Automatic Construction of Deformable Models In-The-Wild

Epameinondas Antonakos, Stefanos Zafeiriou
Department of Computing, Imperial College London
180 Queen's Gate, SW7 2AZ, London, U.K.
{e.antonakos, s.zafeiriou}@imperial.ac.uk

Abstract

Deformable objects are everywhere. Faces, cars, bicycles, chairs etc. Recently, there has been a wealth of research on training deformable models for object detection, part localization and recognition using annotated data. In order to train deformable models with good generalization ability, a large amount of carefully annotated data is required, which is a highly time consuming and costly task. We propose the first - to the best of our knowledge - method for automatic construction of deformable models using images captured in totally unconstrained conditions, recently referred to as "in-the-wild". The only requirements of the method are a crude bounding box object detector and a-priori knowledge of the object's shape (e.g. a point distribution model). The object detector can be as simple as the Viola-Jones algorithm (e.g. even the cheapest digital camera features a robust face detector). The 2D shape model can be created by using only a few shape examples with deformations. In our experiments on facial deformable models, we show that the proposed automatically built model not only performs well, but also outperforms discriminative models trained on carefully annotated data. To the best of our knowledge, this is the first time it is shown that an automatically constructed model can perform as well as methods trained directly on annotated data.

1. Introduction

Many deformable objects exist everywhere around us. Some examples are the human face and body, animals, vehicles such as cars and motorcycles and objects of everyday use like tables, chairs etc. Most of these objects consist of different parts and appear in instances with great variance in shape and appearance. Thus, the concept of a deformable object refers to the deformation of both shape and appearance of an object. For example cars have parts (i.e. doors, windows, tires etc.) with significant changes in shape, size and texture. Furthermore, faces consist of parts (i.e. nose, eyes, mouth etc.) which not only vary with respect to shape and appearance, but can demonstrate a number of expressions due to muscles. Recently, we have witnessed a great progress in object detection, alignment and recognition.

In order to train deformable models with good generalization ability, a large amount of carefully annotated data is needed. Developing useful datasets and benchmarks that can contribute in the progress of an application domain is a highly time consuming and costly procedure. It requires both careful selection of the images, so that they can model the vast amount of an object's variability, and careful annotation of the various parts of the object (or landmarks). The amount of annotation that is required depends on both the object and the application. In faces, for example, where many landmark points are needed in tasks such as facial expression analysis, motion capture and expression transfer, usually more than 60 points are annotated [3, 17, 25, 37]. To illustrate how much time consuming careful face annotation is, according to our experience, a trained annotator may need an average of 8 minutes per image for the manual annotation of 68 landmarks[1]. This means that the annotation of 1000 images requires a total of about 130 hours[2]. Furthermore, fatigue can cause errors on the accuracy and consistency of annotations and they may require correction.

In this paper, we deal with the problem of automatically constructing a robust deformable model using (1) a simple bounding box object detector and (2) a shape by means of a Point Distribution Model (PDM). The detector can be as simple as the Viola-Jones object detector [32][3] which returns only a bounding box of a detected object. Such detectors are widely employed in commercial products (e.g. even the cheapest digital camera has a robust face detector). Other detectors that can be used are efficient subwindow search [15] and deformable part-based models [37]. The annotations that are needed to train the object detector can be acquired very quickly, since only a bounding box con-

[1] This depends on many factors such as the image's illumination and resolution, the presence of occlusions and the face's pose and expression.

[2] It is very difficult to consecutively annotate for more than 4 hours.

[3] The newest versions of Matlab have incorporated a training procedure of Viola-Jones and it is extremely easy to train an object detector.

taining the object is required. Specifically, after selecting the images that are going to be used, the annotation procedure takes a couple of seconds per image. The statistical shape model can be created by using only 40-50 shape examples, which can be produced by either drawing possible shape variations of the 2D shape of the object or projecting 3D CAD model instances of the object on the 2D camera plane (such an example is shown in [38] for cars). Even the annotation of the shape examples is not a time consuming task, due to their small number. Furthermore, there are unsupervised techniques to learn the shape prior (model) directly from images [12, 14].

Due to the fact that manual annotation is a rather costly and labour-intensive procedure, unsupervised and semi-supervised learning of models for the tasks of alignment, landmark localization, tracking and recognition has attracted considerable attention [14, 13, 12, 19, 31, 28, 7, 2, 6, 23, 33, 9, 16, 11, 18, 36, 34]. In this paper, we propose a method to automatically construct deformable models for object alignment and the most related works are [14, 31, 2, 6, 23]. The related family of techniques, known as image-congealing [19, 18, 11, 16], uses implicit models to align a set of images as a whole, which means that both performing alignment to a new image and constructing a model is not straightforward. Our methodology differs from these works because we employ an explicit texture model which is learned through the process.

