Cascaded Continuous Regression for Real-Time Incremental Face Tracking

Enrique Sánchez-Lozano(✉), Brais Martinez, Georgios Tzimiropoulos, and Michel Valstar

Computer Vision Laboratory, University of Nottingham, Nottingham, UK
{psxes1,yorgos.tzimiropoulos,michel.valstar}@nottingham.ac.uk

Abstract. This paper introduces a novel real-time algorithm for facial landmark tracking. Compared to detection, tracking has both additional challenges and opportunities. Arguably the most important aspect in this domain is updating a tracker's models as tracking progresses, also known as incremental (face) tracking. While this should result in more accurate localisation, how to do this online and in real time without causing a tracker to drift is still an important open research question. We address this question in the cascaded regression framework, the state-of-the-art approach for facial landmark localisation. Because incremental learning for cascaded regression is costly, we propose a much more efficient yet equally accurate alternative using continuous regression. More specifically, we first propose cascaded continuous regression (CCR) and show its accuracy is equivalent to the Supervised Descent Method. We then derive the incremental learning updates for CCR (iCCR) and show that it is an order of magnitude faster than standard incremental learning for cascaded regression, bringing the time required for the update from seconds down to a fraction of a second, thus enabling real-time tracking. Finally, we evaluate iCCR and show the importance of incremental learning in achieving state-of-the-art performance. Code for our iCCR is available from http://www.cs.nott.ac.uk/~psxes1.

1 Introduction

The detection of a sparse set of facial landmarks in still images has been a widely-studied problem within the computer vision community. Interestingly, many face analysis methods either systematically rely on video sequences (e.g., facial expression recognition [1]) or can benefit from them (e.g., face recognition [2]). It is thus surprising that facial landmark tracking has received much less attention in comparison. Our focus in this paper is on one of the most important problems in model-specific tracking, namely that of updating the tracker using previously tracked frames, also known as incremental (face) tracking.

The standard approach to face tracking is to use a facial landmark detection algorithm initialised on the landmarks detected at the previous frame. This

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46484-8_39) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 645–661, 2016.
DOI: 10.1007/978-3-319-46484-8_39

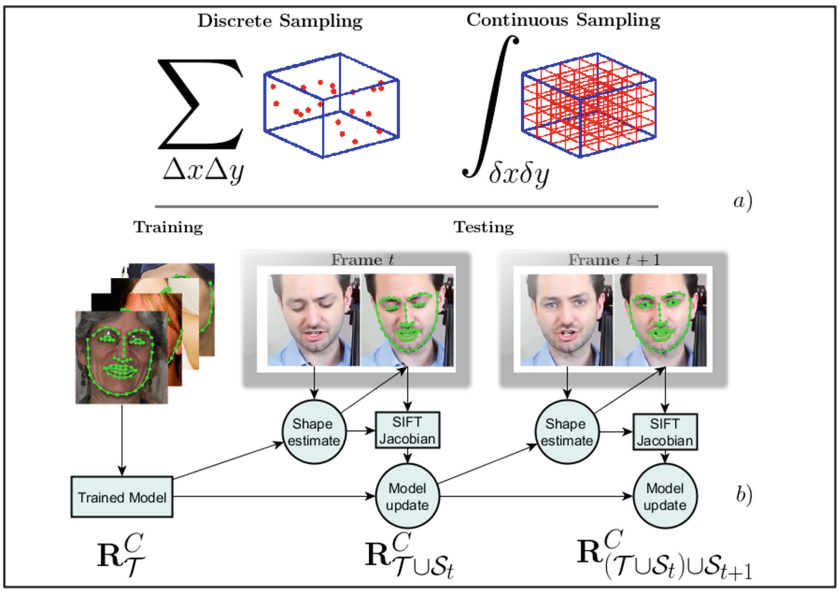

Fig. 1. Overview of our incremental cascaded continuous regression algorithm (iCCR). (a) shows how continuous regression uses all data in a point's neighbourhood, whereas sampled regression uses a finite subset. (b) shows how the originally model $\mathbf{R}_{\mathcal{T}}$ learned offline is updated with each new frame.

exploits the fact that the face shape varies smoothly in videos of sufficiently high framerates: If the previous landmarks were detected with acceptable accuracy, then the initial shape will be close enough for the algorithm to converge to a "good" local optimum for the current frame too. Hence, tracking algorithms are more likely to produce highly accurate fitting results than detection algorithms that are initialised by the face detector bounding box.

