Continuous Regression for Non-Rigid Image Alignment

Enrique Sánchez-Lozano[1] Fernando De la Torre[2]
Daniel González-Jiménez[1]

[1]Multimodal Information Area, Gradiant, Vigo, Pontevedra, 36310. Spain.
{esanchez,dgonzalez}@gradiant.org
[2]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213. USA.
ftorre@cs.cmu.edu

Abstract. Parameterized Appearance Models (PAMs) such as Active Appearance Models (AAMs), Morphable Models and Boosted Appearance Models have been extensively used for face alignment. Broadly speaking, PAMs methods can be classified into generative and discriminative. Discriminative methods learn a mapping between appearance features and motion parameters (rigid and non-rigid). While discriminative approaches have some advantages (e.g., feature weighting, improved generalization), they suffer from two major drawbacks: (1) they need large amounts of perturbed samples to train a regressor or classifier, making the training process computationally expensive in space and time. (2) It is not practical to uniformly sample the space of motion parameters. In practice, there are regions of the motion space that are more densely sampled than others, resulting in biased models and lack of generalization. To solve these problems, this paper proposes a computationally efficient continuous regressor that does not require the sampling stage. Experiments on real data show the improvement in memory and time requirements to train a discriminative appearance model, as well as improved generalization.

1 Introduction

Image alignment [4, 5, 12, 14, 15, 20, 23, 24] is a fundamental building block of many computer vision based systems ranging from robotics to medical diagnosis. Parameterized Appearance Models (PAMs) such as the Lucas-Kanade [24], Eigentracking [6], Active Appearance Models (AAMs) [1, 12, 15, 20, 23], Boosted Appearance Models [13] and Morphable Models [5] are among the most popular methods for appearance-based image alignment. These models have been successfully applied to face alignment, which is a key step for many applications in facial image analysis.

Broadly speaking, there have been two main streams to fit PAMs: generative and discriminative. In generative approaches [1, 4, 15, 22–24], the appearance variation of faces is modeled by performing Principal Component Analysis (PCA) (or kernel extensions) on training samples. Once the model has been

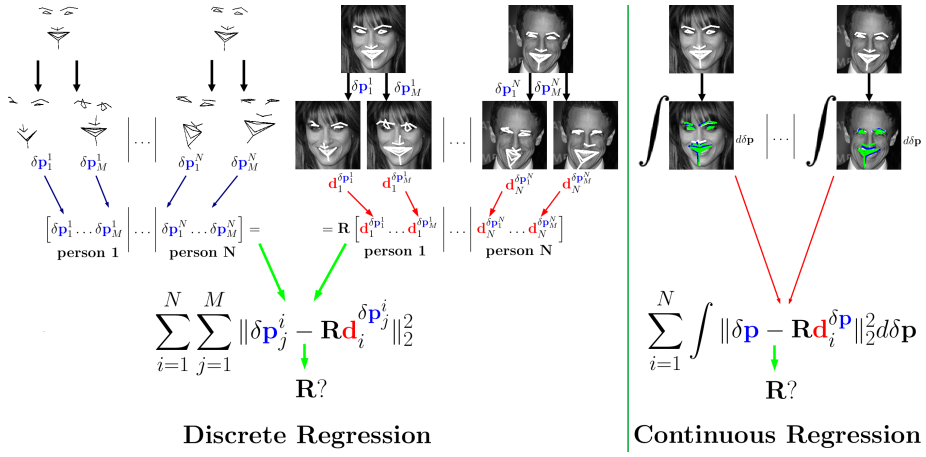

Fig. 1. Differences between discrete (*left*) and continuous regression (*right*). To compute discrete regression, we need to generate many pairs of motion parameters (rigid and non-rigid) and its corresponding features extracted from the image (*left*). Our method avoids the need to explicitly generate pairs of motion and texture samples by integrating over the continuous space of the motion parameters (*right*).

built, alignment is achieved by minimizing a cost function w.r.t. motion parameters (i.e., rigid and non-rigid); this is referred to as the fitting, registration, or alignment process. Although generative approaches achieved good results, these methods emphasize alignment by minimizing reconstruction error. This typically results in local minima [4] and lack of generalization [14, 20]. On the other hand, discriminative approaches [12, 14, 20] directly learn a mapping from image features to rigid and non-rigid parameters, effectively marginalizing the need to minimize the reconstruction error. This mapping can be obtained by linear regression [1], more complex regression models [11, 12, 20] or through a hard decision classifier (e.g., SVM), which outputs whether the model is well aligned [14]. Although widely used, a major problem of standard discriminative approaches is its computational complexity in training. This paper presents a continuous regression approach to efficiently train discriminative methods.

The standard training in discriminative models is as follows; let $\mathbf{d} \in \Re^{p\times 1}$ (see Footnote for the notation [1]) be a vectorized image. Discriminative models learn a mapping, $\mathbf{G}$, between perturbed motion parameters, $\delta\mathbf{p}$ (rigid and non-rigid) and the image features in the perturbed image space, $\mathbf{d}\big(\mathbf{f}(\mathbf{x};\mathbf{p}_0+\delta\mathbf{p})\big)$,

[1] Bold uppercase letters denote matrices ($\mathbf{D}$), bold lowercase letters denote column vectors (e.g., $\mathbf{d}$). $\mathbf{d}_j$ represents the j^{th} column of the matrix $\mathbf{D}$. Non-bold letters represent scalar variables. $tr(\mathbf{D}) = \sum_i d_{ii}$ is the trace of square matrix $\mathbf{D}$. $||\mathbf{d}||_2 = \sqrt{\mathbf{d}^T\mathbf{d}}$ designates Euclidean norm of $\mathbf{d}$. $\|\mathbf{A}\|_F^2 = tr(\mathbf{A}^T\mathbf{A})$ designates the Frobenius norm of matrix $\mathbf{A}$. *diag* is an operator that transforms a vector to a diagonal matrix. $\mathbf{I}_k \in \Re^{k\times k}$ denotes the identity matrix.

where $\mathbf{f}(\mathbf{x};\mathbf{p})$ is a geometric transformation and $\mathbf{p}_0$ represent the ground-truth motion parameters. Mathematically, the mapping is:

$$\delta\mathbf{p} = \mathbf{G}\Big(\mathbf{d}\big(\mathbf{f}(\mathbf{x};\mathbf{p}_0+\delta\mathbf{p})\big)\Big). \tag{1}$$

In the following, and without loss of generality, we will illustrate a drawback of discriminative models using linear regression, i.e. $\mathbf{G}(\mathbf{x}) = \mathbf{R}\mathbf{x}$.

