Synergy between face alignment and tracking via Discriminative Global Consensus Optimization

Muhammad Haris Khan[⋆†] John McDonagh[⋆] Georgios Tzimiropoulos[⋆]
[⋆]Computer Vision Laboratory, University of Nottingham, Nottingham, UK
[†]Electrical Engineering Department, COMSATS Lahore Campus, Pakistan
{Muhammad.Khan3,yorgos.tzimiropoulos}@nottingham.ac.uk

Abstract

An open question in facial landmark localization in video is whether one should perform tracking or tracking-by-detection (i.e. face alignment). Tracking produces fittings of high accuracy but is prone to drifting. Tracking-by-detection is drift-free but results in low accuracy fittings.

To provide a solution to this problem, we describe the very first, to the best of our knowledge, synergistic approach between detection (face alignment) and tracking which completely eliminates drifting from face tracking, and does not merely perform tracking-by-detection. Our first main contribution is to show that one can achieve this synergy between detection and tracking using a principled optimization framework based on the theory of Global Variable Consensus Optimization using ADMM; Our second contribution is to show how the proposed analytic framework can be integrated within state-of-the-art discriminative methods for face alignment and tracking based on cascaded regression and deeply learned features. Overall, we call our method Discriminative Global Consensus Model (DGCM). Our third contribution is to show that DGCM achieves large performance improvement over the currently best performing face tracking methods on the most challenging category of the 300-VW dataset.

1. Introduction

Face alignment is the problem of localizing a set of landmarks on human faces in still images. Face tracking is the problem of localizing the facial landmarks for all frames of a given video. Most face tracking methods are extensions of face alignment methods; the main difference is that, in face alignment, initialization is performed from the bounding box of a face detector, whereas in face tracking, from the shape of the previously tracked frame. Because changes in facial pose and expression from one frame to the next one are typically small, in tracking, initialization is much closer to the "correct" solution, allowing the tracker to track large changes in pose and expression over the whole video. Such changes are typically difficult to accurately detect with a face alignment method that merely uses a bounding box initialization which lacks any information regarding pose, expression and identity. Nonetheless, tracking comes along with an important drawback: drifting. Errors in the tracked shapes can accumulate over time which in turn results in poorer and poorer initialization, and eventually can make the tracker lose tracking. Note that the most common way to deal with drifting is to perform face alignment per each video frame separately: this is known as tracking-by-detection; however, as mentioned earlier, this is sub-optimal because it completely discards shape information from previously tracked frame(s). In this paper, we propose the very

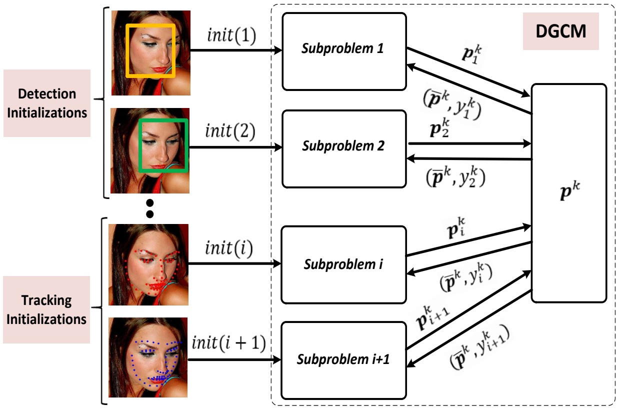

Figure 1. Overview of DGCM: our method performs landmark localization in video via a synergy between face alignment (i.e. detection) and tracking. This synergy is achieved by solving a Global Consensus Optimization problem using ADMM. We assume that different detection and tracking initializations define different optimization sub-problems. At each iteration of DGCM, the independent shape updates for each sub-problem are coupled through the ADMM updates. Notably, both the detector and tracker used in DGCM are trained in a discriminative manner.

first synergistic approach between tracking and detection that completely eliminates drifting from tracking, and does not merely perform tracking by detection.

To improve landmark localization in video, prior work has considered a few fairly orthogonal directions like improving the fitting method used [5, 29, 17, 9, 25, 33, 24], multi-view models [30, 32], exploiting temporal coherency between frames [15], incremental learning [21, 16, 1], tracking-by-detection [28, 6], re-fitting [18] and using multiple initializations [31, 16]. Notably, in most (if not all) of these methods, drifting is handled by just re-initializing the tracker. To the best of our knowledge, there is no prior method proposing a synergistic way to combine tracking and detection in a principled manner. We show that by doing so, significant performance improvement in terms of fitting accuracy can be obtained.

The main idea behind our work is simple: perform face alignment when tracking drifts or, from the opposite perspective, correct detection using the solution from the previously tracked frame. To achieve this synergy, we describe a principled optimization framework for deformable model fitting based on the theory of Global Variable Consensus Optimization using the Alternating Direction Method of Multipliers (ADMM) [3]. Fig. 1 shows an overview of our method. In summary, our contributions are:

1. We propose the very first synergistic approach between tracking and detection that completely eliminates drifting from tracking, and does not merely perform tracking-by-detection.
2. To this end, we propose the very first method integrating the theory of Consensus Optimization using ADMM with the problem of deformable model fitting.
3. Although we derive our framework using analytic gradient descent, we also show how to further integrate the derived formulation into a fully discriminative one based on cascaded regression and deeply learned features. Because of that, we call our method Discriminative Global Consensus Model (DGCM).
4. We show that DGCM achieves significant performance improvement over the currently best performing face tracking methods on the most challenging category of the 300-VW dataset [22].

2. Related work

To improve facial landmark localization in video, a number of orthogonal directions have been proposed in literature. We group them in 5 categories as follows.

The first one is to simply improve the accuracy and the robustness of the face alignment method that is adopted for tracking. To this end, methods based on cascaded regression have recently emerged as the state-of-the-art, see for example [5, 29, 17, 9, 25, 33, 24, 8, 27]. The second line of work is to make use of temporal coherency to create more adaptive and sophisticated fitting algorithms like, for example, the multi-view cascaded regression approaches of [30, 32], the spatio-temporal recurrent encoder-decoder network of [15] as well as the incremental learning approaches of [21, 16, 1]. Note that all methods from the first two categories are purely tracking methods and deal with drifting by typically relying on a SVM to detect when tracking goes off and then re-initialize the tracking procedure.

