Active Pictorial Structures

Epameinondas Antonakos Joan Alabort-i-Medina Stefanos Zafeiriou
Department of Computing, Imperial College London
180 Queens Gate, SW7 2AZ, London, U.K.
{e.antonakos, ja310, s.zafeiriou}@imperial.ac.uk

Abstract

In this paper we present a novel generative deformable model motivated by Pictorial Structures (PS) and Active Appearance Models (AAMs) for object alignment in-the-wild. Inspired by the tree structure used in PS, the proposed Active Pictorial Structures (APS)[1] model the appearance of the object using multiple graph-based pairwise normal distributions (Gaussian Markov Random Field) between the patches extracted from the regions around adjacent landmarks. We show that this formulation is more accurate than using a single multivariate distribution (Principal Component Analysis) as commonly done in the literature. APS employ a weighted inverse compositional Gauss-Newton optimization with fixed Jacobian and Hessian that achieves close to real-time performance and state-of-the-art results. Finally, APS have a spring-like graph-based deformation prior term that makes them robust to bad initializations. We present extensive experiments on the task of face alignment, showing that APS outperform current state-of-the-art methods. To the best of our knowledge, the proposed method is the first weighted inverse compositional technique that proves to be so accurate and efficient at the same time.

1. Introduction

The task of object alignment in terms of landmark points localization under unconstrained conditions is among the most challenging problems in the field of Computer Vision. Such challenging conditions are usually referred to as "in-the-wild". Ongoing research efforts on generic Deformable Models aim to provide robust and accurate techniques that perform in real-time. Such methodologies can have an important impact in human-computer interaction applications, such as multimodal interaction, entertainment, security etc.

One of the most well-studied deformable models are Active Appearance Models (AAMs) [7, 19]. AAMs are statistical generative models of the shape and appearance of an object. The shape model, usually referred to as Point Distribution Model (PDM), is built by applying Principal Component Analysis (PCA) on a set of aligned shapes. Similarly, the appearance model is built by applying PCA on a set of shape-free appearance instances, acquired by warping the training images into a reference shape. AAMs represent the appearance in a holistic/global way, i.e. the whole texture is taken into account. Fitting AAMs involves solving a non-linear least squares problem and it is typically solved using a variant of the Gauss-Newton algorithm [5]. The Simultaneous [14] and Alternating [20, 27] inverse compositional algorithms have proved to be very accurate. They can achieve state-of-the-art performance when combined with powerful features [4]. The Project-Out inverse compositional (POIC) [19] algorithm has a real-time complexity but is inaccurate, which makes it unsuitable for generic settings. Therefore, AAMs have two disadvantages: (1) they are slow and inappropriate for real-time applications, and (2) by employing PCA the appearance of the object is modelled with a single multivariate normal distribution, which, as it will be shown in this paper, restricts the fitting accuracy (Fig. 1).

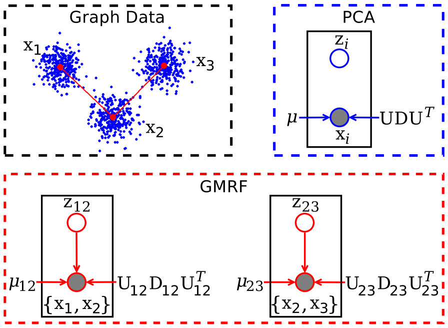


Figure 1: A simple visualization motivating the main idea behind APS[1]. We propose to model the appearance of an object using multiple pairwise distributions based on the edges of a graph (GMRF) and show that this outperforms the commonly used PCA model under an inverse Gauss-Newton optimization framework.

[1] An open-source implementation of Active Pictorial Structures is available within the Menpo Project [1] in `http://www.menpo.org/`.

Mainly due to the high complexity when using a holistic appearance representation, many existing methods employ a part-based one. This means that a local patch is extracted from the neighbourhood around each landmark. Among the most important part-based deformable models are Pictorial Structures (PS) [13, 12, 3], their discriminative descendant Deformable Part Model (DPM) [10, 33] and their extensions like Deformable Structures [34]. PS learn a patch expert for each part and model the shape of the object using spring-like connections between parts based on a tree structure. Thus, a different distribution is assumed for each pair of parts connected with an edge, as opposed to the PCA shape model of AAMs that assumes a single multivariate normal distribution for all parts. The optimization aims to find a tree-based shape configuration for which the patch experts have a minimum cost and is performed using a dynamic programming algorithm based on the distance transform [11]. PS are successfully used for various tasks, such as human pose estimation [32] and face detection [33, 18]. Their biggest advantage is that they find the global optimum, thus they are not dependent neither require initialization. However, in practice, PS have two important disadvantages: (1) inference is very slow, and (2) because the tree structure restricts too much the range of possible realizable shape configurations, the global optimum, even though it is the best solution in the span of the model, it does not always correspond to the shape that best describes the object in reality.

The method proposed in this paper takes advantage of the strengths, and overcomes the disadvantages, of both AAMs and PS. We are motivated by the tree-based structure of PS and we further expand on this concept. Our model can formulate the relations between parts using any graph structure; not only trees. From AAMs we borrow the use of the Gauss-Newton algorihtm in combination with a statistical shape model. Our weighted inverse compositional algorithm with fixed Jacobian and Hessian provides close to real-time cost with state-of-the-art performance. Thus, the proposed model shares characteristics from both AAMs and PS, hence the name Active Pictorial Structures (APS)[1].

Apart from PS and DPM, other important part-based techniques exist in literature. For example, Constrained Local Models (CLMs) [9, 30, 25] and their predecessors Active Shape Models (ASMs) [8] both use a statistical shape model (PDM) and learn a classifier for each part's appearance. Supervised Descent Method [31], which is among the most successful techniques, learns a cascade of consecutive regression steps between the shape coordinates and the feature-based appearance extracted from each part. The recently proposed regression-based methods in [21, 16] also report very accurate and extremely fast performance, but they do not provide publicly available code. The discriminative nature of these techniques indicates that they need loads of training data in order to perform well. This is opposite to the generative nature of APS that require much fewer training examples. The idea of substituting the PCA shape model with a piece-wise linear model has also been proposed for 3D facial models in [26]. The most closely related method to the proposed APS is the Gauss-Newton Deformable Part Model (GN-DPM) [29]. It is a part-based AAM that takes advantage of the efficient inverse alternating Gauss-Newton technique proposed in [28] and reports very accurate performance. The two most important differences between the proposed APS and GN-DPM are that: (1) APS do not model the appearance of an object using PCA but assume a different distribution for each pair of connected parts that proves to perform better, (2) APS employ a weighted inverse compositional algorithm with fixed Jacobian and Hessian, which is by definition at least an order of magnitude faster than the alternating one.