The matrix $\mathbf{R}$ is learned by minimizing $\sum_{i=1}^{N}\sum_{j=1}^{M}||\delta\mathbf{p}_j^i - \mathbf{R}\mathbf{d}_i\big(\mathbf{f}(\mathbf{x};\mathbf{p}_0+\delta\mathbf{p}_j^i)\big)||_2^2$, w.r.t. $\mathbf{R}$, given a large number of images (N) and a large number of perturbations (M). This standard approach to learn discriminative models suffers from two major computational drawbacks. First, ideally the sampling needs to uniformly sample the motion parameter space, $\mathbf{p}$ (typically 30 dimensional space), to achieve good generalization. This results in an exponential number of samples with respect to motion parameters. So, in practice, the sampling is rarely uniform. Second, even with non-uniform sampling, the number of samples needs to be large, which leads to large memory and computational requirements. To solve these issues, this paper proposes a continuous regression method, which computes the regression without the need of sampling the training image. The main idea of the paper is illustrated in Fig. 1. In the left image we illustrate the discrete regression approach, that for each of the N training images, needs M perturbed samples. On the other hand, the continuous regression (right) only needs to sum over N images, and can integrate uniformly over the space of motion parameters.

2 Previous work

This section briefly reviews previous work on discriminative fitting and functional data analysis.

2.1 Discriminative fitting of Appearance Models

Image alignment algorithms have become increasingly important in computer vision. In particular, Parameterized Appearance Models (PAMs) [4, 5, 12, 14, 15, 20, 23, 24] have proven a useful way to register faces, a crucial step in applications such as face recognition, tracking and synthesis.

Broadly speaking, PAMs optimization algorithms can be classified into generative and discriminative. Generative approaches [4, 15, 20, 22, 23] learn a generative model that can reconstruct the image, and the fitting algorithms find the motion parameters that minimize the reconstruction error. Generative approaches can suffer from severe local minima and lack of generalization [2, 14]. On the other hand, discriminative approaches learn a mapping function from the image features to motion parameters [1, 20] or to discrete labels of well-aligned vs. badly-aligned [2, 14]. Recently, several successful approaches combined generative and discriminative methods [29, 30]. There are two main approaches to AAM discriminative fitting. The first set of methods learns a classifier to decide

whether the alignment is correct or not. In this category, Liu *et al.* [2, 14], proposed several algorithms to perform gradient descent on the motion parameters to align the image w.r.t. a classifier score that classifies if the alignment is correct or not. A second set of methods learns a mapping function between image features and motion parameters [9, 16, 18, 20]. To learn this mapping, a variety of regressors have been proposed: Tresadern *et al.*[16] and Sauer *et al.* [18] used a pool of weak classifiers and random forests; Saragih and Göcke [20] proposed to learn a mapping function for each step in the fitting process. Donner et al. [9] proposed to use canonical correlation analysis to learn a fast regressor for AAM. Recently, Rivera and Martinez [10] explored a discriminative approach to learn a manifold from graylevel values directly to landmarks. Alternatively, generative approaches can also be fitted in a discriminative manner [17, 19].

2.2 Linear Discriminative fitting

Let $\mathbf{u} = \mathbf{f}(\mathbf{x}; \mathbf{p}) \in \Re^{2l \times 1}$, denote a geometric transformation of pixel locations, that transforms a vector field $\mathbf{x} \in \Re^{2l \times 1}$ of image coordinates to another vector field $\mathbf{u} \in \Re^{2l \times 1}$, i.e. $[u_1, v_1, \cdots, u_l, v_l]^T$. l represents the number of pixels and $\mathbf{p}$ the vector of motion parameters. For instance, for an affine transformation (u_i, v_i) relates to (x_i, y_i) by:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} a_1 \; a_2 \\ a_3 \; a_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}, \tag{2}$$

where $\mathbf{p} = [a_1 \; a_2 \; .... \; a_6]$. Also, a non-rigid transformation can be represented by considering $\mathbf{x}$ as linear combination of a given set of basis, that is, $[x_1, y_1, \cdots, x_l, y_l]^T = \mathbf{x} + \mathbf{B}_s\mathbf{c}$, where $\mathbf{B}_s$ is the non-rigid shape model learned by performing PCA on a set of registered shapes [17]. In this case, $\mathbf{a}$, $\mathbf{c}$ represent the affine and non-rigid motion parameters respectively, and $\mathbf{p} = [\mathbf{a}; \mathbf{c}] \in \Re^{k \times 1}$, where $k - 6$ represents the number of principal components for the shape. $\mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0 + \delta\mathbf{p}_j^i)) \in \Re^{d \times 1}$ represents a feature or pixel intensities vector extracted from the i^{th} image at the locations given by $\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0 + \delta\mathbf{p}_j^i)$. $\mathbf{p}_i^0$ represents the ground-truth rigid and non-rigid parameters, and $\delta\mathbf{p}_j^i$ is a perturbation of these parameters.

