SUPPLEMENTARY MATERIAL FOR: ADAPTIVE CASCADED REGRESSION

Epameinondas Antonakos[⋆,†]*, Patrick Snape*[⋆,†]*, George Trigeorgis*[†]*, Stefanos Zafeiriou*[†,*]

[†]Department of Computing, Imperial College London, U.K.
[*]Center for Machine Vision and Signal Analysis, University of Oulu, Finland

1. METHOD

In this section we provide a more detailed explanation of the proposed method presented in Section 2 of the main paper. Specifically, we first define our basic notations (Sec. 1.1) and then present details of the discriminative (Sec. 1.2) and generative (Sec. 1.3) models in order to formulate our unified model (Sec. 1.4).

1.1. Shape and Appearance Models

1.1.1. Shape representation and model

In the problem of generic deformable object alignment, the sparse shape of an object consists of n landmark points that are located on semantic parts of the object. By denoting the coordinates of a landmark point within the Cartesian space of an image $\mathbf{I}$ as $\mathbf{x}_i = [x_i, y_i]^T$, then the *shape instance* of the object is given by the $2n \times 1$ vector

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

Given a set of N such shape samples $\{\mathbf{s}^1, \ldots, \mathbf{s}^N\}$, a parametric statistical subspace of the object's shape variance can be retrieved by first applying Generalised Procrustes Analysis on the shapes to normalise them with respect to the global similarity transform (i.e., scale, in-plane rotation and translation) and then using Principal Component Analysis (PCA). The returned shape subspace is further augmented with four eigenvectors that control the global similarity transform of the object's shape. Please refer to [1] for further details about orthonormalising the similarity eigenvectors with the PCA basis. The resulting *shape model* $\{\mathbf{U}_s, \bar{\mathbf{s}}\}$ consists of the orthonormal basis $\mathbf{U}_s \in \mathbb{R}^{2n \times n_s}$ with n_s eigenvectors (including the four similarity components concatenated before the eigenvectors) and the mean shape vector $\bar{\mathbf{s}} \in \mathbb{R}^{2n}$. This parametric model can be used to generate new shape instances as

$$\mathbf{s}(\mathbf{p}) = \bar{\mathbf{s}} + \mathbf{U}_s \mathbf{p} \tag{2}$$

where $\mathbf{p} = [p_1, \ldots, p_{n_s}]^T$ is the $n_s \times 1$ vector of *shape parameters* that control the linear combination of the eigenvectors.

1.1.2. Appearance representation and model

Until recently, the appearance of an object was mainly represented in a holistic way, i.e., the whole appearance information was employed and the texture extraction was performed through a piecewise warp function that maps the pixel coordinates for a shape instance to a common reference space. The scientific community has lately turned towards part-based appearance representation, i.e., extracting appearance patches centred around the landmark coordinates. Although this depends on the object class and application, in general, the part-based representation has proved to be more effective than the holistic as the warp function is replaced by a simple sampling function and thus is more natural for articulated rigid objects. Let us denote the vectorised form of an $h \times w$ image patch that corresponds to the image location $\mathbf{x}_i$ as

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

where $\mathbf{\Omega}_{\mathbf{x}_i}$ is a set of discrete neighbouring pixel locations $\mathbf{z}_j = [x_j, y_j]^T$ within a rectangular region centred at location $\mathbf{x}_i$ and hw is the image patch vector's length. Moreover, let us define a feature extraction function $\mathcal{H} : \mathbb{R}^{hw} \to \mathbb{R}^m$, which extracts a descriptor vector of length m (e.g. SIFT [2], HOG [3]) given an appearance vector. We denote the procedure of extracting a feature-based vector from a patch centred at a given image location by the function

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

Consequently, the appearance vector of length $mn \times 1$ that corresponds to a shape instance is expressed as

$$\boldsymbol{\phi}(\mathbf{s}) = \left[\mathcal{F}(\mathbf{x}_1)^T, \ldots, \mathcal{F}(\mathbf{x}_n)^T\right]^T \tag{5}$$

and involves the concatenation of the vectorised feature-based image patches that correspond to the n landmarks of the shape instance. Similar to the shape case, given a set of N such appearance samples $\{\boldsymbol{\phi}^1, \ldots, \boldsymbol{\phi}^N\}$ and applying PCA, we obtain a parametric statistical *appearance model* $\{\mathbf{U}_a, \bar{\mathbf{a}}\}$ that

[⋆]Alphabetical order due to equal contribution.

E. Antonakos is funded by the EPSRC project EP/J017787/1 (4D-FAB). P. Snape and G. Trigeorgis are funded by a DTA from Imperial College London. S. Zafeiriou is partially supported by the 4D-FAB and EP/L026813/1 (ADAManT) projects.

consists of the orthonormal basis $\mathbf{U}_a \in \mathbb{R}^{mn \times n_a}$ with n_a eigenvectors and the mean appearance vector $\bar{\mathbf{a}} \in \mathbb{R}^{mn}$. An appearance instance can be generated as

$$\mathbf{a}(\mathbf{c}) = \bar{\mathbf{a}} + \mathbf{U}_a \mathbf{c} \tag{6}$$

where $\mathbf{c} = [c_1, \ldots, c_{n_a}]^T$ is the $n_a \times 1$ vector of *appearance parameters*. Finally, let us define

$$\mathbf{P} = \mathbf{E} - \mathbf{U}_a \mathbf{U}_a^T \tag{7}$$

which is the orthogonal complement of the appearance subspace $\mathbf{U}_a$, where $\mathbf{E}$ denotes the $mn \times mn$ identity matrix. This projection operator is used in order to project-out the appearance variance in the following methods.

1.2. Cascaded Regression Discriminative Model

Herein, we present a fully parametric cascaded regression model. We employ an appearance model and learn a regression function that regresses from the object's projected-out appearance to the parameters of a linear shape model. Let us assume that we have a set of N training images $\{\mathbf{I}^1, \ldots, \mathbf{I}^N\}$ and their corresponding annotated shapes $\{\mathbf{s}^1, \ldots, \mathbf{s}^N\}$. By projecting each ground-truth shape to the shape basis $\mathbf{U}_s$, we get the set of ground-truth shape parameters $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_N^*\}$. Moreover, we aim to learn a cascade of K levels, i.e. $k = 1, \ldots, K$. During the training process of each level, we generate a set of P perturbed shape parameters $\mathbf{p}_{i,j}^k,\ j = 1, \ldots, P,\ i = 1, \ldots, N$, which are sampled from a distribution that models the statistics of the detector employed for initialisation. By defining $\Delta\mathbf{p}_{i,j}^k = \mathbf{p}_i^* - \mathbf{p}_{i,j}^k,\ j = 1, \ldots, P,\ i = 1, \ldots, N$ to be a set of shape parameters increments, the least-squares problem that we aim to solve during training at each cascade level k is

$$\arg\min_{\mathbf{W}^k} \sum_{i=1}^{N} \sum_{j=1}^{P} \left\| \Delta\mathbf{p}_{i,j}^k - \mathbf{W}^k \mathbf{P} \left(\boldsymbol{\phi}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) - \bar{\mathbf{a}} \right) \right\|_2^2 \tag{8}$$

where $\mathbf{P}$ is the projection operator defined in Eq. 7 and $\boldsymbol{\phi}_i(\cdot)$ denotes the vector of concatenated feature-based patches extracted from the training image $\mathbf{I}^i$, as defined in Eq. 5. Note that the bias term of the above objective function is substituted by the mean appearance vector $\bar{\mathbf{a}}$. By denoting

$$\hat{\boldsymbol{\phi}}_{i,j,k} = \mathbf{P} \left(\boldsymbol{\phi}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) - \bar{\mathbf{a}} \right) \tag{9}$$

to be the projected-out residual, then the closed-form solution to the above least-squares problem is given by

$$\mathbf{W}^k = \left(\sum_{i=1}^{N} \sum_{j=1}^{P} \Delta\mathbf{p}_{i,j}^k \hat{\boldsymbol{\phi}}_{i,j,k}^T \right) \left(\sum_{i=1}^{N} \sum_{j=1}^{P} \hat{\boldsymbol{\phi}}_{i,j,k} \hat{\boldsymbol{\phi}}_{i,j,k}^T \right)^{-1} \tag{10}$$

for each level of the cascade $k = 1, \ldots, K$.