In summary, the contributions of this paper are:

- The proposed model combines the advantages of PS (graph-based relations between parts) and AAMs (weighted inverse Gauss-Newton optimization with statistical shape model).

- We show that it is more accurate to model the appearance of an object with multiple graph-based normal distributions, thus using a Gaussian Markov Random Field [22] structure, rather than a single multidimensional normal distribution (PCA), as is commonly done in literature. We also prove that this is not beneficial for modelling an object's shape, because the resulting covariance matrix has high rank and the shape subspace has too many dimensions to be optimized. We also show that employing a tree structure for the shape model, as done in PS [12, 10, 33], limits the model's descriptiveness and hampers the performance.

- We use the spring-like shape model of PS and DPM as a shape prior in the Gauss-Newton optimization. This deformation term makes the model more robust as it manages to restrict non-realistic instances of the object's shape.

- We propose, to the best of our knowledge, the best performing weighted inverse compositional Gauss-Newton algorithm with fixed Jacobian and Hessian. As it will be shown, its computational cost reduces to a single matrix multiplication per iteration and is independent of the employed graph structure. We test the proposed method on the task of face alignment, because of the plethora of annotated facial data. However, it can also be applied to other objects, such as eyes, cars etc. Our experiments show that APS outperform the current state-of-the-art methods.

2. Method

In the following, we denote vectors by small bold letters, matrices by capital bold letters, functions by capital calligraphic letters and scalars by small regular-font letters.

2.1. Shape and appearance representation

In the problem of object alignment in-the-wild, the shape of the object is described using n landmark points that are usually located on semantic parts of the object. Let us denote the coordinates of a point within the Cartesian space of an image $\mathbf{I}$ as the 2×1 vector $\boldsymbol{\ell} = [x, y]^T$. A sparse *shape instance* of the object is defined as the $2n \times 1$ vector

$$\mathbf{s} = [\boldsymbol{\ell}_1^T, \ldots, \boldsymbol{\ell}_n^T]^T = [x_1, y_1, \ldots, x_n, y_n]^T \tag{1}$$

The relative location of a landmark point i with respect to a landmark point j is defined as

$$d\boldsymbol{\ell}_{ij} = \boldsymbol{\ell}_i - \boldsymbol{\ell}_j = [x_i - x_j, y_i - y_j]^T \tag{2}$$

Furthermore, let us denote an image patch of size $h \times w$ corresponding to the image location $\boldsymbol{\ell}_i$ in vectorized form as

$$\mathbf{t}_{\boldsymbol{\ell}_i} = [\mathbf{I}(\mathbf{z}_1), \mathbf{I}(\mathbf{z}_2), \ldots, \mathbf{I}(\mathbf{z}_{hw})]^T, \ \{\mathbf{z}_i\}_{i=1}^{hw} \in \boldsymbol{\Omega}_{\boldsymbol{\ell}_i} \tag{3}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\ell}_i}$ is a set of discrete neighbouring pixel locations $\mathbf{z}_i = [x_i, y_i]^T$ within a rectangular region centered at location $\boldsymbol{\ell}_i$ and hw is the image patch vector's length. Moreover, we define $\mathcal{H} : \mathbb{R}^{hw} \rightarrow \mathbb{R}^m$ to be a feature extraction function, which computes a descriptor vector of length m (e.g. SIFT [17]) given an appearance vector. We denote the procedure of extracting a feature-based vector from a patch centred at a given image location by the function

$$\begin{aligned} \mathcal{F}(\boldsymbol{\ell}_i) &= \mathcal{H}(\mathbf{t}_{\boldsymbol{\ell}_i}) = \\ &= \mathcal{H}\left([\mathbf{I}(\mathbf{z}_1), \ldots, \mathbf{I}(\mathbf{z}_k)]^T\right), \ \{\mathbf{z}_i\}_{i=1}^{k} \in \boldsymbol{\Omega}_{\boldsymbol{\ell}_i} \end{aligned} \tag{4}$$

Finally, we define the function

$$\mathcal{A}(\mathbf{s}) = [\mathcal{F}(\boldsymbol{\ell}_1)^T, \mathcal{F}(\boldsymbol{\ell}_2)^T, \ldots, \mathcal{F}(\boldsymbol{\ell}_n)^T]^T \tag{5}$$

which concatenates all the vectorized feature-based image patches corresponding to the n landmarks of a shape in a vector of length mn.

2.2. Graphical model

Let us define an undirected graph between the n landmark points of an object as $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n vertexes and there is an edge $(v_i, v_j) \in E$ for each pair of connected landmark points. Moreover, let us assume that we have a set of random variables $X = \{X_i\}, \forall i : v_i \in V$ which represent an abstract feature vector of length k extracted from each vertex v_i, i.e. $\mathbf{x}_i, i : v_i \in V$ (e.g. the location coordinates, appearance vector etc.). We model the likelihood probability of two random variables that correspond to connected vertexes with a normal distribution

$$\begin{aligned} p(X_i = \mathbf{x}_i, X_j = \mathbf{x}_j | G) \sim \mathcal{N}(\boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij}), \\ \forall i, j : (v_i, v_j) \in E \end{aligned} \tag{6}$$

where $\boldsymbol{\mu}_{ij}$ is the $2k \times 1$ mean vector and $\boldsymbol{\Sigma}_{ij}$ is the $2k \times 2k$ covariance matrix. Consequently, the cost of observing a set of feature vectors $\{\mathbf{x}_i\}, \forall i : v_i \in V$ can be computed using a Mahalanobis distance per edge, i.e.

$$\sum_{\forall i,j:(v_i,v_j) \in E} \left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} - \boldsymbol{\mu}_{ij} \right)^T \boldsymbol{\Sigma}_{ij}^{-1} \left(\begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \end{bmatrix} - \boldsymbol{\mu}_{ij} \right) \tag{7}$$

In practice, the computational cost of computing Eq. 7 is too expensive because it requires looping over all the graph's edges. Especially in the case of a complete graph, it makes it impossible to perform inference in real time.

