

Procrustes Analysis for Shape Alignment. PA [10] is one of the most popular algorithms to align shapes or images to a common reference frame removing rigid transformations. PA registers the training shape samples by finding a common co-ordinate frame while removing rigid transformations. The shape samples are represented as the 2D locations of ℓ points in a matrix $\mathbf{D}_i \in \mathbb{R}^{2\times\ell}$,

$$\mathbf{D}_i = \begin{pmatrix} x_{i1} & \cdots & x_{i\ell} \\ y_{i1} & \cdots & y_{i\ell} \end{pmatrix}.$$

Let $\mathbf{D}$ be a matrix $\mathbf{D} = [\mathbf{D}_1^T, \cdots, \mathbf{D}_m^T]^T \in \mathbb{R}^{2m\times\ell}$ (see notation[1]) containing the set of shape samples, where m is the number of $2D$ training examples, and ℓ refers to the number of landmarks.

PA optimizes over the geometric transformations that aligns each sample ($\mathbf{D}_i$) w.r.t. the mean by minimizing:

$$E_1(\mathbf{M}, \mathbf{A}) = \sum_{i=1}^{m} ||\mathbf{D}_i - \mathbf{A}_i\mathbf{M}||_F^2 = ||\mathbf{D} - \mathbf{AM}||_F^2, \quad (2.1)$$

where $\mathbf{M} \in \mathbb{R}^{2\times\ell}$ represents the mean shape, and each $\mathbf{A}_i$ in $\mathbf{A} = [\mathbf{A}_1^T, \cdots, \mathbf{A}_m^T]^T \in \mathbb{R}^{2m\times 2}$ corresponds to the rigid transformation for the shape sample $\mathbf{D}_i$ (e.g. Affine, Euclidean).

Recently, there has been a lot of interest in extending PA methods to the case where correspondences are unknown, using appearance features, or adding non-rigid deformations. Frey and Jojic [8] proposed a method to learn a factor analysis model 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 this problem, De la Torre and Black [6] proposed parameterized component analysis, a gradient-based method that learns a PCA model invariantly to affine transformations. Baker et al. [1] learned an Active Appearance Models (AAMs) invariantly to rigid and non-rigid motion. De la Torre and Nguyen [7] extended Parameterized component analysis to deal with non-linear appearance representations (using kernels) and non-rigid transformations. More recently, Miller et al. have proposed the Congealing method [15, 12], that uses an entropy measure to align images with respect to the distribution of the data. Cox et al. [5] extended [15, 12] with a least-squares optimization, similar in spirit to [6] but not including the subspace. Kookinos and Yuille [14] proposed a probabilistic framework and extended previous approaches [1, 15, 6] to deal with articulated objects by using a Markov Random Field (MRF) on top of AAMs.

Functional Data Analysis. Our work is related to previous works on functional data analysis [20]. Functional data analysis (FDA) [20] is a branch of statistics that analyzes data providing information about curves, surfaces or functions varying over a continuum. For instance, we can treat images as a bidimensional continuous function and many techniques of FDA will be applicable. In fact, many FDA methods are extensions of classical multivariate methods such as Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), and Analysis of Variance [20]. Most related to our work is Functional PCA [20]. Functional PCA models the way in which a set of functional data

[1] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{D}_i$ represents the i^{th} block matrix of the matrix $\mathbf{D}$. $\mathbf{d}_j$ represents the j^{th} column of the matrix $\mathbf{D}$. All non-bold letters denote scalar variables. $||\mathbf{D}||_F^2 = tr(\mathbf{D}^T\mathbf{D})$ designates the square of the Frobenius norm of a matrix. We assume $m, n, d, l, r \in \mathbb{N}$. $\mathbb{N}$ and $\mathbb{R}$ denote the set of natural and real numbers respectively, and $\mathbb{R}^d$ denotes the set of real vectors of dimension d. The set operation $\mathbf{\Omega} \setminus \mathbf{\Gamma}$ stands for the set difference of $\mathbf{\Omega}$ and $\mathbf{\Gamma}$. $\nabla_{\mathbf{u}}F$ is the gradient operator with respect to u of the function F.

varies from its mean, and in terms of these modes of variability, quantifies the discrepancy from the mean of each individual functional data.

