Fast and exact Newton and Bidirectional fitting of Active Appearance Models

Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic, *Fellow, IEEE*

***Abstract*—Active Appearance Models (AAMs) are generative models of shape and appearance that have proven very attractive for their ability to handle wide changes in illumination, pose and occlusion when trained in the wild, while not requiring large training dataset like regression-based or deep learning methods. The problem of fitting an AAM is usually formulated as a non-linear least squares one and the main way of solving it is a standard Gauss-Newton algorithm. In this paper we extend Active Appearance Models in two ways: we first extend the Gauss-Newton framework by formulating a bidirectional fitting method that deforms both the image and the template to fit a new instance. We then formulate a second order method by deriving an efficient Newton method for AAMs fitting. We derive both methods in a unified framework for two types of Active Appearance Models, holistic and part-based, and additionally show how to exploit the structure in the problem to derive fast yet exact solutions. We perform a thorough evaluation of all algorithms on three challenging and recently annotated in-the-wild datasets, and investigate fitting accuracy, convergence properties and the influence of noise in the initialisation. We compare our proposed methods to other algorithms and show that they yield state-of-the-art results, out-performing other methods while having superior convergence properties.**

***Index Terms*—Active Appearance Models, Newton method, bidirectional image alignment, inverse compositional, forward additive.**

I. Introduction

Active Appearance Models are generative models of shape and appearance widely used and studied in the field of Computer Vision, especially for facial landmark detection. First introduced by [1], AAMs formulate the problem of landmark detection as a non-linear sum of squares minimization. A linear model of both shape and appearance is built in a strongly supervised way and that model is aligned to a new instance to localize landmarks. Fitting an AAM to a new image is then done by reconstructing that object, i.e. reconstructing its appearance and deforming either the image in the *forward* framework or the template in the *inverse* framework so that the difference between the two is as small as possible. The deformation is modelled *via* a motion model, typically a piecewise affine warping, that warps the appearance from a given image to the mean shape. Finding the correct parameters for the affine warping is equivalent to localizing the landmarks on the face.

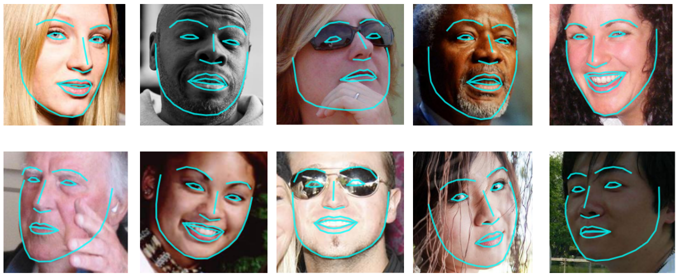


Fig. 1. Example of images from the AFW dataset fitted with our proposed Bidirectional Part-Based AAM.

There are two main approaches to solving the AAM problem: regression based –the goal of which is to learn a function that maps directly the appearance features to the desired target, as the original AAM [1], [2], [3]– and optimisation based, which solve it analytically. In this paper, we focus solely on optimisation based methods which have been shown to produce state-of-the art results [4], [5]. In that case, the problem of fitting an Active Appearance Model is formulated as a non-linear least-squares one and is iteratively solved in a Lucas-Kanade fashion. Prior work has been focusing exclusively on Gauss-Newton methods in either the inverse or the forward framework.

The Lukas-Kanade algorithm was introduced in [6] for image alignment and an appearance-based version was introduced by Hager and Belhumeur [7]. It was first applied to AAM fitting by Matthews and Baker in [8] where they notably introduce the simultaneous inverse compositional (SIC) framework for fitting algorithms to solve the AAM problem. As its name indicates, it works in the inverse compositional framework in the sense that, at each iteration, it deforms the template to align it to the image and composing the inverse of the resulting warp update to the current image warp estimate. However, albeit robust and exact and although it has gained significant interest following the work of [8], its computational cost remained prohibitive for most applications [9], [10].

For that reason, the project-out inverse compositional (POIC) algorithm, introduced in [8] has been for a long time the preferred method for person specific AAMs. In contrast to SIC, POIC is a very fast yet approximate algorithm which has been shown unable to generalise well for the case of large appearance variations.

Besides SIC and POIC, fast versions of exact Gauss-Newton algorithms (both inverse and forward) were recently intro-

J. Kossaifi is with the Department of Computing, Imperial College London, London SW7 2AZ, U.K. (email: jean.kossaifi12@imperial.ac.uk).

G. Tzimiropoulos is with the School of Computer Science, The University of Notthingham, Nottingham NG8 1BB, U.K. (e-mail: yorgos.tzimiropoulos@nottingham.ac.uk).

M. Pantic is with the Department of Computing, Imperial College London, London, U.K., and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede 7522 NB, The Netherlands (e-mail: m.pantic@imperial.ac.uk).

duced in [11]. The proposed methods capitalize on results from optimization theory to provide solutions that are both exact and computationally efficient, making them prime choices.

Most recently, the authors of [4] introduced a part-based model, coined GN-DPM which is built in the same way as the Active Appearance Model but replaces the holistic appearance model by a more flexible, local, patch based one. This method has been showed to produce state-of-the-art results [12], [4], [5], even outperforming regression-based methods such as SDM [3] and its variants [13] while being more robust and more computationally efficient thanks to a sparse formulation.

Active Appearance Models and most recently GN-DPM are therefore widely used in practice, mainly owing to their ability to handle challenging pose illumination and occlusion conditions when trained in the wild. In addition, their generative nature makes it easy to build an instance of the model even with very few training images, which is extremely useful for person-specific modelling, such as in a tracking context [14].

In this work, we depart from the de facto standard approach to AAM fitting using Gauss-Newton optimisation and make several contributions:

- For the first time (to the best of our knowledge), we propose two novel, fast and exact optimisation frameworks for AAM fitting. The first algorithm is a Bidirectional fitting approach which elegantly combines both inverse and forwards formulations. The second algorithm is a fast yet exact second-order method based on Newton Optimisation.
- Naive derivations of these methods result in computationally heavy algorithms, in practice prohibitive for most applications. We show how to address this problem by exploiting the structure in the AAM problem to derive fast and exact solutions.
- We derive these methods for both holistic and part-based Active Appearance Models in a unified framework and extend them to handle robust features.
- We provide comprehensive experiments on three different datasets recently annotated in-the-wild, investigating both fitting accuracy and convergence properties.
- We investigate their robustness to noise in the initialisation.
- We provide comparison with the State-of-the-Art.

A preliminary version of the Newton and Bidirectional methods was previously formulated in [15] and [16] respectively for the simple case of intensity-based holistic Active Appearance Models.

In the rest of the paper, Sec. II introduces rigorously sparse and part-based Active Appearance Models, Sec. III quickly reviews prior work while Sec. IV introduces a unified objective function for fitting the models. Sec. V, details the derivation of the Bidirectional method to solve that problem. Sec. VI shows how the fast version of SIC and Forward can be derived for the weighted case as special cases of the Bidirectional problem and Sec. VII details the derivation of the Newton algorithm. The experimental setting, implementation details, results and analysis of these are presented in Sec. VIII.

II. Building the Active Appearance Models

Active Appearance Models, be them holistic or part-based, are generative models defined by a shape model, an appearance model and a motion model:

- **Shape model:** A linear model of shape, shared by both holistic and part-based AAM.
- **Appearance model:** A linear model of appearance defined in some reference canonical frame that depends on the motion model used (also known as texture model). This appearance model is holistic for AAMs and part-based for GN-DPM / part-based AAMs.
- **Motion model:** This is a function that warps the pixels from the image frame to the reference frame and can be a piecewise affine warp for holistic AAMs [8] or a simple translation one for part-based AAMs [4].

In this work we unify the formulations for both holistic and part-based AAMs and derive the solutions for all main optimisation methods.

A. Shape model

We assume that we have a dataset of D training images represented as functions of their pixels $(I_k(x,y))_{k=1,\cdots,D}$ for which the coordinates $(x,y)^T$ of u landmarks have been annotated (typically manually). For a given object, the set of these u coordinates $(x_1,y_1,\cdots,x_u,y_u)^T \in \mathbb{R}^{2u}$ defines the shape of that object. The shape model is obtained by first aligning the training shapes by applying a generalised Procrustes analysis, which removes *similaritiy transformations* (translation, scaling and rotation). PCA is applied to these similarity-free shapes and the $n-4$ resulting eigenvectors with the highest associated eigenvalues are kept to obtain the shape model defined by the mean shape $\mathbf{s}_0$ and these eigenvectors. Since this model has been built on similarity-free shapes it is unable to model scaling translation and rotation. We address that by appending four similarity eigenvectors and re-orthonormalising the whole set of vector. Finally, we stack these n shape eigenvectors as the columns of the matrix $\mathbf{S} \in \mathbb{R}^{2u,n}$. Instances of this shape model are then expressed as:

$$\mathbf{s}(\mathbf{p}) = \mathbf{s}_0 + \mathbf{S}\mathbf{p}, \tag{1}$$

with $\mathbf{p} = (\mathbf{p}_1,\cdots,\mathbf{p}_n)^T \in \mathbb{R}^n$ containing the shape parameters.

The shape model is built in the same way (as described above) for both AAMs and GN-DPMs. We now detail appearance model, that is built slightly differently for the two methods. However, for both method, the end result is a linear model of appearance, similar to the shape one, which can be summarised by a mean appearance and a set of appearance eigenvectors, allowing unified notations and abstracting away the difference between the two models.

B. Holistic Active Appearance Models

Holistic Active Appearance Models usually use a Piecewise Affine Warping as their motion model $\mathcal{W}$. A piecewise affine warp each defined as follows: first both shape and mean shape are triangulated (eg a Delaunay triangulation). Each triangle in the target shape, together with its the corresponding triangle in the mean shape, define an affine transformation. The collection of all affine transformation defined by all triangle pairs defines the piecewise affine warp. The appearance model is then obtained by warping each training image to the mean shape s_0 which forms the base mesh and we denote $\mathcal{V}$ the set of the N pixels $\mathcal{V} = (\mathbf{v_l})_{l=1,\cdots,N} = \left((x_l, y_l)^T\right)_{l=1,\cdots,N}$ inside that mesh. We then apply PCA on these flattened shape-free images to obtain the appearance model of which we again keep only the first m with the highest associated eigenvalues. The resulting appearance model is described by the mean appearance $\mathbf{A}_0 \in \mathbb{R}^N$ and the appearance eigenvectors stacked as the column of an appearance matrix $\mathbf{A} \in \mathbb{R}^{N,m}$. Note that the appearance eigenvectors can also be considered as functions $A_i(x, y), i \in \{1, \cdots, m\}$ of the pixel locations $\mathbf{v} = (x, y)^T \in \mathcal{V}$. Instances of this appearance model can be expressed as:

$$\mathbf{A}(\mathbf{c}) = \mathbf{A}_0 + \mathbf{A}\mathbf{c}, \tag{2}$$

with $\mathbf{c} = (\mathbf{c}_1, \cdots, \mathbf{c}_m)^T \in \mathbb{R}^m$ containing the appearance parameters.