During testing, given the current estimate of the shape parameters $\mathbf{p}_k$) that was computed at cascade level k, we create the feature-based image vector $\boldsymbol{\phi}(\mathbf{s}(\mathbf{p}_k))$, subtract the mean appearance vector $\bar{\mathbf{a}}$, project-out the appearance variation and estimate the shape parameters increment as

$$\Delta\mathbf{p}_k = \mathbf{W}^k \mathbf{P} \left(\boldsymbol{\phi}(\mathbf{s}(\mathbf{p}_k)) - \bar{\mathbf{a}} \right) \tag{11}$$

Then, the shape parameters vector is updated as

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \Delta\mathbf{p}_{k-1} \tag{12}$$

where we set $\mathbf{p}_0 = \mathbf{0}$ at the first iteration. The computational complexity of Eq. 11 per cascade level is $\mathcal{O}(n_s mn)$, thus the complexity per test image is $\mathcal{O}(K n_s mn)$.

1.3. Gauss-Newton Generative Model

The optimisation of an AAM aims to minimise the reconstruction error of the input image with respect to the shape and appearance parameters, i.e.,

$$\arg\min_{\mathbf{p},\mathbf{c}} \left\| \boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a \mathbf{c} \right\|_2^2 \tag{13}$$

where we employ the appearance model of Eq. 6 and $\boldsymbol{\phi}(\cdot)$ denotes the vectorised form of the input image as defined in Eq. 5. This cost function is commonly optimised in an iterative manner using the Gauss-Newton algorithm. This algorithm introduces an incremental update for the shape and appearance parameters, i.e. $\Delta\mathbf{p}$ and $\Delta\mathbf{c}$ respectively, and solves the problem with respect to $\Delta\mathbf{p}$ by first linearising using first-order Taylor expansion around $\Delta\mathbf{p} = \mathbf{0}$. The Gauss-Newton optimisation can be performed either in a forward or in an inverse manner, depending on whether the incremental update of the shape parameters is applied on the image or the model, respectively. In this paper, we focus on the *inverse* algorithm, however the forward case can be derived in a similar way.

We follow the derivation that was first presented in [4] and later was readily employed in [5, 6]. By applying the incremental shape parameters on the part of the model, the cost function of Eq. 13 becomes

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} \left\| \boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}}(\Delta\mathbf{p}) - \mathbf{U}_a(\Delta\mathbf{p})(\mathbf{c} + \Delta\mathbf{c}) \right\|_2^2 \tag{14}$$

where $\bar{\mathbf{a}}(\Delta\mathbf{p}) = \bar{\mathbf{a}}(\mathbf{s}(\Delta\mathbf{p}))$ and $\mathbf{U}_a(\Delta\mathbf{p}) = \mathbf{U}_a(\mathbf{s}(\Delta\mathbf{p}))$. Given the part-based nature of our model, the compositional update of the parameters at each iteration is reduced to a simple subtraction [6], as

$$\mathbf{p} \leftarrow \mathbf{p} - \Delta\mathbf{p} \tag{15}$$

By taking the first order Taylor expansion around $\Delta\mathbf{p} = \mathbf{0}$, we arrive at

$$\arg\min_{\Delta\mathbf{p},\Delta\mathbf{c}} \left\| \boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_a(\mathbf{c} + \Delta\mathbf{c}) - \mathbf{J}_a \Delta\mathbf{p} \right\|_2^2 \tag{16}$$

where

$$\mathbf{J}_a = \mathbf{J}_{\bar{\mathbf{a}}} + \sum_{i=1}^{m} c_i \mathbf{J}_i \tag{17}$$

is the model Jacobian. This Jacobian consists of the mean appearance Jacobian $\mathbf{J}_{\bar{\mathbf{a}}} = \frac{\partial \bar{\mathbf{a}}}{\partial \mathbf{p}}$ and the Jacobian of each appearance eigenvector denoted as $\mathbf{J}_i,\ i = 1, \ldots, m$.

By employing the projection operator of Eq. 7 in order to work on the orthogonal complement of the appearance subspace $\mathbf{U}_a$ and using the fact that $\mathbf{P}\mathbf{U}_a = \mathbf{P}^T\mathbf{U}_a = \mathbf{0}$, the above cost function can be expressed as

$$\arg\min_{\Delta \mathbf{p}} \|\boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{J}_a \Delta \mathbf{p}\|^2_{\mathbf{P}} \tag{18}$$

The solution to this least-squares problem is

$$\Delta \mathbf{p} = \hat{\mathbf{H}}_a^{-1} \hat{\mathbf{J}}_a^T (\boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}}) \tag{19}$$

where

$$\hat{\mathbf{J}}_a = \mathbf{P}\mathbf{J}_a \text{ and } \hat{\mathbf{H}}_a = \hat{\mathbf{J}}_a^T \hat{\mathbf{J}}_a \tag{20}$$

are the projected-out Jacobian and Hessian matrices respectively. Note that even though $\mathbf{J}_{\bar{\mathbf{a}}}$ and $\mathbf{J}_i$ can be precomputed, the complete model Jacobian $\mathbf{J}_a$ depends on the appearance parameters $\mathbf{c}$ and has to be recomputed at each iteration. Given the current estimate of $\Delta \mathbf{p}$, the solution of $\mathbf{c}$ with respect to the current estimate $\mathbf{c}_c$ can be retrieved as

$$\mathbf{c} = \mathbf{c}_c + \mathbf{U}_a^T \left(\boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}} - \mathbf{U}_{\mathbf{a}}\mathbf{c}_c - \mathbf{J}_a \Delta \mathbf{p}\right) \tag{21}$$

Thus, the computational complexity of computing Eq. 19 per iteration is $\mathcal{O}(n_s n_a mn + n_s^2 mn)$.

The inverse approach that we followed, which was first proposed in [4], is different from the well-known project-out inverse compositional method of [7]. Specifically, in our case, the linearisation of the cost function is performed *before* projecting-out. On the contrary, the authors in [7] followed the approximation of *projecting-out first and then linearising*, which eleminates the need to recompute the appearance subspace Jacobian. However, the project-out method proposed by [7] does not generalise well and is not suitable for generic facial alignment.