The third category is to perform joint re-fitting of all shapes for the whole video (i.e. offline) as in [18]. Such methods are post-processing methods, can be applied only when the whole video has been tracked, are typically very slow, and have been shown to mainly correct very crude errors. A fourth approach is to perform tracking-by-detection, see for example [28, 6]. Notably, we also make use of tracking-by-detection to eliminate drifting but synergistically with a tracker to exploit shape information from the previously tracked frames.

A fifth approach for improving landmark localization is to obtain multiple fittings using different initializations, and then combine the fitted results, as in [31, 16]. Note that both [31, 16] have only been shown how to combine the fitted shapes obtained by applying the *same face alignment* method (i.e. the same detection method), and not how to achieve synergy between detection and tracking as proposed in our work. Also, these methods have not been shown to correct drifting in face tracking. More importantly, both [31, 16] are actually post-processing methods, in which the multiple fittings are obtained independently and then are somehow combined. Hence, both [31, 16] are sub-optimal.

In this work, we propose a sixth orthogonal improvement for facial landmark localization in video: synergy between face alignment and tracking. To this end, we introduce a novel optimization framework that integrates the theory of Consensus Optimization using ADMM with the problem of discriminative fitting of facial deformable models.

3. Discriminative Global Consensus Model

In this section, we describe the proposed synergistic approach for face alignment and tracking. To do so, in section 3.1, we will firstly introduce a generative facial deformable model, which when fitted to a facial image can be used to localize the facial landmarks. Then, in section 3.2, we describe and discuss two optimization problems for fitting our facial deformable model, one for face alignment (i.e. detection) and one for face tracking. These problems will be the sub-problems used in our Consensus Optimization framework for deformable model fitting introduced in section 3.3. In the same section, we also present the solution to the proposed Consensus Optimization problem using an analytic gradient descent formulation. Finally, in section 3.4, we show how to further integrate the derived analytic gradient descent formulation into a fully discriminative one based on cascaded regression and deeply learned features. Overall, we call our method Discriminative Global Consensus Model (DGCM).

3.1. Generative facial deformable model

In our formulation, we used a generative facial deformable model that is based on parametric shape and appearance models. Perhaps, the most notable example of such models is the Active Appearance Model [7, 11]. We chose to use the most recent parts-based formulation of [26] which has been shown to largely outperform the holistic-based approach. Our deformable model is built in a fully supervised manner, from a set of training facial images $\mathbf{I}_i$ each of which is annotated with u fiducial points defining the facial shape, a vector $\in \mathcal{R}^{2u \times 1}$.

Shape model. All training facial shapes are firstly normalized using Procrustes Analysis. Then, the shape model is obtained by applying PCA on all training normalized shapes. The model is defined in terms of a mean shape $\mathbf{s}_0$ and n shape eigenvectors $\mathbf{s}_i$ which form the columns of matrix $\mathbf{S} \in \mathcal{R}^{2u \times n}$. Finally, to model similarity transforms, $\mathbf{S}$ is appended with 4 additional bases [11]. Using this model, a shape can be generated from:

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

where $\mathbf{p} \in \mathcal{R}^{n \times 1}$ is the vector of the shape parameters.

Appearance model. All training facial images are firstly warped to a reference frame so that similarity transformations are removed. Then, around each landmark a descriptor is extracted and all descriptors are stacked in a vector $\in \mathcal{R}^{N \times 1}$ which defines the part-based facial appearance. Finally, the appearance model is obtained by applying PCA on all training part-based facial appearances. The appearance model is defined in terms of the mean appearance $\mathbf{A}_0$ and m appearance eigenvectors $\mathbf{A}_i$ which form the columns of matrix $\mathbf{A} \in \mathcal{R}^{N \times m}$. Using this model, a part-based facial appearance can be generated from:

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

where $\mathbf{c} \in \mathcal{R}^{m \times 1}$ is the vector of the appearance parameters. As for the case of the descriptor used, most commonly, SIFT descriptors [10] are used; notably we show in this work that if one uses deeply learned features, large improvements in fitting accuracy can be obtained.

3.2. Optimization problems for face alignment and tracking

To localize the landmarks in a given facial image, one can fit the generative deformable model of the previous section by solving the following optimization problem:

$$\arg\min_{\mathbf{p},\mathbf{c}} f(\mathbf{p},\mathbf{c}) = \arg\min_{\mathbf{p},\mathbf{c}} ||\mathbf{I}(\mathbf{s}(\mathbf{p})) - \mathbf{A}(\mathbf{c})||^2, \tag{3}$$

where $\mathbf{I}(\mathbf{s}(\mathbf{p})) \in \mathcal{R}^{N \times 1}$ is obtained by concatenating the descriptors extracted from the landmarks of $\mathbf{s}(\mathbf{p})$.

As mentioned earlier, the difference between face alignment and tracking is in the initialization. In face detection, $\mathbf{p}_{init} = \mathbf{p}_{det} = (p_1, 0, p_3, p_4, 0, \dots, 0)^T$, where p_1, and p_2, p_3, are the scale and the translation parameters obtained from the bounding box of the face detector. In face tracking, we are looking to estimate the shape parameters for frame t, so $\mathbf{p}_{init} = \mathbf{p}_{t-1}$ where $\mathbf{p}_{t-1}$ is shape parameter vector as estimated at frame $t-1$. Note that $\mathbf{c}_{init} = \mathbf{0}$ for both cases.

3.3. Global Consensus optimization for deformable model fitting

Because the problem of Eq. (3) is non-convex, a locally optimal solution can be readily found using analytic gradient descent. Because the obtained solution depends on the initialization, we propose to view the different initializations in the optimization problem of Eq. (3) as separate optimization problems. In our case, we treat the problem of face alignment and the problem of face tracking as separate optimization problems because the initialization for these problems is different (although the error to be optimized has the same functional form). Notice that an arbitrary number of initializations can be used, with the hope that some of them will be sufficiently close to the "correct" solution. For example, one can apply some noise to the face detection bounding box, or to the fitted shape of the previously tracked frame to generate an arbitrary number of initializations; the trade-off in this case is a linear increase in computational complexity.

