Feature-Based Lucas-Kanade and Active Appearance Models

Epameinondas Antonakos, Joan Alabort-i-Medina, *Student Member, IEEE,*
Georgios Tzimiropoulos, and Stefanos P. Zafeiriou, *Member, IEEE,*

***Abstract*—Lucas-Kanade and Active Appearance Models are among the most commonly used methods for image alignment and facial fitting, respectively. They both utilize non-linear gradient descent, which is usually applied on intensity values. In this paper, we propose the employment of highly-descriptive, densely-sampled image features for both problems. We show that the strategy of warping the multi-channel dense feature image at each iteration is more beneficial than extracting features after warping the intensity image at each iteration. Motivated by this observation, we demonstrate robust and accurate alignment and fitting performance using a variety of powerful feature descriptors. Especially with the employment of HOG and SIFT features, our method significantly outperforms the current state-of-the-art results on in-the-wild databases.**

***Index Terms*—Lucas-Kanade, Active Appearance Models, dense image feature descriptors, face alignment, face fitting**

I. Introduction

DUE to their importance in Computer Vision and Human-Computer Interaction, the problems of face alignment and fitting have accumulated great research effort during the past decades. The Lucas-Kanade (LK) algorithm [1] is the most important method for the problem of aligning a given image with a template image. The method's aim is to find the parameter values of a parametric motion model that minimize the discrepancies between the two images. Active Appearance Models (AAMs) are among the most commonly used models for the task of face fitting. They are generative deformable statistical models of shape and appearance variation. AAMs were introduced in [2] and they are descendants of Active Contour Models [3] and Active Shape Models [4]. Among the most efficient techniques to optimize AAMs is gradient descent, which recovers the parametric description of a face instance. Gradient descent optimization for AAMs is similar to the LK algorithm, with the difference that the registration is obtained between the input image and a parametric appearance model instead of a static template.

The most common choice for both LK and AAMs matching is the Inverse Compositional (IC) image alignment algorithm [5], [6]. IC is a non-linear, gradient descent optimization technique that aims to minimize the ℓ_2 norm between the warped image texture and a target texture. The target texture is the static template image in the case of LK and a model texture instance in the case of AAMs. Since IC is a gradient descent optimization technique, the registration result is sensitive to initialization and to appearance variation (illumination, object appearance variation, occlusion etc.) exposed in the input and the target images [7]. Especially, in the case of intensity-based AAMs, the model is incapable of adequately generalizing in order to be robust to outliers. Many approaches have been proposed to deal with these issues and improve efficiency [5], [8]–[13], robustness [12], [14]–[19] and generalization [18], [20], [21]. Many of the proposed methods introduce algorithmic improvements. The authors in [8] propose an adaptation on the fitting matrix and the employment of prior information to constrain the IC fitting process. In [7], [19] the ℓ_2 norm is replaced by a robust error function and the optimization aims to solve a re-weighted least squares problem with an iterative update of the weights. Moreover, the method in [15] aligns two images by maximizing their gradient correlation coefficient. However, most of the proposed methods utilize an intensity-based appearance, which is not suitable to create a generic appearance model and achieve accurate image alignment, as is also shown in our experiments.

In this paper, we propose the employment of highly-descriptive, *dense* appearance features for both LK and AAMs. We show that even though the employment of dense features increases the data dimensionality, there is a small raise in the time complexity and a significant improvement in the alignment accuracy. We show that within the IC optimization, there is no need to compute the dense features at each iteration from the warped image. On the contrary, we extract the dense features from the original image once and then warp the resulting multi-channel image at each iteration. This strategy gives better results, as shown in our motivating experiment of Sec. V-A1 and has smaller computational complexity, as explained in Sec. IV and Tab. II. Motivated by this observation, we present very accurate and robust experimental results for both face alignment and fitting with feature-based LK and AAMs, that prove their invariance to illumination and expression changes and their generalization ability to unseen faces. Especially in the case of HOG and SIFT AAMs, we demonstrate results on in-the-wild databases that significantly outperform the current state-of-the-art performance.

Feature-based image representation has gained extended attention for various Computer Vision tasks such as image segmentation and object alignment/recognition. There is ongoing research on the employment of features for both LK [11], [15], [16] and AAMs [16], [21]–[31]. The authors

E. Antonakos, J. Alabort-i-Medina and S. Zafeiriou are with the Department of Computing, Imperial College London, London, SW7 2AZ, U.K. (e-mail: e.antonakos@imperial.ac.uk; ja310@imperial.ac.uk; s.zafeiriou@imperial.ac.uk).

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

in [11] use correspondences between dense SIFT descriptors for scene alignment and face recognition. Various appearance representations are proposed in [25], [26] to improve the performance of AAMs. One of the first attempts for feature-based AAMs is [22]. The authors use novel features based on the orientations of gradients to represent edge structure within a regression framework. Similar features are employed in [21] to create a robust similarity optimization criterion. In [27], the intensities appearance model is replaced by a mixture of grayscale intensities, hue channel and edge magnitude.

Recently, more sophisticated multi-dimensional features are adopted for AAM fitting. The work in [16] proposes to apply the IC optimization algorithm in the Fourier domain using the Gabor responses for LK and AAMs. This is different than the framework proposed in this paper, since in our approach the optimization is carried out in the spatial domain. In [28], a new appearance representation is introduced for AAMs by combining Gabor wavelet and Local Binary Pattern (LBP) descriptor. The work in [23] is the closest to the proposed framework in this paper. The authors employ Gabor magnitude features summed over either orientations or scales or both to build an appearance model. However, even though the optimization is based on the IC technique and carried out in the spatial domain, features are extracted at each iteration from the warped image. Finally, similarly to [23], the authors in [24] model the characteristic functions of Gabor magnitude and phase by using lognormal and Gaussian density functions respectively and utilize the mean of the characteristics over orientations and scales.

The framework proposed in this paper differs from the above works in various aspects. We adopt the concept of highly-descriptive, densely-sampled features within the IC optimization and utilize multi-channel warping at each iteration of the IC optimization which does not greatly increase the computational complexity but significantly improves the fitting performance and robustness. In our previous work [32], we showed that the combination of AAMs with HOG features results in a powerful model with excellent performance. Herein, we apply the above concept for both LK and AAMs by using a great variety of widely-used features, such as Histograms of Oriented Gradients (HOG) [33], Scale-Invariant Feature Transform (SIFT) [34], Image Gradient Orientation kernel (IGO) [15], [20], Edge Structure (ES) [22], Local Binary Patterns (LBP) [35]–[37] with variations [38], and Gabor filters [39]–[41]. We extensively evaluate the performance and behaviour of the proposed framework on the commonly used Yale B Database [42] for LK and on multiple in-the-wild databases (LFPW [43], AFW [44], Helen [45], iBUG [46]) for AAMs. Finally, we compare with the current state-of-the-art methods [47], [48] and report more accurate results.

To summarize, the contributions of this paper are:

- We propose the incorporation of densely-sampled, highly-descriptive features in the IC gradient descent framework. We show that the combination of (1) non-linear least-squares optimization with (2) robust features (e.g. HOG/SIFT) and (3) generative models can achieve excellent performance for the task of face alignment.
- We elaborate on the reasons why it is preferable to warp the features image at each iteration, rather than extracting features at each iteration from the warped image, as it is done in the relevant bibliography.
- Our extended experimental results provide solid comparisons between some of the most successful and widely-used features that exist in the current bibliography for the tasks of interest, by thoroughly investigating the features' accuracy, robustness, and speed of convergence.
- Our proposed HOG and SIFT AAMs outperform current state-of-the-art face fitting methods on a series of cross-database challenging in-the-wild experiments.
- An open-source Python implementation of the described methods is provided in the Menpo Project[1] [49].

The rest of the paper is structured as follows: Section II briefly describes the used features. Section III elaborates on the intensity-based IC algorithm for LK and AAMs. Section IV explains the strategy to combine the IC optimization with dense features. Finally, Section V presents extended experiments for LK and AAMs.

II. Image Features

A feature-based image representation is achieved with the application of a *feature extraction function* that attempts to describe distinctive and important image characteristics. In this work, we require the descriptor function to extract densely-sampled image features, thus compute a feature vector for each pixel location. This means that it transforms a 2D input image to a multi-channel image of the same height and width. Given an input image $\mathbf{T}$ with size $H \times W$, the feature extraction function $\mathcal{F}(\mathbf{T})$ is defined as $\mathcal{F}: \mathbb{R}^{H\times W} \longrightarrow \mathbb{R}^{H\times W\times D}$, where D is the number of channels of the feature image. By denoting the input image in vectorial form $\mathbf{t}$ with size $L_T \times 1$, where $L_T = HW$, the descriptor-based image is $\mathbf{f} = \mathcal{F}(\mathbf{t})$ where the feature extraction function is redefined as

$$\mathcal{F}: \mathbb{R}^{L_T\times 1} \longrightarrow \mathbb{R}^{L_T D\times 1} \tag{1}$$

In the rest of the paper, we will denote the images in vectorized form within the equations.

Many robust multi-dimensional image descriptors have been proposed and applied to various tasks. They can be divided in two categories: those extracted based only on the pixel values and those extracted based on larger spatial neighbourhoods. They all aim to generate features that are invariant to translation, rotation, scale and illumination changes and robust to local geometric distortion. We select nine of the most powerful and successful descriptors, which are briefly described in the following subsections (II-A–II-F). Figure 1 shows the feature-based image representation for each of the employed feature types. The visualized grayscale images are constructed by summing all the D channels of the feature images. Notice how each descriptor handles the illumination changes and the face's distinctive edges. Table I summarizes the parameter values, the number of channels and the neighbourhood size that gets involved in computing the descriptor at each image location for all features.

[1]Open source code of the proposed methods is available as part of the Menpo Project [49] in www.menpo.org.

(a) Original (b) ES (c) IGO (d) HOG (e) SIFT (f) OLBP (g) TPLBP (h) FPLBP (i) Gabor Ang. (j) Gabor Mag.

Fig. 1. Examples of the nine employed dense feature types. The feature images have the same height and width as the original image and D channels. In order to visualize them, we compute the sum over all D channels.

A. Edge Structure (ES)

ES [22] is a measure which captures the orientation of image structure at each pixel, together with an indication of how accurate the orientation estimate is. Assume that $\mathbf{g} = \sqrt{\mathbf{g}_x^2 + \mathbf{g}_y^2}$ is the gradient magnitude, where $\mathbf{g}_x$ and $\mathbf{g}_y$ are the local gradients. Then $\mathbf{f} = f(\mathbf{g})[\mathbf{g}_x, \mathbf{g}_y]$ is evaluated, where $f(\mathbf{g}) = |\mathbf{g}|/(|\mathbf{g}| + \bar{g})$ is a non-linear normalization function ($\bar{g}$ is the mean of $\mathbf{g}$). This feature-based representation has $D = 2$ channels and is effective at favouring strong and distinctive edges (Fig. 1b).

B. Image Gradient Orientation (IGO)

IGO is introduced and successfully applied in [15], [20], [21], [50]. Given the gradients $\mathbf{g}_x$, $\mathbf{g}_y$ of an input image and their orientation $\boldsymbol{\phi}$, we compute the IGO image as $\mathbf{f} = \frac{1}{\sqrt{N}}[\cos\boldsymbol{\phi}^T,\ \sin\boldsymbol{\phi}^T]^T$, where N is the length of the input image and $\cos\boldsymbol{\phi} = [\cos\boldsymbol{\phi}(1), \ldots, \cos\boldsymbol{\phi}(N)]^T$ (the same for $\sin\boldsymbol{\phi}$). The above feature image definition results in $D = 2$ channels. IGO features are robust to outliers and are also low-dimensional compared to other robust features (Fig. 1c).