Given the fact that $\mathbf{P}^T = \mathbf{P}$ and $\mathbf{P}^T\mathbf{P} = \mathbf{P}$, then the solution of Eq. 19 can be expanded as

$$\Delta \mathbf{p} = (\mathbf{J}_a^T \mathbf{P} \mathbf{J}_a)^{-1} \mathbf{J}_a^T \mathbf{P} (\boldsymbol{\phi}(\mathbf{s}(\mathbf{p})) - \bar{\mathbf{a}}) \tag{22}$$

Thus, it is worth mentioning that the solution of the regression-based model in Eq. 11 is equivalent to the Gauss-Newton solution of Eq. 19 if the regression matrix has the form

$$\mathbf{W}^k = (\mathbf{J}_a^T \mathbf{P} \mathbf{J}_a)^{-1} \mathbf{J}_a^T \tag{23}$$

which further reveals the equivalency of the two cost functions of Eqs. 8 and 18.

1.4. Adaptive Cascaded Regression

As previously explained, both the AAMs of Section 1.2 and traditional SDMs as in 1.3 suffer from a number of disadvantages. To address these disadvantages, we propose ACR which combines the two previously described discriminative and generative optimisation problems into a single unified cost function. Specifically, by employing the regression-based objective function of Eq. 8 along with the Gauss-Newton analytical solution of Eq. 19, the training procedure of ACR aims to minimise

$$\sum_{i=1}^{N} \sum_{j=1}^{P} \left\| \Delta \mathbf{p}_{i,j}^k - \left(\lambda^k \mathbf{W}^k - (1 - \lambda^k) \mathbf{H}_{i,j}^{-1} \mathbf{J}_{i,j}^T\right) \hat{\boldsymbol{\phi}}_{i,j,k} \right\|_2^2 \tag{24}$$

with respect to $\mathbf{W}^k$, where

$$\hat{\boldsymbol{\phi}}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) = \mathbf{P}\left(\boldsymbol{\phi}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) - \bar{\mathbf{a}}\right) \tag{25}$$

is the projected-out residual and $\mathbf{H}_{i,j}$ and $\mathbf{J}_{i,j}$ denote the Hessian and Jacobian matrices, respectively, of the Gauss-Newton optimisation algorithm per image $i = 1, \ldots, N$ and per perturbation $j = 1, \ldots, P$. λ_k is a hyperparameter that controls the weighting between the regression-based descent directions and the Gauss-Newton gradient descent directions at each level of the cascade $k = 1, \ldots, K$. The negative sign in front of the gradient descent directions is due to the fact that the shape parameters update within the inverse Gauss-Newton optimisation is performed with subtraction, as shown in Eq. 15.

1.4.1. Training

During training, ACR aims to learn a cascade of K optimal linear regressors given the gradient descent directions of each training image at each level. Let us assume that we have a set of N training images $\{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$ along with the corresponding ground truth shapes $\{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$. We also assume that we have recovered the ground truth shape parameters for each training image $\{\mathbf{p}_1^*, \ldots, \mathbf{p}_N^*\}$ by projecting the ground truth shapes against the shape model.

Perturbations Before performing the training procedure, we generate a set of initialisations per training image, so that the regression function of each cascade level learns how to estimate the descent directions that optimise from these initialisations to the ground truth shape parameters. Consequently, for each training image, we first align the mean shape $\bar{\mathbf{s}}$ with the ground truth shape $\mathbf{s}^i$, project it against the shape basis $\mathbf{U}_s$ and then generate a set of P random perturbations for the first four shape parameters that correspond to the global similarity transform. Thus, we have a set of shape parameter vectors $\mathbf{p}_{i,j}^k,\ \forall i = 1, \ldots, N,\ \forall j = 1, \ldots, P$. Since the random perturbations are applied on the first four parameters, the rest of them remain zero, i.e., $\mathbf{p}_{i,j}^k = [p1_{i,j}^k, p2_{i,j}^k, p3_{i,j}^k, p4_{i,j}^k, \mathbf{0}_{n_s-4\times 1}^T]^T$. Moreover, the

perturbations are sampled from a distribution that models the statistics of the detector that will be used for automatic initialisation at testing time. This procedure is necessary only because we have a limited number of training images and can be perceived as training data augmentation. It could be avoided if we had more annotated images and a single initialisation per image using the detector would be adequate. The perturbations are performed once at the beginning of the training procedure of ACR. The steps that are applied at each cascade level $k = 1, \ldots, K$, in order to estimate $\mathbf{W}^k$, are the following:

Step 1: Shape Parameters Increments Given the set of vectors $\mathbf{p}_{i,j}^k$, we formulate the set of shape parameters increments vectors $\Delta\mathbf{p}_{i,j}^k = \mathbf{p}_i^* - \mathbf{p}_{i,j}^k,\ \forall i = 1, \ldots, N,\ \forall j = 1, \ldots, P$ and concatenate them in a $n_s \times NP$ matrix

$$\Delta\mathbf{P}_k = \left[\Delta\mathbf{p}_{1,1}^k \ \cdots \ \Delta\mathbf{p}_{N,P}^k\right] \tag{26}$$

Step 2: Projected-Out Residuals The next step is to compute the part-based appearance vectors from the perturbed shape locations $\boldsymbol{\phi}_i(\mathbf{s}(\mathbf{p}_{i,j}^k))$ and then the projected-out residuals of Eq. 25 $\forall i = 1, \ldots, N,\ \forall j = 1, \ldots, P$. These vectors are then concatenated in a single $mn \times NP$ matrix as

$$\hat{\boldsymbol{\Phi}}_k = \left[\hat{\boldsymbol{\phi}}_1(\mathbf{s}(\mathbf{p}_{1,1}^k)) \ \cdots \ \hat{\boldsymbol{\phi}}_N(\mathbf{s}(\mathbf{p}_{N,P}^k))\right] \tag{27}$$

Step 3: Gradient Descent Directions Compute the Gauss-Newton solutions for all the images and their perturbed shapes and concatenate them in a $n_s \times NP$ matrix as

$$\mathbf{G}_k = (1 - \lambda^k) \begin{bmatrix} [\mathbf{H}_{1,1}^{-1}\mathbf{J}_{1,1}^T\hat{\boldsymbol{\phi}}_1(\mathbf{s}(\mathbf{p}_{1,1}^k))]^T \\ \vdots \\ [\mathbf{H}_{i,j}^{-1}\mathbf{J}_{i,j}^T\hat{\boldsymbol{\phi}}_i(\mathbf{s}(\mathbf{p}_{i,j}^k))]^T \\ \vdots \\ [\mathbf{H}_{N,P}^{-1}\mathbf{J}_{N,P}^T\hat{\boldsymbol{\phi}}_N(\mathbf{s}(\mathbf{p}_{N,P}^k))]^T \end{bmatrix}^T \tag{28}$$

Based on the expanded solution of Eq. 22, the calculation of the Jacobian and Hessian per image involves the estimation of the appearance parameters using Eq. 21 and then

$$\begin{aligned} \mathbf{J}_{i,j} &= \mathbf{J}_a \\ \mathbf{H}_{i,j} &= \mathbf{J}_{i,j}^T\mathbf{P}\mathbf{J}_{i,j} \end{aligned} \tag{29}$$

where $\mathbf{J}_a$ is computed based on Eq. 17 for each image.