Standard regression methods for alignment minimize the following error:

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \|\delta\mathbf{p}_j^i - \mathbf{R}\,\mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0 + \delta\mathbf{p}_j^i))\|_2^2, \tag{3}$$

w.r.t. $\mathbf{R} \in \Re^{k \times p}$, where i indexes images and j is the perturbation number. After re-arranging the features (i.e. $\mathbf{d}_i$) into the columns of $\mathbf{D} \in \Re^{p \times (NM)}$, and all the perturbations into the matrix $\mathbf{P} \in \Re^{k \times (NM)}$, the previous problem can be formulated as: $\min_{\mathbf{R}} \|\mathbf{P} - \mathbf{R}\mathbf{D}\|_F^2$. The optimal $\mathbf{R}$ is $\mathbf{R} = \mathbf{P}\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}$ (assuming that the inverse exists). However this approach has three main problems: (1) It is computationally costly in space ($O(NMp)$) and time ($O(N^2M^2)$). (2) It is not practical to sample uniformly the motion parameter space. Non-uniform sampling can lead to biased models that lack generalization. (3) The matrix $\mathbf{D}\mathbf{D}^T$

is typically rank deficient. To solve problem (3), there are several approaches. For instance, it is possible to compute principal component regression by projecting $\mathbf{D}$ onto the principal subspace. Alternative, reduced rank regression, the pseudo-inverse or regularized regression can be done to solve the rank deficient problem [7]. However, it remains unclear how to solve problem (1) and (2). Recently Huang et al. [8] proposed a regression method robust to non-uniformly sampling of the data space; however, the method does not scale well for high-dimensional data such as non-rigid image alignment. In the following sections, we describe a method to compute regression without need of sampling.

2.3 Functional Data Analysis

Our work is related to previous works on Functional Data Analysis (FDA) [25]. FDA is a branch of statistics that analyzes data providing information about curves, surfaces or functions varying over a continuum. For instance, images can be modeled as bidimensional continuous function and multivariate analysis methods (e.g., PCA, LDA) can be extended to the continuous domain. In the context of computer vision, Levin and Sashua [26] used a continuous formulation of PCA to solve the bias of learning PCA from discrete non-uniformly distributed samples. Recently, Igual and De la Torre [3] extended Procrustes Analysis to Continuous Procrustes Analysis in order to learn a 2D shape model from a 3D object deformation. In this paper, we will apply similar ideas and propose the continuous regression method.

3 Continuous Regression

A major limitation of standard regression methods is the need to sample the motion parameter space. Our main contribution is to provide a functional analysis framework [25] for regression. Our main assumption is that $\delta\mathbf{p}$ is a continuous variable, and we can integrate Equation 3 over the set of motion parameters $\delta\mathbf{p}$. That is, continuous regression minimizes:

$$\min_{\mathbf{R}} \sum_{i=1}^{N} \int \|\delta\mathbf{p} - \mathbf{R}\,\mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0 + \delta\mathbf{p}))\|_2^2 d\delta\mathbf{p}, \tag{4}$$

where the limits of the integral are finite. Once we have formulated learning the discrete models as a continuous problem, the next step is to solve the integrals. In this paper, we will focus on solving the integral expressions for non-rigid parameters, that is, $\mathbf{p} = \mathbf{c}$. We estimate the rigid parameters separately using discrete regression, because when using the face detector as an initialization, the variation of the rigid parameters is small (locally behaves linearly). In this case, the regression matrix can be estimated using few samples. On the other hand, non-rigid parameters are highly non-linear and a better sampling strategy is needed. So, in the following, $\delta\mathbf{p}$ will represent a perturbation in the $k-6$ dimensional shape parameter space with fixed rigid parameters [2].

[2] Onwards, k represents only the non-rigid shape parameters

Let $E_i(\mathbf{R}, \delta\mathbf{p}) = \|\delta\mathbf{p} - \mathbf{R}\,\mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0 + \delta\mathbf{p}))\|_2^2$ be the error function associated to the i^{th} training sample, then we can re-write Equation 4 as:

$$\min_{\mathbf{R}} \sum_{i=1}^{N} \int_{-\sigma_1\sqrt{\lambda_1}}^{\sigma_1\sqrt{\lambda_1}} \int_{-\sigma_2\sqrt{\lambda_2}}^{\sigma_2\sqrt{\lambda_2}} \cdots \int_{-\sigma_k\sqrt{\lambda_k}}^{\sigma_k\sqrt{\lambda_k}} E_i(\mathbf{R}, \delta\mathbf{p})\, d\delta p_1 \delta p_2 \cdots \delta p_k, \tag{5}$$

where λ_j is the eigenvalue associated to the j^{th} shape bases, and σ_j is the parameter that determines the number of standard deviations considered in the integral (typically between 2.5 and 3).

In order to find an analytic solution to previous integrals, we do a linear approximation, which is common in many alignment algorithms [5, 6, 24]:

$$\mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0 + \delta\mathbf{p})) \approx \mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0)) + \mathbf{J}_i^{\mathbf{p}} \delta\mathbf{p}, \tag{6}$$

where $\mathbf{J}_i^{\mathbf{p}} \in \Re^{p \times k}$ is the Jacobian matrix of $\mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0))$ w.r.t. $\mathbf{p}$, evaluated at $\mathbf{p}_i^0$. Onwards, and as a convenient abuse of notation, $\mathbf{d}_i = \mathbf{d}_i(\mathbf{f}(\mathbf{x}; \mathbf{p}_i^0))$. Using this approximation, we can further expand Equation 5 as:

$$\begin{aligned} \mathbf{R} = \arg\min_{\mathbf{R}} & \int \sum_{i=1}^{N} \delta\mathbf{p}^T \delta\mathbf{p}\, d\delta\mathbf{p} - 2 \int \sum_{i=1}^{N} \delta\mathbf{p}^T \mathbf{R}(\mathbf{d}_i + \mathbf{J}_i^{\mathbf{p}} \delta\mathbf{p})\, d\delta\mathbf{p} + \\ & + \int \sum_{i=1}^{N} (\mathbf{d}_i + \mathbf{J}_i^{\mathbf{p}} \delta\mathbf{p})^T \mathbf{R}^T \mathbf{R} (\mathbf{d}_i + \mathbf{J}_i^{\mathbf{p}} \delta\mathbf{p})\, d\delta\mathbf{p}. \end{aligned} \tag{7}$$