The two most closely related works to the proposed method are the automatic construction of Active Appearance Models (AAMs) in [2] and the so-called RASL methodology in [23] for person-specific face alignment. There are two main differences between our framework and [2]. (1) We use a predefined statistical shape model instead of trying to find both the shape and appearance models. We believe that with the current available optimization techniques, it is extremely difficult to simultaneously optimize for both the texture and shape parameters. (2) We employ the robust component analysis of [30] for the appearance which deals with outliers. Thus, even though our method is similar in concept to [2], these two differences make the problem feasible to solve. In particular, the methodology in [2] fails to create a generic model even in controlled recording conditions, due to extremely high dimensionality of the parameters to be found and to the sensitivity of the subspace method to outliers. This was probably one of the reasons why the authors demonstrate very limited and only person-specific experiments. Furthermore, our methodology bypasses some of the limitations of [23], which requires the presence of only one low-rank subspace, hence it has been shown to work only for the case of congealing images of a single person. Finally, we argue that in order for an automatically constructed AAM methodology to be robust to both within-class and out-of-class outliers[4], which cannot be avoided in totally unsupervised settings, statistical component analysis techniques should be employed [2]. We summarize our contributions as follows:

- We propose the first, to the best of our knowledge, methodology for automatic construction of both a generative and a discriminative AAM given only a dataset of images with the respective bounding boxes and a statistical shape model (PDM). Until recently, mainly due to the Project-Out Inverse Compositional (POIC) [20] algorithm, it was widely believed that AAMs do not possess good generalization capabilities [10, 27, 37]. We show that this is far from being true, demonstrating that even an AAM that is constructed fully automatically, not only performs well, but it outperforms some state-of-the-art discriminative methodologies trained on manually annotated data [1, 37]. Even though our method uses a similar texture model to [29], it is considerably different, since in that work an AAM is built using only annotated data, while our technique constructs the texture model in a fully automatic manner.

- We propose a discriminatively trained AAM methodology using the robust component analysis in [30]. Inspired by the recent success in applying a cascade of regressors [8, 35, 4, 26] to discriminatively learn a model for face alignment, we follow a similar line of research. The proposed discriminative AAM uses the robust component analysis [30] due to the fact it is trained on automatically annotated data, hence it needs to be robust to all kinds of outliers.

- Overall, the proposed methodology constructs a very powerful model, by iteratively training a generative fully automatically built AAM and then a discriminative AAM learned from the fitted shapes of the generative AAM.

We present experimental results on the task of face alignment. We choose this application because there exist many in-the-wild facial databases with a large number of images and annotated landmarks, hence solid quantitative evaluations can be performed.

2. AAM Automatic Construction In-The-Wild

Assuming the existence of a statistical shape model of an object (PDM), our method automatically trains a generative AAM and in extension a discriminative AAM, by only using a dataset of totally unconstrained in-the-wild images containing the object and the corresponding bounding

[4] Within-class outliers refer to outliers present in the image of an object such as occlusion. Out-of-class outliers refer to images of irrelevant objects or to background.

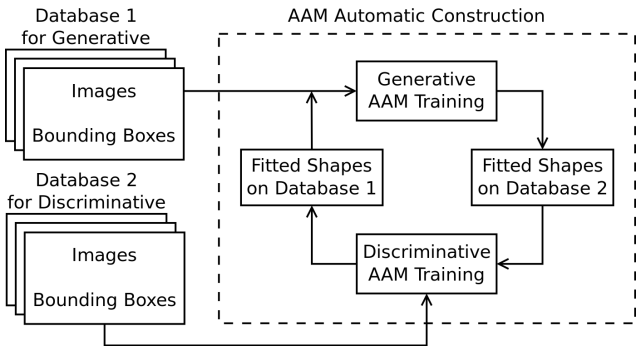

Figure 1: Automatic construction of deformable models. Given two sets of disjoint in-the-wild images and the object detector bounding boxes, our method automatically trains an AAM by training a generative and a discriminative model in an alternating manner.

boxes. This is achieved by alternatingly constructing a generative and a discriminative deformable model. At each iteration, the training of each of the two models utilizes the fitted shapes computed with the other already trained model. This iterative procedure is demonstrated in Fig. 1.

Specifically, we separate our set of images and the corresponding bounding boxes in two disjoint equally-sized datasets, referred to as the *generative* and the *discriminative* that are used for the training of the respective models. The first generative model is trained on the initial shapes extracted by initializing the PDM mean shape in the bounding boxes. At each iteration, the currently trained generative model is used to find the fitted shapes on the discriminative database's images. Then, a discriminative model is trained on these shapes. At the next iteration, the currently trained discriminative model is applied on the images of the generative database to extract the shapes estimations. A new version of the generative model is then trained based on these extracted shapes of the generative dataset. At the end of this iterative procedure, we train a final generative and discriminative AAM on the unified database of both datasets.

This alternating training of each model followed by the supply of updated shapes to the other and vice versa manages to continuously improve the fitted shapes, leading to more accurate models. The role of the discriminative model is crucial, as it moves the generative model from the local optimum that it stuck. Next, in Sec. 2.1 and 2.2 we present the generative and discriminative models respectively.