C. Histograms of Oriented Gradients (HOG)

HOG descriptors [33] cluster the gradient orientations in different bins for localized sub-windows of an input image. Thus, the shape and texture of the image are described by histograms of local edge directions, which are also characterized by photometric invariance. The HOG features extraction begins by computing the image gradient. Two spatial neighbourhoods are used at the region of each pixel: cells and blocks. A cell is a small sub-window from which we create a histogram of the gradient's orientations weighted by the gradient magnitude. The histogram has N_{bins} bins and trilinear interpolation is applied between the votes of neighbouring bin centres with respect to orientation and position. A block is a larger spatial region that consists of $N_{block} \times N_{block}$ cells. We apply contrast normalization between the cells that are grouped within a block, based on the Euclidean norm. The final descriptor vector extracted from each block is composed by concatenating the normalized histograms of the cells, thus it has length $D = N_{bins}N_{block}^2$. We compute dense features, which means that we use a sampling step of one pixel and we extract a descriptor vector from the block centered at each such location. This ends up in a very powerful representation that is descriptive on the important facial parts and flat on the rest of the face (Fig. 1d). By using cells of size 8×8 pixels with $N_{block} = 2$ and $N_{bins} = 9$, we have $D = 36$ channels.

D. Scale-Invariant Feature Transform (SIFT)

SIFT descriptors [34] are similar to HOGs, with the difference that the orientations histograms are computed with respect to each pixel's dominant orientation. Assume that $\mathbf{L}(x, y, \sigma)$ is the Gaussian-smoothed image at the scale σ of the location (x, y). We calculate the gradient magnitude and direction for every pixel in a neighbourhood around the point in $\mathbf{L}$ and form an orientation histogram, where each orientation is weighted by the corresponding gradient magnitude and by a Gaussian-weighted circular window with standard deviation proportional to the pixel's σ. Then, we take the orientations that are within a percentage (80%) of the highest bin. If these orientations are more than one, then we create multiple points and assign them each orientation value. Eventually, the final descriptor vector is created by sampling the neighbouring pixels at the image $\mathbf{L}(x, y, \sigma)$ with scale closest to the point's scale, rotating the gradients and coordinates by the previously computed dominant orientation, separating the neighbourhood in $N_{block} \times N_{block}$ sub-regions and create a Gaussian-weighted orientations histogram for each sub-region with N_{bins} bins. Finally, the histograms are concatenated in a single vector with length $D = N_{bins}N_{block}^2$ that is normalized to unit length. In general, SIFT are invariant to scale, rotation, illumination and viewpoint (Fig. 1e). We use the same parameters as in HOGs ($N_{block} = 2$, $N_{bins} = 9$ and 8×8 cells), thus $D = 36$ channels.

E. Local Binary Patterns (LBP)

The basic idea behind LBP [35]–[37] is to encode the local structure in an image by comparing each pixel's intensity value with the pixel intensities within its neighbourhood. For each pixel, we define a neighbourhood radius r centered at the pixel and compare the intensities of S circular sample points to its intensity. The sampling is done clockwise or counter-clockwise, starting from a specific angle, and we apply interpolation on sample points that are not discrete. If the center pixel's intensity is greater or equal than the sample's, then we denote it by 1, otherwise by 0. Thus, we end up with a binary number (LBP code) for each pixel, with S digits and 2^S possible combinations, which is converted to decimal. In the original LBP formulation, the output is a descriptor vector describing the whole image with a normalized histogram of the decimal codes. We instead use N_{radius} number of values for the radius parameter, r. Then we sample $N_{samples}$ sets of points S from the circle of each radius value and concatenate the LBP codes in a vector. This means that our dense feature image has $D = N_{radius}N_{samples}$ channels. We also employ the extension of rotation-invariant uniform LBPs. Uniform

TABLE I
FEATURES PARAMETERS, NEIGHBOURHOOD SIZE THAT CONTRIBUTES IN EACH PIXEL'S COMPUTATION AND NUMBER OF CHANNELS.

Feature Type	*Parameters Values*	*Neighbourhood Size (in pixels)*	*Ch. (D)*
IGO, ES	–	–	2
HOG SIFT	$N_{bins} = 9$, $N_{cell} = 2$ $cell = 8 \times 8$ pixels	256	36
OLBP [a]	$N_{radius} = 8$, $N_{samples} = 8$	64	8
TPLBP [a] FPLBP [b]	$N_{radius} = 8$, $N_{samples} = 8$ $N_{patch} = 2$	64	16
Gabor	$N_{sc} = 4$, $N_{or} = 9$	–	36

[a] Radius takes values $\{1, 2, \ldots, 8\}$, patch sizes are 2 and 4 and for each radius we sample a single set of 8 points.
[b] Inner and outer radius are $\{[1, 5], [2, 6], \ldots, [8, 12]\}$, patch sizes are 2 and 4 and for each radius we sample a single set of 8 points.

LBPs are binary codes with at most two circular 0-1 and 1-0 transitions. In the computation of the final LBP patterns, there is a separate label for each uniform code and all the non-uniform codes are labelled with a single label. By setting $r = \{1, 2, \ldots, 8\}$ ($N_{radius} = 8$) and sampling $N_{samples} = 8$ points for each radius value, we end up with $D = 8$ channels.

Moreover, apart from the original LBP (OLBP), we also use the variations of Three-Patch LBP (TPLBP) and Four-Patch LBP (FPLBP), introduced in [38]. TPLBP and FPLBP encode in the binary codes the similarities between neighbouring patches (for details, please refer to [38]). Thus, the number of channels in this case also depends on the employed number of patches N_{patch} with different sizes, hence $D = N_{radius} N_{samples} N_{patch}$. With the parameters we use, we end up with $D = 16$ channels. The three LBP derivatives are visualized in Figs. 1f-1h.

F. Gabor Magnitude and Angle

Herein, we employ the log-Gabor filter (wavelet) [39]–[41]. In the log-polar coordinates of the Fourier domain (ρ, θ), this is defined as $G_{(s,o)}(\rho, \theta) = \exp\left(-\frac{1}{2}\left(\frac{\rho-\rho_s}{\sigma_\rho}\right)^2\right) \exp\left(-\frac{1}{2}\left(\frac{\theta-\theta_{(s,o)}}{\sigma_\theta}\right)^2\right)$, where σ_ρ and σ_θ are the bandwidths in ρ and θ respectively and (s, o) are the indices of each filter's scale and orientation. Thus, by using N_{sc} scales and N_{or} orientations, we have a filterbank of log-Gabor filters with $s = 1, \ldots, N_{sc}$ and $o = 1, \ldots, N_{or}$. The reason why log-Gabor filter is preferred over Gabor is that it has no DC component and its transfer function is extended at a high frequency range. Given an image, we compute its convolution with each log-Gabor filter for all scales and orientations. Then, we create two feature images by concatenating the convolution's magnitude and phase respectively (Figs. 1i and 1j). Both feature versions have $D = N_{sc} N_{or}$ channels. We use the log-Gabor filters implementation available in [51] with $N_{sc} = 4$ and $N_{or} = 9$, thus $D = 36$.

G. Features Function Computational Complexity

As mentioned before, the presented features can be separated in two categories: (1) the ones that are computed in a pixel-based fashion (e.g. ES, IGO), and (2) the ones that are computed in a window-based mode, thus they depend on the values of a larger spatial neighbourhood for each location (e.g. HOG, SIFT, LBP). Given an image $\mathbf{t}$ in vectorial form with length L_T, the computational cost of extracting dense D-channel features of the first category is $\mathcal{O}(L_T D)$. Respectively, the complexity of extracting the features of the second category, using a window of size $h \times w$ for each pixel, is $\mathcal{O}(L_T L_w D)$, where $L_w = hw$ is the window's area. However, since the window's dimensions h and w take values of the same order as D, hence $hw \approx D^2$, the cost of the second case can also be expressed as

$$\mathcal{O}(L_T D^3) \tag{2}$$

This gives an intuition on the complexity difference between the two cases. In the following sections, we will use the window-based features complexity of Eq.2 as the worst-case scenario, since it is more expensive than the pixel-based one.

III. INVERSE-COMPOSITIONAL ALIGNMENT ALGORITHM

The optimization technique that we employ for both LK and AAMs is the efficient gradient descent Inverse Compositional (IC) Image Alignment [5], [6]. In this section, we firstly refer to the problem of LK (III-A) and then elaborate on AAMs (III-B). In order to explain the IC algorithm, we first present the forwards-additive (FA) and forwards-compositional (FC) ones. Note that all the algorithms in this section are presented based on pixel intensities, thus we assume that we have images with a single channel.

The gradient descent image alignment aims to find the optimal parameters values of a parametric motion model. The motion model consists of a Warp function $\mathcal{W}(\mathbf{x}, \mathbf{p})$ which maps each point $\mathbf{x}$ within a target (reference) shape to its corresponding location in a shape instance. The identity warp is defined as $\mathcal{W}(\mathbf{x}, \mathbf{0}) = \mathbf{x}$. In AAMs, we employ the Piecewise Affine Warp (PWA) which performs the mapping based on the barycentric coordinates of the corresponding triangles between the source and target shapes that are extracted using Delaunay triangulation. In the following sections we will denote the warp function as $\mathcal{W}(\mathbf{p})$ for simplicity.

A. Lucas-Kanade Optimization

Herein, we first define the optimization techniques for the LK face alignment problem, in order to describe the IC optimization for AAMs in the following Sec. III-B. The aim of image alignment is to find the location of a constant template $\bar{\mathbf{a}}$ in an input vectorized image $\mathbf{t}$. This is mathematically expressed as minimizing the ℓ_2-norm cost function

$$\underset{\mathbf{p}}{\operatorname{argmin}} \|\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p}))\|^2 \tag{3}$$

with respect to the motion model parameters $\mathbf{p}$. The proposed gradient descent optimization techniques [5], [7] are categorized as: (1) *forwards* or *inverse* depending on the direction of the motion parameters estimation and (2) *additive* or *compositional* depending on the way the motion parameters are updated.

1) Forwards-Additive: Lucas and Kanade proposed the FA gradient descent in [1]. By using an additive iterative update of the parameters, i.e. $\mathbf{p} \leftarrow \mathbf{p}+\Delta\mathbf{p}$, and having an initial estimate of $\mathbf{p}$, the cost function of Eq. 3 is expressed as minimizing $\text{argmin}_{\Delta\mathbf{p}} \|\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p}+\Delta\mathbf{p}))\|^2$ with respect to $\Delta\mathbf{p}$. The solution is given by first linearising around $\mathbf{p}$, thus using first order Taylor series expansion at $\mathbf{p}+\Delta\mathbf{p} = \mathbf{p} \Rightarrow \Delta\mathbf{p} = \mathbf{0}$. This gives $\mathbf{t}(\mathcal{W}(\mathbf{p}+\Delta\mathbf{p})) \approx \mathbf{t}(\mathcal{W}(\mathbf{p})) + \mathbf{J_t}|_{\mathbf{p}=\mathbf{p}}\Delta\mathbf{p}$, where $\mathbf{J_t}|_{\mathbf{p}=\mathbf{p}} = \nabla\mathbf{t}\frac{\partial\mathcal{W}}{\partial\mathbf{p}}$ is the *image Jacobian*, consisting of the *image gradient* evaluated at $\mathcal{W}(\mathbf{p})$ and the *warp jacobian* evaluated at $\mathbf{p}$. The final solution is given by

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_\mathbf{t}^T|_{\mathbf{p}=\mathbf{p}}\left[\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p}))\right]$$

where $\mathbf{H} = \mathbf{J}_\mathbf{t}^T|_{\mathbf{p}=\mathbf{p}}\mathbf{J_t}|_{\mathbf{p}=\mathbf{p}}$ is the Gauss-Newton approximation of the *Hessian matrix*. This method is forwards because the warp projects into the image coordinate frame and additive because the iterative update of the motion parameters is computed by estimating a $\Delta\mathbf{p}$ incremental offset from the current parameters. The algorithm is very slow with computational complexity $\mathcal{O}({N_S}^3 + {N_S}^2 L_A)$, because the computationally costly Hessian matrix and its inverse depend on the warp parameters $\mathbf{p}$ and need to be evaluated in every iteration.