A necessary condition for the minimum of Equation 7 is that the derivative vanishes w.r.t. $\mathbf{R}$. After some linear algebra, it can be shown that

$$\begin{aligned} \frac{\partial E}{\partial \mathbf{R}} = & -\left(\int \delta\mathbf{p} d\delta\mathbf{p} \right) \sum_{i=1}^{N} \mathbf{d}_i^T - \left(\int \delta\mathbf{p} \delta\mathbf{p}^T d\delta\mathbf{p} \right) \sum_{i=1}^{N} (\mathbf{J}_i^{\mathbf{p}})^T \\ & + \mathbf{R} \left(\int d\delta\mathbf{p} \right) \sum_{i=1}^{N} \mathbf{d}_i \mathbf{d}_i^T + 2\mathbf{R} \sum_{i=1}^{N} \mathbf{d}_i \left(\int \delta\mathbf{p} \right)^T (\mathbf{J}_i^{\mathbf{p}})^T d\delta\mathbf{p} \\ & + \mathbf{R} \sum_{i=1}^{N} (\mathbf{J}_i^{\mathbf{p}}) \left(\int \delta\mathbf{p} \delta\mathbf{p}^T d\delta\mathbf{p} \right) (\mathbf{J}_i^{\mathbf{p}})^T = \mathbf{0}_{k \times p}. \end{aligned} \tag{8}$$

The analytic solution for these integrals is as follows:

$$\int d\delta\mathbf{p} = 2^k \prod_{i=1}^{k} \sigma_i \sqrt{\lambda_i}\ ,\ \int \delta\mathbf{p}\, d\delta\mathbf{p} = \mathbf{0}\ ,\ \int \delta\mathbf{p} \delta\mathbf{p}^T\, d\delta\mathbf{p} = 2^k \boldsymbol{\Lambda}(\boldsymbol{\sigma}) \prod_{i=1}^{k} \sigma_i \sqrt{\lambda_i} \tag{9}$$

where $\boldsymbol{\Lambda}(\boldsymbol{\sigma}) = diag(\frac{1}{3}\{\sigma_i^2 \lambda_i\}_{i=1}^k) \in \Re^{k \times k}$, and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_k)^T$. See Appendix A for the derivation of these integrals. Once the integrals are solved, Equation 4 has a closed-form solution as:

$$\mathbf{R}(\boldsymbol{\sigma}) = \boldsymbol{\Lambda}(\boldsymbol{\sigma}) \Big(\sum_{i=1}^{N} (\mathbf{J}_i^{\mathbf{p}})^T \Big) \Big(\sum_{i=1}^{N} \big(\mathbf{d}_i \mathbf{d}_i^T + (\mathbf{J}_i^{\mathbf{p}}) \boldsymbol{\Lambda}(\boldsymbol{\sigma}) (\mathbf{J}_i^{\mathbf{p}})^T \big) \Big)^{-1}. \tag{10}$$

Observe that in Equation 10, we sum over images ($i = 1 \dots N$) but not over the perturbations.

A closer look to Equation 10 reveals an interesting phenomenon of the proposed method. Let us consider $\mathbf{K}_i = \mathbf{d}_i\mathbf{d}_i^T + (\mathbf{J}_i^{\mathbf{p}})\boldsymbol{\Lambda}(\boldsymbol{\sigma})(\mathbf{J}_i^{\mathbf{p}})^T$ and $\mathbf{S} = \sum_{i=1}^{N}\big(\mathbf{K}_i\big)$. The matrix $\mathbf{S}$ is the covariance of the image features (in the linearization point) plus the weighted covariance of the Jacobians. Unlike the discrete method, that computes the covariance as the sum of perturbed samples, the continuous method approximates this covariance by the weighted outer product of Jacobians. Recall the dependence of the regression matrix with $\boldsymbol{\sigma}$, that is, $\mathbf{R}(\boldsymbol{\sigma})$ indicates that the regression matrix is a continuous function of the limit integral. This continuous regression matrix can be used within an annealing strategy in the fitting process, where in the first iterations the $\boldsymbol{\sigma}$ value is higher and it is lowered over iterations. Note that computing the regression matrix for different $\boldsymbol{\sigma}$'s is trivial.

Observe that for $\mathbf{S} \in \Re^{p\times p}$ to be invertible the $rank(\mathbf{S})$ must be equal to p, where p is the number of pixels. However, each image can, at most, contribute a matrix with rank $k+1$ to $\mathbf{S}$, where k is the number of non-rigid parameters. This is because each image is expressed as a linear combination of at most $k+1$ independent basis, that is ($rank(\mathbf{K}_i) \leq k+1$). Then, the $rank(\mathbf{S}) \leq N(k+1)$. To ensure full-rank, the minimum number of images that is required will be:

$$N \geq \frac{p}{k+1}. \tag{11}$$

In most applications, $p \gg k+1$, and it is not practical to label that many images. A common approach to solve the small sample size problem is to use principal component regression, reduced rank regression or regularization methods [7]. In this paper, we will use principal component regression (PCR) and regularization for its effectiveness and easiness of implementation. In principal component regression, the data $\mathbf{D}$ is projected into the principal components $\mathbf{B}_a^T \in \Re^{p\times a}$, that preserve a certain % of variance. In our case, we preserve 90% of the energy. Recall that the optimal transformation is Canonical Correlation Analysis (CCA), but in presence of few training samples, PCR typically removes noise and outperforms CCA without regularization. After projecting the data and regularizing onto the principal component, Equation 10 becomes:

$$\mathbf{R} = \boldsymbol{\Lambda}\Big(\sum_{i=1}^{N}(\mathbf{B}_a^T\mathbf{J}_i^{\mathbf{p}})^T\Big)\Big(\sum_{i=1}^{N}\big(\mathbf{B}_a^T(\mathbf{d}_i-\mathbf{d}_0)(\mathbf{d}_i-\mathbf{d}_0)^T\mathbf{B}_a+(\mathbf{B}_a^T\mathbf{J}_i^{\mathbf{p}})\boldsymbol{\Lambda}(\mathbf{B}_a^T\mathbf{J}_i^{\mathbf{p}})^T\big)+\lambda\mathbf{I}_a\Big)^{-1}.$$

Now, $\mathbf{K}_i = \big(\mathbf{B}_a^T(\mathbf{d}_i - \mathbf{d}_0)(\mathbf{d}_i - \mathbf{d}_0)^T\mathbf{B}_a + (\mathbf{B}_a^T\mathbf{J}_i^{\mathbf{p}})\boldsymbol{\Lambda}(\mathbf{B}_a^T\mathbf{J}_i^{\mathbf{p}})^T\big)$. If $\lambda = 0$, then $\mathbf{S} = \sum_{i=1}^{N}\mathbf{K}_i \in \Re^{a\times a}$, and the number of images minimum to ensure that $rank(\mathbf{S}) = a$ becomes

$$N \geq \frac{a}{k+1} \ll \frac{p}{k+1}. \tag{12}$$

4 Experiments

This section describes experimental results that compare our continuous regression method to fit discriminative appearance models versus standard approaches.

In the first experiment, we compared the memory requirements for each method. In the second experiment, we compared the time to compute the continuous regression versus the discrete one, as a function of the number of images and perturbations. Finally, the last experiment shows the improvement of our continuous approach vs. standard methods in fitting unseen faces. For the memory and timing experiments, we used the public available face database LFPW [21] and the associated labels. This database consists of 1432 faces downloaded from the web using simple text queries on sites such as google.com, flickr.com and yahoo.com. Each image was labeled with 29 landmarks. For the generalization experiment we have also used the BioID database [28], that consists of 1521 images, which are labeled with 20 landmarks. The shape model was learned using 2500 frontal and profile images of the Multi-pie database [27], and has 25 non-rigid shape basis. The rigid parameters are learned using discrete regression, and they will be the same for all methods. In all cases, we have used normalized graylevel values as feature vectors. For each landmark, we extracted a patch of 10×10 pixels. The feature vector consists on the whole set of patches, rearranged as column vectors.

4.1 Memory Requirements

The first experiment compares the memory requirements for the discrete vs. our proposed method. In the continuous regression we only needed memory to compute the matrices $\mathbf{S} \in \Re^{a \times a}$, $\mathbf{J} \in \Re^{a \times k}$ and $\boldsymbol{\Lambda} \in \Re^{k \times k}$, where a is the number of principal components (≈ 100), and k the shape parameters (25). Typically, $a >> k$, and hence the cost in memory is approximately $O(a^2)$. In the case of the discrete regression, the computational cost is $O(NMa)$, where N is number of images and M number of perturbations. Recall that both methods used the same PCA projections, and hence we exclude the memory requirements to compute PCA. PCA was applied to the texture set, retaining 90% of energy ($a = 173$). Fig. 2 (left) represents the memory requirements as a function of the number of the training images and the number of perturbations. We selected 5 sets for training using $15, 25, 50, 100$ and 200 training images respectively. For each set, we used five perturbations (1, 10, 20, 50, 100) for the discrete regression. As can be seen, as the number of samples increases, memory requirements grows linearly (*pink*) for the standard regression. In contrast, our continuous regression keeps memory constant and has a computational cost equivalent to one perturbation.

4.2 Training Time

This experiment tests the training time for both discrete and continuous methods. The most computationally expensive part of the continuous regression method is computing the inverse matrix, with a computational cost of $O(pk^2)$. For the discrete regression, the computational cost is $O(N^2M^2)$. Fig. 2 (right) shows the mean and standard deviation training times for the same protocol described in Experiment 1. We run the experiment 10 times and show the mean

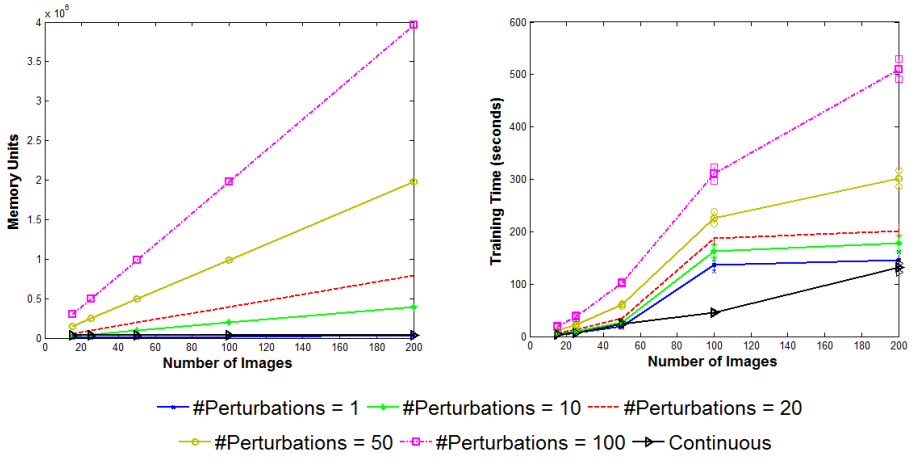

Fig. 2. Left: Number of training images (*x-label*) vs. Memory Units (*y-label*). Each line represents the memory requirements for different number of images and perturbations per image, see text. **Right:** Number of training images (*x-label*) vs. time, in seconds (*y-label*).

and the standard deviation. As expected, the continuous regression and the discrete regression with one perturbation are the less computationally expensive algorithms.