To make our point clearer, we follow the same notation as in [12], to emphasize the point that gradient descent optimization is used to solve the non-convex problem of Eq. (3) and hence using initialization $\mathbf{p}_{init}(i)$ can be interpreted as solving a different optimization problem $f_i(\mathbf{p}_i)$:

$$\arg\min_{\mathbf{p}_i,\mathbf{c}_i} f_i(\mathbf{p}_i,\mathbf{c}_i) = \mathrm{gd}\arg\min_{\substack{\mathbf{p}_i=\mathbf{p}_{init}(i)\\ \mathbf{c}_i=\mathbf{0}}} ||\mathbf{I}(\mathbf{s}(\mathbf{p}_i)) - \mathbf{A}(\mathbf{c}_i)||^2, \tag{4}$$

where $\mathrm{gd}\arg\min_{\mathbf{p}_i=\mathbf{p}_{init}(i)}$ means "perform gradient descent starting from $\mathbf{p}_{init}(i)$" ($\mathbf{c}_i$ is always initialized to $\mathbf{0}$). Notice that a different parameter vector $\mathbf{p}_i$ is used for the $i-$th problem f_i (the same holds also for $\mathbf{c}_i$).

Assume M different initializations $\mathbf{p}_{init}(i)$, $i = 1, \dots, M$, each of which defining sub-problem f_i as in Eq. (4). Then, we propose to perform joint optimization so that all sub-problems converge to the same solution. To this end, we can formulate the following Global Variable Consensus Optimization problem [3]:

$$\arg\min_{\substack{\mathbf{p}_1,\dots,\mathbf{p}_M\\ \mathbf{c}_1,\dots,\mathbf{c}_M}} \sum_{i=1}^{M} f_i(\mathbf{p}_i,\mathbf{c}_i), \text{ s.t. } \mathbf{p}_i = \mathbf{p},, i = 1,\dots,M. \tag{5}$$

Note that there is no need to impose constraints on $\mathbf{c}_i$. The reason for this is that if the global constraints for $\mathbf{p}_i$ of Eq. (5) are satisfied, then necessarily $\mathbf{c}_i = \mathbf{c}, , i = 1, \ldots, M$. Actually, as we show below, $\mathbf{c}_i = , i = 1, \ldots, M$ can be completely eliminated from the optimization problem of Eq. (5). The Consensus Optimization problem of Eq. (5) can be solved using the Alternating Direction Method of Multipliers (ADMM). Following [3], the ADMM solution for the $i-$th sub-problem at iteration $k+1$ is:

$$\begin{aligned} \mathbf{p}_i^{k+1}, \mathbf{c}_i^{k+1} &:= \arg\min_{\mathbf{p}_i, \mathbf{c}_i} \{ f_i(\mathbf{p}_i, \mathbf{c}_i) \\ &+ (\mathbf{y}_i^k)^T (\mathbf{p}_i - \bar{\mathbf{p}}^k) + \frac{\rho}{2} ||\mathbf{p}_i - \bar{\mathbf{p}}^k||^2 \} \\ \mathbf{y}_i^{k+1} &:= \mathbf{y}_i^k + \rho(\mathbf{p}_i^{k+1} - \bar{\mathbf{p}}_i^{k+1}), \end{aligned} \tag{6}$$

where ρ is the penalty parameter, $\bar{\mathbf{p}}^k = \sum_{i=1}^M \mathbf{p}_i$ is the average of the shape parameters at iteration k, and y_i^{k+1} are the auxiliary (dual) variables driving the shape parameters of each sub-problem into consensus. In the next paragraphs, we describe our modifications to Eq. (6) and how to solve the optimization problem of Eq. (6).

Shape-based penalty. ADMM theory [3] allows one to replace the quadratic term $(\rho/2)||\mathbf{r}||^2$ with $(1/2)\mathbf{r}^T\mathbf{Q}\mathbf{r}$, where $\mathbf{Q}$ is a symmetric positive definite matrix. In our case, we chose $\mathbf{Q} = \Lambda^{-1}$, where Λ is the diagonal matrix containing the eigenvalues of the shape model computed using PCA. This is equivalent of using the Mahalanobis distance in the shape parameter space. It is more intuitive than using the standard form of the penalty and we found it necessary for the algorithm to behave well.

Inexact minimization. The formulation of Eq. (6) requires finding the optimal value for $\mathbf{p}_i, \mathbf{c}_i$ at each iteration k, which is costly because at each iteration, for each sub-problem, an iterative Gauss-Newton minimization would be required. However, ADMM converges even when the minimization is not carried out exactly [4], allowing us to perform a single Gauss-Newton update per iteration.

Gauss-Newton update. To perform inexact minimization for each sub-problem (given in the first row of Eq. (6)), we follow prior work on deformable model fitting using analytic gradient descent, in particular, Gauss-Newton optimization [11, 2, 26]. Given the current estimate $\mathbf{p}_i^k$ and $\mathbf{c}_i^k$ for iteration k, we firstly perform a first-order Taylor approximation. Then, the ADMM updates are given by:

$$\begin{aligned} \Delta\mathbf{p}_i, \Delta\mathbf{c}_i &:= \arg\min_{\Delta\mathbf{p}_i, \Delta\mathbf{c}_i} \{ ||\mathbf{I}(\mathbf{s}(\mathbf{p}_i^k)) + \mathbf{J}_i^k \Delta\mathbf{p}_i \\ &- \mathbf{A}_0 - \mathbf{A}\mathbf{c}_i^k - \mathbf{A}\Delta\mathbf{c}_i||^2 \\ &+ (\mathbf{y}_i^k)^T (\mathbf{p}_i^k + \Delta\mathbf{p}_i - \bar{\mathbf{p}}^k) \\ &+ \frac{\Lambda^{-1}}{2} ||\mathbf{p}_i^k + \Delta\mathbf{p}_i - \bar{\mathbf{p}}^k||^2 \} \\ \mathbf{p}_i^{k+1}, \mathbf{c}_i^{k+1} &:= \mathbf{p}_i^k + \Delta\mathbf{p}_i, \mathbf{c}_i^k + \Delta\mathbf{c}_i \\ \mathbf{y}_i^{k+1} &:= \mathbf{y}_i^k + \Lambda^{-1}(\mathbf{p}_i^{k+1} - \bar{\mathbf{p}}_i^{k+1}), \end{aligned} \tag{7}$$