Let $\mathbf{v} = (x, y) \in \mathcal{V}$ and $\mathbf{s} = (x_1, y_1, \cdots, x_u, y_u)$. The derivative of $\mathcal{W}(\mathbf{v}, \mathbf{p})$ with respect to the shape parameter $\mathbf{p}$ depends on the shape vertices.

$$\begin{aligned}\frac{\partial \mathcal{W}(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}} &= \left(\frac{\partial \mathcal{W}_1(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}}, \ \frac{\partial \mathcal{W}_2(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}}\right)^T \\ &= \left(\sum_{k=1}^{u} \frac{\partial \mathcal{W}_1}{\partial x_k} \frac{\partial x_k}{\partial \mathbf{p}}, \sum_{k=1}^{u} \frac{\partial \mathcal{W}_2}{\partial y_k} \frac{\partial y_k}{\partial \mathbf{p}}\right)^T.\end{aligned}$$

For more detail on how to compute the derivatives for the case of a piecewise affine please refer to [11].

C. Part-Based Active Appearance Model

Part-based Active Appearance Models on the other hand use a translational motion model $\mathcal{W}$. First similarities are removed from the training images by warping them to a reference frame. Then, around each landmark, a patch of size $N_s \times N_s$ is extracted. The resulting u patches are concatenated and flattened to form a *warped* image of size $u \times N_s^2$. The appearance model is then obtained in the same way described for holistic AAMs by applying PCA on that set of warped images, and again the appearance space is described by the mean appearance $\mathbf{A}_0 \in \mathbb{R}^N$ and the appearance eigenvectors stacked as the column of an appearance matrix $\mathbf{A} \in \mathbb{R}^{N,m}$, with an instance of that model given by (2). As previously done for holistic AAMs, we denote $\mathcal{V}$ the set of the $N = N_s \times N_s$ pixels $\mathbf{v} = (x, y)^T$ inside the patches, $\mathcal{V} = \left(\mathbf{v}_l = (x_l, y_l)^T\right)_{l=1,\cdots,N}$.

As now the motion model is a translational one, its derivative is simpler than that of a piecewise affine warping: with $v = (x, y) \in \mathcal{V}$;

$$\frac{\partial \mathcal{W}(\mathbf{v}, \mathbf{p})}{\partial \mathbf{p}} = \sum_{k=1}^{u} \delta_v^k \mathbf{S}_k, \tag{3}$$

where $\delta_v^k = 1$ if $\mathbf{v}$ is in the patch extracted around s_k, 0 otherwise and $\mathbf{S}_k$ is the $2 \times n$ matrix of parameters of the k^{th} landmark.

For more detail on GN-DPM, refer to [12], [4].

III. Background work

The problem of fitting an Active Appearance Model is traditionally expressed as a non-linear least squares problem:

$$\arg \min_{\mathbf{\Delta p}, \mathbf{\Delta c}} \frac{1}{2} \sum_{l=1}^{N} \left[I(\mathcal{W}(\mathbf{v}_l, \mathbf{p})) - A_0 - \sum_{i=1}^{m} \mathbf{c}_i A_i \right]^2 \tag{4}$$

This problem has been previously solved using a Gauss-Newton method, either in the inverse framework or in the forward framework.

A. Inverse Framework

The Simultaneous Inverse Compositional algorithm solves (4) by linearising the model around a parameter $\mathbf{p} = 0$ and computing at each iteration an optimal update $\mathbf{\Delta p}$.

The resulting optimisation problem is:

$$\begin{aligned}\arg \min_{\mathbf{\Delta p}, \mathbf{\Delta c}} \frac{1}{2} \sum_{l=1}^{N} [I(\mathcal{W}(\mathbf{v}_l, \mathbf{p})) - A_0 - \sum_{i=1}^{m} \mathbf{c}_i A_i \\ - \mathbf{J}_{A_0} \mathbf{\Delta p} - \sum_{i=1}^{m} \mathbf{c}_i \mathbf{J}_{A_i} \mathbf{\Delta p}]^2,\end{aligned} \tag{5}$$

where for all $i = \{0, \cdots, n\}$, $\mathbf{J}_{A_i}$ is the matrix of derivatives of A_i with respect to $\mathbf{p}$, with $\mathbf{J}_{A_i} \in \mathbb{R}^{1,n}$. All the terms will be introduced in more detail in the next section.

Typically, (5) is solved over a single parameter $\begin{pmatrix} \mathbf{p} \\ \mathbf{c} \end{pmatrix} \in \mathbb{R}^{n+m}$ that combines both shape and appearance parameters appearance parameters. The shape parameter is then updated in an inverse compositional way, $\mathbf{p} = \mathbf{p} \circ \mathbf{\Delta p}^{-1}$. This results in complexity $O((n+m)^2 N)$ which is prohibitive for most applications [8].

Fast-SIC adopts a smarter approach in solving the same problem by capitalizing on optimization theory [17]. The result is an algorithm of $O(nmN + n^2 N)$ which is much less than $O((n+m)^2 N)$ for the original SIC. We generalise based on [17] to derive fast and exact solutions for our Newton and bidirectional AAM fitting algorithms.

B. Forward Framework

In the forward framework, the image rather than the template is linearized by re-writing the problem as:

$$\arg \min_{\mathbf{\Delta q}, \mathbf{c}} \frac{1}{2} \sum_{l=1}^{N} \left[I(\mathcal{W}(\mathbf{v}_l, \mathbf{q})) + \mathbf{J}_I \mathbf{\Delta q} - A_0 - \sum_{i=1}^{m} \mathbf{c}_i A_i \right]^2, \tag{6}$$

where $\mathbf{J}_I$ is the matrix of derivatives of $I(\mathcal{W}(\mathbf{v}_l, \mathbf{q}))$ with respect to $\mathbf{q}$, with $\mathbf{J}_I \in \mathbb{R}^{1,n}$.

Again, at each iteration, optimal updates $\mathbf{\Delta q}$ and $\mathbf{\Delta c}$ are obtained for the shape and the texture parameter, respectively.

The shape parameter is then updated in a forward additional way, $\mathbf{q} = \mathbf{q} + \mathbf{\Delta q}$.

Fast-Forward works in a similar way as Fast-SIC by also capitalizing on (14) to solve problem (6). Again, one can show that solving the above optimization problem has a cost $O(nmN + n^2N)$ [11].

IV. Unified objective function

To formulate our Newton and Bidirectional methods, we first introduce here a unified framework in which we derive all methods by formulating the problem of fitting an Active Appearance Model as a more general weighted non-linear least squares problem. For this purpose we introduce a parameter $\mathbf{q}$ used to deform the image, not the template. Note that all the calculations are done in the coordinate frame of the mean shape.

The goal is then to solve the following optimization problem:

$$\arg\min_{\mathbf{p},\mathbf{q},\mathbf{c}} \frac{1}{2}\sum_{l=1}^{N} f(\mathbf{v}_l, \mathbf{p}, \mathbf{q}, \mathbf{c}) = \arg\min_{\mathbf{p},\mathbf{q},\mathbf{c}} \frac{1}{2}\sum_{l=1}^{N} \mathbf{W}_{ll} \times g(\mathbf{v}_l, \mathbf{p}, \mathbf{q}, \mathbf{c})^2, \tag{7}$$

where

$$g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c}) = [I(\mathcal{W}(\mathbf{v}, \mathbf{q})) - A_0(\mathcal{W}(\mathbf{v}, \mathbf{p})) - \sum_{i=1}^{m} \mathbf{c}_i A_i(\mathcal{W}(\mathbf{v}, \mathbf{p}))]. \tag{8}$$

and $\mathbf{W}$ is a weight matrix, i.e. a diagonal matrix which diagonal elements $\mathbf{W}_{ll}, l \in \{1, \cdots, N\}$ define the weights associated with each pixel. In this work we set that $\forall l \in \{1, \cdots, N\}, \mathbf{W}_{ll} \in \{0, 1\}$, therefore allowing for sparsity. In particular, we define a sparse grid over $\mathcal{V}$ by considering only every K-th pixel (in practice $K = 2$ or $K = 4$). This reduces drastically the speed as it divides by 2 or 4 the number of features of the appearance model and results in computationally much more efficient algorithms, with virtually no decrease in performance.

We now provide the derivatives needed to compute the forward, inverse and bidirectional algorithms. For all $\mathbf{v} \in \mathcal{V}$, the derivatives of g with respect to its different parameters are given by:

$$\begin{aligned}
\frac{\partial g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{c}} &= -\, A(\mathcal{W}(\mathbf{v}, \mathbf{p})) \\
\frac{\partial g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{p}} &= -\, \nabla T(\mathcal{W}(\mathbf{v}, \mathbf{p})) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}}\right) \\
&= -\, \nabla A_0(\mathcal{W}(\mathbf{v}, \mathbf{p})) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}}\right) \\
&\quad - \sum_{i=1}^{m} \mathbf{c}_i (\nabla A_i(\mathcal{W}(\mathbf{v}, \mathbf{p}))) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}}\right) \\
\frac{\partial g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{q}} &= -\, \nabla I(\mathcal{W}(\mathbf{v}, \mathbf{q})) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{q}}\right)
\end{aligned}$$

where

$$\begin{aligned}
&\nabla A_i(\mathcal{W}(\mathbf{v}, \mathbf{p})) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{q}}\right) \in \mathbb{R}^{1,n} \\
&\quad = [A_{i,x}(\mathcal{W}(\mathbf{v}, \mathbf{p})) \;\; A_{i,y}(\mathcal{W}(\mathbf{v}, \mathbf{p}))] \begin{pmatrix} \frac{\partial \mathcal{W}_1(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}} \\ \frac{\partial \mathcal{W}_2(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}} \end{pmatrix}
\end{aligned}$$

and

$$\nabla A(\mathcal{W}(\mathbf{v}, \mathbf{p})) = \begin{pmatrix} \nabla A_1(\mathcal{W}(\mathbf{v}, \mathbf{p})) \\ \vdots \\ \nabla A_m(\mathcal{W}(\mathbf{v}, \mathbf{p})) \end{pmatrix} \in \mathbb{R}^{m,2}.$$

- $A_{i,x}$ and $A_{i,y}$ are the x and y gradients of $A_i(\mathcal{W}(v, \mathbf{p}))$, the i^{th} appearance vector $A_i(x, y)$ for pixel $\mathbf{v} = (x, y)$.
- $\frac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}} \in \mathbb{R}^{2,n}$ is the derivative of the motion model $(W)(\mathbf{v}, \mathbf{p})$.