4.3 Generalization

The last experiment evaluates the fitting and generalization performance. For the LFPW database, we have selected 789 training images from the training set and 227 testing images from the testing set (those that were available), whereas for the BioID we randomly selected 1012 images for training and 506 for testing. For the discrete regression, we used 20, 50 and 100 perturbations for each image. PCA was computed using the training images preserving 90% of the energy. The λ of the regularization was chosen by cross-validation. For the testing images, the ground truth landmarks are known, and we randomly perturb the non-rigid parameters (the rigid parameters are fixed) with $\sigma = 3$, that is equivalent to perturb the parameters three standard deviations. First row of Fig. 4 and Fig. 5 show some examples of the initial perturbations for each database. After convergence, we computed the mean square error (MSE) between the ground-truth landmarks and the ones returned by our algorithm, divided by the inter-ocular distance. All methods use the same initialization.

We compared the performance of three methods: the discrete algorithm (for 3 ranges of perturbation), the continuous one, and approximating the regression matrix for the Jacobian, i.e., $\mathbf{R} = \left((\mathbf{J}_i^{\mathbf{p}})^T(\mathbf{J}_i^{\mathbf{p}})\right)^{-1}(\mathbf{J}_i^{\mathbf{p}})^T$. The last approximation was first proposed by Cootes et al. [1]. In the case of having several training

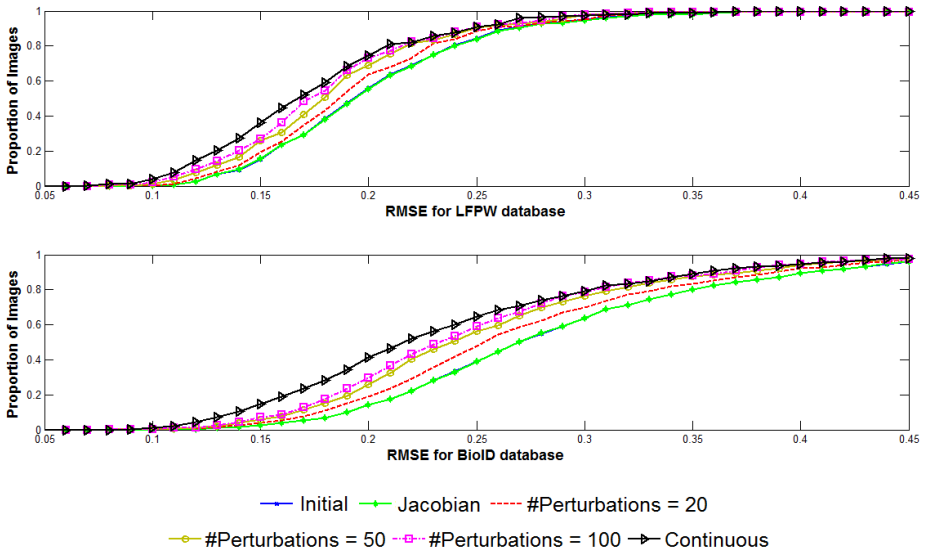

Fig. 3. Cumulative RMSE for the LFPW (**top**) and BioID (**bottom**) databases. As can be seen, the continuous method (*black*) outperforms the other methods. The Jacobian approximation is not valid when the initialization is far from the ground-truth data.

images, the regression matrix would be the average:

$$\mathbf{R} = \left(\sum_{i=1}^{N} (\mathbf{J}_i^{\mathbf{P}})^T (\mathbf{J}_i^{\mathbf{P}}) \right)^{-1} \left(\sum_{i=1}^{N} \mathbf{J}_i^{\mathbf{P}} \right)^T . \tag{13}$$

Fig. 3 shows the cumulative error for both databases. Fig. 4 and Fig. 5 show how different algorithms fit six images. The first row illustrate the initialization of the algorithm, the second row the solution provided by the best discrete regression (100 perturbation), and the third row the continuous regression. It is worth noting that the Jacobian approximation only works when the initialization is close to the ground-truth.

5 Conclusions and future work

This paper presents a continuous method to train a linear regressor for discriminative fitting. The key idea is to assume a continuous motion parameter space and integrate over it, rather than discretely sampling the motion parameter space. Three benefits follow: it is more computationally efficient in space and time, and provides a better generalization. Although the method has worked well in practice, a major limitation is the range of displacements that it can learn, due to the linealization in Equation 6.