where $\mathbf{J}_i^k \in \mathcal{R}^{N \times n}$ is the image Jacobian with respect to the shape parameters $\mathbf{p}_i^k$. Notice that in the above minimization problem, $\Delta\mathbf{c}_i$ does not appear in the second and third terms. Hence, we can apply the same Gauss-Newton approach of [26] which by-passes the calculation of $\Delta\mathbf{c}$, and at each iteration solves only for $\Delta\mathbf{p}$ (for more details on why this is possible see [26]). By doing so, the ADMM updates are:

$$\begin{aligned} \Delta\mathbf{p}_i &:= \arg\min_{\Delta\mathbf{p}_i} \{ ||\mathbf{I}(\mathbf{s}(\mathbf{p}_i^k)) + \mathbf{J}_i^k \Delta\mathbf{p}_i - \mathbf{A}_0||^2_{\mathbf{P}} \\ &+ (\mathbf{y}_i^k)^T (\mathbf{p}_i^k + \Delta\mathbf{p}_i - \bar{\mathbf{p}}^k) \\ &+ \frac{\Lambda^{-1}}{2} ||\mathbf{p}_i^k + \Delta\mathbf{p}_i - \bar{\mathbf{p}}^k||^2 \} \\ \mathbf{p}_i^{k+1} &:= \mathbf{p}_i^k + \Delta\mathbf{p}_i \\ \mathbf{y}_i^{k+1} &:= \mathbf{y}_i^k + \Lambda^{-1}(\mathbf{p}_i^{k+1} - \bar{\mathbf{p}}_i^{k+1}), \end{aligned} \tag{8}$$

where $||\mathbf{x}||^2_{\mathbf{P}} = \mathbf{x}^T\mathbf{P}\mathbf{x}$ is the weighted ℓ_2-norm of a vector $\mathbf{x}$, $\mathbf{P} = \mathbf{E} - \mathbf{A}\mathbf{A}^T$ is a projection operator that projects-out the appearance variation, and $\mathbf{E}$ is the identity matrix. Notice that because we work in a subspace orthogonal to the appearance variation, $\Delta\mathbf{c}$ has been completely eliminated from the optimization problem. Finally, by solving the optimization problem of Eq. (8), we obtain the final ADMM updates rules for our Consensus Optimization problem:

$$\begin{aligned} \Delta\mathbf{p}_i &:= -(\mathbf{H}^k_{i,P})^{-1} \{ (\mathbf{J}^k_{i,P})^T (\mathbf{I}(\mathbf{s}(\mathbf{p}_i^k)) - \mathbf{A}_0) + \mathbf{y}_i^k \\ &+ \Lambda^{-1}(\mathbf{p}_i^k - \bar{\mathbf{p}}^k) \} \\ \mathbf{p}_i^{k+1} &:= \mathbf{p}_i^k + \Delta\mathbf{p}_i \\ \mathbf{y}_i^{k+1} &:= \mathbf{y}_i^k + \Lambda^{-1}(\mathbf{p}_i^{k+1} - \bar{\mathbf{p}}_i^{k+1}), \end{aligned} \tag{9}$$

where $\mathbf{J}^k_{i,P} = \mathbf{P}\mathbf{J}_i^k$ and $\mathbf{H}^k_{i,P} = \mathbf{J}^T_{i,P}\mathbf{J}_{i,P} + \Lambda^{-1}$ are the projected-out image-based Jacobian and Hessian for the i-th sub-problem at iteration k.

Overall algorithm. At each iteration k, and for the $i-$th sub-problem/initialization, the proposed algorithm firstly computes the image-based $\mathbf{J}^k_{i,P}$, $\mathbf{H}^k_{i,P}$ (and its inverse), and then uses the first and second rows of Eq. (9) to update the shape parameters for this sub-problem. This process is repeated for all sub-problems, and then the new average of the shape parameter vector $\mathbf{p}_i^{k+1}$ is obtained. Then, this is used to update the auxiliary variables for each sub-problem from the last row of Eq. (9).

3.4. Discriminative Global Consensus optimization

In the previous section, we derived the ADMM updates for solving the Consensus Optimization problem for deformable model fitting using analytic gradient descent. As it has been noted in a number of works in literature (e.g. [5, 29, 17, 9, 25, 33, 24]), fitting algorithms based on analytic gradient have a small basin of attraction and hence they can be trapped in local minima when initialization is far from the correct solution. Additionally, such algorithms are relatively slow because the image-based Jacobian, Hessian and its inverse need to be re-computed per iteration. To circumvent this problem, recent state-of-the-art face alignment methods have suggested learning "averaged" descent directions in a discriminative manner using the framework of cascaded regression [5, 29, 17, 9, 25, 33, 24]. In this section, we show how to incorporate this type of discriminative training into the proposed analytic formulation for Consensus Optimization introduced in the previous section.

While most works in Cascaded Regression estimate a direct mapping between image features and shape updates, the ADMM update for $\Delta\mathbf{p}_i$ in Eq. (9) does not allow this requiring the calculation of $\mathbf{H}^k_{i,P}$ at each iteration. Therefore, we will proceed based on PO-CR [25] which is the only method that explicitly calculates averaged Jacobians (and then Hessians) at each iteration from data.