Inference can be much faster if we convert this cost to an equivalent matrical form as

$$(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{8}$$

This is equivalent to modelling the set of random variables X with a Gaussian Markov Random Field (GMRF) [22]. A GMRF is described by an undirected graph, where the vertexes stand for random variables and the edges impose statistical constraints on these random variables. Thus, the GMRF models the set of random variables with a multivariate normal distribution

$$p(X = \mathbf{x} | G) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{9}$$

where $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^T, \ldots, \boldsymbol{\mu}_n^T]^T = [E(X_1)^T, \ldots, E(X_n)^T]^T$ is the $nk \times 1$ mean vector and $\boldsymbol{\Sigma}$ is the $nk \times nk$ overall covariance matrix. We denote by $\mathbf{Q}$ the block-sparse precision matrix that is the inverse of the covariance matrix, i.e. $\mathbf{Q} = \boldsymbol{\Sigma}^{-1}$. By applying the GMRF we make the assumption that the random variables satisfy the three Markov properties (pairwise, local and global) and that the blocks of the precision matrix that correspond to disjoint vertexes are zero, i.e. $\mathbf{Q}_{ij} = \mathbf{0}_{k \times k}, \forall i, j : (v_i, v_j) \notin E$. By defining $\mathcal{G}_i = \{(i-1)k + 1, (i-1)k + 2, \ldots, ik\}$ to be a set of indices for sampling a matrix, we can prove that the structure of the precision matrix is

$$\mathbf{Q} = \begin{cases} \displaystyle\sum_{\forall j:(v_i,v_j) \in E} \boldsymbol{\Sigma}_{ij}^{-1}(\mathcal{G}_1, \mathcal{G}_1) + & \\ \displaystyle\sum_{\forall j:(v_j,v_i) \in E} \boldsymbol{\Sigma}_{ji}^{-1}(\mathcal{G}_2, \mathcal{G}_2), \ \forall v_i \in V, & \text{at } (\mathcal{G}_i, \mathcal{G}_i) \\ \boldsymbol{\Sigma}_{ij}^{-1}(\mathcal{G}_1, \mathcal{G}_2), \ \forall i, j : (v_i, v_j) \in E, & \text{at } (\mathcal{G}_i, \mathcal{G}_j) \\ & \text{and } (\mathcal{G}_j, \mathcal{G}_i) \\ 0, & \text{elsewhere} \end{cases} \tag{10}$$

Using the same assumptions and given a directed graph (cyclic or acyclic) $G = (V, E)$, where $(v_i, v_j) \in E$ denotes the relation of v_i being the parent of v_j, we can show that

$$(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{Q} (\mathbf{x} - \boldsymbol{\mu}) = \\ = \sum_{\forall i,j:(v_i,v_j)\in E} \left(\mathbf{x}_i - \mathbf{x}_j - \boldsymbol{\mu}_{ij}\right)^T \boldsymbol{\Sigma}_{ij}^{-1} \left(\mathbf{x}_i - \mathbf{x}_j - \boldsymbol{\mu}_{ij}\right) \tag{11}$$

is true if

$$\mathbf{Q} = \begin{cases} \sum\limits_{\forall j:(v_i,v_j)\in E} \boldsymbol{\Sigma}_{ij}^{-1} + & \\ \sum\limits_{\forall j:(v_j,v_i)\in E} \boldsymbol{\Sigma}_{ji}^{-1},\ \forall v_i \in V, & \text{at } (\mathcal{G}_i, \mathcal{G}_i) \\ -\boldsymbol{\Sigma}_{ij}^{-1},\ \forall i,j:(v_i,v_j)\in E, & \text{at } (\mathcal{G}_i, \mathcal{G}_j) \\ & \text{and } (\mathcal{G}_j, \mathcal{G}_i) \\ 0, & \text{elsewhere} \end{cases} \tag{12}$$

where $\boldsymbol{\mu}_{ij} = E(X_i - X_j)$ and $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^T, \dots, \boldsymbol{\mu}_n^T]^T = [E(X_1)^T, \dots, E(X_n)^T]^T$. In this case, if G is a tree, then we have a Bayesian network. Please refer to the supplementary material for detailed proofs of Eqs. 10 and 12.

2.3. Model training

APS differ from most existing generative object alignment methods because they assume a GMRF structure in order to model the appearance and the deformation of an object. As we show in the experiments, this assumption is the key that makes the proposed method efficient and accurate.

Shape model APS use a statistical shape model built using PCA, similar to the PDM employed in most existing parametric methods such as AAMs, CLMs and GN-DPMs. The procedure involves the alignment of the training shapes with respect to their rotation, translation and scaling (similarity transform) using Procrustes analysis, the subtraction of the mean shape and the application of PCA. We further augment the acquired subspace with four eigenvectors that control the global similarity transform of the object, re-orthonormalize [19] and keep the first n_S eigenvectors. Thus, we end up with a linear shape model $\{\bar{\mathbf{s}}, \mathbf{U} \in \mathbb{R}^{2n \times n_S}\}$, where $\bar{\mathbf{s}} = \left[E(\boldsymbol{\ell}_1)^T, \dots, E(\boldsymbol{\ell}_n)^T\right]^T$ is the $2n \times 1$ mean shape vector and $\mathbf{U}$ denotes the orthonormal basis.

We define a function $\mathcal{S} \in \mathbb{R}^{2n}$ that generates a shape instance given the linear model's basis, an input shape and a parameters' vector (weights) as

$$\mathcal{S}(\mathbf{U}, \mathbf{s}, \mathbf{p}) = \mathbf{s} + \mathbf{U}\mathbf{p} \tag{13}$$

where $\mathbf{p} = [p_1, p_2, \dots, p_{n_S}]^T$ are the parameters' values. Similarly, we define the set of functions $\mathcal{S}_i \in \mathbb{R}^2$, $\forall i = 1, \dots, n$ that return the coordinates of the i^{th} landmark of the shape instance as

$$\mathcal{S}_i(\mathbf{U}, \mathbf{s}, \mathbf{p}) = \mathbf{s}_{2i-1,2i} + \mathbf{U}_{2i-1,2i}\mathbf{p},\ \forall i = 1, \dots, n \tag{14}$$

where $\mathbf{s}_{2i-1,2i}$ denotes the coordinates' vector of the i^{th} landmark point, i.e. $\boldsymbol{\ell}_i = [x_i, y_i]^T$, and $\mathbf{U}_{2i-1,2i}$ denotes the $2i - 1$ and $2i$ row vectors of the shape subspace $\mathbf{U}$. Note that from now onwards, for simplicity, we will write $\mathcal{S}(\mathbf{s}, \mathbf{p})$ and $\mathcal{S}_i(\mathbf{s}, \mathbf{p})$ instead of $\mathcal{S}(\mathbf{U}, \mathbf{s}, \mathbf{p})$ and $\mathcal{S}_i(\mathbf{U}, \mathbf{s}, \mathbf{p})$ respectively.