2) Forwards-Compositional: Compared to the FA version, in the FC gradient descent we have the same warp direction for computing the parameters, but a compositional update of the form $\mathcal{W}(\mathbf{p}) \leftarrow \mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\Delta\mathbf{p})$. The minimization cost function in this case takes the form $\text{argmin}_{\Delta\mathbf{p}} \|\bar{\mathbf{a}} - \mathbf{t}\left(\mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\Delta\mathbf{p})\right)\|^2$ and the linearisation is $\|\bar{\mathbf{a}} - \mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{J_t}|_{\Delta\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}\|^2$, where the composition with the identity warp is $\mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\mathbf{0}) = \mathcal{W}(\mathbf{p})$. The image Jacobian in this case is expressed as $\mathbf{J_t}|_{\mathbf{p}=\mathbf{0}} = \nabla\mathbf{t}(\mathcal{W}(\mathbf{p}))\left.\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$. Thus, with this formulation, the warp Jacobian is constant and can be precomputed, because it is evaluated at $\mathbf{p} = \mathbf{0}$. This precomputation slightly improves the algorithm's computational complexity compared to the FA case, even though the compositional update is more expensive than the additive one.

3) Inverse-Compositional: In the IC optimization, the direction of the warp is reversed compared to the two previous techniques and the incremental warp is computed with respect to the template $\bar{\mathbf{a}}$ [5], [10]. Compared to Eq. 3 the goal in this case is to minimize

$$\underset{\Delta\mathbf{p}}{\text{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))\|^2 \tag{4}$$

with respect to $\Delta\mathbf{p}$. The incremental warp $\mathcal{W}(\Delta\mathbf{p})$ is computed with respect to the template $\bar{\mathbf{a}}$, but the current warp $\mathcal{W}(\mathbf{p})$ is still applied on the input image. By linearising around $\Delta\mathbf{p} = \mathbf{0}$ and using the identity warp, we have

$$\|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}\|^2 \tag{5}$$

where $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} = \nabla\bar{\mathbf{a}}\left.\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$. Consequently, similar to the FC case, the increment is $\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_{\bar{\mathbf{a}}}^T|_{\mathbf{p}=\mathbf{0}}\left[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}\right]$ where the Hessian matrix is $\mathbf{H} = \mathbf{J}_{\bar{\mathbf{a}}}^T|_{\mathbf{p}=\mathbf{0}}\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}$. The compositional motion parameters update in each iteration is

$$\mathcal{W}(\mathbf{p}) \leftarrow \mathcal{W}(\mathbf{p}) \circ \mathcal{W}(\Delta\mathbf{p})^{-1} \tag{6}$$

Since the gradient is always taken at the template, the warp Jacobian $\left.\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$ and thus the Hessian matrix's inverse remain constant and can be precomputed. This makes the IC algorithm both fast and efficient with a total computational complexity of $\mathcal{O}(N_S^2 + N_S L_A)$.

B. Active Appearance Models Optimization

AAMs are deformable statistical models of shape and appearance that recover a parametric description of a certain object through optimization. A shape instance is represented as $\mathbf{s} = [x_1, y_1, \ldots, x_{L_S}, y_{L_S}]^T$, a $2L_S \times 1$ vector consisting of L_S landmark points coordinates $(x_i, y_i),\ \forall i = 1, \ldots, L_S$. An appearance instance is expressed as a $L_A \times 1$ vector $\mathbf{a}(\mathbf{x}),\ \mathbf{x} \in \mathbf{s}$ consisting of the appearance values of the L_A column-wise pixels inside the shape graph. The *shape model* is constructed by first aligning a set of training shapes using Generalized Procrustes Analysis and then applying Principal Component Analysis (PCA) on the aligned shapes to find an orthonormal basis of N_S eigenvectors $\mathbf{U}_S \in \mathbb{R}^{2L_S \times N_S}$ and the mean shape $\bar{\mathbf{s}}$. The first four eigenshapes correspond to the similarity transform parameters that control the global rotation, scaling and translation and the rest are the PCA eigenvectors with maximum variance. The *appearance model* is trained similarly in order to find the corresponding N_A eigentextures subspace $\mathbf{U}_A \in \mathbb{R}^{L_A \times N_A}$ and the mean appearance $\bar{\mathbf{a}}$. Note that the training images are warped into the mean shape in order to apply PCA, thus L_A denotes the number of pixels that belong inside the mean shape, $\mathbf{x} \in \bar{\mathbf{s}}$. Synthesis is achieved through linear combination of the eigenvectors weighted by the according model parameters, thus

$$\mathbf{s_p} = \bar{\mathbf{s}} + \mathbf{U}_S\mathbf{p} \quad \text{and} \quad \mathbf{a}_{\boldsymbol{\lambda}} = \bar{\mathbf{a}} + \mathbf{U}_A\boldsymbol{\lambda} \tag{7}$$

where $\mathbf{p} = [p_1, \ldots, p_{N_S}]^T$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{N_A}]^T$ are the shape and appearance parameters respectively.

The basic difference between the IC algorithm employed for LK and AAMs is that the template image $\bar{\mathbf{a}}$ is not static, but it includes a linear appearance variation controlled by the appearance parameters $\boldsymbol{\lambda}$ as shown in Eq. 7. Consequently, the minimization cost function of Eq. 3 now becomes

$$\underset{\mathbf{p},\boldsymbol{\lambda}}{\text{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_A\boldsymbol{\lambda}\|^2 \tag{8}$$

We present three algorithms for solving the optimization problem: Simultaneous, Alternating and Project-Out.

1) Project-Out Inverse-Compositional: The Project-Out IC (POIC) algorithm [6] decouples shape and appearance by solving Eq. 8 in a subspace orthogonal to the appearance variation. This is achieved by "projecting-out" the appearance variation, thus working on the orthogonal complement of the appearance subspace $\hat{\mathbf{U}}_A = \mathbf{I} - \mathbf{U}_A\mathbf{U}_A^T$. The cost function of Eq. 8 takes the form

$$\underset{\Delta\mathbf{p}}{\text{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))\|^2_{\mathbf{I}-\mathbf{U}_A\mathbf{U}_A^T} \tag{9}$$

and first-order Taylor expansion over $\Delta\mathbf{p} = \mathbf{0}$ is $\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) \approx \bar{\mathbf{a}} + \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}$. The incremental update of the warp parameters is computed as $\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_{POIC}^T[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}]$ where $\mathbf{H}^{-1} = \mathbf{J}_{POIC}^T\mathbf{J}_{POIC}$ and $\mathbf{J}_{POIC} = (\mathbf{I} - \mathbf{U}_A\mathbf{U}_A^T)\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}$.

The appearance parameters can be retrieved at the end of the iterative operation as $\boldsymbol{\lambda} = \mathbf{U}_A^T[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}]$ in order to reconstruct the appearance vector. The POIC algorithm is very fast with $\mathcal{O}(N_S L_A + N_S^2)$ computational complexity, because the Jacobian, the Hessian matrix and its inverse are constant and can be precomputed. However, the algorithm is not robust, especially in cases with large appearance variation or outliers.

2) Simultaneous Inverse-Compositional: In the Simultaneous IC (SIC) [18] we aim to optimize simultaneously for $\mathbf{p}$ and $\boldsymbol{\lambda}$ parameters. Similar to the Eq. 4 of the LK-IC case, the cost function of Eq. 8 now becomes

$$\underset{\Delta\mathbf{p},\Delta\boldsymbol{\lambda}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{a}_{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}\mathcal{W}(\Delta\mathbf{p}))\|^2 \tag{10}$$

where $\mathbf{a}_{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}(\mathcal{W}(\Delta\mathbf{p})) = \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) + \mathbf{U}_A(\mathcal{W}(\Delta\mathbf{p}))(\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})$. We denote by $\Delta\mathbf{q} = [\Delta\mathbf{p}^T, \Delta\boldsymbol{\lambda}^T]^T$ the vector of concatenated parameters increments with length $N_S + N_A$. As in Eq. 5, the linearisation of $\mathbf{a}_{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}(\mathcal{W}(\Delta\mathbf{p}))$ around $\Delta\mathbf{p} = \mathbf{0}$ consists of two parts: the mean appearance vector approximation $\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) \approx \bar{\mathbf{a}} + \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}$ and the linearised basis $\mathbf{U}_A(\mathcal{W}(\Delta\mathbf{p})) \approx \mathbf{U}_A + [\mathbf{J}_{\mathbf{u}_1}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}, \ldots, \mathbf{J}_{\mathbf{u}_{N_A}}|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}]$, where $\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}} = \nabla\mathbf{u}_i \left.\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}$ denotes the Jacobian with respect to the i^{th} eigentexture at $\Delta\mathbf{p} = \mathbf{0}$. Then the final solution at each iteration is

$$\Delta\mathbf{q} = \mathbf{H}^{-1}\mathbf{J}_{SIC}^T[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_A\boldsymbol{\lambda}] \tag{11}$$

where the Hessian matrix is $\mathbf{H} = \mathbf{J}_{SIC}^T\mathbf{J}_{SIC}$ and the Jacobian is given by $\mathbf{J}_{SIC} = \left[\mathbf{J}_{\mathbf{a}_{\boldsymbol{\lambda}}}|_{\mathbf{p}=\mathbf{0}}, \mathbf{U}_A\right]$ with $\mathbf{J}_{\mathbf{a}_{\boldsymbol{\lambda}}}|_{\mathbf{p}=\mathbf{0}} = \mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} + \sum_{i=1}^{N_A}\lambda_i\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}$. At every iteration, we apply the compositional motion parameters update of Eq. 6 of the LK-IC and an additive appearance parameters update $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$. The individual Jacobians $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}$ and $\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}$, $\forall i = 1, \ldots, N_A$ are constant and can be precomputed. However, the total Jacobian $\mathbf{J}_{\mathbf{a}_{\boldsymbol{\lambda}}}|_{\mathbf{p}=\mathbf{0}}$ and hence the Hessian matrix depend on the current estimate of the appearance parameters $\boldsymbol{\lambda}$, thus they need to be computed at every iteration. This makes the algorithm very slow with a total cost of $\mathcal{O}((N_S + N_A)^2 L_A + (N_S + N_A)^3)$.

3) Alternating Inverse-Compositional: The Alternating IC (AIC) algorithm, proposed in [8], instead of minimizing the cost function simultaneously for both shape and appearance as in the SIC algorithm, it solves two separate minimization problems, one for the shape and one for the appearance optimal parameters, in an alternating fashion. That is

$$\begin{cases} \underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{a}_{\boldsymbol{\lambda}}(\mathcal{W}(\Delta\mathbf{p}))\|^2_{\mathbf{I}-\mathbf{U}_A\mathbf{U}_A^T} \\ \underset{\Delta\boldsymbol{\lambda}}{\operatorname{argmin}} \|\mathbf{t}(\mathcal{W}(\mathbf{p})) - \mathbf{a}_{\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}}(\mathcal{W}(\Delta\mathbf{p}))\|^2 \end{cases} \tag{12}$$

The minimization in every iteration is achieved by first using a fixed estimate of $\boldsymbol{\lambda}$ to compute the current estimate of the increment $\Delta\mathbf{p}$ and then using the fixed estimate of $\mathbf{p}$ to compute the increment $\Delta\boldsymbol{\lambda}$. More specifically, similar to the previous cases and skipping the linearisation steps, given the current estimate of $\boldsymbol{\lambda}$, the warp parameters increment is computed from the first cost function as

$$\Delta\mathbf{p} = \mathbf{H}^{-1}\mathbf{J}_{AIC}^T[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_A\boldsymbol{\lambda}] \tag{13}$$

where $\mathbf{J}_{AIC} = (\mathbf{I} - \mathbf{U}_A\mathbf{U}_A^T)[\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}} + \sum_{i=1}^{N_A}\lambda_i\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}]$ and $\mathbf{H}^{-1} = \mathbf{J}_{AIC}^T\mathbf{J}_{AIC}$. Then, given the current estimate of the motion parameters $\mathbf{p}$, AIC computes the optimal appearance parameters as the least-squares solution of the second cost function of Eq. 12, thus

$$\Delta\boldsymbol{\lambda} = \mathbf{U}_A^T[\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})) - \mathbf{U}_A(\mathcal{W}(\Delta\mathbf{p}))\boldsymbol{\lambda}] \tag{14}$$

This alternating optimization is repeated at each iteration. The motion parameters are compositionally updated as in Eq. 6 and the appearance parameters are updated in an additive mode, i.e. $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$. AIC algorithm is slower than POIC, but more accurate as it also optimizes with respect to the appearance variance. Although the individual Jacobians $\mathbf{J}_{\mathbf{u}_i}|_{\mathbf{p}=\mathbf{0}}$, $\forall i = 1, \ldots, N_A$ and $\mathbf{J}_{\bar{\mathbf{a}}}|_{\mathbf{p}=\mathbf{0}}$ can be precomputed, the total Jacobian $\mathbf{J}_{AIC}$ and the Hessian need to be evaluated at each iteration. Following the Hessian matrix computation technique proposed in [8], which improves the cost from $\mathcal{O}(N_S^2 L_A)$ to $\mathcal{O}(N_S^2 N_A^2)$ (usually $L_A > N_A^2$) and taking into account the Hessian inversion ($\mathcal{O}(N_S^3)$), the total cost at each iteration is $\mathcal{O}(N_S^2 N_A^2 + (N_S + N_A)L_A + N_S^3)$.