In the computer vision literature, Ormoneit et al. [18] presented a robust method for learning functional PCA to model cyclic 3D human motion such as walking in motion-capture data. The pose of the body is represented by a time series of joint angles, which are automatically segmented into a sequence of motion cycles. The mean and the principal components of these cycles are computed using a new algorithm that enforces smooth transitions between the cycles by operating in the Fourier domain. This model is used for Bayesian tracking of 3D human motion. Also in the context of PCA, Levine and Shashua [16] used a continuous formulation of PCA that integrates over the convex hull of the sample data. Strictly speaking, the eigenvectors that the authors compute are not functions. By integrating over the convex hull of the data, they avoid biasing the principal components due to bad sampling of the input space.

3. Continuous Procrustes Analysis

This section proposes Continuous Procrustes Analysis (CPA) to learn an unbiased 2D shape model using as input the 3D structure of an object. CPA extends PA by adapting a continuous formulation that incorporates the information of all 3D rigid transformations.

3.1. Energy Functional for Continuous Procrustes Analysis

Let $\mathbf{D}$ be a matrix $\mathbf{D} = [\mathbf{D}_1^T, \cdots, \mathbf{D}_n^T]^T \in \mathbb{R}^{3n\times\ell}$, where each matrix $\mathbf{D}_i \in \mathbb{R}^{3\times\ell}$ is a 3D shape described by ℓ landmark points

$$\mathbf{D}_i = \begin{pmatrix} x_{i1} & & x_{i\ell} \\ y_{i1} & \cdots & y_{i\ell} \\ z_{i1} & & z_{i\ell} \end{pmatrix},$$

and n is the number of $3D$ training examples. Let $\mathbf{\Omega} = \{\boldsymbol{\omega} = (\phi, \theta, \psi)\} \subset \mathbb{R}^3$ be the set of 3D rotations, where $\boldsymbol{\omega}$ contains the Euler angles. CPA minimizes the following energy functional:

$$E_2(\mathbf{M}, \mathbf{A}) = \sum_{i=1}^{n} \int_{\Omega} F_2(\mathbf{M}, \mathbf{A}_i(\boldsymbol{\omega})) d\boldsymbol{\omega} =$$

$$= \sum_{i=1}^{n} \int_{\Omega} ||\mathbf{P}\mathbf{R}(\boldsymbol{\omega})\mathbf{D}_i - \mathbf{A}_i(\boldsymbol{\omega})\mathbf{M}||_F^2 d\boldsymbol{\omega}, \qquad (3.1)$$

where $\mathbf{M} \in \mathbb{R}^{2\times\ell}$ is the mean shape and each matrix $\mathbf{A}_i$ in $\mathbf{A} = [\mathbf{A}_1^T, \cdots, \mathbf{A}_n^T]^T \in \mathbb{R}^{2n\times 2}$, is an affine transformation. The matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ corresponds to the Cartesian parametrization of the 3D rotation depending on the Euler angle $\boldsymbol{\omega}$ and $\mathbf{P} \in \mathbb{R}^{2\times 3}$ is the matrix describing the orthographic projection onto the plane $Z = 0$. Note that for Euler angles (ϕ, θ, ψ), $d\boldsymbol{\omega} = \sin(\theta) d\phi d\theta d\psi$.

The functional in Eq. (3.1) is similar to the energy function in Eq. (2.1), with three main differences: (i) the affine transformations $\mathbf{A}_i$ are functions depending on the Euler angles $\boldsymbol{\omega}$, whereas in Eq. (2.1) $\mathbf{A}_i$ are variables, (ii) the 2D shape projection depends directly on the 3D structure of the object $\mathbf{D}_i$ and the 3D transformation parameters, (iii) it is a continuous formulation and discrete sums are replaced by integrals. We assume the 3D object has been centered and the translation can be removed. Because it is an orthographic projection, the Z translations are not modeled but could be compensated with a scale factor in the projection.

3.2. Optimization for CPA

This section describes an alternating procedure to minimize the CPA energy functional defined in Eq. (3.1):

$$\min_{\mathbf{M}, \mathbf{A}_1, \ldots, \mathbf{A}_n} E_2(\mathbf{M}, \mathbf{A}_1, \ldots, \mathbf{A}_n). \qquad (3.2)$$

The main difference of the present formulation with respect to PA is that $\mathbf{A}_i : \Omega \to \mathbb{R}^{3\times 3}$ are functions and not parameters. Moreover, it is worth noticing that the dependence of E_2 with the functions $\mathbf{A}_i$ is non-linear. This makes the minimization of E_2 (Eq. (3.2)) a non-linear variational problem. For simplicity in notation, we will not explicitly denote the dependency of $\mathbf{A}_i$ on $\boldsymbol{\omega}$ when it is obvious.