Another way to build the shape model is by using the GMRF structure (Fig. 1). Specifically, given an undirected graph $G^s = (V^s, E^s)$ and assuming that the pairwise locations' vector of two connected landmarks follows a normal distribution as in Eq. 6, i.e. $[\boldsymbol{\ell}_i^T, \boldsymbol{\ell}_j^T]^T \sim \mathcal{N}(\boldsymbol{\mu}_{ij}^s, \boldsymbol{\Sigma}_{ij}^s)$, $\forall i,j : (v_i^s, v_j^s) \in E^s$, we formulate a GMRF. Following Eq. 9 and using the shape vector of Eq. 1, this can be expressed as

$$p(\mathbf{s}|G^s) \sim \mathcal{N}(\bar{\mathbf{s}}, \boldsymbol{\Sigma}^s) \tag{15}$$

where the precision matrix $\mathbf{Q}^s$ is structured as shown in Eq. 10 with $\mathbf{x}_i = \boldsymbol{\ell}_i$ and $k = 2$. Then, after constructing the precision matrix, we can invert it and apply PCA on the resulting covariance matrix $\boldsymbol{\Sigma}^s = (\mathbf{Q}^s)^{-1}$ in order to obtain a linear shape model. Even though, as we show below, the GMRF-based modelling creates a more powerful appearance model representation, it does not do the same for the shape model. Our experiments suggest that the single Gaussian PCA shape model is more beneficial than any other model that assumes a GMRF structure. This can be explained by the fact that $\boldsymbol{\Sigma}^s$ ends up having a high rank, especially if G^s has many edges. As a result, most of its eigenvectors correspond to non-zero eigenvalues and they express a small percentage of the whole data variance. This means that during fitting we need to employ a large number of eigenvectors ($n_S \approx 2n$), much more than in the case of a single multivariate distribution, which makes the Gauss-Newton optimization very unstable and ineffective.

Appearance model In most AAM-like formulations, the appearance model is built by warping all textures to a reference frame, vectorizing and building the PCA model. In this work, we propose to model the appearance of an object using a GMRF graphical model, as presented in Sec. 2.2. In contrast to the shape model case, the GMRF-based appearance model is more powerful than its PCA counterpart. Specifically, given an undirected graph $G^a = (V^a, E^a)$ and assuming that the concatenation of the appearance vectors of two connected landmarks can be described by a normal distribution (Eq. 6), i.e. $\left[\mathcal{F}(\boldsymbol{\ell}_i)^T, \mathcal{F}(\boldsymbol{\ell}_j)^T\right]^T \sim \mathcal{N}(\boldsymbol{\mu}_{ij}^a, \boldsymbol{\Sigma}_{ij}^a)$, $\forall i,j : (v_i^a, v_j^a) \in E^a$, we form a GMRF that, using Eq. 5, can be expressed as

$$p(\mathcal{A}(\mathbf{s})|G^a) \sim \mathcal{N}(\bar{\mathbf{a}}, \boldsymbol{\Sigma}^a) \tag{16}$$

where $\bar{\mathbf{a}} = \left[E(\mathcal{F}(\boldsymbol{\ell}_1))^T, \ldots, E(\mathcal{F}(\boldsymbol{\ell}_n))^T\right]^T$ is the $mn \times 1$ mean appearance vector and $\mathbf{Q}^a = (\boldsymbol{\Sigma}^a)^{-1}$ is the $mn \times mn$ precision matrix that is structured as shown in Eq. 10 with $\mathbf{x}_i = \mathcal{F}(\boldsymbol{\ell}_i)$ and $k = m$. During the training of the appearance model, we utilize the low rank representation of each edgewise covariance matrix $\boldsymbol{\Sigma}^a_{ij}$ by using the first n_A singular values of its SVD factorization. Given $\bar{\mathbf{a}}$ and $\mathbf{Q}^a$, the cost of an observed appearance vector $\mathcal{A}(\mathbf{s})$ corresponding to a shape instance $\mathbf{s} = \mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})$ in an image is

$$\begin{aligned} &\|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} = \\ =&\left[\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\right]^T \mathbf{Q}^a \left[\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\right] \end{aligned} \tag{17}$$

Our experiments show that all the tested GMRF-based appearance models greatly outperform the PCA-based one.

Deformation prior Apart from the shape and appearance models, we also employ a deformation prior that is similar to the deformation models used in [12, 33]. Specifically, we define a directed (cyclic or acyclic) graph between the landmark points as $G^d = (V^d, E^d)$ and model the relative locations between the parent and child of each edge with the GMRF of Eq. 11. We assume that the relative location between the vertexes of each edge, as defined in Eq. 2, follows a normal distribution $\boldsymbol{\ell}_i - \boldsymbol{\ell}_j \sim \mathcal{N}(\boldsymbol{\mu}^d_{ij}, \boldsymbol{\Sigma}^d_{ij}), \ \forall (i,j) : (v^d_i, v^d_j) \in E^d$ and model the overall structure with a GMRF that has a $2n \times 2n$ precision matrix $\boldsymbol{Q}^d$ given by Eq. 12 with $k = 2$. The mean relative locations vector used in this case is the same as the mean shape $\bar{\mathbf{s}}$, because $\boldsymbol{\mu}^d_{ij} = E(\boldsymbol{\ell}_i - \boldsymbol{\ell}_j) = E(\boldsymbol{\ell}_i) - E(\boldsymbol{\ell}_j)$. As mentioned in [12], the normal distribution of each edge's relative locations vector in some sense controls "the stiffness of a spring connecting the two parts". In practice, this spring-like model manages to constrain extreme shape configurations that could be evoked during fitting with very bad initialization, leading the optimization process towards a better result. Given $\bar{\mathbf{s}}$ and $\mathbf{Q}^d$, the cost of observing a shape instance $\mathbf{s} = \mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})$ is