Recently it was shown that AIC and SIC are theoretically equivalent (i.e., Eqs. 13, 14 are exactly the same as Eq. 11) and that the only difference is their computational costs. That is the SIC algorithm requires to invert the Hessian of the concatenated shape and texture parameters ($O((N_S + N_A)^3)$). However, using the fact that $min_{x,y}f(x,y) = min_x\,(min_y f(x,y))$ and solving first for the texture parameter increments, it was shown that (1) the complexity of SIC can be reduced dramatically and (2) SIC is equivalent to AIC algorithm [52] (similar results can be shown by using the Schur's complement of the Hessian of texture and shape parameters).

IV. Feature-Based Optimization

In this section we describe the combination of the IC algorithm with the feature-based appearance of Eq. 1. The keypoint of this combination is that there are two different ways of conducting the composition of the features function $\mathcal{F}$ and the warp function $\mathcal{W}$ on an image. Given an image $\mathbf{t}$ and the warp parameters $\mathbf{p}$, the warped feature-based image $\mathbf{f}$ can be obtained with the two following composition directions:

- *Features from warped image:*

$$\mathbf{f} = \mathcal{F}(\mathbf{t}(\mathcal{W}(\mathbf{p}))) \tag{15}$$

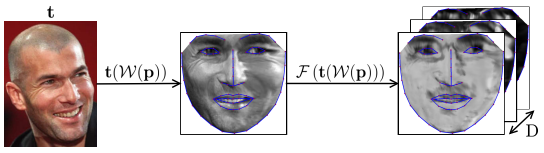
- *Warping on features image:*

$$\mathbf{f} = \mathbf{t}_{\mathcal{F}}(\mathcal{W}(\mathbf{p})) \text{ where } \mathbf{t}_{\mathcal{F}} = \mathcal{F}(\mathbf{t}) \tag{16}$$

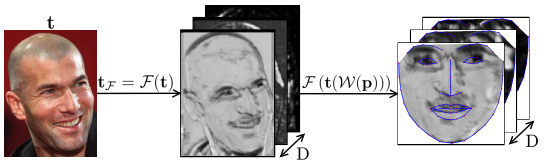
The composition order of these two cases is shown in Fig. 2. In the following subsections we present the incorporation of these two functions compositions in the IC algorithm and explain why the second one is preferable. For simplicity, we use the LK-IC algorithm (Sec. III-A3) for face alignment that does not include appearance variation.

A. Warp Function Computational Complexity

As shown in Sec. II-G, the computational cost of the feature extraction function $\mathcal{F}(\mathbf{t})$ is $\mathcal{O}(L_T D^3)$, where L_T is



(a) Features from warped image.



(b) Warping on features image.

Fig. 2. The two possible composition directions of the feature extraction function $\mathcal{F}$ and the warp function $\mathcal{W}(\mathbf{p})$.

the resolution of the image $\mathbf{t}$. Regarding the warp function, we need to consider that the warping of a D-channel image, $\mathbf{t}(\mathcal{W}(\mathbf{p}))$, includes the three following steps:

1) Synthesis of the shape model instance $\mathbf{s_p}$, as in Eq. 7, using the weights $\mathbf{p}$, which has a cost of $\mathcal{O}(2L_S N_S)$.
2) Computation of the mapping of each pixel in the mean shape $\bar{\mathbf{s}}$ to the synthesized shape instance $\mathbf{s_p}$. This firstly involves the triangulation of the shape instance in N_{tr} number of triangles (same as the number of triangles of the mean shape) using Delaunay triangulation. Then, six affine transformation parameters are computed for each triangle based on the coordinates of the corresponding triangles' vertices. Finally, the transformed location of each point within each triangle is evaluated. Thus, the complexity of this step is $\mathcal{O}(6N_{tr}\frac{L_A}{N_{tr}}) = \mathcal{O}(6L_A)$.
3) Copying the values of all channels D for all pixels from the input image to the reference frame $\bar{\mathbf{s}}$ ($\mathcal{O}(DL_A)$).

Consequently, taking into account that $(6+D)L_A \gg 2L_S N_S$, the overall computational complexity of warping a multi-channel image is $\mathcal{O}((6+D)L_A)$.

B. Optimization with Features from Warped Image

From Eqs. 4 and 15 we get the cost function of minimizing

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathcal{F}(\mathbf{t}(\mathcal{W}(\mathbf{p}))) - \mathcal{F}(\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p})))\|^2$$

with respect to $\Delta\mathbf{p}$. Thus, the first-order Taylor expansion of this expression around $\Delta\mathbf{p} = \mathbf{0}$ is $\mathcal{F}(\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))) \approx \mathcal{F}(\bar{\mathbf{a}}) + \frac{\partial\mathcal{F}}{\partial\bar{\mathbf{a}}}\nabla\bar{\mathbf{a}}\,\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\Big|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p}$. Since it is not possible to compute $\frac{\partial\mathcal{F}}{\partial\bar{\mathbf{a}}}$, we make the approximation $\frac{\partial\mathcal{F}}{\partial\bar{\mathbf{a}}}\nabla\bar{\mathbf{a}} \approx \nabla\mathcal{F}(\bar{\mathbf{a}})$ and the linearisation becomes

$$\mathcal{F}(\bar{\mathbf{a}}(\mathcal{W}(\Delta\mathbf{p}))) \approx \mathcal{F}(\bar{\mathbf{a}}) + \nabla\mathcal{F}(\bar{\mathbf{a}})\,\frac{\partial\mathcal{W}}{\partial\mathbf{p}}\Big|_{\mathbf{p}=\mathbf{0}}\Delta\mathbf{p} \qquad (17)$$

Consequently, in every IC repetition step, the warping is performed on the intensities image ($D = 1$) with the current parameters estimate ($\mathcal{O}(7L_A)$) and is followed by the feature extraction ($\mathcal{O}(L_A D^3)$), ending up to a cost of $\mathcal{O}(L_A(7+D^3))$ per iteration. Hence, by applying k iterations of the algorithm and given that $D^3 \gg 7$, the overall complexity of warping and features extraction is

$$\mathcal{O}(kL_A D^3) \qquad (18)$$

Note that this is only a part of the final cost, as the IC algorithm complexity also needs to be taken into account. Moreover, in the AAMs case, it is difficult to extract window-based features (e.g. HOG, SIFT, LBP) from the mean shape template image, as required from the above procedure. This is because, we have to pad the warped texture in order to compute features on the boundary, which requires extra triangulation points.

C. Optimization with Warping on Features Image

The combination of Eqs. 4 and 16 gives the cost function

$$\underset{\Delta\mathbf{p}}{\operatorname{argmin}} \|\mathbf{t}_{\mathcal{F}}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}}_{\mathcal{F}}(\mathcal{W}(\Delta\mathbf{p}))\|^2$$

where $\mathbf{t}_{\mathcal{F}} = \mathcal{F}(\mathbf{t})$ and $\bar{\mathbf{a}}_{\mathcal{F}} = \mathcal{F}(\bar{\mathbf{a}})$ are the multi-channel feature-based representations of the input and the template images respectively. The linearisation around $\Delta\mathbf{p} = \mathbf{0}$ has the same form as in Eq. 17 of the previous case. However, in contrast with the previous case, the warping is performed on the feature-based image. This means that the feature extraction is performed *once* on the input image and the resulting multi-channel image is warped during each iteration. Hence, the computational complexity of feature extraction and warping is $\mathcal{O}((6+D)L_A)$ per iteration and $\mathcal{O}(k(6+D)L_A + L_T D^3)$ overall per image for k iterations, where L_T is the resolution of the input image.

The above cost greatly depends on the input image dimensions L_T. In order to override this dependency, we firstly resize the input image with respect to the scaling factor between the face detection bounding box and the mean shape resolution. Then, we crop the resized image in a region slightly bigger than the bounding box. Thus, the resulting input image has resolution approximately equal to the mean shape resolution L_A, which leads to an overall complexity of

$$\mathcal{O}(kL_A(6+D) + L_A D^3) \qquad (19)$$

for k iterations. Another reason for resizing the input image is to have correspondence on the scales on which the features are extracted, so that they describe the same neighbourhood.

The computational complexities of Eqs. 18 and 19 are approximately equal for small number of channels D (e.g. for ES and IGO). However, this technique of warping the features image has much smaller complexity for large values of D (e.g. for HOG, SIFT, LBP, Gabor). This is because $k(D+6) < D^3$ for large values of D, so $kL_A(6+D)$ can be eliminated in Eq. 19. Consequently, since $kL_A D^3 \gg L_A D$, it is more advantageous to compute the features image once and then warp the multi-channel image at each iteration. In the experiments (Sec. V), we report the timings that prove the above conclusion. Finally, we carried out an extensive experiment comparing the two methods for face alignment (LK) in Sec. V-A1 (Fig. 4). The results indicate that warping the multi-channel features image performs better, which is an additional reason to choose this composition direction apart from the computational complexity.

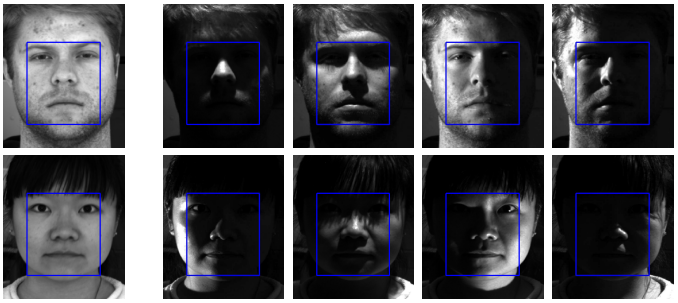


Fig. 3. Yale B Database images examples. The template image (left) is corrupted with extreme illumination in the testing images for each subject.

V. Experimental Results

Herein, we present extended experiments for both face alignment (LK, Sec. V-A) and face fitting (AAMs, Secs. V-B and V-C) using the IC framework. We employ all the dense features described in Sec. II with the parameters of Tab. I.

A. Face Alignment (Lucas-Kanade)

In this section, we conduct experiments for the task of face alignment using the LK-IC algorithm. In Sec. V-A1 we show a motivating experiment in which we compare the performance of IC with warping the features image at each iteration vs. extracting features from the warped image. In Sec. V-A2, we compare the performance of IC with warping the features image for all features types. For both experiments, we use the Yale Face Database B [42], which consists of 10 subjects with 576 images per subject under different viewing conditions. We select 1 template image and 10 testing images for each subject (100 image pairs) that are corrupted with extreme illumination conditions (Fig. 3).

We use the evaluation framework proposed in [5]. Specifically, we define three canonical points within a region of interest for each image. These points are randomly perturbed using a Gaussian distribution with standard deviation $\sigma = \{1, 2, \ldots, 9\}$. Then, we create the affine distorted image based on the affine warp defined between the original and perturbed points. After applying 30 iterations of the IC optimization algorithm, we compute the RMS error between the estimated and the correct locations of the three canonical points. The optimization is considered to have converged if the final RMS error is less than 3 pixels. Additionally, for each value of σ, we perform 100 experiments with different randomly perturbed warps. We evaluate the performance by plotting the average frequency of convergence and the average mean RMS error of the converged cases with respect to each value of σ. The results are averaged over the 100 experiment repetitions with different random warps.