Although from a theoretical point of view the existence of a solution $(\mathbf{M}^*, \mathbf{A}_1^*, \ldots, \mathbf{A}_n^*)$ to the problem in Eq. (3.2) is guaranteed, in general, it may not be easy to find its explicit expression. For this reason, we propose the following alternating minimization algorithm that is guaranteed to converge to a critical point of the error function (typically a local minima). First, we start with an initial value $\mathbf{M}^* = \mathbf{M}^0$ and optimize over the functions $\mathbf{A}_1, \ldots, \mathbf{A}_n$ obtaining a solution $\mathbf{A}^* = [\mathbf{A}_1^{*T}, \ldots, \mathbf{A}_n^{*T}]^T$. In the next step, we minimize over $\mathbf{M}$ the functional $E_2(\mathbf{M}, \mathbf{A}^*)$. See Algorithm 1 for a sketch of the optimization algorithm.

```
Given an initialization $\mathbf{M}^* = \mathbf{M}^0$;
while not convergence of $\{\mathbf{A}^*, \mathbf{M}^*\}$ do
    Step 1: solve $\mathbf{A}^* = \text{argmin}_{\mathbf{A}} E_2(\mathbf{M}^*, \mathbf{A})$;
    Step 2: solve $\mathbf{M}^* = \text{argmin}_{\mathbf{M}} E_2(\mathbf{M}, \mathbf{A}^*)$ ;
end
```

Algorithm 1: CPA optimization algorithm

Step 1: Optimizing $E_2(\mathbf{M}, \mathbf{A})$ over the functions $\mathbf{A}_i, i = 1, \ldots, n$ is a Calculus of Variations problem that can be solved using the following equation $\nabla_{\mathbf{A}_i} F_2(\mathbf{M}, \mathbf{A}) = \mathbf{0}$, where $\nabla_{\mathbf{A}_i}$ is the gradient operator with respect to the unknown parameters of $\mathbf{A}_i$. The solution of these equations is $\mathbf{A}_i^*(\boldsymbol{\omega}) = \mathbf{P}\mathbf{R}(\boldsymbol{\omega})\mathbf{D}_i\mathbf{M}^T\mathbf{L}$, where $\mathbf{L} = (\mathbf{M}\mathbf{M}^T)^{-1}$.

Step 2: To optimize $E_2(\mathbf{M},\mathbf{A})$ over $\mathbf{M}$ the necessary condition is $\nabla_{\mathbf{M}} E_2(\mathbf{M},\mathbf{A}) = \mathbf{0}$.

Given that $\mathbf{M}$ and $\mathbf{D}_i$ do not depend on the rotation, the functional can be rewritten, involving three definite integrals, $\mathbf{Y}^{(1)} \in \mathbb{R}^{3\times 3}, \mathbf{Y}_i^{(2)} \in \mathbb{R}^{2\times 2}$ and $\mathbf{Y}_i^{(3)} \in \mathbb{R}^{3\times 2}$ as follows:

$$E_2(\mathbf{M},\mathbf{A}) = \sum_{i=1}^{n} tr\Big[\mathbf{D}_i^T \underbrace{\Big(\int_\Omega (\mathbf{PR})^T(\mathbf{PR})d\boldsymbol{\omega}\Big)}_{\mathbf{Y}^{(1)}} \mathbf{D}_i +$$

$$+\mathbf{M}^T \underbrace{\Big(\int_\Omega \mathbf{A}_i{}^T\mathbf{A}_i d\boldsymbol{\omega}\Big)}_{\mathbf{Y}_i^{(2)}}\mathbf{M} - 2\mathbf{D}_i^T \underbrace{\Big(\int_\Omega (\mathbf{PR})^T\mathbf{A}_i d\boldsymbol{\omega}\Big)}_{\mathbf{Y}_i^{(3)}}\mathbf{M}\Big].$$

Taking derivatives with respect to $\mathbf{M}$, the first term of the functional cancels out, therefore, we only need to compute the integrals $\mathbf{Y}_i^{(2)}$ and $\mathbf{Y}_i^{(3)}$. Let us call $\mathbf{X} = (\mathbf{PR})^T\mathbf{PR}$, and rewrite the integral form as follows:

$$\mathbf{Y}_i^{(2)} = \int_\Omega \mathbf{A}_i^T\mathbf{A}_i d\boldsymbol{\omega} = \mathbf{L}^T\mathbf{M}\mathbf{D}_i^T\Big(\int_\Omega \mathbf{X}d\boldsymbol{\omega}\Big)\mathbf{D}_i\mathbf{M}^T\mathbf{L},$$

$$\mathbf{Y}_i^{(3)} = \int_\Omega (\mathbf{PR})^T\mathbf{A}_i d\boldsymbol{\omega} = \Big(\int_\Omega \mathbf{X}d\boldsymbol{\omega}\Big)\mathbf{D}_i\mathbf{M}^T\mathbf{L}.$$