In addition, to derive the Newton method, we will need the second order derivatives that are given by:

$$\begin{aligned}
\frac{\partial^2 g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{p}^2} &= -\, \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{q}}\right)^T \left(\nabla^2 T(\mathcal{W}(\mathbf{v}, \mathbf{q}))\right) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}}\right) \\
&\quad - \nabla A_0(\mathcal{W}(\mathbf{v}, \mathbf{p})) \left(\tfrac{\partial^2 \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}^2}\right) \\
&\quad - \sum_{i=1}^{m} \mathbf{c}_i (\nabla A_i(\mathcal{W}(\mathbf{v}, \mathbf{p}))) \left(\tfrac{\partial^2 \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}^2}\right) \\
\frac{\partial^2 g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{c}^2} &= 0 \\
\frac{\partial^2 g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{c} \partial \mathbf{p}} &= -\, \nabla A(\mathcal{W}(\mathbf{v}, \mathbf{p})) \left(\tfrac{\partial \mathcal{W}}{\partial \mathbf{q}}\right)
\end{aligned}$$

$\forall i \in \{0, \cdots, m\}$, $\nabla^2 A_i(\mathcal{W}(\mathbf{v}, \mathbf{p})$ contains the second order derivatives of $A_i(\mathcal{W}(\mathbf{v}, \mathbf{p})$:

$$\nabla^2 A_i(\mathcal{W}(\mathbf{v}, \mathbf{p}) = \begin{pmatrix} A_{i,xx}(\mathcal{W}(\mathbf{v}, \mathbf{p})) & A_{i,xy}(\mathcal{W}(\mathbf{v}, \mathbf{p})) \\ A_{i,yx}(\mathcal{W}(\mathbf{v}, \mathbf{p})) & A_{i,yy}(\mathcal{W}(\mathbf{v}, \mathbf{p})) \end{pmatrix}$$

where $A_{i,xx}$ and $A_{i,xy}$ are the x and y gradients of $A_{i,x}(\mathcal{W}(\mathbf{v}, \mathbf{p}))$ and $A_{i,yx}$ and $A_{i,yy}$ are the x and y gradients of $A_{i,y}(\mathcal{W}(\mathbf{v}, \mathbf{p}))$.

Finally, since the second order derivative of the motion model $\mathcal{W}$ is null, the second order derivative of g with respect to $\mathbf{p}$ simplifies to

$$\frac{\partial^2 g(\mathbf{v}, \mathbf{p}, \mathbf{q}, \mathbf{c})}{\partial \mathbf{p}^2} = -\left(\tfrac{d\mathcal{W}(\mathbf{v},\mathbf{p})}{d\mathbf{q}}\right)^T \left(\nabla^2 T(\mathcal{W}(\mathbf{v}, \mathbf{q}))\right) \left(\tfrac{\partial \mathcal{W}(\mathbf{v},\mathbf{p})}{\partial \mathbf{p}}\right).$$

A. Vectorised form

We vectorise the calculations over all the pixels by rewriting:

$$\mathbf{\Phi}(\mathbf{p}, \mathbf{q}, \mathbf{c}) = \mathbf{I}[\mathbf{q}] - \mathbf{A}_0[\mathbf{p}] - \sum_{i=1}^{m} \mathbf{c}_i \mathbf{A}_i[\mathbf{p}] = \mathbf{I}[\mathbf{q}] - \mathbf{T}[\mathbf{p}]. \tag{9}$$

We denote N the number of pixels $\mathbf{v} \in \mathcal{V}$ in the mean shape coordinate frame. $\mathbf{I}[\mathbf{q}] \in \mathbb{R}^{N,1}$ is the vectorised warped image $[I(\mathcal{W}(\mathbf{v}, \mathbf{q})]_{v \in \mathcal{V}}$ and $\mathbf{T}[\mathbf{p}] = \mathbf{A}_0[\mathbf{p}] - \sum_{i=1}^{m} \mathbf{c}_i \mathbf{A}_i[\mathbf{p}] \in \mathbb{R}^{N,1}$ is the vectorised template $[A_0(\mathcal{W}(\mathbf{v}, \mathbf{p})) + \sum_{i=1}^{m} \mathbf{c}_i A_i(\mathcal{W}(\mathbf{v}, \mathbf{p}))]_{\mathbf{v} \in \mathcal{V}}$.

We can then write f as:

$$\mathbf{f}(\mathbf{p},\mathbf{q},\mathbf{c}) = \frac{1}{2}\|\mathbf{\Phi}\|^2_{\mathbf{W}} = \mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi}. \tag{10}$$

We also stack the first order derivatives for each pixels into a vector form to obtain the following terms:

$$\begin{aligned} \mathbf{J_c} &= -\,\mathbf{A} \in \mathbb{R}^{N,m} \\ \mathbf{J_p} &= -\,\nabla\mathbf{T}\left(\frac{\partial\mathcal{W}}{\partial\mathbf{q}}\right) \in \mathbb{R}^{N,n} \\ \mathbf{J_q} &= \nabla\mathbf{I}\left(\frac{\partial\mathcal{W}}{\partial\mathbf{q}}\right) \in \mathbb{R}^{N,n} \end{aligned}$$

Minimising $\mathbf{f}$ is usually done using the Gauss-Newton method. The main idea is to linearise, using a first order Taylor expansion, either the template around $\mathbf{p} = 0$ as

$$\mathbf{T}[\mathbf{p}+\mathbf{\Delta p}] = \mathbf{A}_0[\mathbf{p}] + \mathbf{A}[\mathbf{p}]\mathbf{c} + \mathbf{J_p}\mathbf{\Delta p}. \tag{11}$$

or the image as

$$\mathbf{I}[\mathbf{q}+\mathbf{\Delta q}] = \mathbf{I}[\mathbf{q}] + \mathbf{J_q}\mathbf{\Delta q}, \tag{12}$$

The former is called inverse framework while the latter is called forward framework. Note that the template is already linear with respect to the appearance parameter $\mathbf{c}$.

By abuse of notation we will denote $\mathbf{T}[\mathbf{p}], \mathbf{A}_0[\mathbf{p}], \mathbf{A}[\mathbf{p}]$ and $\mathbf{I}[\mathbf{q}]$ by simply $\mathbf{T}, \mathbf{A}_0, \mathbf{A}$ and $\mathbf{I}$, respectively.

B. Robust descriptors

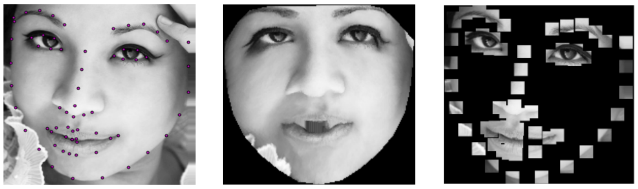
We present the results of holistic AAM and part-based AAM using robust features which prove more robust to changes in illumination and occlusion [5], [18]. This is easily done using vector notation by flattening, for each pixel, the descriptor vector and considering each of its items as additional pixels. Assuming a dense descriptor:

$$\begin{aligned} \Psi: \quad & \mathbb{R}^{\mathcal{V}} \times \mathcal{V} \to \mathbb{R}^{N_p} \\ & (I,\mathbf{v}) \longmapsto \Psi(I(\mathbf{v})). \end{aligned}$$

that maps each pixel of an image to a descriptor of size N_p, we can rewrite $\mathbf{I}$ as the vectorized flattened feature-image $[\Psi(I(\mathbf{v}))]_{\mathbf{v}\in\mathcal{V}}$. In this work we used SIFT features [19] which were shown to perform best [5]. In particular, we used a compact representation with $N_p = 8$ where each feature is extracted from an eight by eight window as in [4]. Therefore, in the rest of the paper we will simply use the term AAM to mention SIFT-AAM.

C. Parameters update

The appearance parameter update is straightforward and, at each iteration, given an update $\mathbf{\Delta c}$, the texture parameter is updated as $\mathbf{c} = \mathbf{c} + \mathbf{\Delta c}$. In the forward case, at each iteration, an update $\mathbf{\Delta p}$ is computed and the shape parameter is updated as $\mathbf{p} = \mathbf{p} + \mathbf{\Delta p}$. In the inverse case, an update $\mathbf{\Delta q}$ is estimated by deforming the template rather than the image and the update is done in an inverse compositional way as $\mathbf{p} = \mathbf{p} \circ \mathbf{\Delta q}^{-1}$. Note that in the case of a part-based AAM (or GN-DPM), composition update is equivalent to a simple addition [4] and $\mathbf{p} \circ \mathbf{\Delta q}^{-1} = \mathbf{p} - \mathbf{\Delta q}$.



(a) $\mathbf{I}$ (b) AAM: $\mathbf{I}[\mathbf{p}]$ (c) DPM: $\mathbf{I}[\mathbf{p}]$

Fig. 2. Example of an image $\mathbf{I}$ and the corresponding warped image $\mathbf{I}[\mathbf{p}]$ for an AAM (2b) and a part-based AAM (2c). Notice the deformation induced by the piecewise affine warping, a deformation that is avoided by the translational model of the part-based AAM.