However, in this setting the tracker still employs a generic deformable model of the face built offline using a generic set of annotated facial images, which does not include the subject being tracked. It is well known that person-specific models are far more constrained and easier to fit than generic ones [3]. Hence one important problem in tracking is how to improve the generic model used to track the first few frames into an increasingly person-specific one as more frames are tracked.

This problem can be addressed with incremental learning, which allows for the smart adaptation of pre-trained generic appearance models. Incremental learning is a common resource for generic tracking, being used in some of the state-of-the-art trackers [4,5], and incremental learning for face tracking is by no means a new concept, please see Ross et al. [6] for early work on the topic. More recently, incremental learning within cascaded regression, the state-of-the-art approach for facial landmark localisation, was proposed by Xiong and De la

Torre [7] and independently by Asthana et al. [8]. However, in both [7,8] the model update is far from being sufficiently efficient to allow real-time tracking, with [8] mentioning that the model update requires 4.7 s per frame. Note that the actual tracking procedure (without the incremental update) is faster than 25 frames per second, clearly illustrating that the incremental update is the bottleneck impeding real-time tracking.

If the model update cannot be carried out in real time, then incremental learning might not be the best option for face tracking - once the real-time constraint is broken in practice one would be better off creating person-specific models in a post-processing step [9] (e.g., re-train the models once the whole video is tracked and then track again). That is to say, without the need and capacity for real-time processing, incremental learning is sub-optimal and of little use.

Our main contribution in this paper is to propose the first incremental learning framework for cascaded regression which allows real-time updating of the tracking model (Fig. 1). To do this, we build upon the concept of continuous regression [10] as opposed to standard sampling-based regression used in almost all prior work, including [7,8]. We note that while we tackle the facial landmark tracking problem, cascaded regression has also been applied to a wider range of problems such as pose estimation [11], model-free tracking [5] or object localisation [12], thus making our methodology of wider interest. We will release code for training and testing our algorithm for research purposes.

1.1 Contributions

Our main contributions are as follows:

- We propose a complete **new formulation for Continuous Regression**, of which the original continuous regression formulation [10] is a special case. Crucially, our method is now formulated by means of a **full covariance matrix capturing real statistics** of how faces vary between consecutive frames rather than on the shape model eigenvalues. This makes our method particularly suitable for the task of tracking, something the original formulation cannot deal with.
- We incorporate continuous regression in the Cascaded Regression framework (coined Cascaded Continuous Regression, or **CCR**) and demonstrate its performance is equivalent to sampling-based cascaded regression.
- We derive the **incremental learning for continuous regression**, and show that it **is an order of magnitude faster** than its standard incremental SDM counterpart.
- We evaluate the incremental Cascaded Continuous Regression (**iCCR**) on the 300VW data set [13] and show the importance of incremental learning in achieving state-of-the-art performance, especially for the case of very challenging tracking sequences.

1.2 Prior Work on Face Alignment

Facial landmark tracking methods have often been adaptations of facial landmark detection methods. For example, Active Appearance Models (AAM) [14,15], Constrained Local Models (CLM) [16] or the Supervised Descent Method (SDM) [17] were all presented as detection algorithms. It is thus natural to group facial landmark tracking algorithms in the same way as the detection algorithms, i.e. splitting them into discriminative and generative methods [8].

On the generative side, AAMs have often been used for tracking. Since the model fitting relies on gradient descent, it suffices to start the fitting from the last solution[1]. Tracking is particularly useful to AAMs since they are considered to have frequent local minima and a small basin of attraction, making it important that the initial shape is close to the ground truth. AAMs have further been regarded as very reliable for person specific tracking, but not for generic tracking (i.e., tracking faces unseen during training) [3]. Recently [19] showed however that an improved optimisation procedure and the use of in-the-wild images for training can lead to well-behaving person independent AAM. Eliminating the piecewise-affine representation and adopting a part-based model led to the Gauss-Newton Deformable Part Model (GN-DPM) [20], which is the AAM state of the art.