$$\begin{aligned} &\|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\|^2_{\mathbf{Q}^d} = \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \mathcal{S}(\bar{\mathbf{s}}, \mathbf{0})\|^2_{\mathbf{Q}^d} = \\ =&\mathcal{S}(\mathbf{0}, \mathbf{p})^T \mathbf{Q}^d \mathcal{S}(\mathbf{0}, \mathbf{p}) \end{aligned} \tag{18}$$

where we used the properties $\mathcal{S}(\bar{\mathbf{s}}, \mathbf{0}) = \bar{\mathbf{s}} + \mathbf{U}\mathbf{0} = \bar{\mathbf{s}}$ and $\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}} = \bar{\mathbf{s}} + \mathbf{U}\mathbf{p} - \bar{\mathbf{s}} = \mathcal{S}(\mathbf{0}, \mathbf{p})$.

2.4. Gauss-Newton optimization

The trained shape, appearance and deformation models can be combined to localize the landmark points of an object in a new testing image $\mathbf{I}$. Specifically, given the appearance and deformation costs of Eqs. 17 and 18, the cost function to be optimized is

$$\arg\min_{\mathbf{p}} \|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}\|^2_{\mathbf{Q}^a} + \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \bar{\mathbf{s}}\|^2_{\mathbf{Q}^d} \tag{19}$$

We minimize the cost function with respect to the shape parameters $\mathbf{p}$ using a variant of the Gauss-Newton algorithm [15, 19, 5]. The optimization procedure can be applied in two different ways, depending on the coordinate system in which the shape parameters are updated: (1) *forward* and (2) *inverse*. Additionally, the parameters update can be carried out in two manners: (1) *additive* and (2) *compositional*, which we show that in the case of our model they are identical. However, the forward additive algorithm is very slow compared to the inverse one. This is the reason why herein we only present and experiment with the inverse case (for a derivation of the forward case please refer to the supplementary material).

Inverse-Compositional The compositional update has the form $\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) \leftarrow \mathcal{S}(\mathbf{s}, \mathbf{p}) \circ \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})^{-1}$. As also shown in [29], by expanding this expression we get

$$\mathcal{S}(\mathbf{s}, \mathbf{p}) \circ \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})^{-1} = \mathcal{S}(\mathcal{S}(\bar{\mathbf{s}}, -\Delta\mathbf{p}), \mathbf{p}) = \mathcal{S}(\bar{\mathbf{s}}, \mathbf{p} - \Delta\mathbf{p})$$

Consequently, due to the translational nature of our motion model, the compositional parameters update is reduced to the parameters subtraction, as $\mathbf{p} \leftarrow \mathbf{p} - \Delta\mathbf{p}$, which is equivalent to the additive update. By using this compositional update of the parameters and having an initial estimate of $\mathbf{p}$, the cost function of Eq. 19 is expressed as minimizing

$$\begin{aligned} \arg\min_{\Delta\mathbf{p}} &\|\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p})) - \bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p}))\|^2_{\mathbf{Q}^a} + \\ &+ \|\mathcal{S}(\bar{\mathbf{s}}, \mathbf{p}) - \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})\|^2_{\mathbf{Q}^d} \end{aligned}$$

with respect to $\Delta\mathbf{p}$. With some abuse of notation due to $\bar{\mathbf{a}}$ being a vector, $\bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p}))$ can be described as

$$\bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})) = \begin{bmatrix} \boldsymbol{\mu}^a_1(\mathcal{S}_1(\bar{\mathbf{s}}, \Delta\mathbf{p})) \\ \vdots \\ \boldsymbol{\mu}^a_n(\mathcal{S}_n(\bar{\mathbf{s}}, \Delta\mathbf{p})) \end{bmatrix}$$

where $\boldsymbol{\mu}^a_i = E(\mathcal{F}(\boldsymbol{\ell}_i)), \forall i = 1, \ldots, n$. This formulation gives the freedom to each landmark point of the mean shape to slightly move within its reference frame. The reference frame of each landmark is simply the $h \times w$ patch neighbourhood around it, in which $\boldsymbol{\mu}^a_i$ is defined. In order to find the solution we need to linearize around $\Delta\mathbf{p} = \mathbf{0}$ as

$$\begin{cases} \bar{\mathbf{a}}(\mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p})) \approx \bar{\mathbf{a}} + \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} \Delta\mathbf{p} \\ \mathcal{S}(\bar{\mathbf{s}}, \Delta\mathbf{p}) \approx \bar{\mathbf{s}} + \mathbf{J}_{\mathcal{S}}|_{\mathbf{p}=\mathbf{0}} \Delta\mathbf{p} \end{cases}$$

where $\mathbf{J}_{\mathcal{S}}|_{\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\mathcal{S}} = \frac{\partial \mathcal{S}}{\partial \mathbf{p}} = \mathbf{U}$ is the $2n \times n_S$ shape Jacobian and $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\bar{\mathbf{a}}}$ is the $mn \times n_S$ appearance Jacobian

$$\mathbf{J}_{\bar{\mathbf{a}}} = \nabla\bar{\mathbf{a}} \frac{\partial \mathcal{S}}{\partial \mathbf{p}} = \nabla\bar{\mathbf{a}}\mathbf{U} = \begin{bmatrix} \nabla\boldsymbol{\mu}^a_1 \mathbf{U}_{1,2} \\ \vdots \\ \nabla\boldsymbol{\mu}^a_n \mathbf{U}_{2n-1,2n} \end{bmatrix}$$

where $\mathbf{U}_{2i-1,2i}$ denotes the $2i-1$ and $2i$ row vectors of the basis $\mathbf{U}$. Note that we make an abuse of notation by writing $\nabla\boldsymbol{\mu}_i^a$ because $\boldsymbol{\mu}_i^a$ is a vector. However, it represents the gradient of the mean patch-based appearance that corresponds to landmark i and it has size $m \times 2$. By substituting, taking the partial derivative with respect to $\Delta\mathbf{p}$, equating it to $\mathbf{0}$ and solving for $\Delta\mathbf{p}$ we get