V. Fast Bidirectional algorithm

We formulate here a bidirectional Gauss-Newton algorithm for AAM fitting that combines forward and inverse approaches and works by deforming both the image and the template at each iteration. Both template (11) and image (12) are linearised and the optimization is done jointly over $\mathbf{\Delta q}, \mathbf{\Delta p}$ and $\mathbf{\Delta c}$:

$$\arg\min_{\mathbf{\Delta q},\mathbf{\Delta p},\mathbf{\Delta c}} \frac{1}{2}\|\mathbf{I} + \mathbf{J_q}\mathbf{\Delta q} - \mathbf{A}_0 - \mathbf{A}\mathbf{c} - \mathbf{A}\mathbf{\Delta c} - \mathbf{J_p}\Delta\mathbf{p}\|^2_{\mathbf{W}}. \tag{13}$$

The problem is solved by capitalizing on optimization theory [17] and using:

$$\min_{x,y,z} f(x,y,z) = \min_x[\min_y[\min_z f(x,y,z)] \tag{14}$$

Therefore, (13) is first optimised with respect to $\Delta\mathbf{c}$ which yields

$$\Delta\mathbf{c} = \left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}(\mathbf{I} + \mathbf{J_q}\mathbf{\Delta q} - \mathbf{A}_0 - \mathbf{A}\mathbf{c} - \mathbf{J_p}\mathbf{\Delta p}). \tag{15}$$

Plugging the result back into (13) gives the following optimization problem:

$$\arg\min_{\mathbf{\Delta q},\mathbf{\Delta p}} \|\mathbf{I} + \mathbf{J_q}\mathbf{\Delta q} - \mathbf{A}_0 - \mathbf{J_p}\mathbf{\Delta p}\|^2_{\mathbf{P}}, \tag{16}$$

using the projection operator $\mathbf{P} = (\mathbf{W} - \mathbf{W}\mathbf{A}\left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^T)$, where, as specified earlier, we write $\|\mathbf{x}\|^2_{\mathbf{P}}$ to denote the weighted ℓ_2-norm $\mathbf{x}^T\mathbf{P}\mathbf{x}$.

We go on by optimizing (16) with respect to $\mathbf{\Delta q}$. This gives

$$\mathbf{\Delta q} = -\mathbf{H_q}^{-1}\mathbf{G_q}^T(\mathbf{I} - \mathbf{A}_0 - \mathbf{J_p}\mathbf{\Delta p}), \tag{17}$$

where the projected-out Jacobian and Hessian matrices are given by $\mathbf{G_q} = \mathbf{P}\mathbf{J_q} \in \mathbb{R}^{N,n}$ and $\mathbf{H_q} = \mathbf{G_q}^T\mathbf{G_q} \in \mathbb{R}^{n,n}$, respectively.

Next, we plug (17) into (16), to get the following optimization problem

$$\arg\min_{\mathbf{\Delta p}} \|\mathbf{I} - \mathbf{A}_0 - \mathbf{J_p}\Delta\mathbf{p}\|^2_{\mathbf{R}}, \tag{18}$$

where $\mathbf{R} = \mathbf{P}(\mathbf{E} - \mathbf{Q})$ and $\mathbf{Q} = \mathbf{G_q}\mathbf{H_q}^{-1}\mathbf{G_q}^T$. The final step is to optimize (18) with respect to $\mathbf{\Delta p}$. This gives:

$$\mathbf{\Delta p} = \mathbf{H_p}^{-1}\mathbf{G_p}^T(\mathbf{I} - \mathbf{A}_0), \tag{19}$$

where the projected-out Jacobian and Hessian matrices are given by $\mathbf{G_p} = \mathbf{R}\mathbf{J_p} \in \mathbb{R}^{N,n}$ and $\mathbf{H_p} = \mathbf{G_p}^T\mathbf{G_p} \in \mathbb{R}^{n,n}$, respectively.

Finally the shape and appearance parameters are updated as $\mathbf{q} \leftarrow \mathbf{q} \circ \mathbf{\Delta p}^{-1} + \mathbf{\Delta q}$ and $\mathbf{c} \leftarrow \mathbf{c} + \mathbf{\Delta c}$.

The overall complexity per iteration for computing these updates is readily given by $O(nmN + n^2N)$.

VI. Weighted Fast-SIC and Fast-Forward

Having introduced our new bidirectional algorithm, Fast-SIC and Fast-Forward are simply special cases of (13) obtained by ignoring some of the terms.

FAST-SIC: With the unified notations, SIC can be obtained by ignoring parameter $\mathbf{\Delta q}$ and solving the following simplified problem:

$$\arg\min_{\mathbf{\Delta p},\mathbf{\Delta c}} \frac{1}{2}\|\mathbf{I} - \mathbf{A}_0 - \mathbf{Ac} - \mathbf{A\Delta c} + \mathbf{J_p\Delta p}\|^2_{\mathbf{W}} \qquad (20)$$

By using the same strategy as for bidirectional we obtain the following update rules:

$$\mathbf{\Delta c} = \left(\mathbf{A}^T\mathbf{WA}\right)^{-1}\mathbf{A}^T\mathbf{W}(\mathbf{I} - \mathbf{A}_0 - \mathbf{Ac} - \mathbf{J_p}\Delta\mathbf{p}). \quad (21)$$

And for the shape parameter:

$$\mathbf{\Delta p} = \mathbf{H_p}^{-1}\mathbf{G_p}^T(\mathbf{W}(\mathbf{I}[\mathrm{q}] - \mathbf{T}[\mathrm{p}])), \qquad (22)$$

with $\mathbf{G_p} = \mathbf{PG_p}$ and $\mathbf{H_p} = \mathbf{G_p}^T\mathbf{G_p}$.

FAST-Forward: Similarly, we rewrite (13) by ignoring the terms in $\mathbf{\Delta p}$ and solve the following simplified problem:

$$\arg\min_{\{\mathbf{\Delta q},\mathbf{\Delta c}\}} \frac{1}{2}\|\mathbf{I} + \mathbf{J_q}\Delta\mathbf{q} - \mathbf{A}_0 - \mathbf{Ac} - \mathbf{A\Delta c}\|^2_{\mathbf{W}}, \qquad (23)$$

At each iteration, the optimal $\mathbf{\Delta c}$ is given by

$$\mathbf{\Delta c} = \left(\mathbf{A}^T\mathbf{WA}\right)^{-1}\mathbf{A}^T\mathbf{W}(\mathbf{I} + \mathbf{J}_q\Delta\mathbf{q} - \mathbf{A}_0 - \mathbf{Ac}). \quad (24)$$

The update for the shape parameters is:

$$\mathbf{\Delta q} = -\mathbf{H_q}^{-1}\mathbf{G_q}^T(\mathbf{W}(\mathbf{I}[\mathrm{q}] - \mathbf{A}_0]), \qquad (25)$$

with $\mathbf{G_q} = \mathbf{PG_q}$ and $\mathbf{H_q} = \mathbf{G_q}^T\mathbf{G_q}$.

VII. Fast Newton algorithm

Newton differs from the previous Gauss-Newton based algorithms in that it performs a Taylor expansion to the second order of the whole objective function $\mathbf{f}$ rather than simply a first order expansion on $\mathbf{\Phi}$ (in other words, it approximates the objective function with a quadratic function rather than approximating $\mathbf{\Phi}$ with a linear one).

The Newton update rules for minimising f can be obtained by solving:

$$\begin{pmatrix}\mathbf{H^f_{pp}} & \mathbf{H^f_{pc}}\\ \mathbf{H^f_{cp}} & \mathbf{H^f_{cc}}\end{pmatrix}\begin{pmatrix}\mathbf{\Delta p}\\ \mathbf{\Delta c}\end{pmatrix} = \begin{pmatrix}-\mathbf{J^f_p}^T\\ -\mathbf{J^f_c}^T\end{pmatrix}, \qquad (26)$$

We detail here the derivation of the Newton update rules for the inverse framework. The first order derivatives of f are easily derived from those of $\mathbf{\Phi}$:

$$\begin{aligned}\mathbf{J^f_p} = \mathbf{\Phi}^T\mathbf{WJ_p} &= -\,\mathbf{\Phi}^T\mathbf{W}\nabla\mathbf{T}\left(\tfrac{\partial\mathcal{W}}{\partial\mathbf{q}}\right)\\ \mathbf{J^f_c} = \mathbf{\Phi}^T\mathbf{WJ_c} &= -\,\mathbf{\Phi}^T\mathbf{WA}\end{aligned}$$

Since the Newton method uses the exact term for the Hessian and not only the Gauss-Newton approximation, we also need the second order derivatives of $\mathbf{\Phi}$ and $\mathbf{f}$.

We introduce the terms:

$$\begin{aligned}\mathbf{H_{cc}} &= \sum_{l=1}^{N}\mathbf{W}_{ll} \times g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})\frac{\partial^2 g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})}{\partial\mathbf{c}^2} = \mathbf{0}\\ \mathbf{H_{pp}} &= \sum_{l=1}^{N}\mathbf{W}_{ll} \times g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})\frac{\partial^2 g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})}{\partial\mathbf{p}^2} \in \mathbb{R}^{n,n}\\ &= -\sum_{l=1}^{N}\mathbf{W}_{ll} \times g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})\\ &\quad\times\left(\tfrac{\partial\mathcal{W}(\mathbf{v}_l,\mathbf{p})}{\partial\mathbf{q}}\right)^T\left(\nabla^2 T(\mathcal{W}(\mathbf{v},\mathbf{q}))\right)\left(\tfrac{\partial\mathcal{W}(\mathbf{v}_l,\mathbf{p})}{\partial\mathbf{p}}\right)\\ \mathbf{H_{cp}} &= \sum_{l=1}^{N}\mathbf{W}_{ll} \times g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})\frac{\partial^2 g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})}{\partial\mathbf{c}\partial\mathbf{p}} \in \mathbb{R}^{m,n}\\ &= -\sum_{l=1}^{N}\mathbf{W}_{ll} \times g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})\nabla A(\mathcal{W}(\mathbf{v}_l,\mathbf{p}))\left(\tfrac{d\mathcal{W}}{\partial\mathbf{q}}\right)\end{aligned}$$

From these we easily get the Hessians of $\mathbf{f}$:

$$\begin{aligned}\mathbf{H^f_{pp}} &= \mathbf{H_{pp}} + \mathbf{J_p}^T\mathbf{WJ_p}\\ \mathbf{H^f_{cp}} &= \mathbf{H_{cp}} + \mathbf{J_c}^T\mathbf{WJ_p}\\ \mathbf{H^f_{cc}} &= \mathbf{J_c}^T\mathbf{WJ_c} = \mathbf{A}^T\mathbf{WA}\end{aligned}$$

Note that $\forall l \in \{1,\cdots,N\}$, $\mathbf{W}_{ll} \times \frac{\partial^2 g(\mathbf{v}_l,\mathbf{p},\mathbf{q},\mathbf{c})}{\partial\mathbf{c}\partial\mathbf{p}}$ can be precomputed, leaving only a dot product to compute at each iteration. The cost of computing $\mathbf{H^f_{cp}}$ is therefore $O(mnN)$. The computational cost of $\mathbf{H^f_{pp}}$ is simply $O(n^2N)$.