1) Warping of features image vs Features from warped image: In the experiment of Fig. 4 we compare the performance of the two possible combination techniques between the features extraction function and the warp function, as presented in Sec. IV. The figure shows only HOG, SIFT, IGO and LBP cases, though we get the same results with the rest of features types. The comparison indicates that the method of extracting the features from the original image

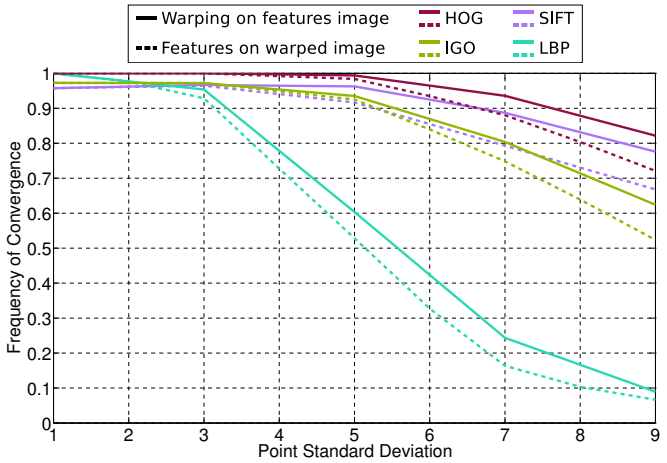


Fig. 4. Comparison between the techniques of warping the features image and extracting features from the warped image. The plot shows results for HOG, SIFT, IGO and LBP features, however the rest of the features demonstrate the same behaviour.

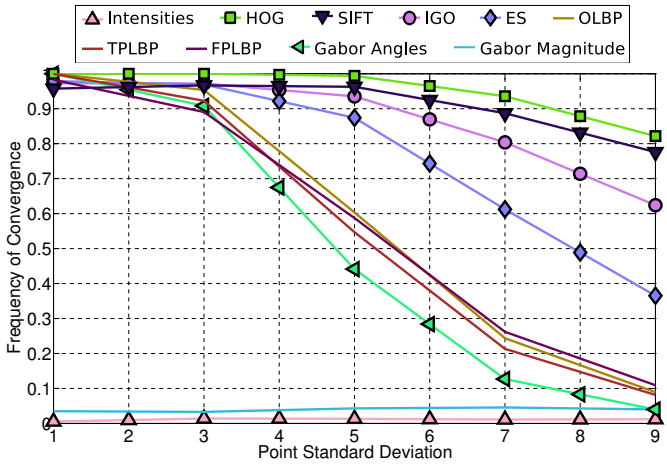
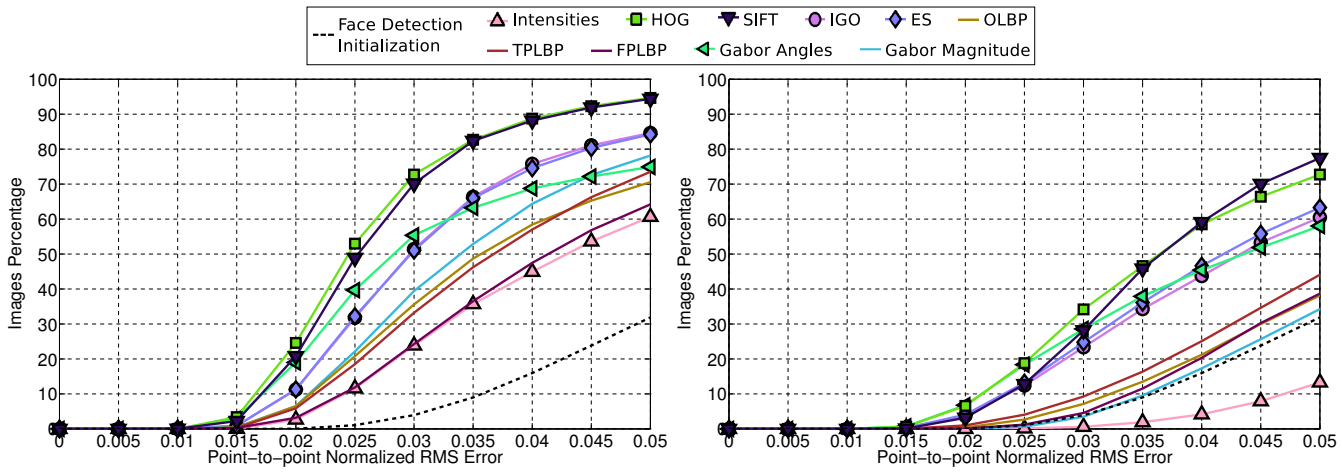


Fig. 5. Face alignment (Lucas-Kanade) results on Yale B database using the inverse compositional framework. The figure shows the frequency of convergence with respect to the standard deviation σ.

outperforms the one of extracting the features from the warped image, especially for large values of σ. The reason behind this behaviour is that the warping of an image provokes some distortion on the texture which partly destroys the local structure. This has negative consequences on the computation of all the employed features, because the descriptor of each pixel depends on the structure of its neighbourhood.

2) Features Comparison: Figure 5 provides an evaluation of the robustness of each feature by showing the average frequency of convergence with respect to each value of σ. This experiment clearly indicates that Intensities or Gabor Magnitude features are totally inappropriate for such a task. HOG is the most robust feature with remarkable convergence frequency, followed by SIFT, IGO and ES. Finally, the LBPs family and Gabor Angles are not robust, but they can achieve decent results when the initialization is good.



(a) Alternating IC (b) Project-Out IC

Fig. 6. Face fitting (AAMs) accuracy on in-the-wild databases (3026 test images) using the alternating and project-out inverse compositional frameworks, evaluated on 68 landmark points.

B. Face Fitting (Active Appearance Models)

In this section we compare the performance of the selected features using AAMs for the task of face fitting with cross-database experiments. We investigate *which* features are more suitable for the task by comparing them with respect to their accuracy (Sec. V-B1), speed of convergence (Sec. V-B2) and computational cost (Sec. V-B3). We also shed light on *why* some features perform better by comparing them with respect to the number of appearance components (Sec. V-B4), the neighbourhood size per pixel (Sec. V-B5) and the smoothness of their cost function (Sec. V-B6).

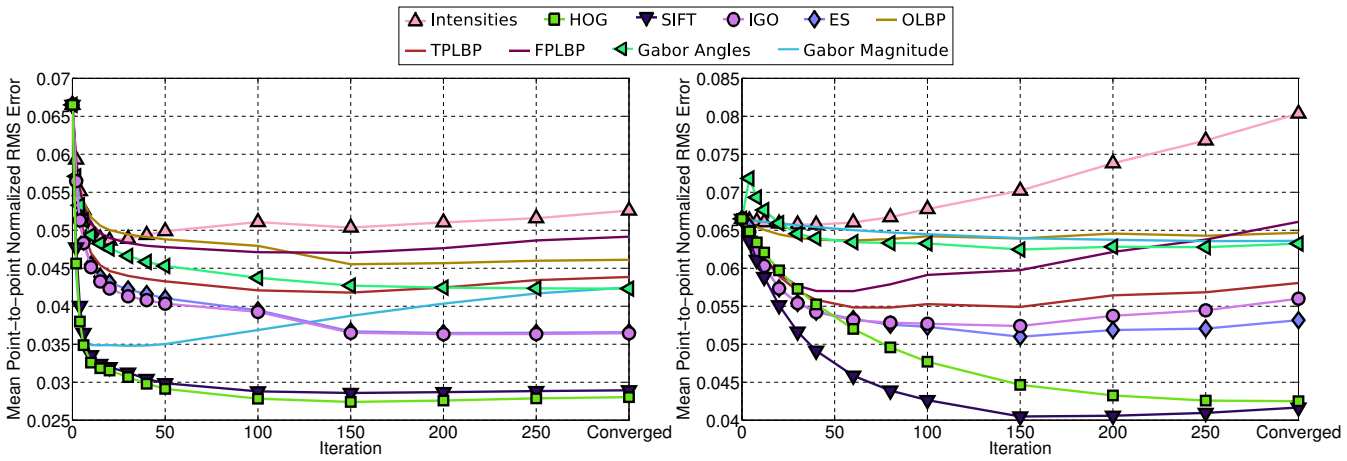
As explained in Sec.III-B3, AIC and SIC algorithms are theoretically equivalent and the only difference between them is that SIC is significantly slower. Specifically, the updates of SIC (Eq. 11) and AIC (Eqs. 13 and 14) are theoretically guaranteed to be the same [52]. Thus, herein we employ the AIC and POIC algorithms.

We use four popular in-the-wild databases, which contain images downloaded from the web that are captured in totally unconstrained conditions and exhibit large variations in pose, identity, illumination, expressions, occlusion and resolution. Specifically, we use the Labelled Faces Parts in the Wild (LFPW) [43] training set in order to train a model for each feature type. As some of the database's images URLs are invalid, we acquired only 811 training images. The testing is performed on the very challenging in-the-wild databases of Annotated Faces in the Wild (AFW) [44], LFPW testing set [43], Helen training and testing set [45] and iBUG [53] which consist of 337, 224, 2000, 330 and 135 images respectively. Thus, the testing is performed on 3026 in-the-wild images. We acquired the groundtruth annotations of 68 points for all databases from the 300 Faces In-The-Wild Challenge [46], [53], [54].

The fitting process is always initialized by computing the face's bounding box using Cascade Deformable Part Models (CDPM) face detector [55] and then estimating the appropriate global similarity transform that fits the mean shape within the bounding box boundaries. Note that this initial similarity transform only involves a translation and scaling component and not any in-plane rotation. The accuracy of the fitting result is measured by the point-to-point RMS error between the fitted shape and the groundtruth annotations, normalized by the face size, as proposed in [44]. Denoting $\mathbf{s}^f = [x_1^f, y_1^f, \ldots, x_{L_S}^f, y_{L_S}^f]^T$ and $\mathbf{s}^g = [x_1^g, y_1^g, \ldots, x_{L_S}^g, y_{L_S}^g]^T$ as the fitted shape and the groundtruth shape respectively, then the error between them is expressed as RMSE $= \frac{\sum_{i=1}^{L} \sqrt{(x_i^f - x_i^g)^2 + (y_i^f - y_i^g)^2}}{s_f L_S}$, where $s_f = (\max x_i^g - \min x_i^g + \max y_i^g - \min y_i^g)/2$ defines the face's size.

1) Accuracy: Figures 6a and 6b compare the accuracy of AIC and POIC respectively on all the databases (3026 testing images) for all the features types. The fitting procedure is performed using the methodology of Sec. IV-C and keeping $N_S = 15$ eigenshapes and $N_A = 100$ eigentextures, regardless of the feature type. The results are plotted in the form of Cumulative Error Distributions (CED). Note that this experiment intends to make a fair comparison of the accuracy between the various features by letting the fitting procedure converge for all feature types. The results indicate that HOG and SIFT features are the most appropriate for the task. HOG features perform better in the case of AIC and the SIFT ones are more robust for POIC, however the differences between them are very small. IGO and ES features have a sufficiently good performance. Moreover, similar to the face alignment case, Gabor Angles are not robust, but they achieve very accurate fitting result when they converge, especially in the POIC case. On the contrary, even though Gabor Magnitude features demonstrate a decent performance in the AIC, they completely diverge in the POIC case. This observation, combined with their performance with the LK algorithm, indicates that they are unsuitable for image alignment without a linear appearance variation model. The same fact stands for intensities as well. Finally, the LBPs family has relatively poor performance. Figure 14 shows some indicative fitting examples from the very challenging iBUG database for all features with AIC.

2) Convergence: Herein, we examine the frequency of convergence achieved by each feature type. We assume that



(a) Alternating IC (b) Project-Out IC

Fig. 7. Mean point-to-point normalized RMS fitting error with respect to iteration number on in-the-wild databases (3026 test images). The plot aims to compare the speed of convergence of each feature type. Please refer to Table II (columns 5-10) for the computational cost of each feature-based method.