2.1. Automatic Construction of a Generative AAM

AAMs are deformable statistical models of shape and appearance that recover a parametric description of an object through optimization [5, 20]. A shape instance is denoted as a $2L_S \times 1$ vector $\mathbf{s} = [x_1, y_1, \ldots, x_{L_S}, y_{L_S}]^T$ with the coordinates of the L_S landmark points that correspond to the object's parts. The PDM *shape model* consists of an orthonormal basis of $4 + N_S$ eigenvectors $\mathbf{U}_S \in \mathbb{R}^{2L_S \times (4+N_S)}$ and the mean shape $\bar{\mathbf{s}}$. The first four eigenvectors correspond to the similarity transform that controls the global rotation, scaling and translation of the shape. Synthesis is achieved through linear combination of the eigenvectors weighted by the shape parameters $\mathbf{p} = [p_1, \ldots, p_{4+N_S}]^T$, thus

$$\mathbf{s_p} = \bar{\mathbf{s}} + \mathbf{U}_S \mathbf{p}$$

The warp function $\mathcal{W}(\mathbf{x}; \mathbf{p})$ represents the *motion model*. For each point $\mathbf{x}$ within a source shape it aims to find its corresponding location in the mean shape $\bar{\mathbf{s}}$ that plays the role of a reference template for aligning the appearance vectors.

In the generative AAM, we use a robust representation of appearance. Specifically, the appearance model is trained by employing the robust subspace analysis proposed in [30], which uses the image gradient orientations. Given an image $\mathbf{t}$ in vectorial form with size $L_A \times 1$, the so-called *normalized gradients* feature extraction function $\mathbf{g}(\mathbf{t})$ involves the computation of the image gradients $\mathbf{g}_x$, $\mathbf{g}_y$ and the corresponding gradient orientation $\boldsymbol{\phi} = \arctan(\mathbf{g}_y / \mathbf{g}_x)$ as

$$\mathbf{g}(\mathbf{t}) = \frac{1}{\sqrt{L_A}} [\cos\boldsymbol{\phi},\ \sin\boldsymbol{\phi}]^T \tag{1}$$

where $\cos\boldsymbol{\phi} = [\cos\boldsymbol{\phi}(1), \ldots, \cos\boldsymbol{\phi}(L_A)]$ and $\sin\boldsymbol{\phi} = [\sin\boldsymbol{\phi}(1), \ldots, \sin\boldsymbol{\phi}(L_A)]$. We denote the feature-based warped appearance vector as

$$\mathbf{a}(\mathbf{p}) \equiv \mathbf{g}\left(\mathbf{t}(\mathcal{W}(\mathbf{x}; \mathbf{p}))\right)$$

that has size $2L_A \times 1$, where L_A is the number of pixels inside the reference (i.e. mean) shape. An appearance model is then trained by performing Principal Component Analysis (PCA) on a set of training appearance vectors that results in a subspace of N_A eigenvectors $\mathbf{U}_A \in \mathbb{R}^{2L_A \times N_A}$ and the mean appearance $\bar{\mathbf{a}}$. This model can be used to synthesize shape-free texture instances, as $\mathbf{a}_{\boldsymbol{\lambda}} = \bar{\mathbf{a}} + \mathbf{U}_A \boldsymbol{\lambda}$, where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{N_A}]^T$ is the appearance parameters vector.

The employment of the robust kernel of Eq.1 has a key role in the successful performance of the proposed method, because it cancels-out both within-class and out-of-class outliers [30]. This is shown in the "toy" example of Fig. 2. In this experiment we have a dataset of 50 aligned face images. We replace 20% of these with the same baboon image and apply PCA on intensities and normalized gradients. Figure 2 shows that the PCA eigenvectors on intensities (top row) are corrupted with the baboon information. On the contrary, the employment of normalized gradients manages to separate the baboon information from the facial subspace and isolate it (second row). In our case, during the automatic training of the generative model, we expect to have both within-class and out-of-class outliers. Since the training images are captured in totally unconstrained conditions

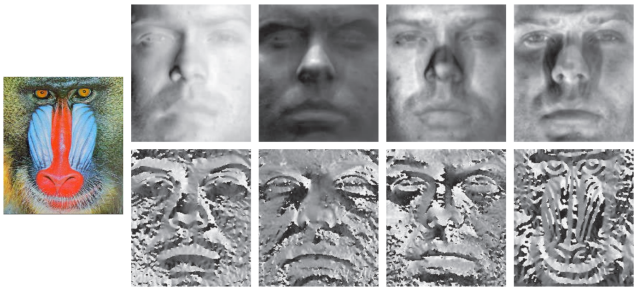

Figure 2: Robust kernel. Having a face dataset with 20% of the images replaced by the baboon, the top and bottom rows show 4 principal components of the PCA on intensities and normalized gradients respectively. Note that contrary to the normalized gradients subspace where the baboon is isolated, most intensities eigentextures are corrupted with the baboon.

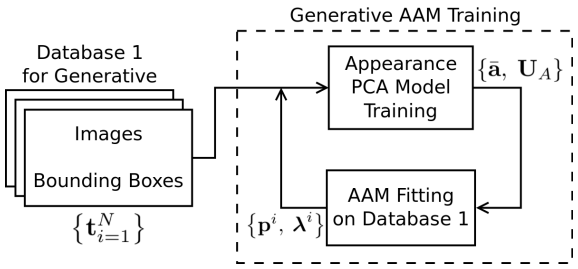

Figure 3: Automatic training of appearance model of Generative AAM. This diagram demonstrates the operation of Generative AAM Training step of Fig. 1. Given a set of images and the corresponding bounding boxes from the object detector, the method iteratively re-trains the appearance PCA model and re-performs AAM fitting on the images set to update the shapes.