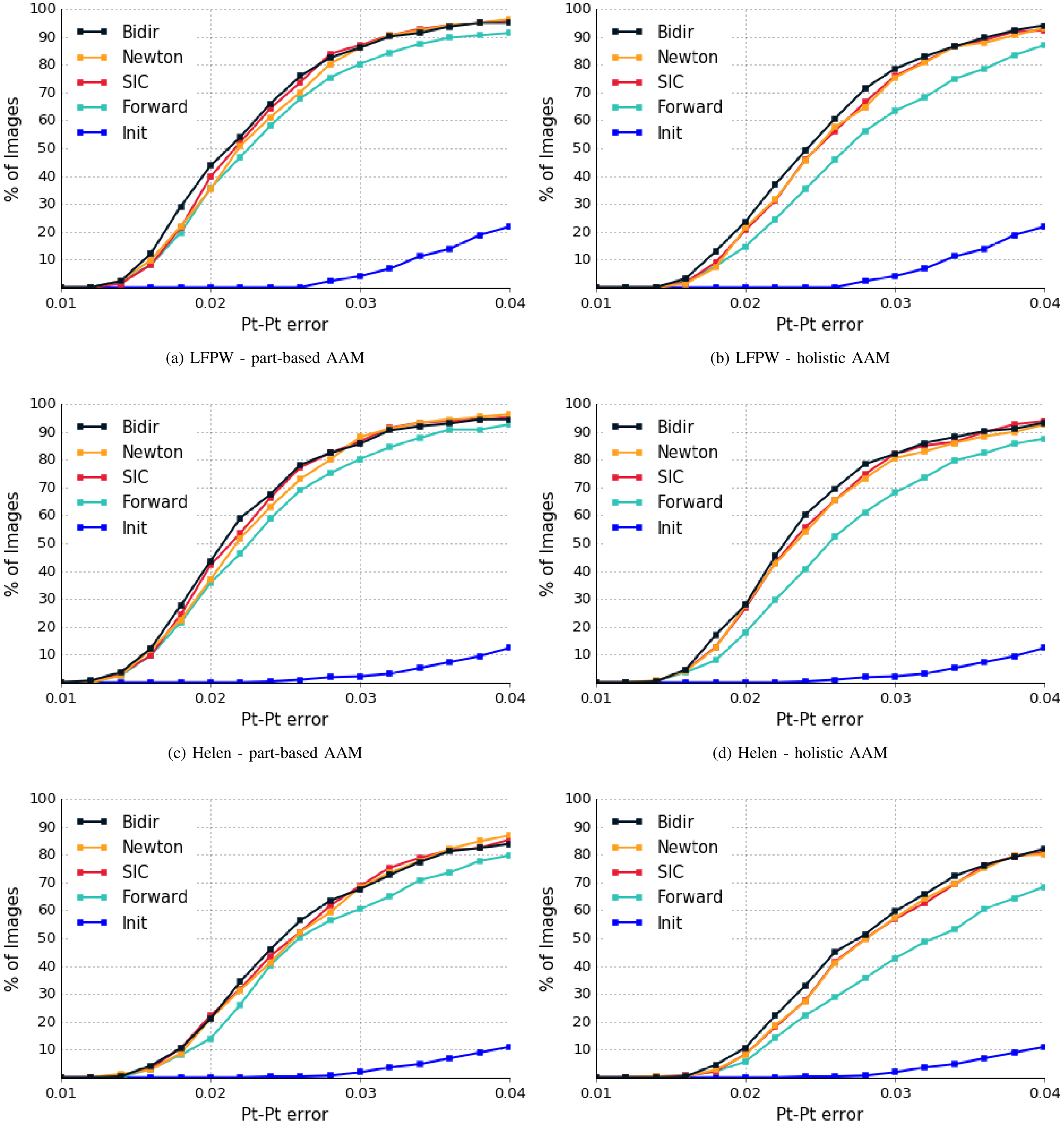
We can now solve the original optimisation problem (26) and, using Schur's complement, the following update rules are obtained:

$$\begin{aligned}\mathbf{\Delta p} &= \left(\mathbf{H^f_{pp}} - \mathbf{H^f_{pc}}\mathbf{H^f_{cc}}^{-1}\mathbf{H^f_{cp}}\right)^{-1}\left(-\mathbf{J^f_p}^T + \mathbf{H^f_{pc}}\mathbf{H^f_{cc}}^{-1}\mathbf{J^f_c}^T\right),\\ \mathbf{\Delta c} &= \mathbf{H^f_{cc}}^{-1}\left(-\mathbf{J^f_c}^T - \mathbf{H^f_{cp}}\mathbf{\Delta p}\right).\end{aligned}$$

The Gauss-Newton method can be derived from the Newton formulation by simplifying the second order terms and approximating the Hessian as $\mathbf{H^{GN}_{pp}} = \mathbf{J_p}^T\mathbf{WJ_p}$ and $\mathbf{H^{GN}_{cp}} = \mathbf{J_c}^T\mathbf{WJ_p}$.

VIII. Experimental results

In this section we provide a comprehensive comparison of holistic Active Appearance Models and Part-Based Active Appearance Models for all four fitting algorithms: Fast-Forward (*Forward*), Fast-SIC (*SIC*), Fast-Newton (*Newton*) and Fast-Bidirectional (*Bidir*). We test the methods on three challenging datasets recently annotated with 68 landmarks in the same configuration as the Multi-Pie dataset: LFPW [20], Helen [21] and AFW [22]. We compare our Part-Based AAM with other state-of-the-art methods and with all competitors of the recently held 300 Faces In-The-Wild Challenge [23].



(a) LFPW - part-based AAM

(b) LFPW - holistic AAM

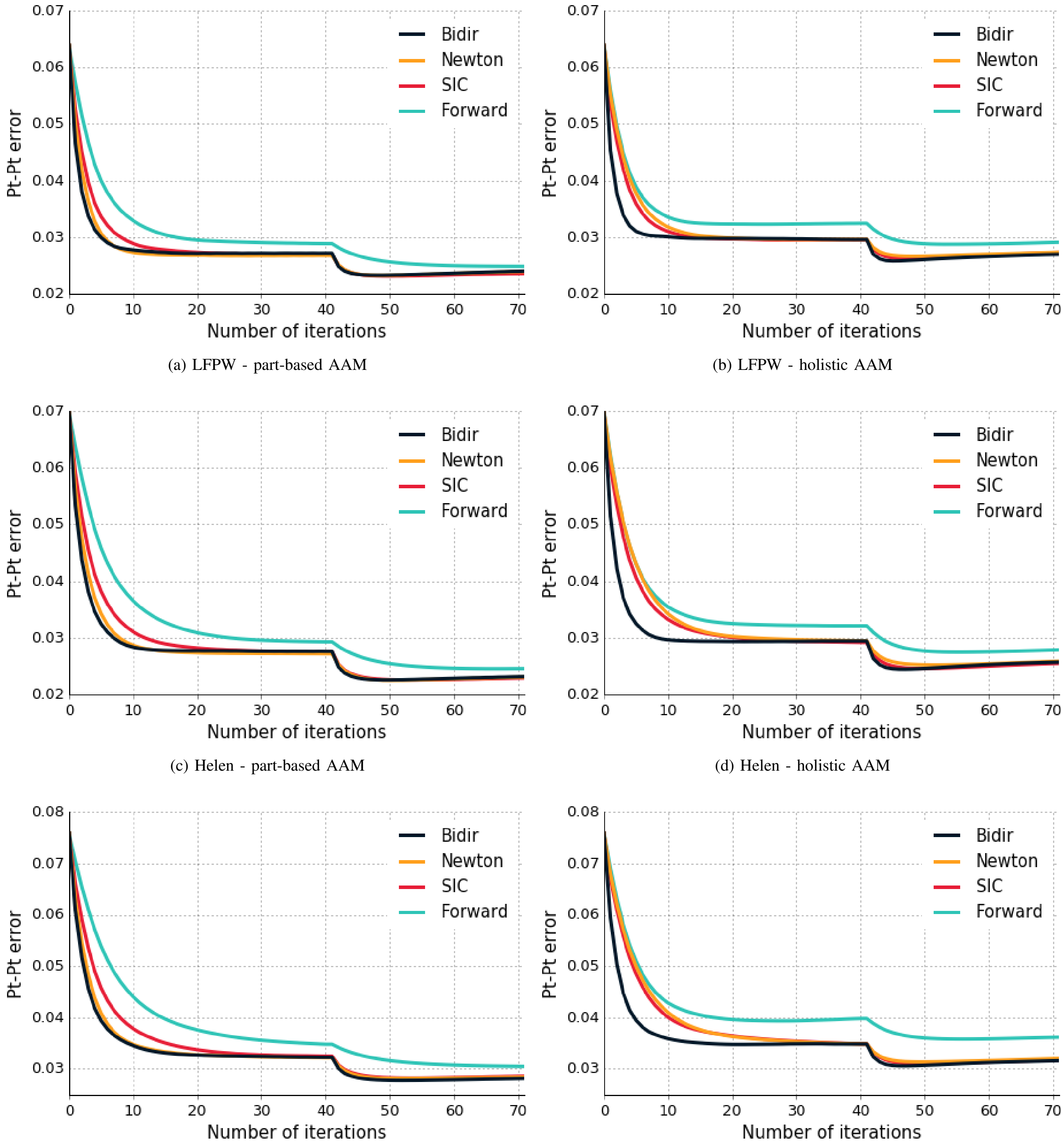
(c) Helen - part-based AAM

(d) Helen - holistic AAM

(e) AFW - part-based AAM

(f) AFW - holistic AAM

Fig. 3. Performance on 68 points for a small noise in the initialisation for part-based AAM (first column) and holistic AAM (second column)