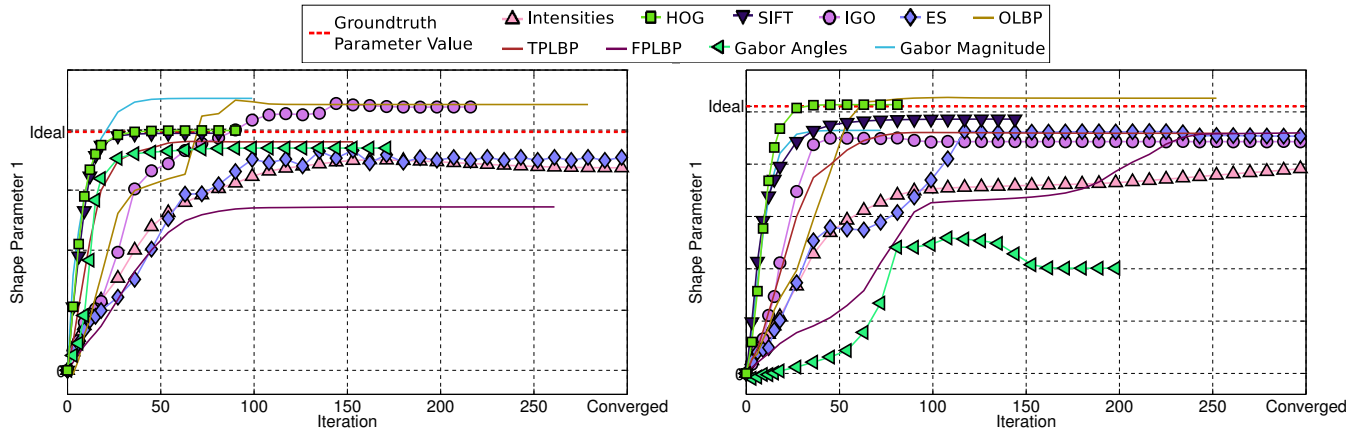


Fig. 8. Indicative examples of the speed of convergence of each feature. The plots show how fast the 1st parameter value of the shape model moves towards its ideal (groundtruth) value. The example images are `image_0022.png` (*left*) and `image_0028.png` (*right*) from LFPW testing set.

a fitting procedure has converged when either the cost function error incremental or the landmarks mean displacement are very small. The cost incremental criterion is defined as $\frac{abs(error_{k-1}-error_k)}{error_{k-1}} < \epsilon$, where $error_k$ is the cost function error from Eq. 8 at current iteration k and $\epsilon = 10^{-5}$. The mean displacement criterion is defined as the mean point-to-point normalized Euclidean distance between the shapes of current and previous iterations, thus $\frac{\sum_{i=1}^{L}\sqrt{(x_i^k-x_i^{k-1})^2+(y_i^k-y_i^{k-1})^2}}{s_f L_S} < \epsilon$ with $\epsilon = 10^{-4}$. Figure 7 shows the mean point-to-point normalized RMS fitting error overall 3026 images with respect to the iteration number by allowing the optimization procedure to converge. The results indicate that HOG and SIFT features converge faster to a more accurate optimum compared to all the other feature types. Indicative examples of the convergence speed of each feature are shown in Fig. 8. Specifically, these plots show how fast the parameter value that corresponds to the 1st eigenvector of the shape subspace $\mathbf{U}_S$ moves towards its ideal (groundtruth) value. This eigenshape controls the face's pose over the yaw angle. These examples demonstrate the advantages of HOG and SIFT features, which reach the ideal value in very few iterations. Note that in all these experiments we want the algorithms to converge, thus we let them execute many iterations. However, this is not necessary in a practical application, because as the iterations advance, the improvements in the fitted shape get much smaller.

3) Timings: Table II reports the timings for each feature type using the two compositional scenarios explained in Sec. IV within the AAMs optimization framework. It presents the computational cost per iteration and the total cost of running the optimization for 50 and 100 iterations. Note that the AAMs framework used for those experiments is developed without any code optimization. The reference frame (mean shape $\mathbf{s}_0$) has size 170×170.

The table justifies the computational analysis presented in Sec. IV. As expected, it is faster to compute the features once and warp the features image (Eq. 19) rather than extracting features from each warped image at each iteration (Eq. 18). This is because, in most features cases, it is more expensive to extract features than warp a multi-channel image ($\mathcal{O}(\mathcal{F}) > \mathcal{O}(\mathcal{W})$). This happens with all the multi-channel features. The only exception is the SIFT features case, because the optimized implementation of [56] is faster than the

TABLE II
COMPUTATIONAL COSTS OF THE FEATURE EXTRACTION FUNCTIONS, THE WARP FUNCTION AND THE AAM FITTING USING BOTH COMPOSITION WAYS OF THE TWO FUNCTIONS FOR ALL FEATURE TYPES. ALL THE REPORTED TIMES ARE MEASURED IN SECONDS.

				Warping on features image						*Features from warped image*					
				Alternating IC			*Project-Out IC*			*Alternating IC*			*Project-Out IC*		
Feature Type	*Ch.*	*Feature function Cost ($\mathcal{F}$)*	*Warp function Cost ($\mathcal{W}$)*	*number of iterations*			*number of iterations*			*number of iterations*			*number of iterations*		
				1	*50*	*100*	*1*	*50*	*100*	*1*	*50*	*100*	*1*	*50*	*100*
Intensities	1	–	0.01	0.02	1.0	2.0	0.02	1.0	2.0	0.02	1.0	2.0	0.02	1.0	2.0
IGO, ES	2	0.01	0.01	0.05	2.0	4.0	0.04	1.5	3.0	0.04	2.0	4.0	0.03	1.5	3.0
OLBP	8	0.07	0.03	0.2	6.6	13.1	0.17	5.1	10.1	0.18	9.0	18.0	0.15	7.5	15.0
TPLBP	16	1.25	0.05	1.48	12.8	24.3	1.43	10.3	19.3	1.44	72.0	144.0	1.39	69.5	139.0
FPLBP		1.82		2.05	13.3	24.8	2.0	10.8	19.8	2.01	100.5	201.0	1.96	98.0	196.0
HOG	36	1.32	0.11	1.84	27.3	53.3	1.72	21.3	41.3	1.74	87.7	174.0	1.62	81.0	162.0
SIFT		0.07		0.59	26.1	52.1	0.47	20.1	40.1	0.49	24.5	49.0	0.37	18.5	37.0
Gabor		0.12		0.64	26.1	52.1	0.52	20.1	40.1	0.54	27.0	54.0	0.42	21.0	42.0

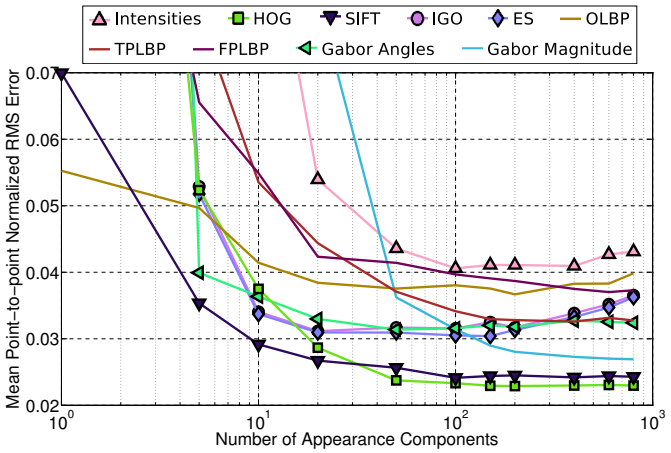


Fig. 9. Mean point-to-point normalized RMS fitting error with respect to number of appearance components on the LFPW testset in-the-wild database. Note that we use logarithmic scale on the horizontal axis.

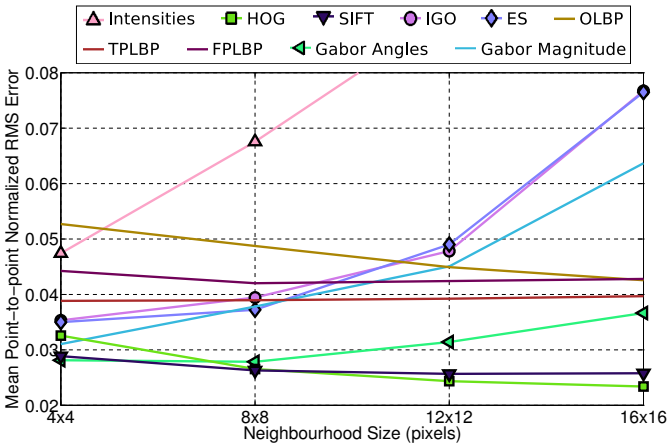


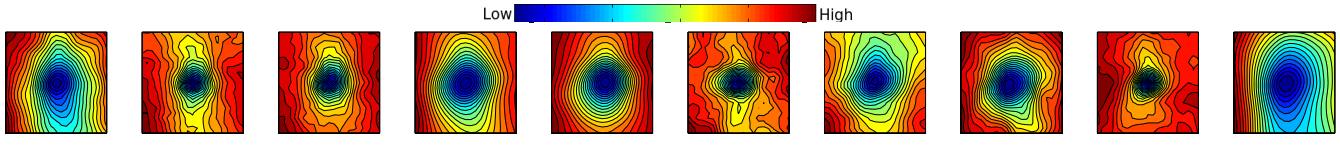
Fig. 10. Mean point-to-point normalized RMS fitting error with respect to neighbourhood size on the LFPW testset in-the-wild database.

unoptimized warping of the 36 channels ($\mathcal{O}(\mathcal{F}) < \mathcal{O}(\mathcal{W})$). Moreover, the combination of Tab. II with Fig. 7 suggests that even though high-dimensional features like HOG and SIFT converge really fast, their computational cost is quite similar to features with less channels that require multiple iterations until convergence. Note that it is not in the scope of this paper to provide an optimized implementation of AAMs or features. Faster AAM optimization can be achieved with the framework proposed in [52] and one could also use GPU or parallel programming to achieve much faster performance and eliminate the cost difference between various features and also between the two composition scenarios of $\mathcal{F}$ and $\mathcal{W}$.

4) Number of Appearance Components: Figure 9 shows the mean point-to-point normalized RMS fitting error with respect to the number of appearance components, i.e. N_A, for LFPW testset using logarithmic scale on the horizontal axis. The results indicate that for most features, except IGO, ES and Intensities, the fitting performance is improved by increasing the number of appearance components. SIFT features can achieve very accurate results by using very few appearance components (even less than 10), thus with small computational cost. Additionally, note that Gabor Magnitude features can achieve significantly good accuracy (close to HOG and SIFT) if one keeps their whole eigenspectrum.

5) Neighbourhood Size: Figure 10 plots the mean point-to-point normalized RMS fitting error with respect to the neighbourhood size from which the feature value of each pixel is computed. For HOG and SIFT this is done by changing the cell size. In the case of the LBPs family, we alter the radius values (N_{radius}). For the rest of features (IGO, ES, Gabor, Inensities), we simply downscale the image. This experiment proves that the spatial neighbourhood covered by each feature does not massively affect its performance. HOG, SIFT and LBP features are more accurate when applied to largest regions, as more information is accumulated to their channels. On the contrary, ES, IGO and Gabor features are not assisted by increasing the neighbourhood size.

6) Cost Function: Figure 11 illustrates the cost function for each feature type in 2D contour plots. The plots are generated by translating the groundtruth shape of an image within a grid of $\pm 15\%$ (pixels) of the face size along the x and y axis and evaluating the cost of Eq. 8, where $\boldsymbol{\lambda}$ are the projection parameters $\boldsymbol{\lambda} = \mathbf{U}_A^T(\mathbf{t}(\mathcal{W}(\mathbf{p})) - \bar{\mathbf{a}})$. The plotted costs are



(a) Intensities (b) ES (c) IGO (d) HOG (e) SIFT (f) OLBP (g) TPLBP (h) FPLBP (i) Gabor Ang. (j) Gabor Mag.

Fig. 11. Contour plots of the cost function for each feature. The plots show the mean cost function over 100 images after translating the groundtruth shape over the x and y axis by $\pm 15\%$ (pixels) of the face size.

averaged over 100 images. For each feature we use $N_A = 100$ appearance components, so that the experiment is fair and can be combined with the accuracy results of Sec. V-B1. These plots are very informative. The cost functions of IGO, ES and Gabor Angles have a very narrow region of small errors, which means that they can be accurate only when their initialization is close to the global optimum. On the contrary, Gabor Magnitude features have a very broad low error region, which means that they can quickly reach a small error but they will get stuck to a local minimum that is probably far from the global optimum. This can also be observed in Fig. 7a, where Gabor Magnitude features converge very fast to a low error but then start to diverge, due to the multiple local minima of their cost function. Finally, HOG and SIFT features have a smooth cost and the region of minimum values is large enough to facilitate fast and accurate convergence.