$$\Delta\mathbf{p} = \mathbf{H}^{-1}[\mathbf{J}_{\bar{\mathbf{a}}}{}^T \mathbf{Q}^a \left(\mathcal{A}(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p})) - \bar{\mathbf{a}}\right) + \mathbf{H}_{\mathcal{S}}\mathbf{p}] \qquad (20)$$

where

$$\left.\begin{array}{l} \mathbf{H}_{\bar{\mathbf{a}}} = \mathbf{J}_{\bar{\mathbf{a}}}{}^T \mathbf{Q}^a \mathbf{J}_{\bar{\mathbf{a}}} \\ \mathbf{H}_{\mathcal{S}} = \mathbf{J}_{\mathcal{S}}{}^T \mathbf{Q}^d \mathbf{J}_{\mathcal{S}} = \mathbf{U}^T \mathbf{Q}^d \mathbf{U} \end{array}\right\} \Rightarrow \mathbf{H} = \mathbf{H}_{\bar{\mathbf{a}}} + \mathbf{H}_{\mathcal{S}}$$

is the combined $n_{\mathcal{S}} \times n_{\mathcal{S}}$ Hessian matrix and we use the property $\mathbf{J}_{\mathcal{S}}{}^T \mathbf{Q}^d \left(\mathcal{S}(\bar{\mathbf{s}},\mathbf{p}) - \bar{\mathbf{s}}\right) = \mathbf{U}^T \mathbf{Q}^d \mathbf{U}\mathbf{p} = \mathbf{H}_{\mathcal{S}}\mathbf{p}$. Note that $\mathbf{J}_{\bar{\mathbf{a}}}$, $\mathbf{H}_{\bar{\mathbf{a}}}$, $\mathbf{H}_{\mathcal{S}}$ and $\mathbf{H}^{-1}$ of Eq. 20 can be precomputed. The computational cost per iteration is only $\mathcal{O}(mnn_{\mathcal{S}})$. The cost is practically reduced to a multiplication between a $n_{\mathcal{S}} \times mn$ matrix and a $n_{\mathcal{S}} \times 1$ vector that leads to a close to real-time performance, similar to the one of the very fast SDM method [31].

2.4.1 Derivation of existing methods

The APS model shown in the cost function of Eq. 19 is an abstract formulation of a generative model from which many existing models from the literature can be derived.

PS [12]**, DPM** [33]. The proposed cost function written in summation form (using Eqs. 7 and 11) is equivalent to PS and DPM. The only difference is that these methods employ a dynamic programming technique to find the global optimum, instead of optimizing with respect to the parameters of a motion model to find a local optimum. Moreover, these methods are limited to use only tree structures for the deformation cost (G^d) and assume an empty graph for the appearance cost (G^a), as opposed to APS that can utilize any graph structure without affecting its computational cost.

AAM-POIC [19]. By removing the deformation prior from Eq. 19 and using a single multidimensional normal distribution in the shape and appearance models, the proposed APS are equivalent to AAMs. After performing an eigenanalysis on the appearance covariance matrix ($\boldsymbol{\Sigma}^a = \mathbf{W}\mathbf{D}\mathbf{W}^T$), the POIC optimization of an AAM can be derived from the presented inverse algorithm by using as precision matrix the complement of the texture subspace, i.e. $\mathbf{Q}^a = \mathbf{I} - \mathbf{W}\mathbf{W}^T$. The part-based AAM of [29] uses an alternating optimization similar to [27]. Its project-out equivalent can be derived by using the above precision matrix.

BAAM-POIC [2]. Similar to the AAM-POIC, the Bayesian AAM can be formulated by replacing the precision matrix with $\mathbf{Q}^a = \mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T + \frac{1}{\sigma^2}(\mathbf{I} - \mathbf{W}\mathbf{W}^T)$. This precision matrix is derived by applying the Woodbury formula on the covariance matrix $\mathbf{W}\mathbf{D}\mathbf{W}^T + \sigma^2\mathbf{I}$, where σ^2 is the variance of the noise in the appearance subspace $\mathbf{W}$.

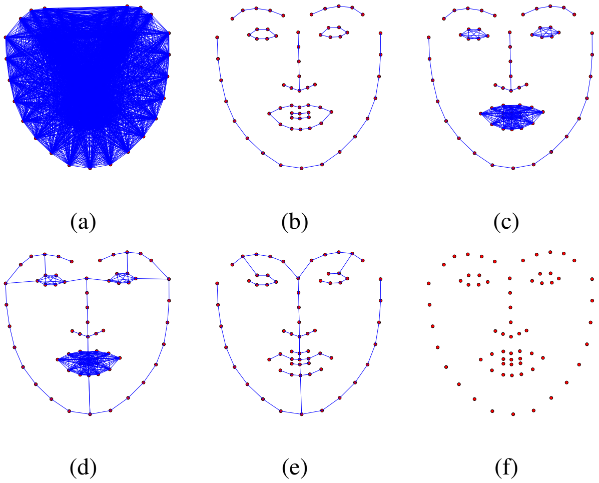


Figure 2: Employed graph structures. *(a)* Complete graph. *(b)* Chain per area. *(c)* Chain and complete per area. *(d)* Chain and complete per area with connections between them. *(e)* Minimum spanning tree. *(f)* Empty graph.

The above highlight the flexibility and strengths of the proposed model. As shown in Sec 3.2, the proposed GMRF-based appearance model makes our inverse technique, to the best of our knowledge, the best performing one among all inverse algorithms with fixed Jacobian and Hessian (e.g. POIC).

3. Experiments

In this section we present a comprehensive evaluation of the different ways in which APS can be used to model the shape and appearance of an object and compare their performance against state-of-the-art deformable models. The experiments are carried out for the popular task of face alignment for which there is a plethora of large annotated databases. In all presented cases, the proposed APS are built using a two-level pyramid. We keep about 92% of the shape variance and set $n_A = 150$ for both levels that corresponds to about 80% of the appearance variance. The appearance is represented either by pixel intensities or dense SIFT [17] with 8 channels and the extracted patch size is 17×17. The accuracy of the fitting results is measured by the point-to-point RMS error between the fitted shape and the ground truth annotations, normalized by the face size, as proposed in [33]. Note that our Python implementation of APS[1] runs at $50ms$ per frame, which is very close to real-time. We believe that with further code optimization, APS are likely to be capable of running in real-time on high end desktop/laptop machine. Their time complexity is independent of the graph structure that is employed.