Step 4: Regression Descent Directions By using the matrices definitions of Eqs. 26, 27 and 28, the cost function of ACR in Eq. 24 takes the form

$$\arg\min_{\mathbf{W}^k} \left\|\Delta\mathbf{P}_k - \lambda^k\mathbf{W}^k\hat{\boldsymbol{\Phi}}_k + \mathbf{G}_k\right\|_2^2 \tag{30}$$

The closed-form solution of the above least-squares problem is

$$\mathbf{W}^k = \frac{1}{\lambda^k}\left(\Delta\mathbf{P}_k + \mathbf{G}_k\right)\left(\hat{\boldsymbol{\Phi}}_k^T\hat{\boldsymbol{\Phi}}_k\right)^{-1}\hat{\boldsymbol{\Phi}}_k^T \tag{31}$$

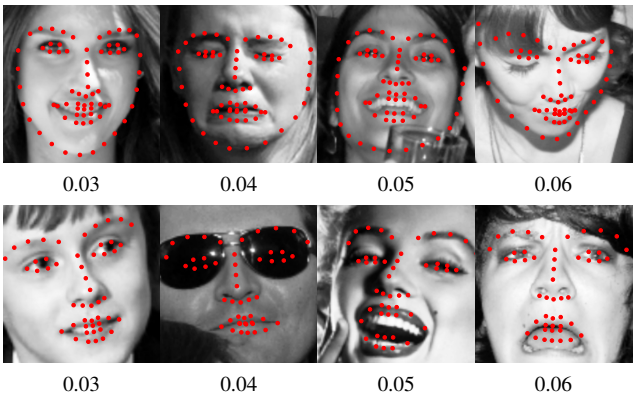

Fig. 1: Representative examples of increasing normalised errors. (*top*) 68-points. (*bottom*) 49-points.

Note that the regression matrix of this step is estimated only in case $\lambda_k \geq 0$. If $\lambda_k = 0$, then we directly set $\mathbf{W}_k = \mathbf{0}_{n_s \times mn}$

Step 5: Shape Parameters Update The final step is to generate the new estimates of the shape parameters per training image. By employing Eqs. 31 and 29, this is achieved as

$$\mathbf{p}_{i,j}^{k+1} = \mathbf{p}_{i,j}^k + \left(\lambda_k\mathbf{W}^k - (1 - \lambda_k)\mathbf{H}_{i,j}^{-1}\mathbf{J}_{i,j}^T\right)\boldsymbol{\phi}_i(\mathbf{s}(\mathbf{p}_{i,j}^k)) \tag{32}$$

$\forall i = 1, \ldots, N$ and $\forall j = 1, \ldots, P$. After obtaining $\mathbf{p}_{i,j}^{k+1}$, steps 1-5 are repeated for the next cascade level.

1.4.2. Fitting

In the fitting phase, given an unseen testing image $\mathbf{I}$ and its initial shape parameters $\mathbf{p}^0 = [p_1^0, p_2^0, p_3^0, p_4^0, \mathbf{0}]^T$, we compute the parameters update at each cascade level $k = 1, \ldots, K$ as

$$\mathbf{p}^k = \mathbf{p}^{k-1} + \left(\lambda_k\mathbf{W}^k - (1 - \lambda_k)\mathbf{H}^{-1}\mathbf{J}^T\right)\boldsymbol{\phi}(\mathbf{s}(\mathbf{p}^{k-1})) \tag{33}$$

where the Jacobian and Hessian are computed as described in Step 3 of the training procedure (Eq. 29). The computational complexity per iteration is $\mathcal{O}(n_s mn(n_a + n_s + 1))$.

2. EXPERIMENTAL RESULTS

This section complements Section 3 of the main paper by presenting additional experimental results.

Datasets We use the 68-point annotations provided by [8, 9, 10] for a number of existing databases including LFPW [11], HELEN [12] and AFW [13]. The 300-W competition [8, 10] also introduced a new challenging dataset called IBUG, annotated with the same 68-points. For all experiments, we used the bounding boxes provided by the 300-W competition [8] for initialisation in both training and testing.

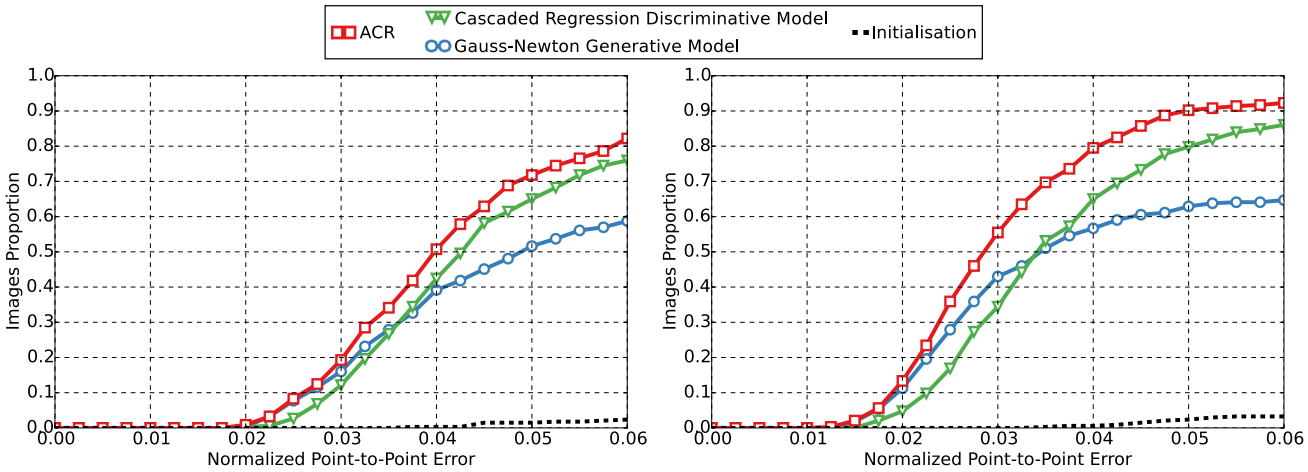

Fig. 2: ACR, AAM (Gauss-Newton) and SDM (Discriminative), trained identically, tested on the images of AFW. (*left*) 68-point error. (*right*) 49-point error. Initialisation given by the bounding boxes of [8].

Evaluation To maintain consistency with the results of the original 300-W competition, we report cumulative error distribution (CED) graphs using the point-to-point error normalised by the interocular distance defined by the outer eye corners. The mean error often reported in recent works [14, 15] is highly biased by alignments that completely fail. Therefore, we believe that the failure rate as shown in [16] is a much more informative error metric. To complement the failure rate, we propose the area under the curve (AUC), which enables simpler comparison of CED curves that are otherwise difficult to compare. We fix a maximum error that we believe represents a failed fitting, and thus the higher the AUC, the more fittings are concentrated within this acceptable fitting area. In all experiments, CED curves and AUC errors are reported up to 0.06. Examples of different errors are given in Figure 1, which shows that 0.06 represents an alignment failure.

Implementation Details The following settings were used for training ACR. 20 components were kept for the shape model and 300 for the appearance model. After running extended cross-validation experiments, we found that the best performance is obtained by using a cascade of 14 levels and setting $\lambda = [1,\ 0.75,\ 0.5,\ 0.25]$ for the first four and $\lambda = 0$ for the rest. Intuitively, this means that the regression-based descent directions need to dominate the optimisation on the first few iterations, as they are able to move towards the correct direction with steps of large magnitude. After that, the gradient descent steps are sufficient in order to converge to an accurate local minimum. The first two were performed on the image at half scale, the rest at full scale. The patch sizes were $[(32 \times 32), (24 \times 24), (24 \times 24), (16 \times 16)]$ for the first four cascades and (24×24) for the rest. Dense SIFT [17, 2] features were used for all methods. When performing a regression, a ridge parameter of 100 was used. In order to increase the size of the training data, we augment it by perturbing the provided bounding boxes of the 300-W competition with uniform noise of 0.005 for scaling and 0.07 for translation (scaled by the bounding box size). The same options were used for training the generative model (AAM) and the discriminative cascaded-regression (SDM) using the implementations in the Menpo Project [18].