(i.e. random images from the web), we expect many of them to have occluded objects, thus within-class outliers. Furthermore, in the cases where the fitted shape is either very inaccurate or even scrambled, the warped appearance consists an out-of-class outlier. However, the employment of the robust component analysis manages to remove such outliers from the appearance subspace.

2.1.1 Automatic Construction of Generative Appearance Model

In this section we present how to construct a robust generative AAM using only a shape prior and a set of images with initialization bounding boxes. We formulate an iterative optimization problem that aims to automatically construct an optimal generative appearance model that minimizes the mean AAM fitting ℓ_2^2 norm error over all given images. Specifically, given a set of N training images $\{\mathbf{t}^i\},\ i = 1, \ldots, N$ and a statistical shape model $\{\bar{\mathbf{s}}, \mathbf{U}_S\}$, we automatically train an AAM appearance model by iteratively solving

$$\begin{aligned} \underset{\bar{\mathbf{a}},\mathbf{U}_A,\mathbf{p}^i,\boldsymbol{\lambda}^i}{\operatorname{argmin}} \quad & \frac{1}{N}\sum_{i=1}^{N} \left\|\mathbf{a}^i(\mathbf{p}^i) - \bar{\mathbf{a}} - \mathbf{U}_A\boldsymbol{\lambda}^i\right\|^2 \\ \text{subject to} \quad & \mathbf{U}_A^T\mathbf{U}_A = \mathbf{I}_{eye} \end{aligned} \tag{2}$$

in order to find the optimal appearance subspace $\mathbf{U}_A$ and mean vector $\bar{\mathbf{a}}$ that minimize the mean ℓ_2^2 norm of the application of AAM fitting $(\mathbf{p}^i,\ \boldsymbol{\lambda}^i)$ over all images. $\mathbf{a}^i(\mathbf{p}^i)$ is the warped feature representation of the training image $\mathbf{t}^i$ and $\mathbf{I}_{eye}$ denotes the identity matrix. The explanation of this optimization procedure is visualized in Fig. 3. In brief, the algorithm iteratively trains a new PCA appearance model $\{\bar{\mathbf{a}}, \mathbf{U}_A\}$ based on the current estimate of the N shapes and then re-estimates the parameters $\{\mathbf{p}^i, \boldsymbol{\lambda}^i\},\ i = 1, \ldots, N$ by minimizing the ℓ_2^2 norm between each warped image and the appearance model instance. Consequently, the optimization is solved in two steps:

(a) Fix $\{\mathbf{p}^i, \boldsymbol{\lambda}^i\}$ and minimize w.r.t. $\{\bar{\mathbf{a}}, \mathbf{U}_A\}$ In this step we have a current estimate of $\{\mathbf{p}^i, \boldsymbol{\lambda}^i\}$ for each image $i = 1, \ldots, N$. From the shape parameters estimate we extract the warped feature-based image vectors $\{\mathbf{a}^i(\mathbf{p}^i)\}$ on which we train a new PCA appearance model $\{\bar{\mathbf{a}}, \mathbf{U}_A\}$. The updated subspace is orthogonal, thus $\mathbf{U}_A^T\mathbf{U}_A = \mathbf{I}_{eye}$. In this paper, we keep 150 eigenvectors per iteration.

(b) Fix $\{\bar{\mathbf{a}}, \mathbf{U}_A\}$ and minimize w.r.t. $\{\mathbf{p}^i, \boldsymbol{\lambda}^i\}$ In this step we have a currently trained statistical appearance model $\{\bar{\mathbf{a}}, \mathbf{U}_A\}$ and aim to estimate the shape and appearance parameters $\{\mathbf{p}^i, \boldsymbol{\lambda}^i\}$ for each image $i = 1, \ldots, N$ so that the ℓ_2^2 norm between each warped image and its reconstruction is minimized. Thus, we optimize

$$\underset{\mathbf{p}^i,\boldsymbol{\lambda}^i}{\operatorname{argmin}} \left\|\mathbf{a}^i(\mathbf{p}^i) - \bar{\mathbf{a}} - \mathbf{U}_A\boldsymbol{\lambda}^i\right\|^2,\ \forall i = 1, \ldots, N \tag{3}$$

This minimization can be solved with the efficient Gauss-Newton algorithm of Inverse Compositional Image Alignment (IC) [20]. Within the IC framework, Eq. 3 is written as $\operatorname{argmin}_{\mathbf{p}^i,\boldsymbol{\lambda}^i} \left\|\mathbf{a}^i(\mathbf{p}^i) - \mathbf{a}_{\boldsymbol{\lambda}^i}(\Delta\mathbf{p}^i)\right\|^2$ where $\mathbf{a}_{\boldsymbol{\lambda}^i} = \bar{\mathbf{a}} + \mathbf{U}_A\boldsymbol{\lambda}$ is the model instance and $\Delta\mathbf{p}^i$ is the increment used to inverse-compositionally update the shape parameters as $\mathcal{W}(\mathbf{x};\mathbf{p}^i) \leftarrow \mathcal{W}(\mathbf{x};\mathbf{p}^i) \circ \mathcal{W}(\mathbf{x};\Delta\mathbf{p}^i)^{-1}$. The two most commonly used IC optimization techniques are Project-Out IC (POIC) [20], where the shape and appearance parameters are decoupled and the Simultaneous IC (SIC) [10] where the optimization is done simultaneously for the shape and appearance parameters.