We will firstly review PO-CR [25]: Assuming H training images $\mathbf{I}_i$, $i = 1, \dots, H$ with ground truth shape parameters $\mathbf{p}^*_i$, and K perturbed shapes for each image $\mathbf{p}^k_{i,j}$, $j = 1, \dots, K$ at level (iteration) k of the cascade, and denoting $\Delta\mathbf{p}_{i,j} = \mathbf{p}^*_i - \mathbf{p}^k_{i,j}$, PO-CR learns an averaged projected-out Jacobian $\widehat{\mathbf{J}}^k_P = \mathbf{P}\widehat{\mathbf{J}}^k$ at level (iteration) k of the cascade by solving the following optimization problem

$$\arg\min_{\widehat{\mathbf{J}}^k_P} \sum_{i=1}^{H} \sum_{j=1}^{K} ||\mathbf{I}(\mathbf{s}(\mathbf{p}^k_{i,j})) + \mathbf{J}^k \Delta\mathbf{p}_{i,j} - \mathbf{A}_0||^2_{\mathbf{P}}. \quad (10)$$

After computing $\widehat{\mathbf{J}}^k_P$, PO-CR further computes the averaged Hessian $\widehat{\mathbf{H}}^k_P = (\widehat{\mathbf{J}}^k_P)^T\widehat{\mathbf{J}}^k_P$ and its inverse. The averaged descent directions are $\mathbf{R}^k = (\widehat{\mathbf{H}}^k_P)^{-1}(\widehat{\mathbf{J}}^k_P)^T$ and for each training sample the shape update is given by $\Delta\mathbf{p}_{i,j} = \mathbf{R}^k(\mathbf{I}(\mathbf{s}(\mathbf{p}^k_{i,j}) - \mathbf{A}_0))$. We note that the problem of Eq. (10) is derived from the problem of deformable model fitting of Eq. (3) that has no constraints. We also note that training a model for face alignment or tracking depends on how the perturbed shapes are produced. For face alignment, the perturbed shapes at the first iteration are produced in order to capture the statistics of the face detector used whereas for tracking the shape changes between consecutive frames.

We now describe the training of the detection and tracking models within our Consensus Optimization framework. For each training image $\mathbf{I}_i$, we assume K_d perturbed shapes for training the detection model and K_{tr} perturbed shapes for training the tracking model. In our case, and in analogy to the unconstrained optimization problem of Eq. (10), to estimate the averaged Jacobian for detection (or tracking) at iteration k, the optimization problem of Eq. (8) will be used. In particular, $\widehat{\mathbf{J}}^{d,k}_P$ for the detection model is estimated (using the K_d perturbed shapes for each training image) from

$$\begin{aligned}\widehat{\mathbf{J}}^{d,k}_P &= \arg\min_{\widehat{\mathbf{J}}^k_P} \sum_{i=1}^{H} \sum_{j=1}^{K_d} \{||\mathbf{I}(\mathbf{s}(\mathbf{p}^k_{i,j})) + \mathbf{J}^k \Delta\mathbf{p}_{i,j} - \mathbf{A}_0||^2_{\mathbf{P}} \\ &+ (\mathbf{y}^k_i)^T(\mathbf{p}^k_{i,j} + \Delta\mathbf{p}_{i,j} - \bar{\mathbf{p}}^k_i) \\ &+ \frac{\Lambda^{-1}}{2}||\mathbf{p}^k_{i,j} + \Delta\mathbf{p}_{i,j} - \bar{\mathbf{p}}^k_i||^2\}, \quad (11)\end{aligned}$$

and similarly using the tracking initializations K_{tr} one can compute $\widehat{\mathbf{J}}^{tr,k}_P$ for the tracking model. It is evident from Eq. (11) that $\widehat{\mathbf{J}}^k_P$ does not depend on the second and third terms and hence the optimization problem reduces to the one of Eq. (10). Hence, at each iteration, we estimate $\widehat{\mathbf{J}}^{d,k}_P$ and $\widehat{\mathbf{J}}^{tr,k}_P$ separately. Following this, we then estimate the averaged ADMM Hessian $\widehat{\mathbf{H}}^{d,k}_P = (\widehat{\mathbf{J}}^{d,k}_P)^T\widehat{\mathbf{J}}^{d,k}_P + \Lambda^{-1}$ and its inverse, and similarly we do the same for $\widehat{\mathbf{H}}^{tr,k}_P$. Hence, at each iteration, the averaged descent directions for the detection and tracking models are estimated independently, and the shape updates are also calculated in a similar fashion. This results in a very intuitive algorithm: during training, the averaged descent directions for detection and tracking are estimated dis-jointly giving rise to independent models which during testing are then forced to agree through the coupling of the ADMM updates of Eq. (9) [1]. More specifically, testing is performed as follows: For each sub-problem/initialization, at iteration k, we use the detection model $\{\widehat{\mathbf{J}}^{d,k}_P, (\widehat{\mathbf{H}}^{d,k}_P)^{-1}\}$ or the tracking model $\{\widehat{\mathbf{J}}^{tr,k}_P, (\widehat{\mathbf{H}}^{tr,k}_P)^{-1}\}$ to update $\Delta\mathbf{p}_i$ from the first row of Eq. (9), depending on whether that particular sub-problem/initialization corresponds to face alignment or tracking. Then, the average shape parameter vectors $\mathbf{p}^{k+1}_i$ over all initializations are computed and the auxiliary variables for each sub-problem $\mathbf{y}^{k+1}_i$ are also updated through the last row of Eq. (9).

The complexity of the proposed method at test time depends linearly on the number of initializations used. For M initializations, the complexity per iteration is $O(MnN)$, where $O(nN)$ is the complexity of the original PO-CR.

4. Experiments

We primarily evaluate the performance of the proposed DGCM on the 300-VW test set [22]. This is by far the most challenging and large-scale face tracking dataset to date containing 121,278 frames. Note that this is the only publicly available large scale face tracking dataset. Additionally, in section 4.4, we also report the results of another

[1] Note that one could also apply the ADMM updates of Eq. (9) during training as follows: rather than updating each sample independently, one could group all $K_d + K_{tr}$ perturbed shapes for each training image, and then at each iteration compute the ADMM updates of Eq. (9). However, there is no real benefit to enforce this kind of consensus as during training all perturbations for all training images will converge anyway.

experiment on the 300-W test set [19] and our cats dataset, illustrating how DGCM can enhance the performance of human and animal face alignment for the case of poor face detection initializations (note that this is not a tracking experiment).