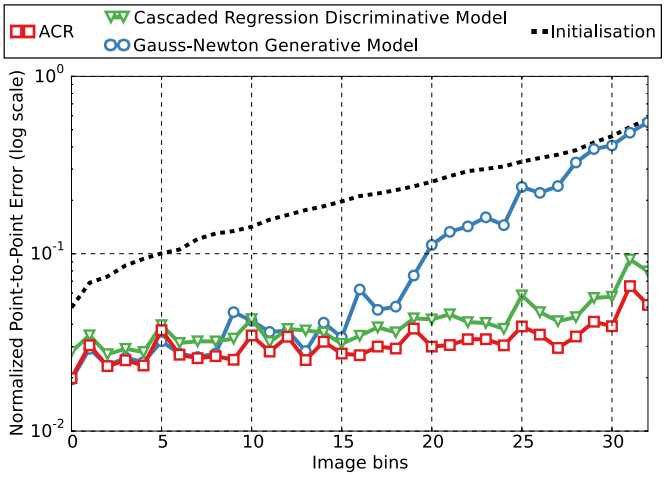

Fig. 3: Sorted initial errors of 10 random initialisations of each image in the AFW dataset. As the initial error increases, the AAM is unable to converge, whereas ACR is both robust to initialisations and consistently accurate.

2.1. Self Evaluation

In the following experiments we performed self evaluations, comparing ACR to both the generative AAM and the discriminative SDM. In each case, we trained the SDM or AAM in the same manner as the corresponding part of ACR. We trained all 3 of the methods on LFPW (training, 811 images),

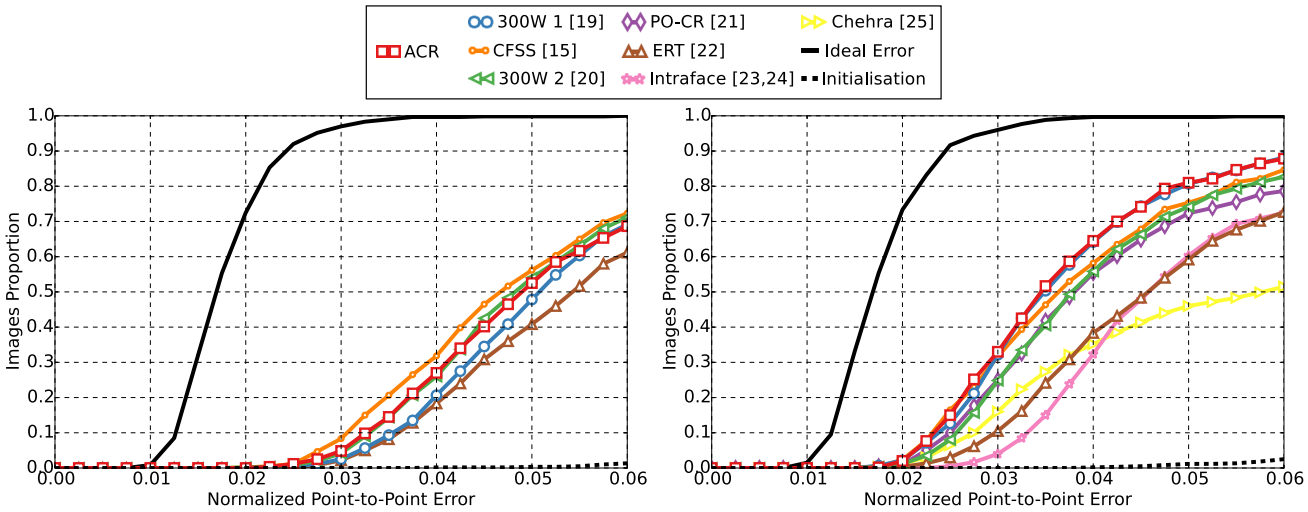

Fig. 4: Normalized error for the testing dataset of 300-W challenge [8, 10]. This database was unseen by all participants and thus represents a fair benchmark for state-of-the-art face alignment methods. (*left*) 68-point error. (*right*) 49-point error.

Method	*AUC*	*Failure rate (%)*
ACR	**0.43**	11.0
300W 1 [19]	0.42	**9.3**
CFSS [15]	0.40	13.5
300W 2 [20]	0.38	14.2
PO-CR [21]	0.37	17.7
ERT [22]	0.28	23.7
Intraface [23, 24]	0.27	23.8
Chehra [25]	0.24	46.8
Initialisation	0.01	96.8

Table 1: The area under the curve (AUC) and percentage failure rate for the 49-point CED curve given in Figure 4. Failure rate is the % of images with error > 0.06.

HELEN (training, 2000 images) and IBUG (135 images). The testing database was chosen as AFW (337 images) as recent works (e.g. [21]) have shown that AFW is still a challenging dataset. Figure 2 shows the CED curves for the SDM, AAM and ACR for both the 68-point and 49-point errors. Figure 2 clearly shows the improved performance of ACR over both SDM and AAM. To demonstrate the sensitivity of generative methods to initialisations, we repeated the experiment on AFW by generating 10 initialisations per image and then sorted the initialisation errors (low-to-high). We then binned the initialisation errors and plotted the final error of the SDM, AAM and ACR with respect to increasing initial errors. Figure 3 shows the results of this initialisation experiment. Here we can clearly see that, as the initialisation error increases, the AAM is incapable of converging towards an acceptable local-minima. It also shows that, although the SDM performs well, ACR outperforms it across all initialisation errors.

2.2. Comparison with state-of-the-art

Herein, we compare the proposed method with the current state-of-the-art techniques on various testing databases. Specifically, we test on the testsets of Labelled Faces In-th-Wild (LFPW) [11] and HELEN [12] databases, as well as the testing dataset of the 300-W challenge [8, 10]. The following subsections are separated with respect to the employed testing database.

For all the experiments, we used the same implementation details as the ones mentioned in the main paper and we also employed the annotations provided by [8, 9, 10]. Additionally, all methods are once again initialised with the bounding boxes that are provided by the 300-W competition [8, 10]. The methods that we compare with are the same as the ones of the main paper, i.e., Zhou et al. (300W 1) [19], Yan et al. (300W 2) [20], Coarse-to-fine Shape Searching (CFSS) [15], Project-Out Cascaded Regression (PO-CR) [21], Ensemble of Regression Trees (ERT) [22], Intraface [23, 24] and Chehra [25]. All the results are reported using the error metric of the 300-W competition [8, 10] based on 68 and 49 points. However, note that the public implementations of PO-CR, Intraface and Chehra only return 49-points, and thus they are not included in the 68-point error results. In all the experiemnts, ACR was trained using LFPW (training), HELEN (training), AFW and IBUG.