We instead perform IC, by optimizing separately for shape and appearance parameters in an alternating mode, similar to [21, 29]. At each iteration, we have a fixed estimate of $\mathbf{p}^i$ and compute the optimal appearance parameters

as the least-squares solution

$$\boldsymbol{\lambda}^i = \mathbf{U}_A^T \left[\mathbf{a}^i(\mathbf{p}^i) - \bar{\mathbf{a}}\right] \quad (4)$$

Then, given the current estimate of $\boldsymbol{\lambda}^i$ and taking the Taylor expansion around $\mathbf{p}^i = \mathbf{0}$, we solve for the shape increment

$$\Delta\mathbf{p}^i = -\left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\left[\mathbf{a}^i(\mathbf{p}^i) - \mathbf{a}_{\boldsymbol{\lambda}^i}\right]$$

where $\mathbf{J} = \nabla\mathbf{a}_{\boldsymbol{\lambda}^i}\frac{\partial\mathcal{W}}{\partial\mathbf{p}^i}$ is the Jacobian matrix with the steepest descent images as its columns. The algorithm requires the computation of the inverse Hessian matrix $\mathbf{H} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}$ and the current estimate of appearance parameters at each iteration which results in a total cost of $\mathcal{O}((N_A + N_S + 4)L_A + (4 + N_S)^2 L_A)$.

Even though the initial PCA model is expected to have many outliers, this optimization technique combined with the robust kernel of Eq. 1 iteratively results in an appearance model that eliminates the initial outliers. By keeping a small number of eigenvectors at each iteration, we ensure that the textures corresponding to inaccurate or scrambled shapes will not be included in our subspace. The convergence rate of this procedure is shown in Sec. 3.2.

A drawback of the optimization procedure is that it will stuck in a local minimum. In the following, in order to move the generative model from the local minimum, we will train a discriminative model using the already trained generative. We work under the assumption that the trained generative model is reliable enough to provide us with a sufficient number of good fittings in a new disjoint set. It is obvious that we need a disjoint set to train the discriminative model, since training it in the same dataset as the generative would result in overfitting.

2.2. Robust Discriminative AAM

Motivated by the recent application of a cascade of regressors [8, 35, 4, 26] to discriminatively learn a model for face alignment, we propose a parametric discriminatively trained AAM. Even though discriminatively trained AAMs have appeared before, the difference between our method and, for example [26], is that we use simple cascaded linear regression, as in [35], and the robust component analysis [30]. Note that other feature descriptors can also be used, such as HOG and SIFT. Intuitively, the goal of the discriminative model is to move the generative model from the local minimum that it converged in the previous iteration and boost it towards a better minimum. We automatically select the appearance vectors on which it is trained so that as few outliers as possible are included. This selection is achieved by keeping the textures with the best ℓ_2^2 norm fitting error.

2.2.1 Fitting Discriminative AAM

During the training procedure, the method aims to learn a number of K regression steps so that the initial shape parameters of all the training images converge to their groundtruth values. Each of these cascade solutions consists of a generic descent direction term $\mathbf{R}_k$ and a bias term $\mathbf{b}_k$. Given an unseen image, the fitting process involves K additive steps to find an updated vector of shape and similarity parameters

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{R}_{k-1}\boldsymbol{\lambda}_{k-1} + \mathbf{b}_{k-1},\ k = 1, \ldots, K \quad (5)$$

where the appearance parameters are retrieved from the inverse projection of the image's warped feature-based texture to a given appearance subspace as in Eq. 4. In the first step, the update $\Delta\mathbf{p}_1 = \mathbf{R}_0\boldsymbol{\lambda}_0 + \mathbf{b}_0$ is added to the initial parameters vectors as $\mathbf{p}_1 = \mathbf{p}_0 + \Delta\mathbf{p}_1$. The initial shape parameters vector $\mathbf{p}_0$ is computed from the image's bounding box, which practically initializes the rotation, translation and scaling values and leaves the rest equal to zero, thus $\mathbf{p}_0 = \left[p_0^1, \ldots, p_0^4, \mathbf{0}^{1:N_S}\right]^T$. The fitting algorithm has a real-time computational complexity of $\mathcal{O}((4 + N_S)(N_A + 2L_A))$ per iteration.

2.2.2 Training Discriminative AAM

Assume we have a set of N training images $\{\mathbf{t}^i\}$, $i = 1, \ldots, N$ and their groundtruth shapes $\{\mathbf{s}_{tr}^i\}$ which correspond to a set of parameters $\{\mathbf{p}_{tr}^i\}$. For each image in the database, we generate M different parameters initializations $\{\mathbf{p}_0^{i,j}\}$, $j = 1, \ldots, M$. This is done by sampling M different bounding boxes from a Normal distribution trained to describe the variance of various face detectors and retrieving the corresponding initialization shape parameters. To learn the sequence of generic descent directions and bias terms, we employ the Monte Carlo approximation of the ℓ_2^2-loss which results in solving the least-squares problem

$$\underset{\mathbf{R}_k, \mathbf{b}_k}{\operatorname{argmin}} \sum_{i=1}^{N}\sum_{j=1}^{M}\left\|\mathbf{p}_{tr}^i - \mathbf{p}_k^{i,j} - \mathbf{R}_k\boldsymbol{\lambda}_k^{i,j} - \mathbf{b}_k\right\|^2$$

for $k = 1, \ldots, K$. At each iteration and for each image, we update the parameters vector $\mathbf{p}_k^{i,j}$ using the rule of Eq. 5 and compute the current appearance parameters from Eq. 4.