Historically, discriminative methods relied on the training of local classifier-based models of appearance, with the local responses being then constrained by a shape model [16,21,22]. These algorithms can be grouped into what is called the Constrained Local Models (CLM) framework [16]. However, the appearance of discriminative regression-based models quickly transformed the state-of-the-art for face alignment. Discriminative regressors were initially used within the CLM framework substituting classifiers, showing improved performance [23,24]. However, the most important contributions came with the adoption of cascaded regression [11] and direct estimation of the full face shape rather than first obtaining local estimates [17,25]. Successive works have further shown the impressive efficiency [26,27] and reliable performance [28,29] of face alignment algorithms using cascaded regression. However, how to best exploit discriminative cascaded regression for tracking and, in particular, how to best integrate incremental learning, is still an open problem.

2 Linear Regression Models for Face Alignment

In this section we revise the preliminary concepts over which we build our method. In particular, we describe the methods most closely related to ours, to wit the incremental supervised descent method [8] and the continuous regressor [10], and motivate our work by highlighting their limitations.

[1] Further "implementation tricks" can be found in [18], which provides a very detailed account of how to optimise an AAM tracker.

2.1 Linear Regression

A face image is represented by $\mathbf{I}$, and a face shape is a $n \times 2$ matrix describing the location of the n landmarks considered. A shape is parametrised through a Point Distribution Model (PDM) [30]. In a PDM, a shape $\mathbf{s}$ is parametrised in terms of $\mathbf{p} = [\mathbf{q}, \mathbf{c}] \in \mathbb{R}^m$, where $\mathbf{q} \in \mathbb{R}^4$ represents the rigid parameters and $\mathbf{c}$ represents the flexible shape parameters, so that $\mathbf{s} = t_{\mathbf{q}}(\mathbf{s}_0 + \mathbf{B}_s\mathbf{c})$, where t is a Procrustes transformation parametrised by $\mathbf{q}$. $\mathbf{B}_s \in \mathbb{R}^{2n \times m}$ and $\mathbf{s}_0 \in \mathbb{R}^{2n}$ are learned during training and represent the linear subspace of flexible shape variations. We will sometimes use an abuse of notation by referring treating shape $\mathbf{s}$ also as function $\mathbf{s}(\mathbf{p})$. We also define $\mathbf{x} = f(\mathbf{I}, \mathbf{p}) \in \mathbb{R}^d$ as the feature vector representing shape $\mathbf{s}(\mathbf{p})$. An asterisk represents the ground truth, e.g., $\mathbf{s}_j^*$ is the ground truth shape for image j.

Given a test image $\mathbf{I}$, and a current shape prediction $\mathbf{s}(\mathbf{p}^* + \delta\mathbf{p})$, the goal of Linear Regression for face alignment is to find a mapping matrix $\mathbf{R} \in \mathbb{R}^{m \times d}$ able to infer $\delta\mathbf{p}$, the increment taking directly to the ground truth, from $f(\mathbf{I}, \mathbf{p}^* + \delta\mathbf{p})$. By using M training images, and K perturbations per image, the mapping matrix $\mathbf{R}$ is typically learned by minimising the following expression w.r.t. $\mathbf{R}$:

$$\sum_{j=1}^{M}\sum_{k=1}^{K} \|\delta\mathbf{p}_{j,k} - \mathbf{R}f(\mathbf{I}_j, \mathbf{p}_j^* + \delta\mathbf{p}_{j,k})\|_2^2, \tag{1}$$

where the bias term is implicitly included by appending a 1 to the feature vector[2].

In order to produce K perturbed shapes $\mathbf{s}(\mathbf{p}_j^* + \delta\mathbf{p}_{j,k})$ per image, it suffices to draw the perturbations from an adequate distribution, ideally capturing the statistics of the perturbations encountered at test time. For example, during detection, the distribution should capture the statistics of the errors made by using the face detection bounding box to provide a shape estimation.