300-W challenge The 300-W face alignment challenge [8, 10] utilises a private dataset of testing images to perform evaluations. The dataset includes 600 "in-the-wild" testing images and is described as being drawn from the same distribution as the IBUG dataset. The results are reported in Figure 4. We believe this shows that face alignment is still very challenging when the images in the testing set are totally unseen by all participants. In Figure 4, we see that the recently pro-

Method	*mean* $\pm$ *std*	*median*	*mad*	*max*	*AUC*	*Failure rate (%)*
ACR	$\mathbf{0.0267 \pm 0.0092}$	**0.0248**	**0.0045**	0.0841	**0.60**	1.3
CFSS [15]	0.0283 ± 0.0079	0.0270	0.0046	**0.0688**	0.58	**0.4**
PO-CR [21]	0.0386 ± 0.0790	0.0279	0.0046	0.8041	0.56	2.2
ERT [22]	0.0353 ± 0.0147	0.0318	0.0060	0.1238	0.48	4.0
Intraface [23, 24]	0.0666 ± 0.1071	0.0314	0.0050	0.6062	0.46	13.4
Chehra [25]	0.0761 ± 0.1185	0.0284	0.0080	0.7344	0.44	23.7
Initialisation	0.1749 ± 0.1098	0.1449	0.0593	0.7273	0.01	94.2

Table 2: Various statistical measures, area under the curve (AUC) and percentage failure rate for the 49-point CED curve given in Figure 6 for LFPW testset. Failure rate is the $\%$ of images with error > 0.06.

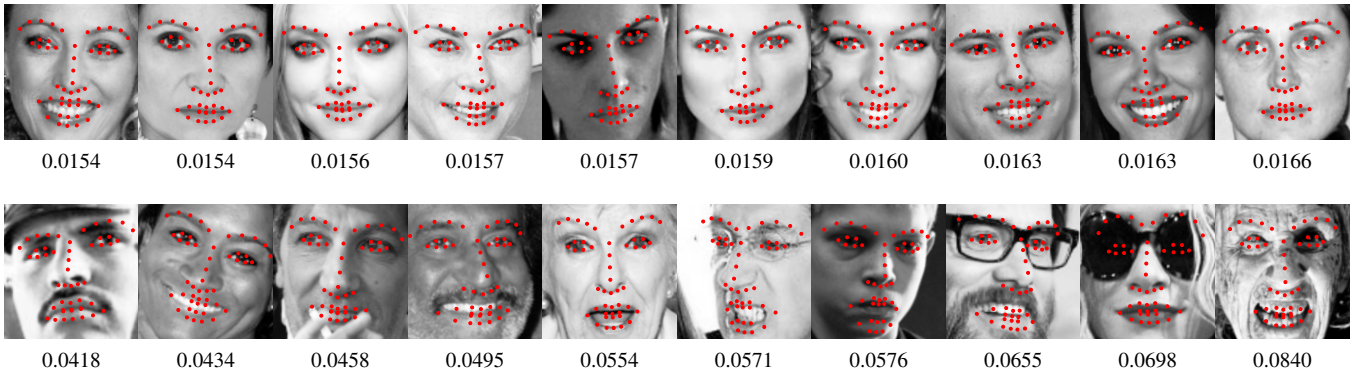

Fig. 5: 10 *best* (top), and 10 *worst* (bottom) fitting results of ACR on LFPW testset.

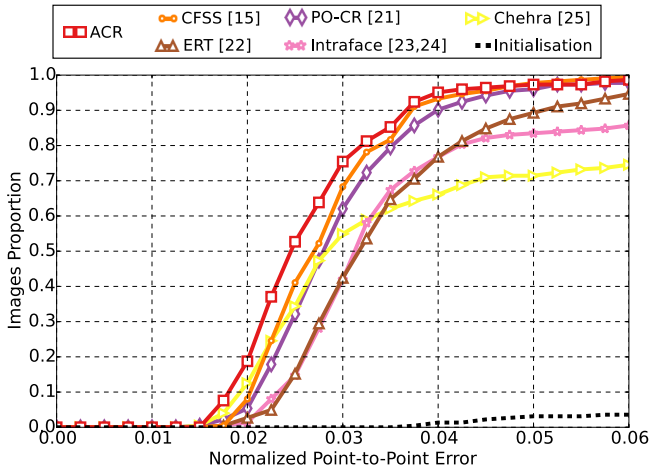

Fig. 6: Normalised error for the testing LFPW dataset based on 49 points.

posed CFSS method is currently the best performing method for 68-points. However, for the 49-points, ACR is the most accurate technique and slightly outperforms (300W 1), which is a much more complex deep learning method provided by industry. Table 1 reinforces the results of Figure 4 by showing that ACR is highly accurate for the 49-points and slightly less robust than the method of [19] over all images.

LFPW Testset The LFPW testset [11] consists of 224 in-the-wild images captured from the web. Figure 6 shows the accuracy of each method in the form of a Cumulative Error Distribution (CED) curve. Table 2 reports some statistical measures (mean, standard deviation, median, median absolute deviation, max), the area under the curve (AUC) and the failure rate of all methods based on Fig. 6. As explained in the main paper, we found, by visual inspection, that 0.06 is the maximum error that corresponds to adequate fitting results. Thus we plot the CED curves and compute the failure rate up to this error limit. Note that ACR is more accurate than all the other methods by a large margin. Especially in the band of low errors, it achieves an improvement of even about 10%. ACR is also slightly less robust than CFSS. Another interesting observation is the very high maximum errors for all the cascaded regression methods (PO-CR, Chehra, Intraface) that indicate that in case of a fitting failure, the final shape is completely scrambled.

Figure 10 reports the mean and standard deviation of the error per landmark point for all the methods. The numbering and colouring of each landmark point is linked with the mean shape of Figure 9. Once again, note that we only take into consideration the fittings with final error smaller than 0.06. ACR is very accurate on all facial parts. On the contrary, all the cascaded-regression based techniques (PO-CR, Intraface,

Method	*mean* $\pm$ *std*	*median*	*mad*	*max*	*AUC*	*Failure rate (%)*
ACR	$\mathbf{0.0262 \pm 0.0104}$	**0.0240**	0.0050	**0.0968**	**0.61**	**1.2**
CFSS [15]	0.0288 ± 0.0318	0.0244	**0.0048**	0.5644	0.60	1.5
PO-CR [21]	0.0299 ± 0.0287	0.0260	0.0051	0.5199	0.58	0.6
ERT [22]	0.0323 ± 0.0236	0.0280	0.0055	0.3732	0.54	1.8
Intraface [23, 24]	0.0666 ± 0.1094	0.0336	0.0060	0.7718	0.45	11.5
Chehra [25]	0.0391 ± 0.0507	0.0251	0.0054	0.4853	0.55	9.4
Initialisation	0.1757 ± 0.1050	0.1475	0.0603	0.5656	0.02	90.9

Table 3: Various statistical measures, area under the curve (AUC) and percentage failure rate for the 49-point CED curve given in Figure 8 for HELEN testset. Failure rate is the % of images with error > 0.06.