3. Experimental Results

The proposed method of automatic AAM construction can be applied to any deformable object. However, we choose to present experimental results using a facial deformable model, because there are numerous in-the-wild, large and fully annotated facial databases that allow us to provide quantitative evaluation. After briefly presenting these databases (Sec. 3.1), we show the convergence of the automatic model construction (Sec. 3.2) and compare its performance with models trained on manually annotated images (Sec. 3.3).

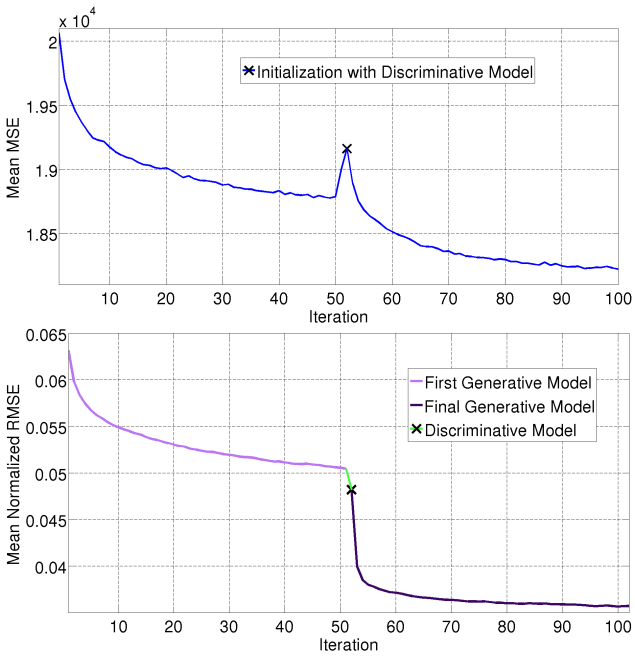

Figure 4: *Top:* Plot of the cost function per iteration. The marked point x denotes the beginning of the second iteration of the generative model. *Bottom:* Plot of the respective point-to-point normalized RMSE.

3.1. Databases

We automatically build our model using the images of the in-the-wild databases Labeled Face Parts in the Wild (LFPW) [3] and Helen [17]. They both consist of a trainset and testset with images downloaded from the web (e.g. Flickr) using simple text queries. The trainset/testset number of images is 810/224 and 2000/330 for LFPW and Helen respectively. We reserve the testing sets, along with the AFW database [37] that consists of 337 images, in order to compare our automatically trained model. All the above databases are manually annotated with a facial mask of 68 landmark points [24, 25] (annotations are available online).

3.2. Convergence of AAM Automatic Construction

Firstly, in order to create a facial shape PDM, we use 50 annotated images of the LFPW database, appropriately selected to demonstrate various deformations and expressions, and apply PCA. Note that one could also project shape instances of a statistical 3D shape model (e.g. [22]) to the 2D plane. Then, we automatically build a facial AAM with the proposed method (Fig. 1) using the images of LFPW and Helen training sets (2800 images in total). In order to perform the iteration between generative and discriminative model, we split these images in two equal disjoint subsets, each consisting of half of the images of each database, thus 405 and 1000 from LFPW and Helen respectively. We retrieve the bounding boxes by using Google Picasa's face detection.

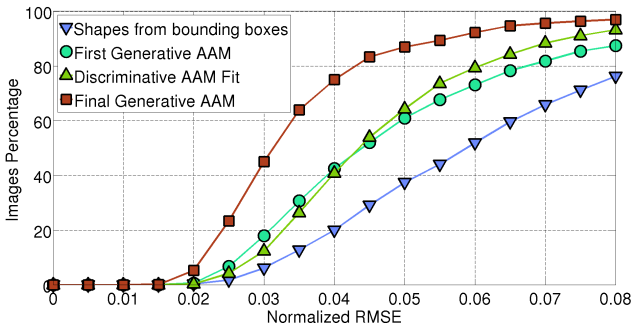

Figure 5: Automatic construction of AAM with a single application of the discriminative model. The plot shows the accuracy evolution of the generative database's shapes compared with their manual annotations.

We execute the overall proposed methodology for 2 iterations in total, which involves an iterative generative model automatic construction followed by a discriminative model and then the final automatic generative model. Our experiments show that the method converges quickly and only a single application of the discriminative model is sufficient to move the generative model to a satisfactory minimum. Figure 4 (top) plots the cost function vs. the number of iterations of the first generative model training on the generative database, the initialization with the first discriminative model (marked with an x) and the application of the final generative model. As can be seen the application of the discriminative step acts as a perturbation over the local optimum which in the end results to a better solution (similar to random perturbations in Simulated Annealing).