The minimisation in Eq. 1 has a closed-form solution. Given M images and K perturbed shapes per training image, let $\mathbf{X} \in \mathbb{R}^{d \times KM}$ and $\mathbf{Y} \in \mathbb{R}^{2n \times KM}$ represent the matrices containing in its columns the input feature vectors and the target output $\delta\mathbf{p}_{j,k}$ respectively. Then, the optimal regressor $\mathbf{R}$ can be computed as:

$$\mathbf{R} = \mathbf{Y}\mathbf{X}^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1}. \tag{2}$$

Given a test shape $\mathbf{s}(\mathbf{p})$, the predicted shape is computed as $\mathbf{s}(\mathbf{p} - \mathbf{R}f(\mathbf{I}, \mathbf{p}))$.

2.2 Continuous Regression

Continuous Regression (CR) [10] is an alternative solution to the problem of linear regression for face alignment. The main idea of Continuous Regression is to treat $\delta\mathbf{p}$ as a continuous variable and to use *all samples* within some finite

[2] It is in practice beneficial to include a regularisation term, although we omit it for simplicity. All of the derivations in this paper hold however for ridge regression.

limits, instead of sampling a handful of perturbations per image. That is to say, the problem is formulated in terms of finite integrals as:

$$\min_{\mathbf{R}} \sum_{j=1}^{M} \int_{-r_1\sqrt{\lambda_1}}^{r_1\sqrt{\lambda_1}} \cdots \int_{-r_{|\mathbf{c}|}\sqrt{\lambda_{|\mathbf{c}|}}}^{r_{|\mathbf{c}|}\sqrt{\lambda_{|\mathbf{c}|}}} \|\delta\mathbf{c} - \mathbf{R}f(\mathbf{I}_j, \mathbf{c}_j^* + \delta\mathbf{c})\|_2^2 d\delta\mathbf{c}, \tag{3}$$

where λ_i is the eigenvalue associated to the i-th flexible parameter of the PDM, $|\mathbf{c}|$ represent the number of flexible parameters, and r_i is a parameter determining the number of standard deviations considered in the integral.

Unfortunately, this formulation does not have a closed-form solution. However, it is possible to solve it approximately in a very efficient manner by using a first order Taylor expansion of the loss function. Following the derivations in [10], we denote $\mathbf{J}_j^*$ as the Jacobian of the image features with respect to the shape parameters evaluated at the ground truth $\mathbf{p}_j^*$, which can be calculated simply as $\mathbf{J}_j^* = \frac{\partial f(\mathbf{I}_j,\mathbf{s})}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial \mathbf{p}}|_{(\mathbf{p}=\mathbf{p}_j^*)}$. A solution to Eq. 3 can then be written as:

$$\mathbf{R}(\mathbf{r}) = \boldsymbol{\Sigma}(\mathbf{r})(\sum_{j=1}^{M} \mathbf{J}_j^{*T}) \left(\sum_{j=1}^{M} \mathbf{x}_j^* \mathbf{x}_j^{*T} + \mathbf{J}_j^* \boldsymbol{\Sigma}(\mathbf{r})\ \mathbf{J}_j^{*T} \right)^{-1}, \tag{4}$$

where $\boldsymbol{\Sigma}(\mathbf{r})$ is a diagonal matrix whose i-th entries are defined as $\frac{1}{3} r_i^2 \lambda_i$. CR formulated in this manner has the following practical limitations:

1. It does not account for correlations within the perturbations. This corresponds to using a fixed (not data-driven) diagonal covariance to model the space of shape perturbations, which is a harmful oversimplification.
2. Because of 1, it is not possible to incorporate CR within the popular cascaded regression framework in an effective manner.
3. Derivatives are computed over image pixels, so more robust features, e.g., HOG or SIFT, are not used.
4. The CR can only account for the flexible parameters, as the integral limits are defined in terms of the eigenvalues of the PDM's PCA space.

In Sect. 3.1 we will solve all of these shortcomings, showing that it is possible to formulate the cascaded continuous regression and that, in fact, its performance is equivalent to the SDM.