Therefore, to compute the value of these integrals we only need to solve the definite integral of $\mathbf{X}$. For instance, if we consider $\boldsymbol{\Omega} = \{(\phi,\theta,\psi) \in \mathbb{R}^3 || \phi|, |\theta|, |\psi| \leq \pi/2\}$, then we obtain:

$$\int_\Omega \mathbf{X}d\boldsymbol{\omega} = \begin{pmatrix} \frac{\pi^2}{8}+\frac{\pi^3}{16} & 0 & 0 \\ 0 & \frac{\pi}{8}+\frac{3\pi^3}{32} & 0 \\ 0 & 0 & \frac{-\pi}{8}+\frac{\pi^2}{8}+\frac{3\pi^3}{32} \end{pmatrix}$$

Finally, the solution for $\mathbf{M}$ can be expressed as follows:

$$\mathbf{M}^* = \Big(\sum_i \mathbf{Y}_i^{(2)}\Big)^{-1}\Big(\sum_i (\mathbf{Y}_i^{(3)})^T\mathbf{D}_i\Big).$$

4. Experiments

This section reports experimental results to validate the performance of CPA and compare it to PA. Note that, CPA only uses 3D training data, whereas PA uses 2D projections from 3D data.

4.1. Experiment 1

In the first experiment, we used the "Stanford Bunny" model from Stanford 3D Scanning Repository [2]. The data consisted of a dense 3D mesh of a real object captured with a 3D scanner. Figure 2 shows the original 3D shape model. Figure 3 displays four projections of the model rotated using the angles $\{\boldsymbol{\omega} = (\phi,\theta,\psi) \in \mathbb{R}^3 | \phi = \psi = 0, \theta \in \{0, \pi/9, 2\pi/9, 7\pi/18\}\}$. The model contained 432 landmark points, but for the purpose of visualization, only frontal landmark points were displayed (linked lines).

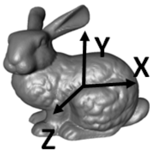

Figure 2. The 3D shape model.

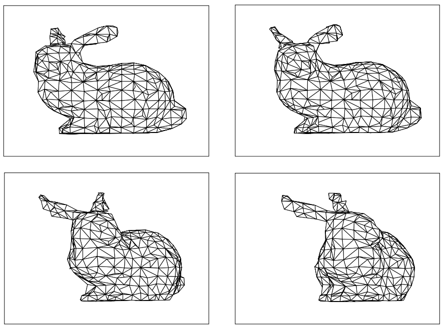

Figure 3. Samples of projections in 2D space used for training.

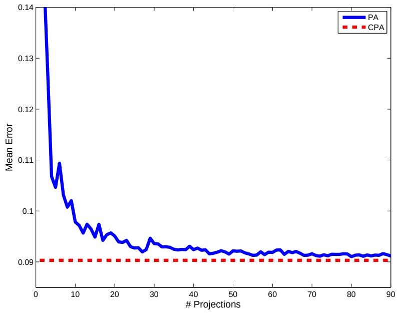

Figure 4. Reconstruction error for PA (solid line) using training sets with an increasing number of randomly chosen projections and reconstruction error for CPA (dashed line) using the 3D shape model as training set.

In order to compare the performance of PA and CPA, we computed the CPA reconstruction error and compared it with the PA reconstruction using different training sets with increasing number of training samples (number of 2D projections). The discrete training sets were built using the 2D projections after rotating the 3D objects using a random sampling of yaw angles in the interval $[0, \pi/2]$. The integration interval for CPA was defined as $\boldsymbol{\Omega} = \{(\phi,\theta,\psi) \in \mathbb{R}^3 | \phi,\psi = 0, 0 \leq \theta \leq \pi/2\}$.