Furthermore, let us define the point-to-point RMSE measure normalized with respect to the face size. Specifically, denoting $\mathbf{s}^f$ and $\mathbf{s}^g$ the fitted and groundtruth shapes respectively, the normalized RMSE between them is $\text{RMSE} = \frac{\sum_{i=1}^{L_S} \sqrt{(x_i^f - x_i^g)^2 + (y_i^f - y_i^g)^2}}{L_S d}$ where $d = (\max_x \mathbf{s}^g - \min_x \mathbf{s}^g + \max_y \mathbf{s}^g - \min_y \mathbf{s}^g)/2$.

Figure 4 (bottom) plots the normalized RMSE over the number of iterations for the generative database. As can be seen, it monotonically decreases. Furthermore, in Fig. 5 we demonstrate the evolution of the fitting curves of the generative database's shapes during this training procedure compared with the manually annotated shapes.

Figure 6 demonstrates the respective evolution of the mean appearance and the three most important eigenvectors. The last row demonstrates the subspace obtained from the PCA on the manual annotations of the generative database. The figure shows that the resulting facial appearance subspace gradually improves and isolates the outliers as expected, due to the employment of the robust compo-

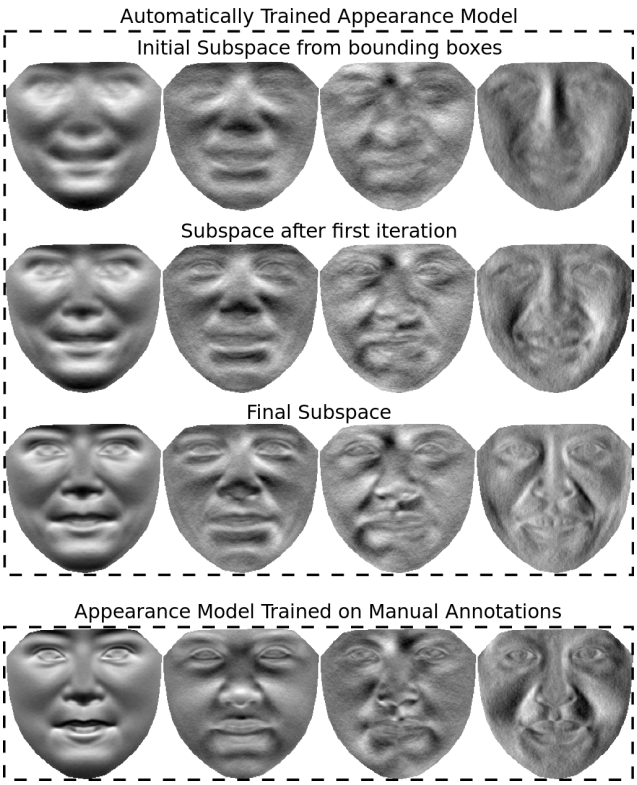

Figure 6: Automatic construction of AAM with a single application of the discriminative model. Visualization of the mean appearance and the three most important eigenvectors for the iterative automatically constructed AAM (top) and the AAM trained on manual annotations (bottom).

nent analysis. This is highlighted by the fact that the facial parts (eyes, nose, mouth etc.) can be distinguished more clearly in the final eigentextures, as opposed to the initial ones. The resulting appearance subspace is very similar to the annotations-based one, even though we performed only two iterations.

3.3. Comparison with Models Trained on Manual Annotations

After completing the iterations demonstrated in Figs. 4 and 6, we train a final generative and discriminative model on the 2800 images of the union of both datasets. We compare the performance of our model with the state-of-the-art method of Robust Discriminative Response Map Fitting (DRMF) for Constrained Local Models [1] and the Deformable Part-Based Models [37]. For both methods, we use the implementation provided by their authors, along with the pre-built models which are discriminatively trained on the manual annotations of much larger datasets than LFPW and Helen datasets. Moreover, we compare with the generative and discriminative AAMs trained on the manual annotations of LFPW and Helen trainsets. Figure 7 shows the normalized RMSE curves on AFW and the union of LFPW and Helen testsets. Note that in both cases, we use Google Picasa's face detection to extract the bounding boxes that initialize the translation and scaling of the mean shape. The results show that our automatically trained models have a very good performance and greatly outperform the discriminative ones trained on manual annotations. See the supplementary material for qualitative fitting results.