4.1. Performance evaluation

300-VW consists of 3 categories of increasing difficulty: A (62,135 frames), B (32,805 frames) and C (26,338 frames). C is by far the most challenging. We report results on category C, and for the results of categories B and A, see supplementary material. Following prior work, results are reported for the 49 inner points.

DGCM consists of two models for landmark localization, one for face alignment (i.e. detection) and one for face tracking, both trained using the discriminative ADMM framework described in section 3.4. Although our framework allows for an arbitrary number of initializations for detection and tracking, we merely used one for detection and one for tracking [2]. This means that, compared to [25], the complexity of the DGCM model used in our evaluations is just doubled.

We trained our face alignment model on all training data from the 300-W competition [20] using also the statistics of the face detector of [13] that was used to initialize our face alignment model for all frames of the video sequences. We trained our tracking model on the same data using also the statistics capturing the shape changes between consecutive frames from the 300-VW training set. Similarly to [1] and [21], a simple SVM was used to detect the cases that the tracker gets lost. We used two types of features for training DGCM: (1) SIFT features [10] as in [25], and (2) `conv-3` features using VGG-16 [23] [3]. We report interesting self-evaluation results in section 4.2 and comparison with the state-of-the-art in section 4.3.

4.2. Self-evaluation

In this section, we compare the performance of DGCM with two related methods of interest. The first method performs tracking-by-detection alone (i.e. face alignment for each frame) using the same face detector for initialization (as the one used in DGCM). The second method performs face tracking alone using the shape initialization from the previously tracked frame. Both the face alignment and tracking methods are trained using PO-CR on the same training data as the one that DGCM was trained on.

Table 1. Comparison between Tracking-`conv` and Detector-`conv` on category C. The table shows the percentage of frames for which one method provides more accurate fitting results over the other.

Method	# frames	% frames	AUC@0.08
Tracking	9682	36.76	55.94
Detection	16656	63.24	58.58

Fig. 3 shows the obtained results from all videos of category C. The results for each video separately, for `conv` features only, are also shown in Fig. 4 and in terms of the Area Under the Curve @ 0.08 error in Table 2. Our main conclusions from these experiments are:

- Using `conv` features results in large performance improvement over SIFT.
- Fig. 3 shows that DGCM consistently results in performance improvement for both SIFT and `conv` features. The biggest improvement is observed for the case of Category C and `conv` features, where the improvement over the detector is about $6-7\%$ almost across the whole spectrum of the pt-pt-error. This performance improvement becomes more notable for specific videos (74, 86, 87, 96, 99, 113) as Fig. 4 and Table 2 show.
- This performance improvement is obtained even though the tracker seems to perform much worse compared to the detector. As shown in Table 1, for Category C and `conv` features, we found that the tracker (alone) outperforms the detector (alone) only for the 36% of the total number of frames. By improving our tracker, further boosting in DGCM performance is expected.
- As expected, the tracker typically produces more accurate fittings compared to the detector when there are gradual large variations in pose and expression. Such examples are included in the first 4 images (from left to right) of Fig. 2. Also, when the tracker's initialization is poor, the detector produces better fitting results. Examples of such cases are displayed in the last 4 images of Fig. 2.
- Finally, with very few exceptions, DGCM works better or at least comparably with the second best performing method. Such examples are displayed in the last row of Fig. 2. One exception to this is video #95, where we observe that DGCM is worse than both the detector and tracker. We attribute this to the following reason: for this video, the detector and tracker seem to work equally poorly. The reason is that the tracker is re-initialized for about 40% of the frames literally "converting" the tracker to detector. However, in our ADMM, the tracker is always initialised from the result of the previous frame which because of the difficulty of the video is poor.

4.3. Comparison with state-of-the-art

In this section, we compare DGCM and the two best performing methods of the 300-VW competition [32, 28],

[2] We also tried more initializations but we observed little difference in performance. The reason for this is that most of the cases that the fittings were no good were cases in which both the detector and the tracker were unable to fit the particular image even when initialisation is perfect, for example some very low quality videos.

[3] The `conv-3` feature maps of the VGG network were up-sampled to have the same resolution as the input and then features were extracted at the landmarks locations. Note that we used the provided pre-trained network.