C. Comparison with state-of-the-art Face Fitting Methods

Herein we compare the performance of our proposed feature-based AAMs (both AIC and POIC) against two state-of-the-art facial trackers: Supervised Descent Method (SDM) [47] and Robust Discriminative Response Map Fitting (DRMF) for Constrained Local Models (CLMs) [48]. For our feature-based AAMs, we employ the HOG and SIFT features because they proved to be the most accurate and robust for both face alignment and fitting. We use the same initialization and experimental setup as in the previous section (Sec. V-B). Specifically, the AAMs are trained on the 811 images of the LFPW trainset, keeping $N_S = 15$ eigenshapes and $N_A = 100$ eigentextures. For the other two methods, we used the implementations provided online by their authors in [57], [58] with their pre-trained models. Note that both these methods are trained on thousands of images, much more than the 811 used to train our AAMs. All methods are initialized using the (CDPM) face detector [55]. In this experiment we report results evaluated on 49 landmark points shape mask instead of 68 points. This is because the SDM framework [57] computes and returns only these 49 points. The 49-point mask occurs by removing the 17 points of the boundary (jaw) and the 2 points the mouth's corners from the 68 points shape mask of [46]. Thus this evaluation scheme emphasizes on the internal facial areas (eyebrows, eyes, nose, mouth).

Figure 13 shows the results on LFPW testset, AFW, iBUG and Helen train and test databases (3026 images in total). A main difference between these two methods and AAMs is that due to their discriminative nature, they both require many data in order to generalize well, whilst the generative shape and appearance models of AAMs perform well with

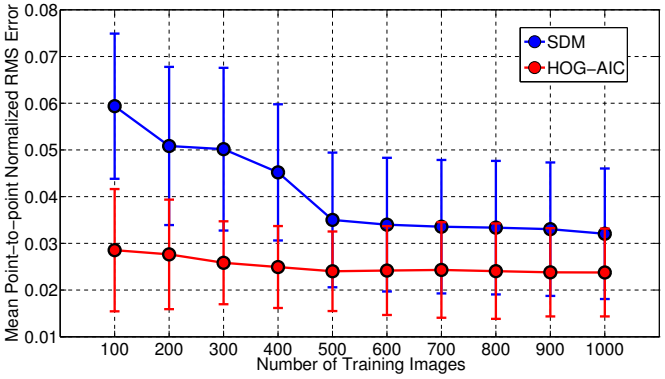


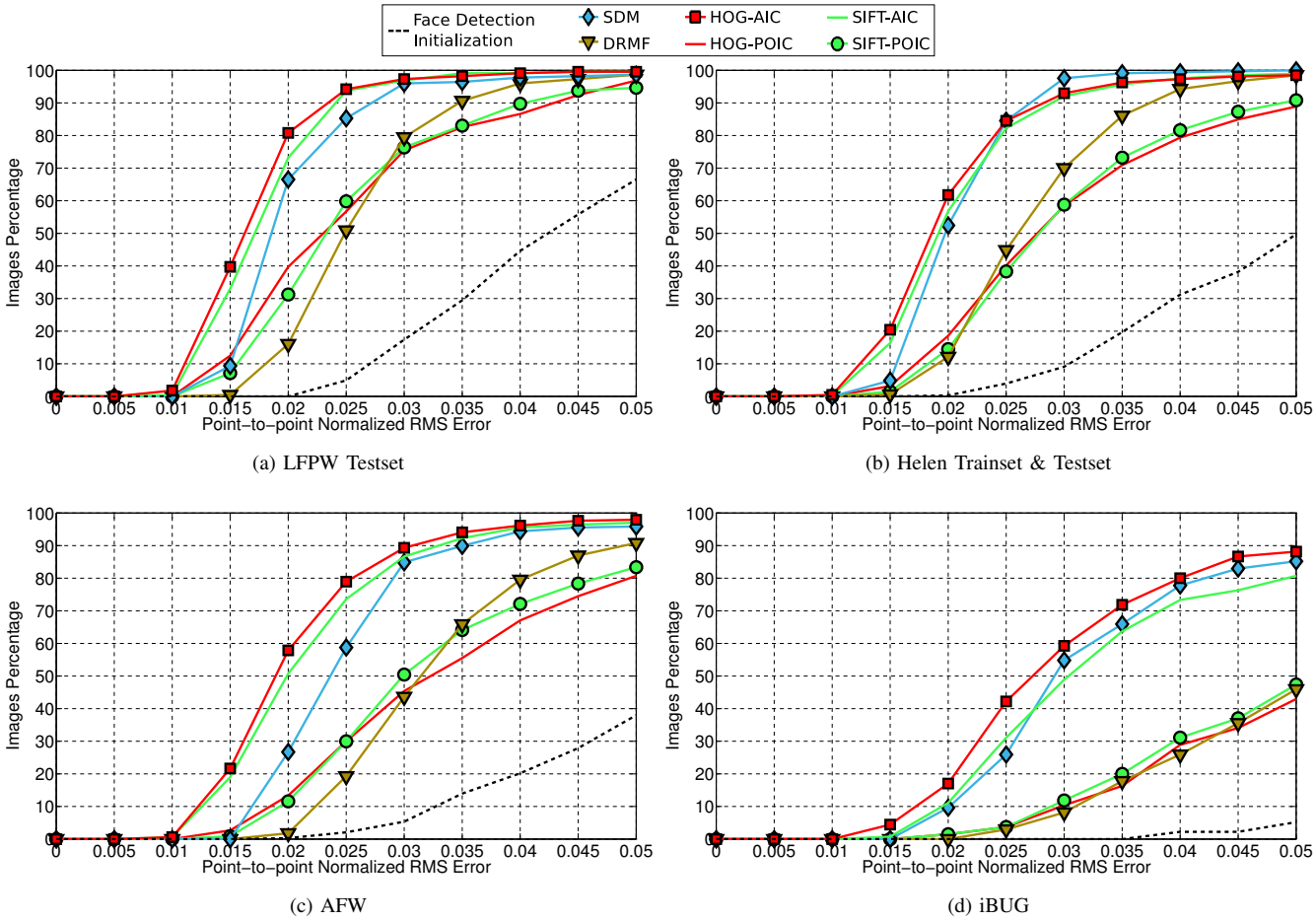
Fig. 12. Performance (mean and standard deviation) of HOG-AIC and SDM with respect to the number of training images. The performance is evaluated on Helen testset and is measured with the mean and standard deviation of the normalized RMS error. In this experiment we use our SDM implementation [49].

much fewer training images. This is shown in Fig. 12 which plots the performance of HOG-AIC and SDM with respect to the number of training images. Since SDMs's authors do not provide any training code [47], for this small experiment we employ our SDM version developed in the Menpo Project [49]. The training images are randomly selected from the 2811 images of LFPW and Helen trainsets and the evaluation is applied on Helen testing set. The graph shows that SDM keeps improving as the number of training images increases whilst the SIFT AAMs performance remains almost the same.

The results indicate that HOG-AIC and SIFT-AIC significantly outperform DRMF and are also more accurate than SDM. They are more accurate especially when they converge as can be seen from the percentage of images with error less or equal than 0.02. Even though SDM and DRMF have smaller computational complexities compared to Tab. II, we find these results remarkable, considering that our feature-based AAMs are trained using much fewer training images. Finally, the results show that the HOG and SIFT POIC models have a similar performance as DRMF.

D. Results Interpretation and Discussion

In general, it is very difficult to find a strict theoretical difference between the various employed non-linear features, such as HOG, SIFT, LBP etc., because the design of features still remains mainly an empirical art rather than an exact science. Nevertheless, we can sketch the difference between the magnitude of Gabor filters in various scales and orientations and SIFT features. Gabor features have been used before in literature [16], [23], however our experiments prove that they



(a) LFPW Testset

(b) Helen Trainset & Testset

(c) AFW

(d) iBUG

Fig. 13. Comparison between our proposed HOG and SIFT AAMs and two state-of-the-art methods: SDM [47] and DRMF [48]. The evaluation is based on 49 points mask, which means it does not include the face boundary (jaw). For SDM and DRMF we use the code provided by their authors.

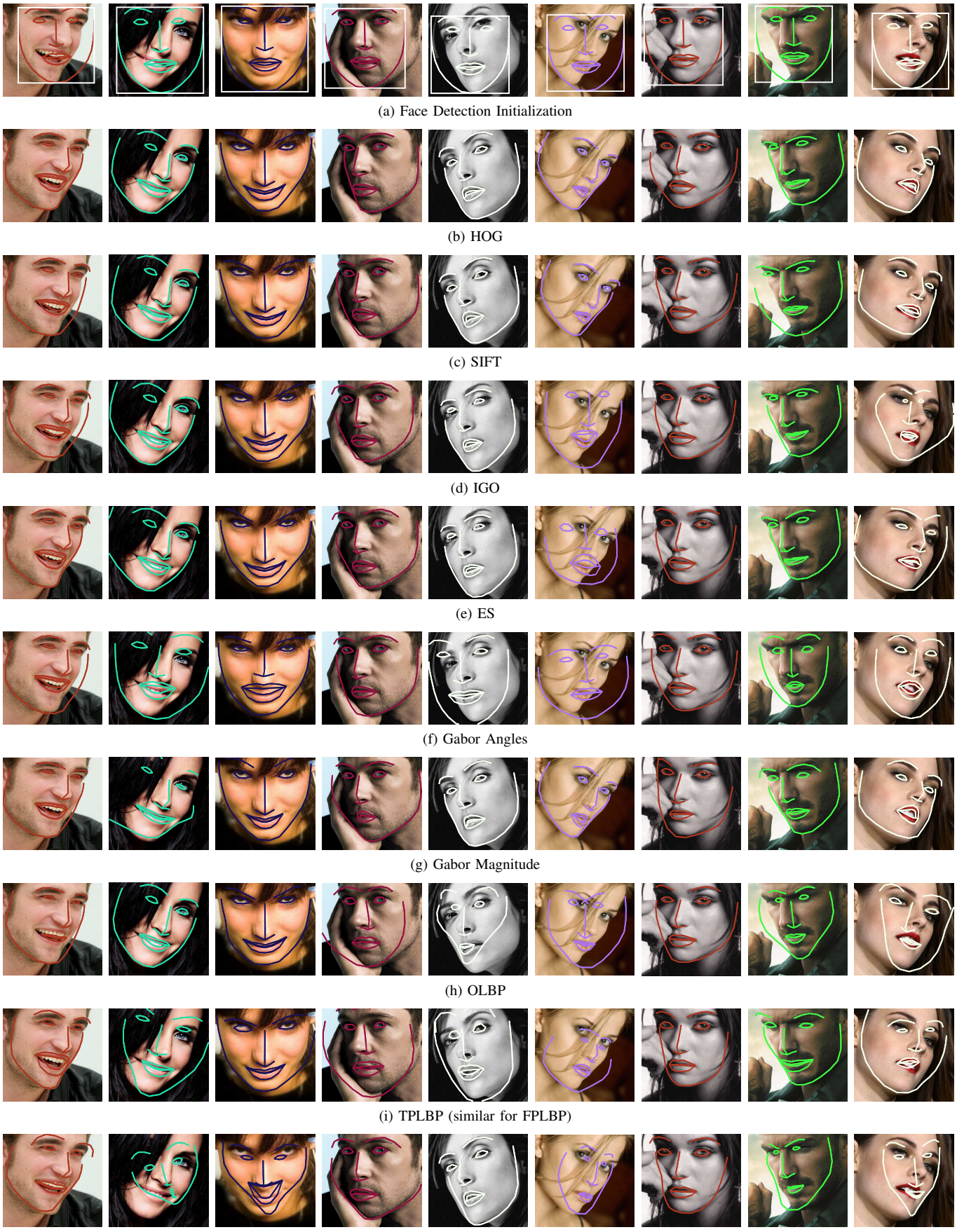
are not efficient for generic face alignment and are probably more suitable for person-specific settings [59], [60].