(a) LFPW - part-based AAM

(b) LFPW - holistic AAM

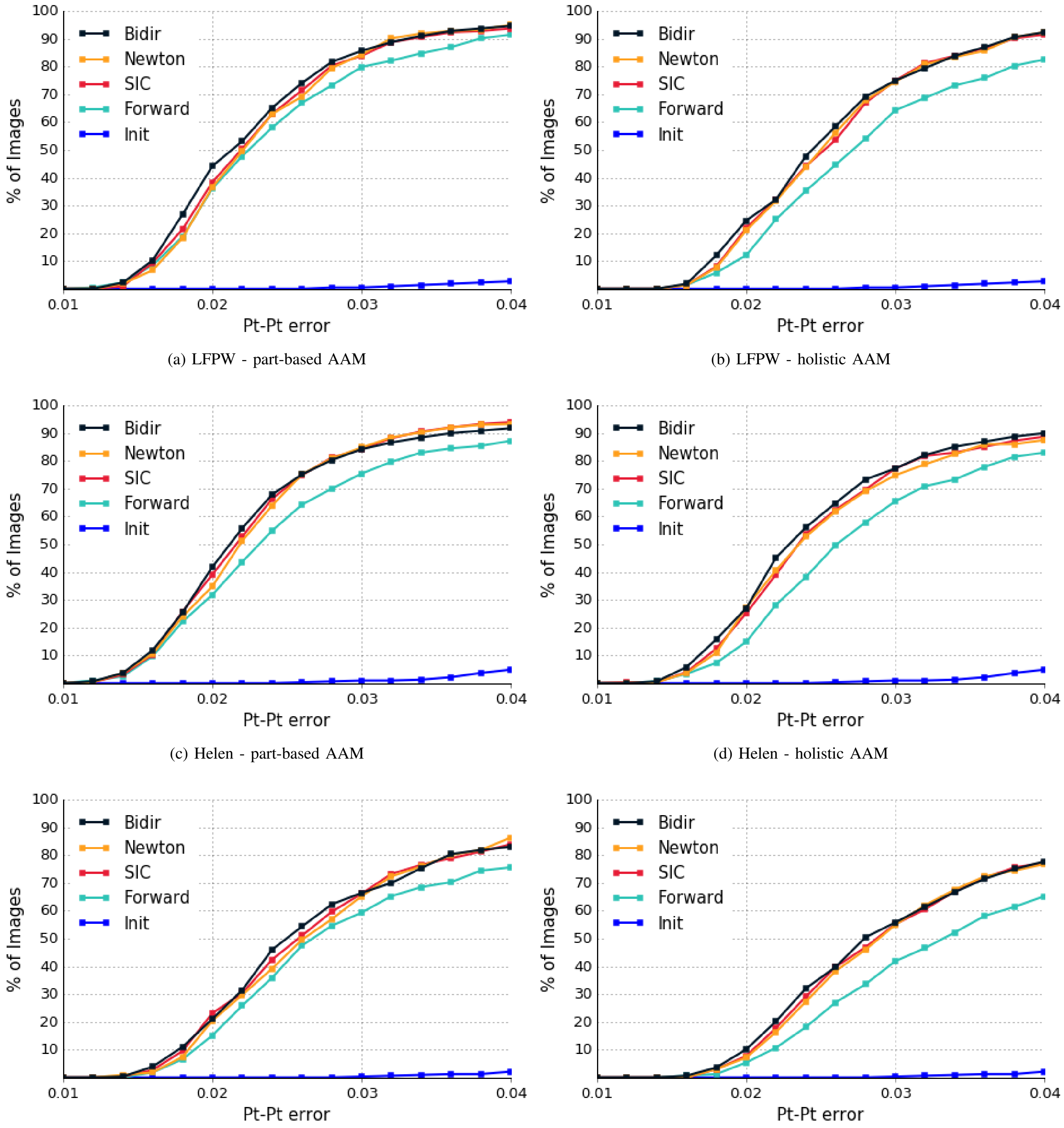
(c) Helen - part-based AAM

(d) Helen - holistic AAM

(e) AFW - part-based AAM

(f) AFW - holistic AAM

Fig. 4. Convergence on 68 points for a small noise in the initialisation for part-based AAM (first column) and holistic AAM (second column)



(a) LFPW - part-based AAM

(b) LFPW - holistic AAM

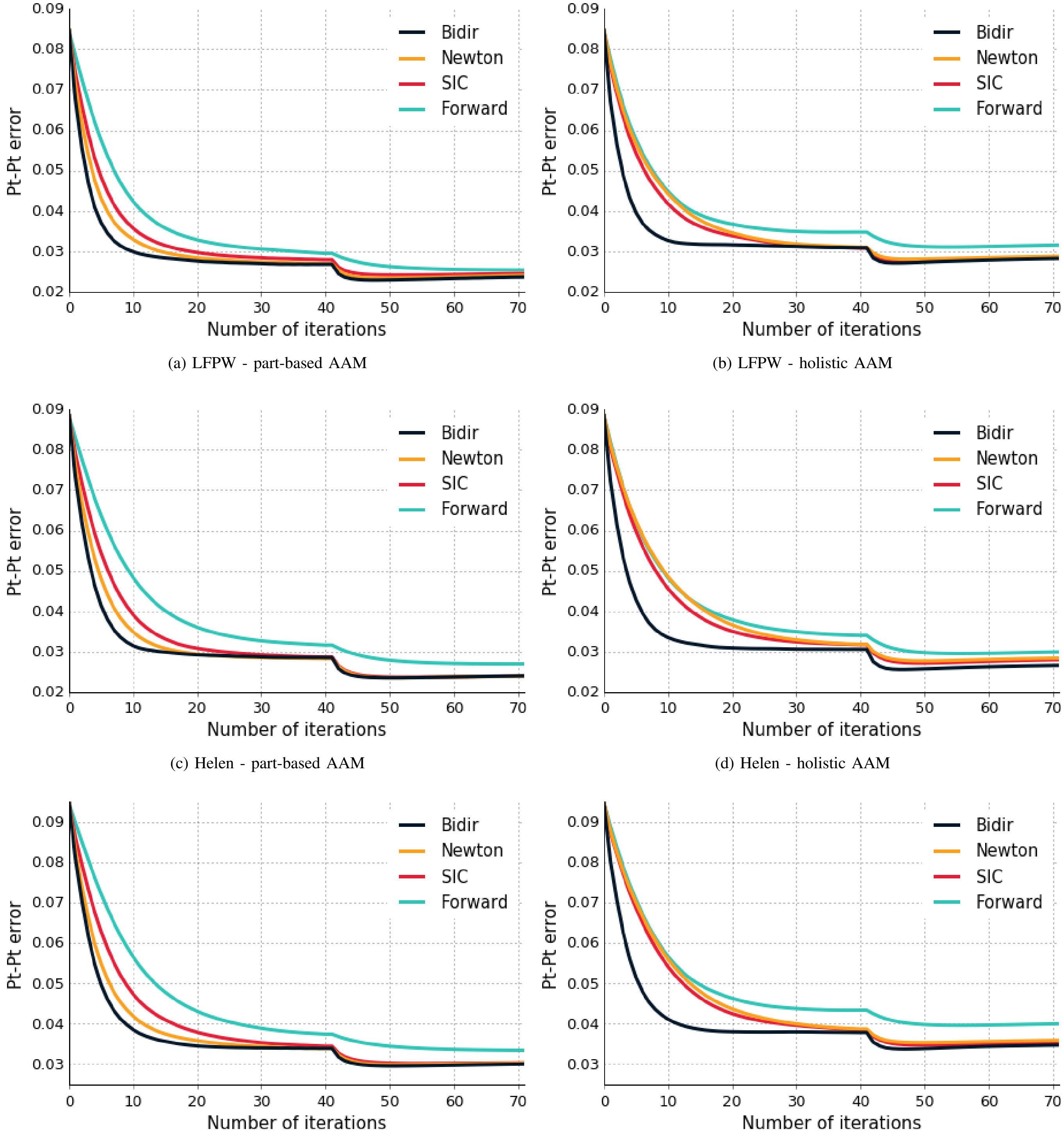
(c) Helen - part-based AAM

(d) Helen - holistic AAM

(e) AFW - part-based AAM

(f) AFW - holistic AAM

Fig. 5. Performance on 68 points for a large noise in the initialisation for part-based AAM (first column) and holistic AAM (second column)



(a) LFPW - part-based AAM

(b) LFPW - holistic AAM

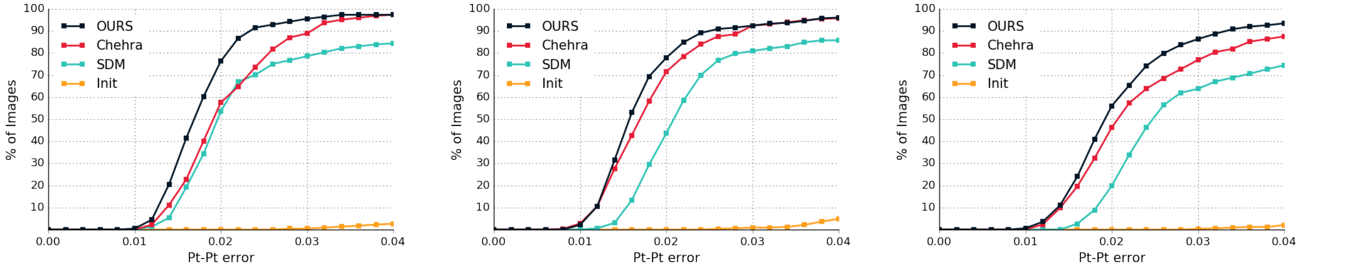
(c) Helen - part-based AAM

(d) Helen - holistic AAM

(e) AFW - part-based AAM

(f) AFW - holistic AAM

Fig. 6. Convergence on 68 points for a large noise in the initialisation for part-based AAM (first column) and holistic AAM (second column)



(a) LFPW - part-based AAM

(b) Helen - holistic AAM

(c) AFW - part-based AAM

Fig. 7. Comparison with the state-of-the art. Our bidirectional part-based AAM largely outperforms both SDM (intraface) [3] and Chehra [13].

A. *Experimental setting*

We conducted two different sets of experiments: first we compare all fitting algorithms presented in the paper for both Part-Based and Holistic AAM on three challenging datasets. In each case we initialised the algorithm using the bounding-box from the face-detector [22]. To make the experiments more realistic, and in order to empirically evaluate the robustness of each method, we added some random translation and scaling to the initialisation, defined by a standard deviation σ_{noise}, following the same protocol as in [24]. We tested two scenarios: adding a small ($\sigma_{\text{noise}} = 1.5$) and a larger ($\sigma_{\text{noise}} = 3$) amount of random noise to these initialisations. Note that the noise is different for each image but the same for each method to allow for a fair comparison.