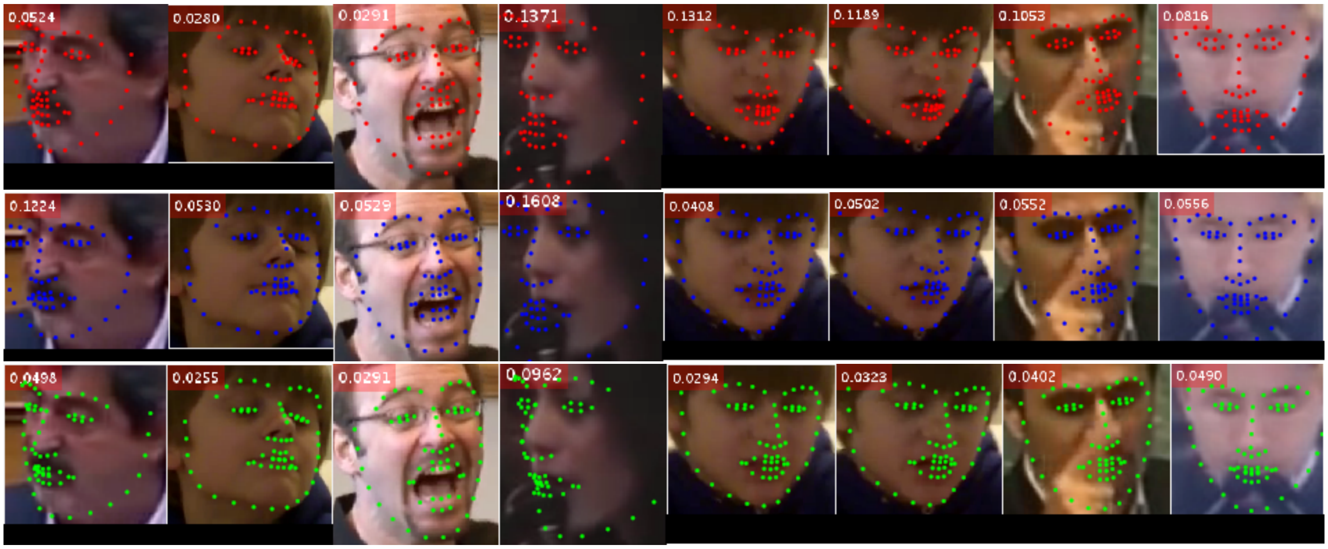

Figure 2. First row (red): Tracking fittings. Second row (blue): Detection fittings. Third row (green): DGCM fittings. The normalized error for each fitting is displayed on the top-left corner. First 4 images (from left to right): Tracker works better than the detector. Last 4 images: Detector works better than the tracker. In all 8 cases, DGCM works better or comparably with the second best performing method.

Table 2. Area under curve @ 0.08 error for Tracking-conv, Detection-conv, and DGCM-conv for each individual video of category C. The numbers are percentages.

Meth./Vi.	#74	#75	#86	#87	#95	#96	#97	#98	#99	#100	#111	#112	#113	#114
Tracking	48.43	55.90	63.37	66.66	46.31	64.45	50.25	55.97	62.75	24.55	61.66	73.99	51.32	47.47
Detection	47.43	**58.35**	62.10	64.46	**46.96**	64.42	55.10	61.03	67.88	**36.98**	**63.80**	75.96	56.00	51.29
DGCM	**51.77**	58.02	**66.13**	**69.08**	44.48	**68.85**	**55.75**	**61.20**	**71.44**	35.01	61.38	**76.86**	**58.43**	**52.87**

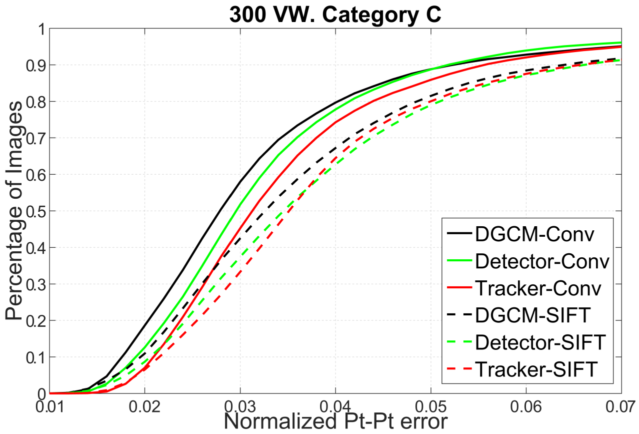

Figure 3. Comparison between DGCM with Detection (alone) and Tracking (alone) on category C of 300-VW.

the state-of-the-art face alignment method of [24], and the state-of-the-art tracker of [21]. Fig. 5 shows the obtained results on category C. We note that (a) DGCM based on conv features is by far the best performing method, and (b) DGCM based on SIFT features is consistently the second best method. Finally, the very competitive method of [21] proposes incremental learning to enhance tracking accuracy. Such an improvement is orthogonal to our framework, and is the focus of future work.

4.4. Detection experiment

In this section, we performed another experiment illustrating how DGCM can be used for enhancing detection (i.e. face alignment) performance for the case of poor initialization. The experiment was performed on two different datasets, namely (a) the publicly available 300-W test set [19], containing 600 images captured in indoor and outdoor environments, and (b) our cats dataset [4] consisting of 2000 images, out of which 200 are kept for testing and the rest for training. In particular, we used two noise levels for initializing a PO-CR detector based on conv features, and DGCM (using also PO-CR and conv features). In both cases, initializations were produced by scaling and translating the ground truth bounding boxes using a noise distribution, defined by σ_{noise}.

When the noise level is low, PO-CR for both human and cat faces produces good results even when only 1 initialization is used (green line in Fig. 6). However, when the noise level is high, the performance of PO-CR drops significantly even when multiple, 5 in these experiments, initializations are used and the median of the 5 fittings for each image is taken as the final fitting (dashed green line in Fig. 6). On the

[4] a subset of the Oxford-IIIT-Pet dataset [14].