The difference between the complex response (i.e., having both the magnitude and the phase) of Gabor filters and other employed features is that the former are produced by the convolution of a bank of linear filters, hence they are not robust to the facial appearance changes [16]. This is the reason why we prefer to extract non-linear features from the responses, i.e. the magnitude (modulus) and the phase. Moreover, the difference between the magnitude of Gabor filters in various scales and orientations and SIFT features can be explained using the theory on invariant scattering networks [61], according to which SIFT features can be very well approximated by the modulus of the coefficients of the wavelet transform using a particular family of wavelets (i.e. partial derivatives of a Gaussian) (for more details please refer to Section 2.3 of [61]). Convolution with Gabor filters with different scales and orientations does not constitute a proper wavelet image transform. In general Gabor filter expansion is not applied in building a wavelet transform, since this requires computation of bi-orthogonal wavelets, which may be very time-consuming. Therefore, usually a filter bank consisting of Gabor filters with various scales and rotations [59], [60], as we do in this work, is created and applied for feature extraction. In general, the results suggest that large-scale features are very robust and have a high convergence frequency even with initializations that are too far from groundtruth. However, when the initialization is close to the optimal solution, higher-frequency features tend to be more accurate. For example the phase filter information may have excellent localization properties when the deformation is small, but it is very sensitive to noise and small perturbations.

Finally, we believe that the advantages of the employed features, especially the multi-channel gradient based ones such as HOG and SIFT, are excellently coupled with the generalization ability of generative models. In fact, we believe that the most important experimental result shown in the previous section is that the combination of (1) non-linear least-squares optimization with (2) robust features and (3) generative models can achieve very good performance without the need of large training datasets, which emphasizes the main advantage of the proposed framework over discriminative methods.

VI. Conclusions

In this paper we present a novel formulation of LK and AAMs alignment algorithms which employs dense feature descriptors for the appearance representation. We showed, both theoretically and experimentally, that by extracting the features from the input image once and then warping the features image has better performance and lower computational complexity



(a) Face Detection Initialization

(b) HOG

(c) SIFT

(d) IGO

(e) ES

(f) Gabor Angles

(g) Gabor Magnitude

(h) OLBP

(i) TPLBP (similar for FPLBP)

(j) Intensities

Fig. 14. Fitting examples using feature-based AIC on very challenging images from iBUG database.

than computing features from the warped image at each iteration. This allows us to take advantage of the descriptive qualities of various features in order to achieve robust and accurate performance for the problems of face alignment and fitting. Our LK experiments prove that feature-based face alignment is invariant to person ID and extreme lighting variations. Our face fitting experiments on challenging in-the-wild databases show that the feature-based AAMs have the ability to generalize well to unseen faces and demonstrate invariance to expression, pose and lighting variations. The presented experiments also provide a comparison between various features and prove that HOG and SIFT are the most powerful. Finally, we report face fitting results using AAMs with HOG and SIFT features that outperform discriminative state-of-the-art methods trained on thousands of images. We believe that the experimental results are among the major contributions of this paper, as they emphasize that the combination of highly-descriptive features with efficient optimization techniques leads to deformable models with remarkable performance.

ACKNOWLEDGMENT

The work of E. Antonakos was 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 was funded by a European DTA studentship from Imperial College London and by the Qualcomm Innovation Fellowship. The authors would like to thank James Booth and Patrick Snape for constructive cooperation on implementing LK and AAMs.

REFERENCES

[1] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proceedings of International Joint Conference on Artificial Intelligence*, vol. 81, 1981, pp. 674–679.

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

[3] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[5] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004.

[6] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[7] S. Baker, R. Gross, I. Matthews, and T. Ishikawa, "Lucas-kanade 20 years on: A unifying framework: Part 2," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-03-01, February 2003.

[8] G. Papandreou and P. Maragos, "Adaptive and constrained algorithms for inverse compositional active appearance model fitting," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2008.

[9] 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*, vol. 20, no. 10, pp. 1025–1039, 1998.

[10] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2001.

[11] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift flow: dense correspondence across different scenes," in *European Conference on Computer Vision*, 2008.

[12] J. Alabort-i-Medina and S. Zafeiriou, "Bayesian active appearance models," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2014.

[13] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2014.

[14] R. Mégret, J. Authesserre, and Y. Berthoumieu, "Bidirectional composition on lie groups for gradient-based image alignment," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2369–2381, 2010.

[15] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Robust and efficient parametric face alignment," in *Proceedings of IEEE International Conference on Computer Vision*, 2011.

[16] S. Lucey, S. Sridharan, R. Navarathna, and A. B. Ashraf, "Fourier lucas-kanade algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1383–1396, 2013.

[17] N. Dowson and R. Bowden, "Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 180–185, 2008.

[18] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 12, pp. 1080–1093, 2005.

[19] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[20] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2454–2466, 2012.

[21] G. Tzimiropoulos, J. Alabort-i-Medina, S. Zafeiriou, and M. Pantic, "Generic active appearance models revisited," in *Asian Conference on Computer Vision*, 2012.

[22] T. F. Cootes and C. J. Taylor, "On representing edge structure for model matching," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2001.

[23] X. Gao, Y. Su, X. Li, and D. Tao, "Gabor texture in active appearance models," *Journal of Neurocomputing*, vol. 72, no. 13, pp. 3174–3181, 2009.

[24] Y. Ge, D. Yang, J. Lu, B. Li, and X. Zhang, "Active appearance models using statistical characteristics of gabor based texture representation," in *Journal of Visual Communication and Image Representation*, 2013.

[25] I. M. Scott, T. F. Cootes, and C. J. Taylor, "Improving appearance model matching using local image structure," in *Information Processing in Medical Imaging*, 2003.

[26] P. Kittipanya-ngam and T. F. Cootes, "The effect of texture representations on aam performance," in *Proceedings of IEEE International Conference on Pattern Recognition*, 2006.

[27] M. B. Stegmann and R. Larsen, "Multi-band modelling of appearance," *Image and Vision Computing*, vol. 21, no. 1, pp. 61–67, 2003.

[28] Y. Su, D. Tao, X. Li, and X. Gao, "Texture representation in aam using gabor wavelet and local binary patterns," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2009.

[29] C. Wolstenholme and C. J. Taylor, "Wavelet compression of active appearance models," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1999.

[30] S. Darkner, R. Larsen, M. B. Stegmann, and B. Ersboll, "Wedgelet enhanced appearance models," in *Proceedings of IEEE Computer Vision and Pattern Recognition Workshop*, 2004.

[31] E. Antonakos and S. Zafeiriou, "Automatic construction of deformable models in-the-wild," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2014.

[32] E. Antonakos, J. Alabort-i-Medina, G. Tzimiropoulos, and S. Zafeiriou, "Hog active appearance models," in *Proceedings of IEEE International Conference on Image Processing*, 2014, pp. 224–228.

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2005.

[34] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of IEEE International Conference on Computer Vision*, 1999.

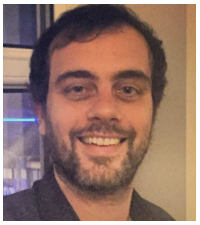
[35] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[36] T. Ojala, M. Pietikäinen, and T. Mäenpää, "A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification," in *International Conference on Advances in Pattern Recognition*, 2001.

[37] ——, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
[38] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *European Conference on Computer Vision (ECCV) Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008.
[39] P. Kovesi, "Symmetry and asymmetry from local phase," in *10th Australian Joint Conf. on Artificial Intelligence*, 1997.
[40] ——, "Image features from phase congruency," *VIDERE: Journal of computer vision research*, vol. 1, no. 3, pp. 1–26, 1999.
[41] T. S. Lee, "Image representation using 2d gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.
[42] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
[43] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2011.
[44] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2012.
[45] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *European Conference on Computer Vision*, 2012.
[46] 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 Intl Conf. on Computer Vision (ICCV-W 2013), 300 Faces in-the-Wild Challenge (300-W)*, 2013.
[47] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2013.
[48] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2013.
[49] 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*. ACM, 2014, pp. 679–682.
[50] G. Tzimiropoulos, J. Alabort-i-Medina, S. Zafeiriou, and M. Pantic, "Active orientation models for face alignment in-the-wild," *IEEE Transactions on Information Forensics and Security, Special Issue on Facial Biometrics in-the-wild*, vol. 9, pp. 2024–2034, December 2014.
[51] P. D. Kovesi, "MATLAB and Octave functions for computer vision and image processing," Centre for Exploration Targeting, Uni. of Western Australia, http://www.csse.uwa.edu.au/$\sim$pk/research/matlabfns/.
[52] G. Tzimiropoulos and M. Pantic, "Optimization problems for fast aam fitting in-the-wild," in *Proceedings of IEEE International Conference on Computer Vision*, 2013.
[53] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proceedings of IEEE Computer Vision and Pattern Recognition Workshop*, 2013.
[54] http://ibug.doc.ic.ac.uk/resources/300-W/.
[55] J. Orozco, B. Martinez, and M. Pantic, "Empirical analysis of cascade deformable models for multi-view face detection," in *Proceedings of IEEE International Conference on Image Processing*, 2013.
[56] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.
[57] http://www.humansensing.cs.cmu.edu/intraface/.
[58] https://sites.google.com/site/akshayasthana/clm-wild-code.
[59] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von Der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.
[60] B. Duc, S. Fischer, and J. Bigun, "Face authentication with gabor information on deformable graphs," *IEEE Transactions on Image Processing*, vol. 8, no. 4, pp. 504–516, 1999.
[61] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

Epameinondas Antonakos received his Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 2011. Since 2013, he is a member of the iBUG group, Department of Computing, Imperial College London, U.K., and he is pursuing the Ph.D. degree in computer vision under the supervision of Dr. Stefanos Zafeiriou. During 2011-2012, he was a Research Assistant in the CVSP group, School of Electrical and Computer Engineering, National Technical University of Athens, Greece, under the supervision of Prof. Petros Maragos. His research interests lie in the fields of computer vision, statistical machine learning and human-computer interaction.

Joan Alabort-i-Medina received the B.Sc. degree in computer science and engineering from the Universitat Autnoma de Barcelona, Barcelona, Spain, in 2008, and the M.Sc. degree in visual information processing from Imperial College London, London, U.K., in 2011, where he is currently pursuing the Ph.D. degree in computer vision. His research interests lie in the fields of computer vision, machine learning, and human–computer interaction.



Georgios (Yorgos) Tzimiropoulos received the M.Sc. and Ph.D. degrees in Signal Processing and Computer Vision from Imperial College London, U.K. He is Assistant Professor with the School of Computer Science at the University of Nottingham, U.K. Prior to this, he was Assistant Professor with the University of Lincoln, U.K. and Senior Researcher in the iBUG group, Department of Computing, Imperial College London. He is currently Associate Editor of the Image and Vision Computing Journal. His main research interests are in the areas of face and object recognition, alignment and tracking, and facial expression analysis.

Stefanos P. Zafeiriou (M'09) is currently a Senior Lecturer (equivalent to Associate Professor) in Pattern Recognition/Statistical Machine Learning for Computer Vision with the Department of Computing, Imperial College London, London, U.K. He was a recipient of the Prestigious Junior Research Fellowships from Imperial College London in 2011 to start his own independent research group. He has received various awards during his doctoral and post-doctoral studies. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and the Image and Vision Computing Journal. He has been a Guest Editor of over five journal special issues and co-organized over five workshops/special sessions in top venues, such as CVPR/FG/ICCV/ECCV. He has coauthored over 40 journal papers mainly on novel statistical machine learning methodologies applied to computer vision problems, such as 2-D/3-D face analysis, deformable object fitting and tracking, shape from shading, and human behaviour analysis, published in the most prestigious journals in his field of research, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the International Journal of Computer Vision, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTERGRAPHICS, and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and many papers in top conferences, such as CVPR, ICCV, ECCV, ICML. His students are frequent recipients of very prestigious and highly competitive fellowships, such as the Google Fellowship, the Intel Fellowship, and the Qualcomm Fellowship.