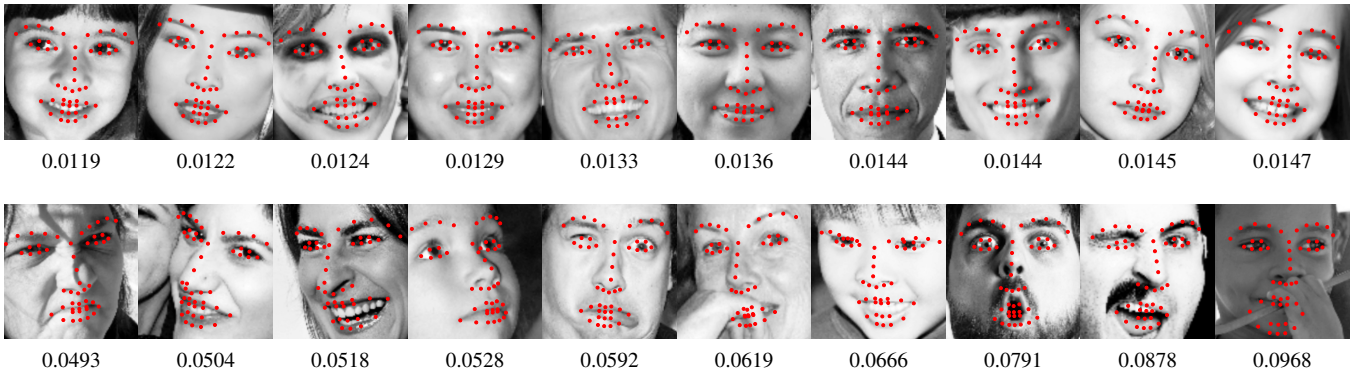

Fig. 7: 10 *best* (top), and 10 *worst* (bottom) fitting results of ACR on HELEN testset.

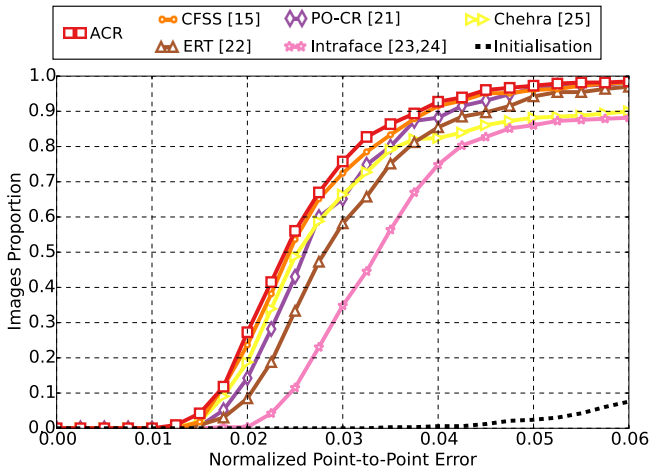

Fig. 8: Normalised error for the testing HELEN dataset based on 49 points.

Chehra) heavily fail on the internal mouth points and are not equally accurate on the eyebrows and eyes. Finally, Fig. 5 shows the 10 best and 10 worst fitting results achieved by ACR. As it can be observed, even the worst results have not heavily failed.

HELEN Testset The HELEN testset [12] consists of 330 in-the-wild images which exhibit larger difficulty than the LFPW ones. Figure 8 shows the accuracy of each method in the form of a Cumulative Error Distribution (CED) curve. Table 3 reports some statistical measures (mean, standard deviation, median, median absolute deviation, max), the area under the curve (AUC) and the failure rate of all methods based on Fig. 6. In this case, ACR is more accurate and more robust than all the other methods, since it achieves the best AUC as well as the lowest failure rate.

Figure 11 reports the mean and standard deviation of the error per landmark point for all the methods. Similar to the LFPW case, the numbering and colouring of each landmark point is linked with the mean shape of Figure 9. The results are again similar and indicate that ACR is more accurate on all facial parts, especially on the mouth region. Finally, Fig. 7 shows the 10 best and 10 worst fitting results achieved by ACR.

3. REFERENCES

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

[2] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp. 1150–1157.

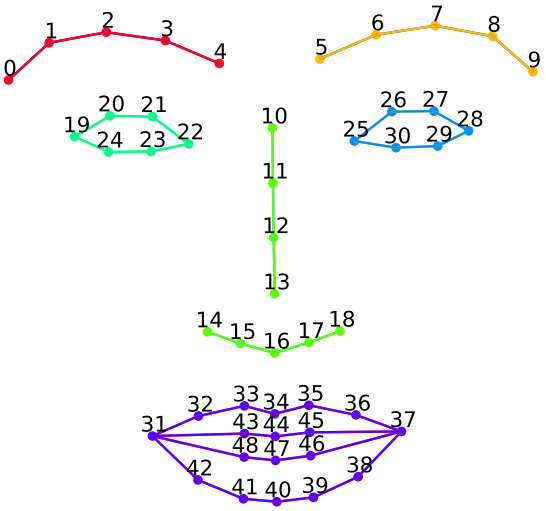

Fig. 9: The numbering and grouping of the landmarks in the 49-points configuration. The colouring and numbering of this figure is to be linked with Figures 10 and 11.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[4] G. Papandreou and P. Maragos, "Adaptive and constrained algorithms for inverse compositional active appearance model fitting," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.

[5] G. Tzimiropoulos and M. Pantic, "Optimization problems for fast aam fitting in-the-wild," in *Proc. of IEEE Int'l Conf. on Computer Vision (ICCV)*, 2013, pp. 593–600.

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

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

[8] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR'W)*, 2013.

[9] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "A semi-automatic methodology for facial landmark annotation," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR'W)*, 2013.

[10] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image and Vision Computing*, 2016.

[11] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[12] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," *Lecture Notes of Computer Science*, vol. 7574, pp. 679–692, 2012.

[13] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2879–2886.

[14] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1685–1692.

[15] S. Zhu, C. Li, C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4998–5006.

[16] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *Proc. of IEEE Int'l Conf. on Computer Vision (ICCV)*, 2013, pp. 1513–1520.

[17] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms," in *Proc. of the Int'l Conf. on Multimedia*. ACM, 2010, pp. 1469–1472.

[18] 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 *Proc. of the ACM Int'l Conf. on Multimedia*, Orlando, Florida, USA, November 2014, pp. 679–682, ACM.

[19] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. of IEEE Int'l Conf. on Computer Vision Workshops (ICCV'W)*, 2013, pp. 386–391.

[20] J. Yan, Z. Lei, D. Yi, and S. Z. Li, "Learn to combine multiple hypotheses for accurate face alignment," in *Proc. of IEEE Int'l Conf. on Computer Vision Workshops (ICCV'W)*, 2013, pp. 392–396.

[21] G. Tzimiropoulos, "Project-out cascaded regression with an application to face alignment," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3659–3667.

[22] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1867–1874.

[23] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 532–539.

[24] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. F. Cohn, "Intraface," in *Automatic Face and Gesture Recognition*, 2015.

[25] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Incremental face alignment in the wild," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1859–1866.