We computed the reconstruction error using the energy function E_1 defined in Eq. (2.1) over the test set, which was built using the 2D projections after rotating uniformly the yaw angles between $[0, \pi/2]$. In particular, we considered the following set of Euler angles $\{\boldsymbol{\omega} = (\phi,\theta,\psi) \in \mathbb{R}^3 | \phi =$

[2] http://graphics.stanford.edu/data/3Dscanrep/

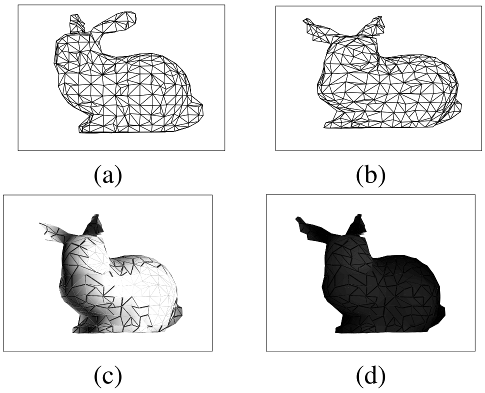

Figure 5. (a) Mean shape computed using PA. (b) Mean shape computed using CPA. (c) Visualization of reconstruction error for PA. (d) Visualization of reconstruction error for CPA. In (c, d) The lighter colors represent higher errors.

	Mean Error (± Std Dev.)	Maximum Error
PA	0.19 (± 0.13)	0.06
CPA	0.092 (± 0.08)	0.05

Table 1. Mean and maximum reconstruction error for PA (top row) and CPA (bottom row) on Bunny data.

$\psi = 0, 0 \leq \theta \leq \pi/2\}$, uniformly sampled every $\pi/180$ radians.

In Figure 4, the solid line represents the mean reconstruction error for PA using different training sets with an increasing size. For each size, the training was repeated 50 times. The dashed line marks the reconstruction error for CPA. Note that, the mean reconstruction error for PA decreases when the size of the training set increases and converged to the continuous error.

Let us show the results of PA when using a training set made by two random projections and compare them to the results obtained with CPA. Figure 5 shows the shape models computed using PA (a) and CPA (b). As can be observed, the PA model (a) was biased towards a certain direction, whereas CPA (b) computed an unbiased mean model near the mean angle $\pi/4$.

We evaluated the reconstruction error in previously untrained samples (test set detailed above). Figure 5 (c,d) shows the reconstructed shape error using PA (c) and CPA (d) on the sample of the test set corresponfing to projection of the rotation $\pi/4$. The gray level in the image represents the error, lighter levels correspond to higher errors. Note that we had error values for each point of the mesh; however, for the purposes of visualization we interpolated to obtain a continuous representation of the error. As can be observed, the PA model had higher reconstruction errors on the untrained sample in comparison to the CPA model. This was expected because the CPA mean was estimated taking into account more 3D transformations.

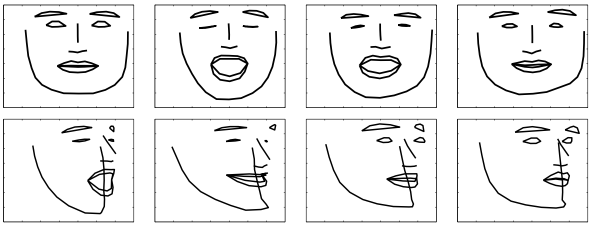

Figure 6. Samples of projections in 2D space of faces of the training set (first row) and the test set (second row).

Table 1 shows quantitative results for the reconstruction error. In order to provide a relative error measure, we computed the mean distance between the nearest neighbor points of the shape mesh (0.0019) and divided the error by this distance. Table 1 shows the mean error per point with the standard deviation and the maximum error per point in the test set for PA and CPA. We can see that the CPA method had lower reconstruction error in comparison to the PA on untrained samples.

4.2. Experiment 2

For this experiment, we used 3D face shape models obtained by performing structure from motion on several subjects from the Multi-PIE database [11]. We considered 30 subjects with three different expressions: neutral, smile and scream (total 90 subjects). See Figure 6 for some examples of projected shape models.

PA was trained on projections on the plane $Z = 0$ from the 90 3D models and CPA was trained on 90 3D shape models. No rotations were performed on this training set of models. The integration interval for CPA was defined as $\mathbf{\Omega} = \{(\phi, \theta, \psi) \in \mathbb{R}^3 | \phi, \theta = 0, 0 \leq \theta \leq \pi/2\}$. Figure 6 (first row) displays four projections of the face models belonging to the training set. The number of landmark points was 66, but we linked the landmarks with lines for better visualization.

The test set was built by projections of rotated models. In particular, the testing angles were $\{\boldsymbol{\omega} = (\phi, \theta, \psi) \in \mathbb{R}^3 | \phi = \psi = 0,\ \theta = \pi/4\}$. Figure 6 (second row) displays four projections of the test set.