Graph type G^a	*mean ± std*	*median*	≤ 0.04
Fig. 2a	0.0399 ± 0.0227	0.0324	68.3%
Fig. 2b	$\mathbf{0.0391 \pm 0.0243}$	**0.0298**	**69.6%**
Fig. 2c	0.0506 ± 0.0371	0.0370	58.9%
Fig. 2d	0.0492 ± 0.0373	0.0354	58.9%
Fig. 2e	0.0413 ± 0.0257	0.0316	65.2%
Fig. 2f	0.0398 ± 0.0246	0.0319	66.5%
PCA	0.0716 ± 0.0454	0.0595	25.5%
Initialization	0.0800 ± 0.0280	0.0768	4.0%

Table 1: Comparison of the GMRF-based and the PCA-based appearance model of APS.

3.1. APS experimental analysis

Herein, we present three experiments as a proof of concept regarding the formulation of APS. Specifically, we aim to examine the contribution of each one of the shape, appearance and deformation models and find an optimal graph structure. The model is trained using the 811 images of Labeled Faces Parts in the Wild (LFPW) [6] train set and tested on the corresponding test set. We use the annotations provided by the 300W competition [23, 24] and evaluate using 66 landmark points which are derived by removing landmarks 61 and 65 from the 68-points mark-up. In this set of experiments, we don't extract any appearance features and only use pixel intensities. Figure 2 shows the graph structures that we employ for the purpose of these experiments. Note that the minimum spanning tree (MST) is computed as shown in [12]. The fitting process of the presented experiments is initialized by adding Gaussian noise to the global similarity transform retrieved from the ground truth annotations (without in-plane rotation) and applying it to the mean shape $\bar{\mathbf{s}}$. We set the standard deviation of the random noise to 0.04, which generates very challenging initializations.

Beginning with the appearance model, Tab. 1 reports the performance when using a GMRF with the graph structures of Fig. 2 and when using a single multivariate normal distribution through PCA. The performance is reported in the form of statistical measures (mean, median and standard deviation) and as the percentage of the testing images that achieved a final error ≤ 0.04 (value at which the result is considered adequately good by visual inspection). For this experiment, we use a PCA shape model and a deformation prior with the MST. The improvement is significantly high. Even the empty graph, which generates a block diagonal precision matrix $\mathbf{Q}^a$, thus it assumes independence between all parts, greatly outperforms the PCA case. The most appropriate graph structure is the one of Fig. 2b, which suggests that, for the case of faces, it is better to connect the landmarks of each facial area (eyes, mouth, nose etc.) between them and avoid relating the areas between each other.

Graph type G^s	*mean ± std*	*median*	≤ 0.04
Fig. 2a	0.0495 ± 0.0273	0.0420	45.5%
Fig. 2b	0.0496 ± 0.0276	0.0438	45.5%
Fig. 2c	0.0503 ± 0.0262	0.0433	44.2%
Fig. 2d	0.0495 ± 0.0257	0.0434	44.6%
Fig. 2e	0.0519 ± 0.0306	0.0437	43.8%
Fig. 2f	0.0492 ± 0.0249	0.0437	42.9%
PCA	$\mathbf{0.0412 \pm 0.0295}$	**0.0301**	**65.6%**
Initialization	0.0800 ± 0.0280	0.0768	4.0%

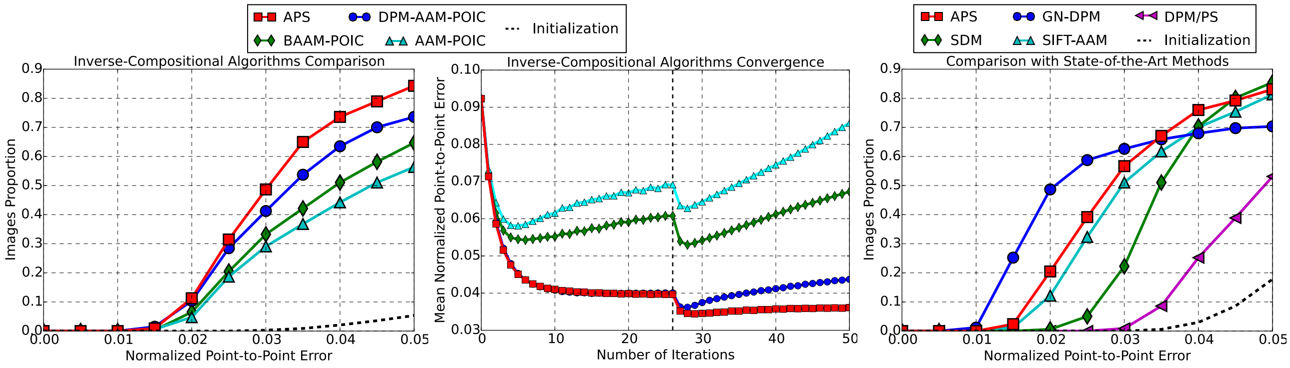
Table 2: Comparison of the GMRF-based and the PCA-based shape model of APS.

Deformation prior G^d	Shape model G^s	
	Fig. 2a	*PCA*
No prior	0.1327 ± 0.0857	0.0429 ± 0.0267
Fig. 2b	0.0524 ± 0.0256	0.0430 ± 0.0240
Fig. 2e	$\mathbf{0.0495 \pm 0.0273}$	$\mathbf{0.0391 \pm 0.0243}$

Table 3: Comparison of the GMRF-based and the PCA-based deformation prior of APS in combination with the GMRF-based and the PCA-based shape model.

Table 2 presents the same experiment for the shape model and the results are opposite to those of the appearance model. However, this is a well expected result. As mentioned in Sec. 2.3, the appearance model utilizes directly the constructed block sparse precision matrix. On the contrary, we need to decompose the covariance matrix ($\mathbf{\Sigma}^s = (\mathbf{Q}^s)^{-1}$) of the shape model in order to learn a parametric subspace that will be used during optimization. However, due to the block sparse formulation, the resulting covariance matrix has high (in some cases full) rank. Most eigenvalues are non-zero and they represent a small percentage of the data variance. Thus by keeping more than 90% of the total variance, the model ends up with too many modes of variation (about 100 in the case of 68 vertexes and depending on the graph structure). Consequently, it is very hard to apply a robust optimization in such a parametric space, as the search space is too large.