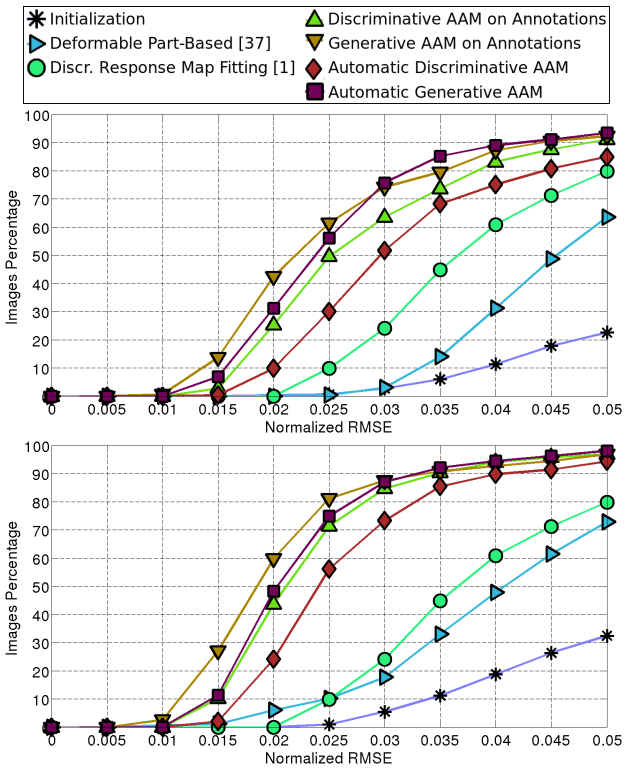

Figure 7: Comparison of automatically constructed deformable models (generative and discriminative) with other models trained on manual annotations. *Top:* AFW database. *Bottom:* LFPW and Helen testing databases.

4. Conclusions

We propose a method for automatic construction of deformable models. The method iteratively trains a generative and a discriminative AAM ending up with a powerful model. The only requirements of the method are a statistical shape model and a set of in-the-wild images with their bounding boxes, which means that it can be applied to any object. Our experiments on faces show that the method outperforms discriminative state-of-the-art methods trained on manual annotations. This is the first, to the best of our knowledge, methodology to automatically building a deformable model that demonstrates such promising results.

5. Acknowledgement

The work of Epameinondas Antonakos and Stefanos Zafeiriou was partially funded by the EPSRC project EP/J017787/1 (4DFAB).

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *IEEE CVPR*, 2013. 2, 7

[2] S. Baker, I. Matthews, and J. Schneider. Automatic construction of active appearance models as an image coding problem. *IEEE TPAMI*, 26(10):1380–1384, 2004. 2

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *IEEE CVPR*, 2011. 1, 6

[4] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *IEEE CVPR*, 2012. 2, 5

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

[6] T. F. Cootes, S. Marsland, C. J. Twining, K. Smith, and C. J. Taylor. Groupwise diffeomorphic non-rigid registration for automatic model building. In *ECCV*. 2004. 2

[7] M. Cox, S. Sridharan, S. Lucey, and J. Cohn. Least squares congealing for unsupervised alignment of images. In *IEEE CVPR*, 2008. 2

[8] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *IEEE CVPR*, 2010. 2, 5

[9] B. J. Frey, M. Jojic, and A. Kannan. Learning appearance and transparency manifolds of occluded objects in layers. In *IEEE CVPR*, 2003. 2

[10] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *IMAVIS*, 23(12):1080–1093, 2005. 2, 4

[11] G. B. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. In *NIPS*, 2012. 2

[12] T. Jiang, F. Jurie, and C. Schmid. Learning shape prior models for object matching. In *IEEE CVPR*, 2009. 2

[13] N. Jojic, J. Winn, and L. Zitnick. Escaping local minima through hierarchical model selection: Automatic object discovery, segmentation, and tracking in video. In *IEEE CVPR*, 2006. 2

[14] I. Kokkinos and A. Yuille. Unsupervised learning of object deformation models. In *IEEE ICCV*, 2007. 2

[15] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE TPAMI*, 31(12):2129–2142, 2009. 1

[16] J. Lankinen and J.-K. Kämäräinen. Local feature based unsupervised alignment of object class images. In *BMVC*, 2011. 2

[17] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*. 2012. 1, 6

[18] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE TPAMI*, 28(2):236–250, 2006. 2

[19] X. Liu, Y. Tong, and F. W. Wheeler. Simultaneous alignment and clustering for an image ensemble. In *IEEE ICCV*, 2009. 2

[20] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004. 2, 3, 4

[21] G. Papandreou and P. Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *IEEE CVPR*, 2008. 4

[22] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *IEEE AVSS*, 2009. 6

[23] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE TPAMI*, 34(11):2233–2246, 2012. 2

[24] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *IEEE ICCV'W*, 2013. 6

[25] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *IEEE CVPR'W*, 2013. 1, 6

[26] J. Saragih and R. Goecke. Iterative error bound minimisation for aam alignment. In *IEEE ICPR*, 2006. 2, 5

[27] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011. 2

[28] Y. Tong, X. Liu, F. W. Wheeler, and P. H. Tu. Semi-supervised facial landmark annotation. *CVIU*, 116(8):922–935, 2012. 2

[29] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic. Generic active appearance models revisited. In *ACCV*. 2013. 2, 4

[30] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Subspace learning from image gradient orientations. *IEEE TPAMI*, 34(12):2454–2466, 2012. 2, 3, 5

[31] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE CVPR*, 1997. 2

[32] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, 2001. 1

[33] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. *ECCV*, 2000. 2

[34] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *IEEE ICCV*, 2005. 2

[35] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE CVPR*, 2013. 2, 5

[36] S. C. Zhu and A. L. Yuille. Forms: a flexible object recognition and modelling system. *IJCV*, 20(3):187–212, 1996. 2

[37] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE CVPR*, 2012. 1, 2, 6, 7

[38] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Revisiting 3d geometric models for accurate object shape and pose. In *IEEE ICCV'W*, 2011. 2