To solve the problem of learning large displacements, in further work we will explore using several linealization points or using additive linealizations. That

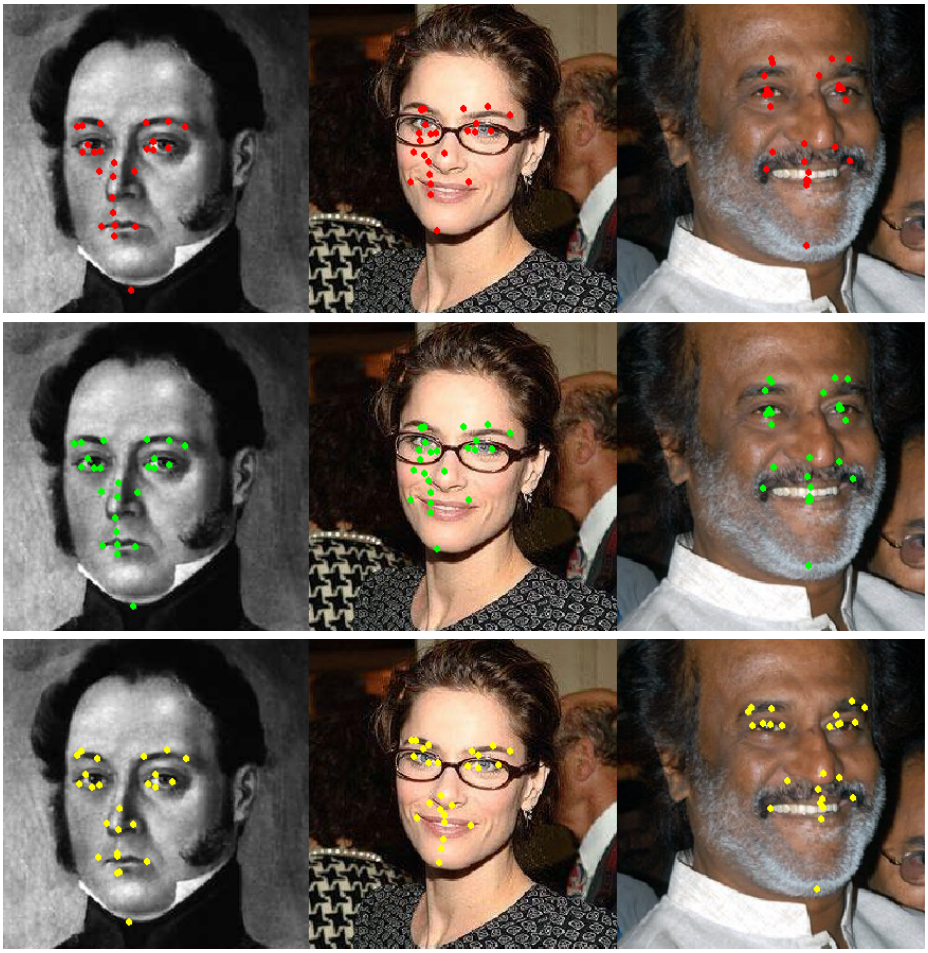

Fig. 4. Images from the LFPW database (best viewed in color). First row shows the perturbed landmarks, second row the estimated landmarks using discrete regression (using 100 perturbations), and the third row the estimated landmarks using our continuous regression.

is, consider $\delta\mathbf{p} = \delta\mathbf{p}_1 + \delta\mathbf{p}_2$. Then, $\mathbf{d}(\mathbf{f}(\mathbf{x}; \delta\mathbf{p}^0 + \delta\mathbf{p}_1 + \delta\mathbf{p}_2)) \approx \mathbf{d}(\mathbf{f}(\mathbf{x}; \delta\mathbf{p}^0 + \delta\mathbf{p}_1)) + \mathbf{J}_\mathbf{p}\delta\mathbf{p}_2 \approx \cdots$. Finally, we have illustrated the benefits of our continuous approach using the linear regression model, but our results are more general and can be extended to other discriminative models rather than linear regression.

Acknowledgments

This work has been partially supported by the European Commission's Seventh Framework Programme (FP7 - Capacities) under grant agreement no.

Fig. 5. Images from the BioID database (best viewed in color). First row shows the perturbed landmarks, second row the estimated landmarks using discrete regression (using 100 perturbations), and the third row the estimated landmarks using our continuous regression.

285901 (LIFTGATE project), and by Xunta de Galicia under projects VISAGE (10TIC008CT) and PROSSAE (10SIN001CT).

References

1. Cootes, T. F., Edwards, G. J., Taylor, C. J.: Active Appearance Models. PAMI **23**(6) (2001) 681–685
2. Hao, W., Liu, X., Doretto, G.: Face alignment via boosted ranking model. CVPR (2008)
3. Igual, L., De la Torre, F.: Continuous Procrustes Analysis to Learn 2D Shape Models from 3D Objects. 3rd Workshop on Non-Rigid Shape and Deformable Image Alignment, in Conjunction with CVPR (2010) 17–22
4. Nguyen, M.H., De la Torre, F.: Metric Learning for Image Alignment. Int'l Journal of Computer Vision **88**(1) (2010) 69–84
5. Jones, M.J., Poggio T.: Multidimensional Morphable models. ICCV (1998)
6. Black, M.J., Jepson, A.D.: Eigentracking: Robust Matching and Tracking of Objects of Articulated Objects Using a View-Based Representation. Int'l Journal of Computer Vision **26**(1) (1998) 63–64