Finally, Tab. 3 examines the contribution of the deformation prior of Eq. 19. We use the graph of Fig. 2b for the appearance model and we test for two cases of the shape model: PCA and GMRF with a complete graph (Fig. 2a). The results prove that the prior plays an important role in both cases, as it improves the result. Especially in the case of the GMRF, the improvement is significant. Given the previous analysis about the non robust behaviour of a GMRF shape model, this result is expected because the prior term will prevent the shape model from generating non-realistic instances of the face.



(a) (b) (c)

Figure 3: Comparison of APS with other methods on AFW database. *(a, b)*: Comparison of APS accuracy and convergence with other inverse compositional methods with fixed Jacobian and Hessian. The dashed vertical black line in *(b)* denotes the transition from lower to higher pyramidal level. *(c)*: Comparison of APS with current state-of-the-art methods.

APS	*SDM*	*SIFT-AAM*	*GN-DPM*	*DPM/PS*
0.0415	0.0453	0.0423	0.0686	0.0585

Table 4: Mean values of the cumulative error curves reported in Fig. 3c.

3.2. Comparison with state-of-the-art methods

Figures 3a and 3b aim to compare the accuracy and convergence speed of APS against the other existing inverse compositional techniques with fixed Jacobian and Hessian (POIC) mentioned in 2.4.1. AAM-POIC [19] and BAAM-POIC [2] denote the POIC optimization of an AAM and a Bayesian AAM. AAM-DPM-POIC refers to the inverse algorithm that can be combined with the AAM part-based model of [29]. All methods are trained on LFPW database in the same manner, using the same pyramid and extracting dense SIFT features with 8 channels. For all of them we keep $n_S = 5$ and $n_S = 15$ shape components for the low and high levels respectively, that correspond to about 92% of the total shape variance, and $n_A = 150$ appearance components for both levels. The results, which are computed using 66 landmark points, are reported on the challenging Annotated Faces In-The-Wild (AFW) [33] database and indicate that the proposed method in this paper outperforms all existing inverse-compositional techniques by a significant margin. Most importantly, APS need very few number of iterations in order to converge (less than 10 at the first pyramidal level and no more than 4 at the second), which highlights their close to real-time computational complexity.

Figure 3c compares APS against the current state-of-the-art techniques: SDM [31], the recently proposed GN-DPM [29] and SIFT-AAM [4]. The initialization for all methods is done using the bounding box of the landmark points returned by DPM [33] (the black dashed line). For all the methods we used the pre-trained implementations provided by their authors, except SIFT-AAM which we trained using the Menpo Project [1]. Note that all competing methods are trained on much more data than the 811 LFPW images that we use. The result is reported on the AFW database and computed based on 49 points, which is the mark-up that both SDM and GN-DPM return. Table 4 reports the mean values of the cumulative error curves of Fig. 3c. These results show that APS outperform all methods and are more robust. Note that GN-DPM is very accurate when the initialization is close to the ground truth but is not robust against bad initializations, as indicated by its large mean error value. Finally, please refer to the supplementary material for additional experimental results.

4. Conclusion

In this work we proposed a powerful generative model that combines the main ideas behind PS and AAMs. APS employ a graph-based modelling of the appearance and use a variant of the Gauss-Newton technique to optimize with respect to the parameters of a statistical shape model. One of the major contributions of this paper is the proof that modelling the patch-based appearance of an object with a GMRF structure is more beneficial than applying a PCA model. APS also introduce a spring-like deformation prior term that makes them robust to bad initializations. The method has a close to real-time fitting performance, which is the same independent of the graph structure that is employed, and as shown in our experiments needs only a few iterations to converge. In the future, we aim to apply APS on classes of articulated objects (e.g. hands, body pose) in order to test whether the combination of patch-based appearance with the deformation prior can make a significant difference.

5. Acknowledgements

The work of E. Antonakos and S. Zafeiriou is funded by the EPSRC project EP/J017787/1 (4D-FAB). The work of S. Zafeiriou is also partially supported by the EPSRC project EP/L026813/1 Adaptive Facial Deformable Models for Tracking (ADAManT). The work of J. Alabort-i-Medina is funded by a DTA studentship from Imperial College London and by the Qualcomm Innovation Fellowship.

References

[1] J. Alabort-i Medina, E. Antonakos, J. Booth, P. Snape, and S. Zafeiriou. Menpo: A comprehensive platform for parametric image alignment and visual deformable models. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 679–682, New York, NY, USA, 2014. ACM.

[2] J. Alabort-i-Medina and S. Zafeiriou. Bayesian active appearance models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3438–3445, 2014.

[3] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1014–1021, 2009.

[4] E. Antonakos, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou. Hog active appearance models. In *Proceedings of IEEE Conference on Image Processing (ICIP)*, pages 224–228, 2014.

[5] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision (IJCV)*, 56(3):221–255, 2004.

[6] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 545–552, 2011.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):681–685, 2001.

[8] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[9] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645, 2010.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 66–73, 2000.

[12] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.

[13] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[14] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12):1080–1093, 2005.

[15] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(10):1025–1039, 1998.

[16] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874, 2014.

[17] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999.

[18] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European Conference on Computer Vision (ECCV)*, pages 720–735. Springer, 2014.

[19] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision (IJCV)*, 60(2):135–164, 2004.

[20] G. Papandreou and P. Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[21] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692, 2014.

[22] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of IEEE International Conference on Computer Vision Workshopw (ICCV'W)*, pages 397–403, 2013.

[24] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 896–903, 2013.

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

[26] J. R. Tena, F. De la Torre, and I. Matthews. Interactive region-based linear 3d face models. *ACM Transactions on Graphics (TOG)*, 30(4):76, 2011.

[27] G. Tzimiropoulos, J. Alabort-i-Medina, S. Zafeiriou, and M. Pantic. Generic active appearance models revisited. In *Asian Conference on Computer Vision (ACCV)*, pages 650–663. Springer, 2013.

[28] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 593–600, 2013.

[29] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, 2014.

[30] Y. Wang, S. Lucey, and J. F. Cohn. Enforcing convexity for improved alignment with constrained local models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[31] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, 2013.

[32] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(12):2878–2890, 2013.

[33] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886, 2012.

[34] S. Zuffi, O. Freifeld, and M. J. Black. From pictorial structures to deformable structures. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3546–3553, 2012.