Second we compare against the state of the art of these three datasets for available state-of-the-art regression methods (Fig. 7) and on the 300-Faces In-The-Wild challenge against all the competitors, both industry and academia for both 51 and 68 points (Figures 8 and 9).

In the whole paper, the performance is measured in terms of the well-established normalised point-to-point error introduced in [22] and defined as the RMS error normalised by the face size (*pt-pt-error*) (Figs 3, 5, 7, 8, 9). We also evaluate the convergence speed of the AAM fitting algorithms by measuring the averaged normalised point-to-point error over the whole dataset, at each iteration (Figs 4, 6). We report results in performance and convergence for both Part-Based and Holistic AAM for small and large noise in the initialisation.

B. *General observations*

Our Bidirectional Part-Based AAM matches or out-performs other state-of-the-art methods. It also largely out-performs classical AAMs, especially for accurately capturing the boundaries. Reconstructing these boundaries seems to be the hardest part for all methods and especially for the Forward algorithm. Bidirectional performs better than the other fitting methods while having superior convergence properties and being more robust to noise. The difference between SIC and Bidirectional is observed for both holistic and part-based AAM and is especially large for holistic AAMs. Newton performs similarly to SIC with better convergence properties in the case of part-based AAMs. As expected, it performs very well in the vicinity of the solution with a slight decrease in performance as the initialisations become more noisy. However, given enough iterations, all methods seem to converge to more similar solutions.

C. *Implementation details*

Holistic AAM: To increase performance, we used a multi-resolution approach with two levels. The lower level has $m = 50$ appearance vectors and $n = 11$ shape vectors while the higher level has $m = 400$ appearance vectors and $p = 25$ shape vectors. We used a step of 2 effectively dividing by two the number of features.

Part-based AAM: We again used a pyramid of two levels with $m = 70$ appearance vectors and $n = 15$ shape vectors in the lower level. The higher level has $m = 200$ appearance vectors and $p = 25$ shape vectors. We also used a step of 4 effectively dividing by four the number of features. We found that Part-Based AAM worked as well with a larger step (here a step of 4) while the holistic model requires a smaller step (here a step of 2).

These parameters were obtained by performing a randomised grid-search over a small set of parameters and a small validation set. In all cases, both holistic AAM and part-based AAM where trained using the training sets of LFPW [20] and Helen [21]. Note that we never compute derivatives for all pixels $\mathbf{v} \in \mathcal{V}$ but only for the subset of pixels $\{\mathbf{v}_l, l \in \{1, \cdots, N\} \wedge \mathbf{W}_{ll} = 1\}$, i.e. we only store the points for which the corresponding weight is not null. That makes all our algorithms computationally much more efficient. On a standard desktop configuration, initializing the method with [22] takes on average 20 seconds per image, due to the nature of the algorithm, while an iteration of the Holistic and Part-Based AAM takes less than a second.

D. *Small noise*

We present results for 68 facial landmarks, which include boundary points[1]. This is particularly interesting as the boundary points are significantly harder to accurately detect and sometimes ill-defined, in particular for challenging cases such as those with large poses.

For Part-based AAM, we notice that in all cases, Forward performs slightly worse than others methods, Fig 3. We believe this is due to the fact that unlike for other methods, the

[1] We also performed the same experiment with only 49 interior points and arrived to similar conclusions.

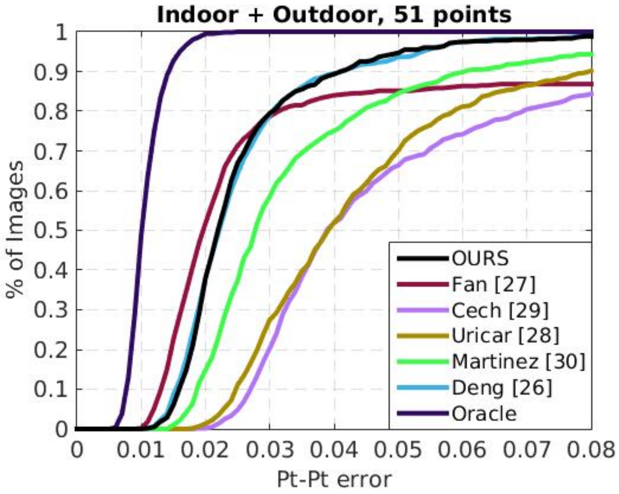


Fig. 8. Results obtained with our Part-based AAM on the 300 Faces In-the-Wild challenge for 51 points.

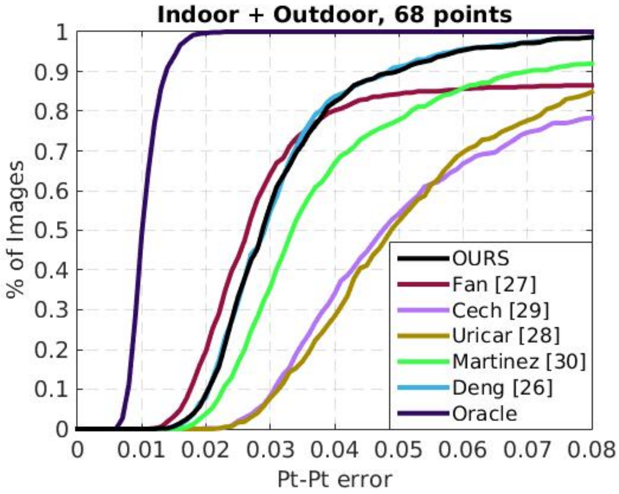


Fig. 9. Results obtained with our Part-based AAM on the 300 Faces In-the-Wild challenge for 68 points.

gradients are extracted directly from the image and not reconstructed with a learned linear model. Therefore, the original boundaries can be potentially far off the correct solution and therefore be very different from the gradients learned from actual faces. Bidirectional consistently outperforms or matches the performance of other methods on LFPW (Fig 3a), Helen (Fig 3c) and AFW (Fig 3e), with a slight advantage for small errors. In term of convergence, Fig 4, there is a clear hierarchy, with bidirectional and Newton both converging much faster in all cases. Forward converges much slower in comparison.

Similar observations can be made for Holistic AAM for which the relative performance of the methods is very similar although the overall fitting accuracy is slightly worse. The advantage of bidirectional is even more noticeable in the case of holistic AAM, where it performs better than all other methods while its convergence advantage is even clearer, while Newton still performs as well as SIC but this time does not match the convergence speed of the bidirectional method.

E. Large noise

We noticed that, for small amounts of noise in the initialisation, SIC and Newton clearly behave best, with SIC following closely and Forward performing worst, on all datasets, for both performance and convergence. However, when increasing the noise, Fig 5, the performance of all methods decrease, but Bidirectional still converges much faster than SIC and Newton's while out-performing them (Fig 5a) or at least matching their performance (Figs 5c, 5e). As theoretically expected, the performance of the Newton method slightly deteriorates, making it significantly slower than Bidirectional, but still faster than SIC and Forward. Similar observations can be made for holistic AAMs with an even more impressive convergence speed for bidirectional which now clearly out-performs all other method in both fitting accuracy and convergence speed.

Finally, Fig 10 shows some representative examples of images taken from AFW along with the initialisation used and the fitted results obtained using this initialisation for each method.

F. Comparison with the state-of-the-art

We provide a comparison of our part-based AAM with state-of-the art methods. Fig 7 shows a comparison of our method with SDM [3] and Chehra [13] on all LFPW, Helen and AFW for $\sigma_{\text{noise}} = 3$. The comparison was done in the same setting as the previous experiments, using the same bounding-box initialisations for all methods and $\sigma_{\text{noise}} = 3$. Results are for the 49 interior points since these are the only landmarks returned by SDM and Chehra. Our method performs significantly better than both methods on all three datasets.

We also compare our methods to the recently published 300 Faces In-The-Wild challenge [23], on both outdoor and indoor images, for 51 and 68 points, Fig 8 and 9. We used the same performance metric as in the previous experiments and obtained the performance curves with that metric for the other methods directly from the organisers of the competition [23]. In order to handle the very large pose present in some of the challenge images, we trained three part-based AAMs, one for approximately frontal poses and two for extreme poses. We used the DPM head detector of [25], [22] to estimate the pose and initialise one of three pose-specific part-based AAMs. As can be seen from Figures 8 and 9, our part-based AAM performs remarkably well. Its performance is on par with that of Deng et al. [26], without employing any complicated multiple initialisation scheme. The work in [26] used a multi-view, multi-scale and multi-component cascade shape regression model using multi-scale HOG. So, to wit, the performance of the work in [26] is not due to the suitability of the proposed model to the task of facial landmark detection so much as it is due to complex engineering of the used algorithm which could also be used in our formulation, but this falls beyond the scope of this paper. On the other hand, the work in [27] outperforms our method in the case of very small errors. However, the opposite is the case for any error larger than 0.02. This is to be expected as the work in [27] is a submission from industry (Megvii company) using cascaded Deep Convolutional Neural Networks trained on undisclosed