7. De la Torre, F.: A least-squares framework for Component Analysis. PAMI **34**(6) (2012) 1041–1055
8. Huang, D., Storer M., De la Torre, F., Bischof H.: Supervised Local Subspace Learning for Continuous Head Pose Estimation CVPR (2011)
9. Donner, R., Reiter, M., Langs, G., Pelloscheck, P., Bischof, H.: Fast active appearance model search using canonical correlation analysis. PAMI **28**(10) (2006) 1690–1694.
10. Rivera, S., Martinez, A.M.: Learning Deformable Shape Manifolds. Pattern Recognition **45**(4) (2012) 1792–1801
11. Valstar, M., Martínez, B., Binefa, X., Pantic, M.: Facial Point Detection using Boosted Regression and Graph Models CVPR (2010)
12. Tian, Y., Narasimhan, S.: Globally Optimal Estimation of Nonrigid Image Distortion. Int'l Journal of Computer Vision (2010)
13. Liu, X.: Discriminative Face Alignment. PAMI **31**(11) (2009) 1941–1954
14. Liu, X.: Optimal gradient pursuit for face alignment. AFGR and Workshops (2011)
15. De la Torre, F., Nguyen, M.H.: Parameterized Kernel Principal Component Analysis: Theory and Applications to Supervised and Unsupervised Image Alignment. CVPR (2008)
16. Tresadern, P., Sauer, P., Cootes, T. F.: Additive Update Predictors in Active Appearance Models. BMVC (2010)
17. Cootes, T. F., Taylor, C. J.: Statistical Models of Appearance for Computer Vision. Online Technical Report Available from http://www.isbe.man.ac.uk/∼bim/Models/app_models.pdf (2004)
18. Sauer, P., Cootes, T. F., Taylor, C. J.: Accurate Regression Procedures for Active Appearance Models. BMVC (2011)
19. Cootes, T. F., Edwards., G. J., Taylor, C. J.: A Comparative Evaluation of Active Appearance Model Algorithms. BMVC **2** (1998)
20. Saragih, J., Göcke., R.: Learning AAM fitting through simulation. Int'l Journal on Pattern Recognition **42**(11) (2009) 2628–2636
21. Belhumeur, P. N., Jacobs, D. W, Kriegman, D. J., Kumar, N.: Localizing Parts of Faces Using a Consensus of Exemplars. CVPR (2011)
22. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. Int'l Journal on Computer Vision **56**(3) (2004) 221–255
23. Matthews, I., Baker, S.: Active Appearance Models Revisited. Int'l Journal on Computer Vision **60**(2) (2004) 135–164
24. Lucas, B., Kanade, T.: An Iterative image registration technique with an application to stereo vision. IJCAI81 (1981)
25. Ramsay, J., Silverman, B.: Functional Data Analysis. Springer (1997)
26. Levin, A., Shashua, A.: Principal Component Analysis over continuous subspaces and intersection of half-spaces. ECCV (2002)
27. Gross, R., Matthews, I., Cohn, J.F., Kanade, T., Baker, S.: Multi-PIE. AFGR (2008)
28. Jesorksy, O., Kirchberg, K. J., Frischhold, R. W.: Robust face detection using the hausdorff distance. Int'l Conference on Audio- and Video-Based Biometric Person Authentication (2001)
29. Saragih J. , Lucey S. and Cohn, J. F.: Deformable Model Fitting by Regularized Landmark Mean-Shift . Int'l Journal of Computer Vision, **91**(2) (2011) 200–215
30. X. Zhu, D. Ramanan. Face detection, pose estimation and landmark localization in the wild CVPR (2012)

A Integral Resolution

This appendix provides details of the computation for the integrals in Equation 9.

A.1 Constant Integral

$$\begin{aligned}\int d\delta\mathbf{p} &= \int_{-\sigma_1\sqrt{\lambda_1}}^{\sigma_1\sqrt{\lambda_1}} \int_{-\sigma_2\sqrt{\lambda_2}}^{\sigma_2\sqrt{\lambda_2}} ... \int_{-\sigma_t\sqrt{\lambda_k}}^{\sigma_t\sqrt{\lambda_k}} d\delta p_1\, d\delta p_2 ... \, d\delta p_k \\ &= 2^k \prod_{i=1}^{k} \sigma_i \sqrt{\lambda_i}.\end{aligned} \tag{14}$$

A.2 Linear Integral

$$\int \delta\mathbf{p}\, d\delta\mathbf{p} = \left(\int \delta p_1\, d\delta\mathbf{p}, ..., \int \delta p_k\, d\delta\mathbf{p} \right)^T. \tag{15}$$

Considering each equation:

$$\begin{aligned}\int \delta p_i\, d\delta\mathbf{p} &= \int_{-\sigma_1\sqrt{\lambda_1}}^{\sigma_1\sqrt{\lambda_1}} ... \int_{-\sigma_k\sqrt{\lambda_k}}^{\sigma_k\sqrt{\lambda_k}} \delta p_i\, d\delta p_1\, d\delta p_2 ... \, d\delta p_k \\ &= 0.\end{aligned} \tag{16}$$

So:

$$\int \delta\mathbf{p} d\delta\mathbf{p} = \Big(0, 0, ..., 0\Big)^T = \mathbf{0}_{k\times 1}. \tag{17}$$

A.3 Quadratic Integral

$$\int \delta\mathbf{p}\delta\mathbf{p}^T\, d\delta\mathbf{p} = \begin{pmatrix} \int \delta p_1^2\, d\delta\mathbf{p} & \int \delta p_1\delta p_2\, d\delta\mathbf{p} & \cdots & \int \delta p_1\delta p_k\, d\delta\mathbf{p} \\ \int \delta p_2\delta p_1\, d\delta\mathbf{p} & \int \delta p_2^2\, d\delta\mathbf{p} & \cdots & \int \delta p_2\delta p_k\, d\delta\mathbf{p} \\ \vdots & \vdots & \ddots & \vdots \\ \int \delta p_k\delta p_1\, d\delta\mathbf{p} & \int \delta p_k\delta p_2\, d\delta\mathbf{p} & \cdots & \int \delta p_k^2\, d\delta\mathbf{p} \end{pmatrix}, \tag{18}$$

where:

$$\int \delta p_i^2\, d\delta\mathbf{p} = \frac{1}{3}\sigma_i^2\lambda_i 2^k \prod_{j=1}^{k} \sigma_j\sqrt{\lambda_j} \tag{19}$$

$$\int \delta p_i\delta p_j\, d\delta\mathbf{p} = 0. \tag{20}$$

Then:

$$\int \delta\mathbf{p}\delta\mathbf{p}^T\, d\delta\mathbf{p} = 2^k \big(\prod_{i=1}^{k} \sigma_i\sqrt{\lambda_i} \big) \boldsymbol{\Lambda}(\boldsymbol{\sigma}). \tag{21}$$