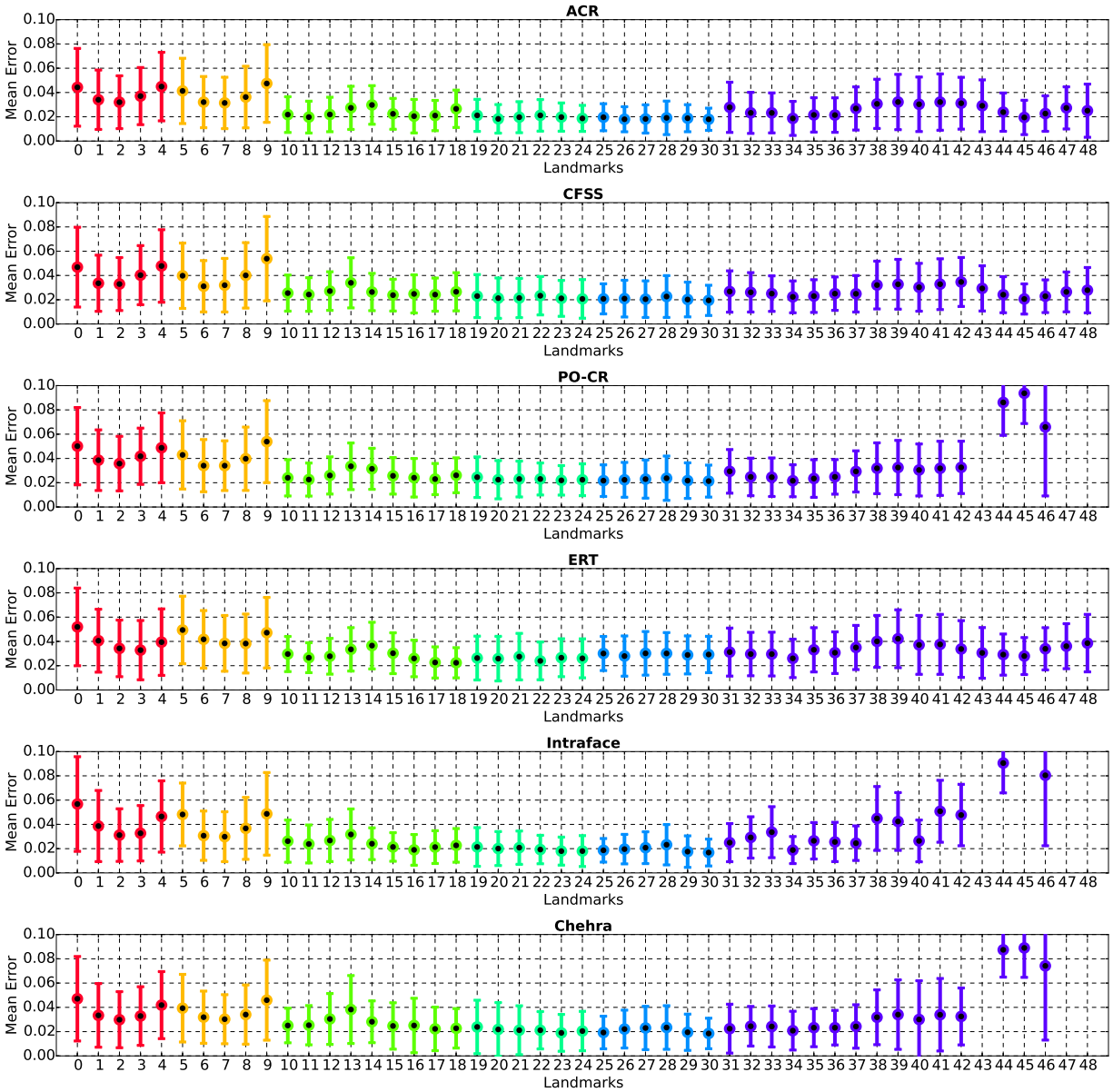

Fig. 10: Mean and standard deviation of the normalised error per landmark point for all the methods on HELEN testset. The colouring and numbering of the landmarks is linked with the mean shape of Figure 9.

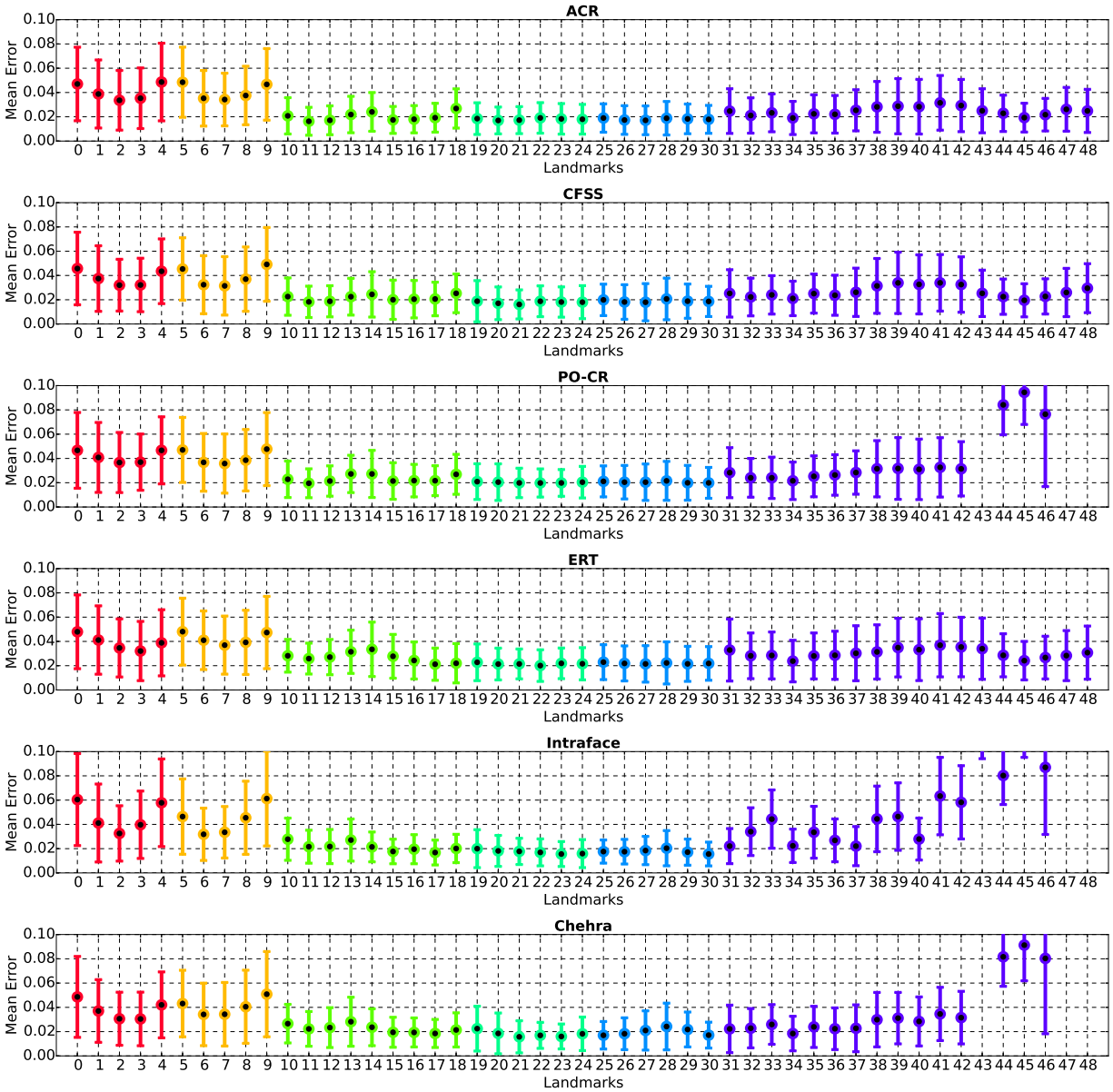

Fig. 11: Mean and standard deviation of the normalised error per landmark point for all the methods on HELEN testset. The colouring and numbering of the landmarks is linked with the mean shape of Figure 9.