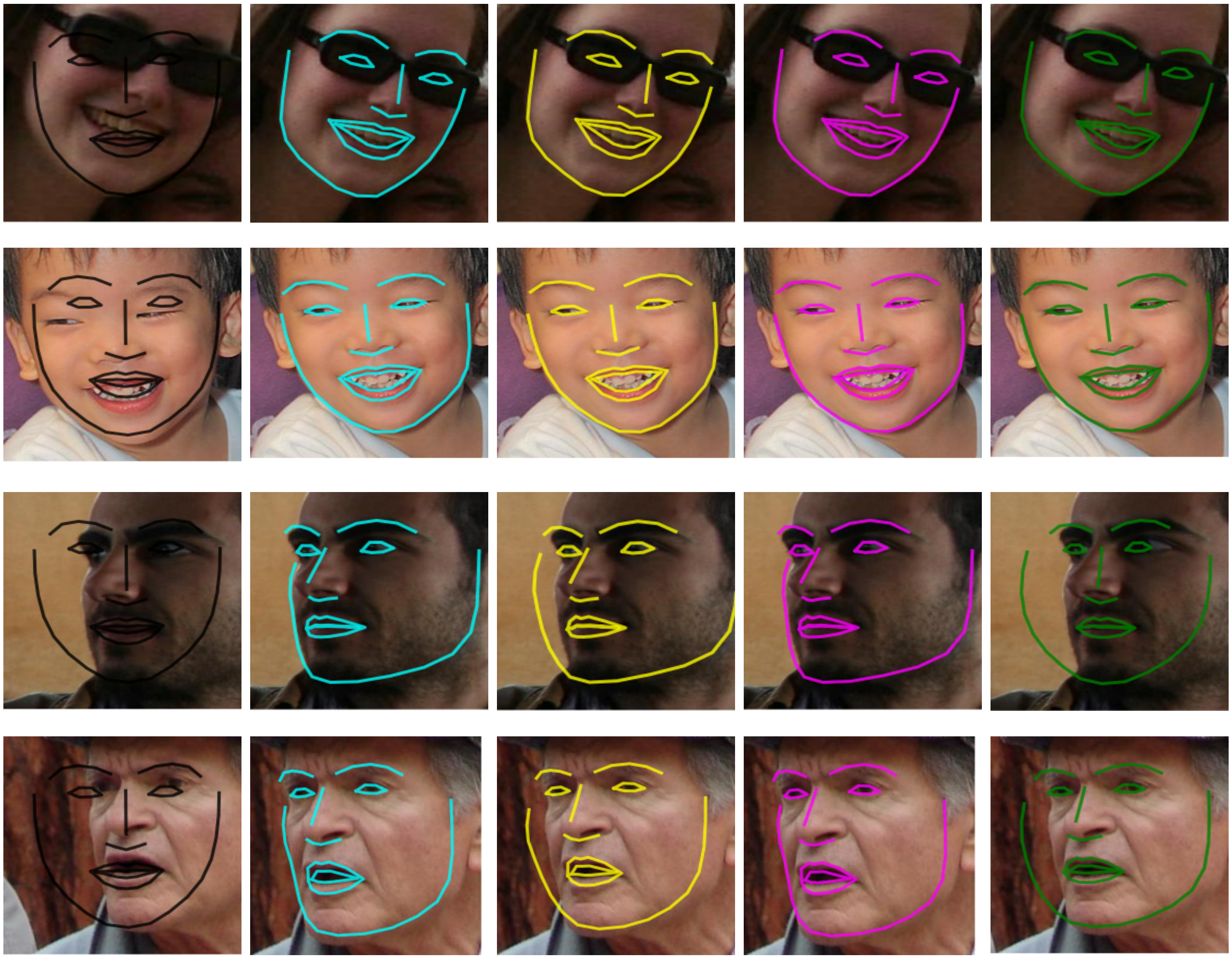


Fig. 10. Example of fitting results from the AFW dataset [22] obtained with a DPM. From left to right: initialisation (black), bidirectional (blue), SIC (yellow), newton (magenta) and forward (green). As illustrated, our SIFT-DPM performs remarkably well even for in-the wild images presenting challenging conditions of illumination, pose and occlusion, even with bad initialisations. Although Forward generally has a tendency to not reconstruct the boundary as well (row 3), it sometimes captures information missed by SIC, while sometimes both SIC and Forward fail to accurately locate the boundary pixels (row 4). Bidirectional advantageously combines the two approaches allowing it to locate correctly locate landmarks when SIC or forward fail (rows 3 and 4). Finally, Newton uses the Hessian to avoid local minimums and in some cases converges to a better solution (rows 3 and 4).

datasets. In [28], a coarse-to-fine with a near frontal DPM is used and learned using structured output SVM, while [29] used a commercial face detector to initialise a structured output SVM-based method that fits a 3D shape model. Finally, [30] used a cascade of regressors modified to use an $\ell_{2,1}$ norm and multiple initialisations. Our method outperforms all these methods while using three models trained on less than 1000 images and using the output of the DPM for initialisation.

IX. Conclusion

We proposed a unified framework for solving both holistic and part-based Active Appearance Models, in which we formulated new Bidirectional and Newton methods. We showed how to exploit the structure of the problem in order to derive exact and computationally efficient algorithms and extended them to handle robust features. We provided a comprehensive study of the performance and convergence of all fitting algorithms for both models on three highly challenging datasets and additionally provided comparison with other methods on these and on the recently published 300 Faces In-The-Wild Challenge database. Our Fast Bidirectional and Fast Newton part-based AAM out-perform or match the performance of other State-of-the-Art methods such as regression, while having superior convergence properties compared to existing AAM fitting algorithms. Going forward, we are planning to extend the same Bidirectional and Newton fitting strategies to the work of [24].

Acknowledgment

This work has been funded by the European Community Horizon 2020 [H2020/2014-2020] under grant agreement no. 645094 (SEWA). The work of Jean Kossaifi is also funded in part by the European Community 7th Framework Programme [FP7/2007-2013] under grant agreement no. 611153 (TERESA).

References

[1] G. J. Edwards, C. J. Taylor, and T. F. Cootes, “Interpreting face images using active appearance models,” in *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 1998, pp. 300–305.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 23, no. 6, 2001, pp. 681–685.

[3] X. Xiong and F. D. la Torre, “Supervised descent method and its applications to face alignment,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 532–539.

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

[5] E. Antonakos, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou, “Feature-based lucas-kanade and active appearance models,” *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2617–2632, 2015.

[6] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th international joint conference on Artificial intelligence (IJCAI)*, 1981, pp. 674 – 670.

[7] 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 (PAMI)*, vol. 20, no. 10, pp. 1025–1039, 1998.

[8] I. Matthews and S. Baker, “Active appearance models revisited,” *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 135 – 164, November 2004.

[9] S. Baker, R. Gross, and I. Matthews, “Lucas-kanade 20 years on: A unifying framework: Part 3,” no. CMU-RI-TR-03-35, November 2003.

[10] R. Gross, I. Matthews, and S. Baker, “Generic vs. person specific active appearance models,” *Image and Vision Computing (IVC)*, vol. 23, no. 12, pp. 1080–1093, 2005.

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

[12] ——, “Fast algorithms for fitting active appearance models to unconstrained images,” *International Journal of Computer Vision*, pp. 1–17, 2016.

[13] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, “Incremental face alignment in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 1859–1866.

[14] J. Shen, S. Zafeiriou, G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, “The first facial landmark tracking in-the-wild challenge: Benchmark and results,” in *Proceedings of IEEE International Conference on Computer Vision, 300 Videos in the Wild (300-VW): Facial Landmark Tracking in-the-Wild Challenge & Workshop (ICCVW’15)*, December 2015, pp. 50–58.

[15] J. Kossaifi, G. Tzimiropoulos, and M. Pantic, “Fast newton active appearance models,” in *Proceedings of the IEEE Intl Conf. on Image Processing (ICIP)*, Paris, France, October 2014, pp. 1420–1424.

[16] ——, “Fast and exact bi-directional fitting of active appearance models,” in *Proceedings of the IEEE Intl Conf. on Image Processing (ICIP)*, Quebec City, QC, Canada, September 2015, pp. 1135–1139.

[17] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[18] A. Asthana, S. Zafeiriou, G. Tzimiropoulos, S. Cheng, and M. Pantic, “From pixels to response maps: Discriminative image filtering for face alignment in the wild,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 37, no. 6, pp. 1312–1320, 2015.

[19] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[20] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, “Localizing parts of faces using a consensus of exemplars,” in *The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 545–552.

[21] J. B. F. Zhou and Z. Lin, “Exemplar-based graph matching for robust facial landmark localization,” in *IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1025–1032.

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

[23] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: Database and results,” *Image and Vision Computing (IVC), Special Issue on Facial Landmark Localisation ”In-The-Wild”*, vol. 47, pp. 3–18, 2016.

[24] G. Tzimiropoulos, “Project-out cascaded regression with an application to face alignment,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3659–3667.

[25] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, “Robust discriminative response map fitting with constrained local models,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3444–3451.

[26] J. Deng, Q. Liu, J. Yang, and D. Tao, “{M3} csr: Multi-view, multi-scale and multi-component cascade shape regression,” *Image Vision Computing (IVC)*, vol. 47, no. C, pp. 19–26, 2016.

[27] H. Fan and E. Zhou, “Approaching human level facial landmark localization by deep learning,” *Image Vision Computing (IVC)*, vol. 47, no. C, pp. 27–35, 2016.

[28] M. Uřičář, V. Franc, D. Thomas, A. Sugimoto, and V. Hlavac, “Multi-view facial landmark detector learned by the structured output svm,” *Image and Vision Computing (IVC)*, pp. 45–59, 2015.

[29] J. Čech, V. Franc, M. Uřičář, and J. Matas, “Multi-view facial landmark detection by using a 3d shape model,” *Image and Vision Computing (IVC)*, pp. 60–70, 2015.

[30] B. Martinez and M. F. Valstar, “L2,1-based regression and prediction accumulation across views for robust facial landmark detection,” *Image and Vision Computing (IVC), Special Issue on Facial Landmark Localisation ”In-The-Wild”*, pp. 36–44, 2016.

Jean Kossaifi is a Research Assistant and PhD student within the Department of Computing, Imperial College London, working as part of the iBUG group under Professor Maja Pantic’s supervision. His current position follows the completion of an MSc in Advanced Computing, obtained with Distinction from Imperial College London, UK. In addition, Jean also holds a French Engineering Diploma/MSc in applied mathematics, finance and computing, obtained in parallel with a BSc in Advanced Mathematics. His research interests are primarily focused on the areas of machine learning, computer vision and pattern recognition, with applications in human-computer interaction, automatic non-verbal behaviour analysis and emotion recognition.

Georgios (Yorgos) Tzimiropoulos received the M.Sc. and Ph.D. degrees in Signal Processing and Computer Vision from Imperial College London, U.K. Following that, he was Post-Doc researcher in the iBUG group at Imperial College London. He is currently Assistant Professor with the School of Computer Science at the University of Nottingham, U.K. He is Associate Editor of the Image and Vision Computing Journal. He has worked on the problems of object detection and tracking, alignment and pose estimation, and recognition with humans and faces being the focal point of his research. His current focus is on Deep Learning.

Maja Pantic is a professor in affective and behavioral computing in the Department of Computing at Imperial College London, United Kingdom, and in the Department of Computer Science at the University of Twente, the Netherlands. She currently serves as the editor in chief of Image and Vision Computing Journal and as an associate editor for both the IEEE Transactions on Pattern Analysis and Machine Intelligence and the IEEE Transactions on Affective Computing. She has received various awards for her work on automatic analysis of human behavior, including the European Research Council Starting Grant Fellowship 2008 and the Roger Needham Award 2011. She is a fellow of the IEEE.