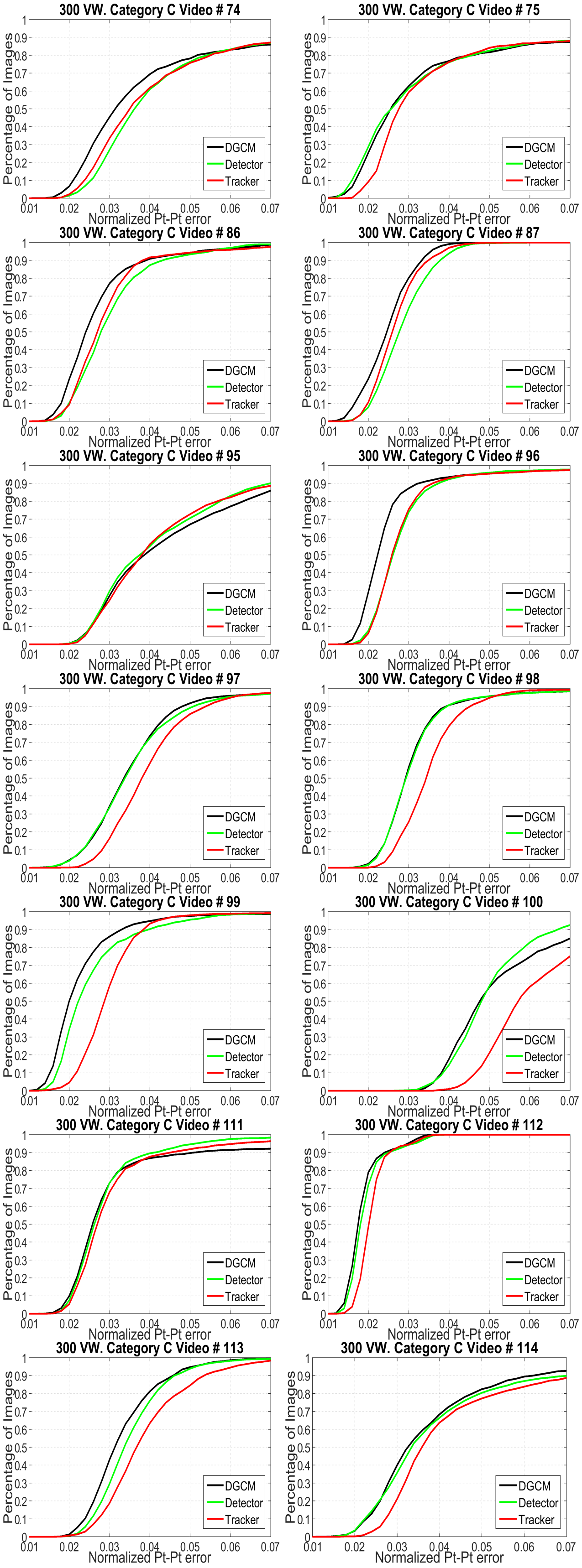

Figure 4. Comparison between DGCM-`conv` with Detection-`conv` (alone) and Tracking-`conv` (alone) on all 14 videos of category C.

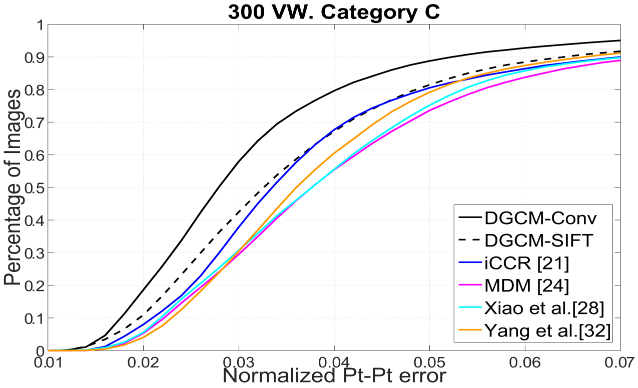

Figure 5. Comparison between DGCM and state-of-the art on category C of 300-VW.

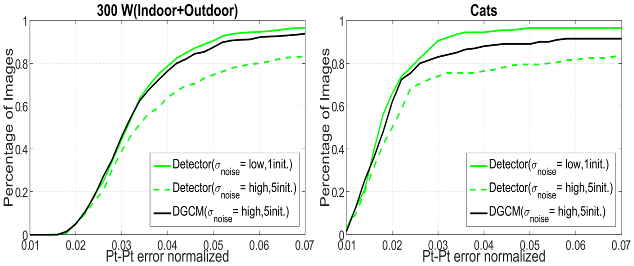

Figure 6. PO-CR detector vs DGCM for different noise levels of initialization on 300-W testset (left) and cats dataset (right). For large noise levels, DGCM shows little loss in performance.

contrary, when 5 initializations are used for DGCM, there is very little loss in performance, as the black line in Fig. 6 illustrates.

5. Conclusions

We proposed a method for achieving synergy between face alignment and tracking based on the principled framework of Global Variable Consensus Optimization using ADMM. We also showed how the proposed formulation can be integrated with state-of-the-art discriminative methods for face alignment and tracking based on cascaded regression and deeply learned features. Contrary to prior work in face tracking, our method is both drifting-free and, at the same time, able to exploit shape information from the previously tracked frame. Finally, we demonstrated that our method results in large performance improvement over the state-of-the-art on the 300-VW dataset.

6. Acknowledgements

This work was supported by the EPSRC project EP/M02153X/1 Facial Deformable Models of Animals. We are also grateful for the access to the University of Nottingham High Performance Computing Facility.

References

[1] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *CVPR*, 2014.

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

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

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

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

[6] G. G. Chrysos, E. Antonakos, P. Snape, A. Asthana, and S. Zafeiriou. A comprehensive performance evaluation of deformable face tracking" in-the-wild". *arXiv preprint arXiv:1603.06015*, 2016.

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

[8] H. Fan and E. Zhou. Approaching human level facial landmark localization by deep learning. *Image and Vision Computing*, 47:27–35, 2016.

[9] V. Kazemi and S. Josephine. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014.

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

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

[12] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):810–815, 2004.

[13] J. McDonagh and G. Tzimiropoulos. Joint face detection and alignment using a deformable hough transform model. In *ECCV-W*, 2016.

[14] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *CVPR*, 2012.

[15] X. Peng, R. S. Feris, X. Wang, and D. N. Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *ECCV*, 2016.

[16] X. Peng, S. Zhang, Y. Yang, and D. N. Metaxas. Piefa: Personalized incremental and ensemble face alignment. In *ICCV*, 2015.

[17] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, 2014.

[18] C. Sagonas, Y. Panagakis, S. Zafeiriou, and M. Pantic. Raps: Robust and efficient automatic construction of person-specific deformable models. In *CVPR*, 2014.

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

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

[21] E. Sánchez-Lozano, B. Martinez, G. Tzimiropoulos, and M. Valstar. Cascaded continuous regression for real-time incremental face tracking. In *ECCV*, 2016.

[22] J. Shen, S. Zafeiriou, G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *ICCV-W*, 2015.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *CVPR*, 2016.

[25] G. Tzimiropoulos. Project-out cascaded regression with an application to face alignment. In *CVPR*, 2015.

[26] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *CVPR*, 2014.

[27] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In *ECCV*, 2016.

[28] S. Xiao, S. Yan, and A. A. Kassim. Facial landmark detection via progressive initialization. In *ICCV-W*, 2015.

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

[30] X. Xiong and F. De la Torre. Global supervised descent method. In *CVPR*, 2015.

[31] J. Yan, Z. Lei, D. Yi, and S. Z. Li. Learn to combine multiple hypotheses for accurate face alignment. In *ICCV-W*, 2013.

[32] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, 2015.

[33] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, 2015.