Figure 7 shows the reconstruction performance using PA and CPA on four of the test samples. PA (first row) was not able to approximate the changes in yaw due to the untrained 3D rotations of the particular test sample. CPA (second row) approximated all the projections better, because functional optimizes over a predefined rotation interval.

Table 2 contains quantitative results of the reconstruction error for PA and CPA. The error measures were given in relation to the eye distance for each shape (mean eye distance $= 0.123$). The second and third columns show the mean relative error per point with the standard deviation and maximum relative error per point respectively. As can

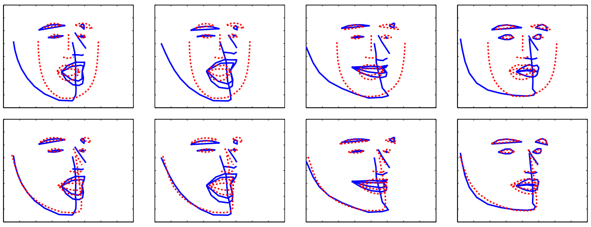

Figure 7. Reconstruction performance using PA (first row) and CPA (second row) on 4 test examples. The solid line represents the test face and the dashed line the reconstructed face.

	Mean Error (± Std Dev.)	Max. Error
PA	1.34e-002 (± 2.10e-003)	2.26e-002
CPA	8.19e-004 (± 5.56e-004)	4.81e-003

Table 2. Results of the reconstruction error for PA (top row) and CPA (bottom row) on facial models.

be seen, CPA outperformed PA.

5. Conclusions and Future Work

This paper proposed CPA, a continuous extension of PA to learn 2D shape models from one or more 3D objects. CPA models the 2D rigid deformations resulting from projections of the 3D object on different viewpoints.

CPA has two main advantages over PA: (i) no need to generate two dimensional samples, (ii) generate unbiased 2D models. Moreover, it is important to notice that the memory requirements for CPA are only the storage of the 3D model and the mean parameters. The experimental results show the advantages of CPA over PA in terms of better generalization to unseen samples. Currently, we are working on incorporating a subspace, rather than only the mean, to complex 3D non-rigid deformations.

Acknowledgments

This work was partially supported by MICINN Grants TIN2009-10435, FIS-PI071188 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

[1] S. Baker, I. Matthews, and J. Schneider. Automatic construction of active appearance models as an image coding problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1380 – 1384, October 2004.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.

[3] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

[4] T. Cootes and C. Taylor. Statistical models of appearance for computer vision. In *Tech. Report.*, 2001.

[5] M. Cox, S. Lucey, S. Sridharan, and J. Cohn. Least squares congealing for unsupervised alignment of images. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[6] F. De la Torre and M. J. Black. Robust parameterized component analysis: theory and applications to 2d facial appearance models. *Computer Vision and Image Understanding*, 91:53 – 71, 2003.

[7] F. De la Torre and M. Nguyen. Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment. In *IEEE Computer Vision and Pattern Recognition*, 2008.

[8] B. J. Frey and N. Jojic. Transformation-invariant clustering and dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.

[9] S. Gong, S. Mckenna, and A. Psarrou. *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, 2000.

[10] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society B*, 53(2):285–339, 1991.

[11] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. The cmu multi-pose, illumination, and expression (multi-pie) face database. Technical report, Carnegie Mellon University Robotics Institute.TR-07-08, 2007.

[12] G. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *International Conference on Computer Vision*, October, Brazil 2007.

[13] M. J. Jones and T. Poggio. Multidimensional morphable models. In *International Conference on Computer Vision*, pages 683–688, 1998.

[14] I. Kookinos and A. Yuille. Unsupervised learning of object deformation models. In *International Conference on Computer Vision*, October, Brazil 2007.

[15] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006.

[16] A. Levin and A. Shashua. Principal component analysis over continuous subspaces and intersection of half-spaces. In *ECCV (3)*, pages 635–650, 2002.

[17] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[18] D. Ormoneit, M. J. Black, T. Hastie, and H. Kjellström. Representing cyclic human motion using functional analysis. *Image Vision Comput.*, 23(14):1264–1276, 2005.

[19] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer-Verlag New York, Inc., 2003.

[20] J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer, 1997.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. Journal of Computer Vision.*, 9(2):137–154, 1992.

[22] S. Ullman and R. Basri. Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):992–1006, 1991.