2.3 Supervised Descent Method

The main limitation of using a single Linear Regressor to predict the ground truth shape is that the training needs to account for too much intra-class variation. That is, it is hard for a single regressor to be simultaneously accurate and robust. To solve this, [31] successfully adapted the cascaded regression of framework of Dollár et al. [11] to face alignment. However, the most widely-used form of face alignment is the SDM [17], which is a cascaded linear regression algorithm.

At **test time**, the SDM takes an input $\mathbf{s}(\mathbf{p}^{(0)})$, and then for a fixed number of iterations computes $\mathbf{x}^{(i)} = f(\mathbf{I}, \mathbf{p}^{(i)})$ and $\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} - \mathbf{R}^{(i)}\mathbf{x}^{(i)}$. The key idea is to use a different regressor $\mathbf{R}^{(i)}$ for each iteration. The input to the **training** algorithm is a set of images $\mathbf{I}_j$ and corresponding perturbed shapes $\mathbf{p}_{j,k}^{(0)}$. The training set i is defined as $\mathbf{X}^{(i)} = \{\mathbf{x}_{j,k}^{(i)}\}_{j=1:M,k=1:K}$, with $\mathbf{x}_{j,k}^{(i)} = f(\mathbf{I}_j, \mathbf{p}_{j,k}^{(i)})$, and $\mathbf{Y}^{(i)} = \{\mathbf{y}_{j,k}^{(i)}\}_{j=1:M,k=1:K}$, with $\mathbf{y}_{j,k}^{(i)} = \mathbf{p}_{j,k}^{(i)} - \mathbf{p}_j^*$. Then regressor i is computed using Eq. 2 on training set i, and a new training set $\{\mathbf{X}^{(i+1)}, \mathbf{Y}^{(i+1)}\}$ is created using the shape parameters $\mathbf{p}_{j,k}^{(i+1)} = \mathbf{p}_{j,k}^{(i)} - \mathbf{R}^{(i)}\mathbf{x}_{j,k}^{(i)}$.

2.4 Incremental Learning for SDM

Incremental versions of the SDM have been proposed by both Xiong and De la Torre [7] and Asthana et al. [8]. The latter proposed the *parallel SDM*, a modification of the original SDM which facilitates the incremental update of the regressors. More specifically, they proposed to alter the SDM training procedure by modelling $\{\mathbf{p}_{j,k}^{(i)} - \mathbf{R}^{(i)}\mathbf{x}_{j,k}^{(i)}\}_{j,k}$ as a Normal distribution $\mathcal{N}(\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)})$, allowing training shape parameters to be sampled for the next level of the cascade as:

$$\mathbf{p}_{j,k}^{(i+1)} \sim \mathcal{N}(\mathbf{p}_j^* + \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)}) \tag{5}$$

Once the parallel SDM is defined, its incremental extension is immediately found. Without loss of generality, we assume that the regressors are updated in an on-line manner, i.e., the information added is extracted from the fitting of the last frame. We thus define $\mathcal{S} = \{\mathbf{I}_j, \{\mathbf{p}_{j,k}\}_{k=1}^K\}$, arrange the matrices $\mathbf{X}_\mathcal{S}$ and $\mathbf{Y}_\mathcal{S}$ accordingly, and define the shorthand $\mathbf{V}_\mathcal{T} = \left(\mathbf{X}_\mathcal{T}\mathbf{X}_\mathcal{T}^T\right)^{-1}$, leading to the following update rules [8]:

$$\mathbf{R}_{\mathcal{T}\cup\mathcal{S}} = \mathbf{R}_\mathcal{T} - \mathbf{R}_\mathcal{T}\mathbf{Q} + \mathbf{Y}_\mathcal{S}\mathbf{X}_\mathcal{S}^T\mathbf{V}_{\mathcal{T}\cup\mathcal{S}} \tag{6}$$

$$\mathbf{Q} = \mathbf{X}_\mathcal{S}\mathbf{U}\mathbf{X}_\mathcal{S}^T\mathbf{V}_\mathcal{T} \tag{7}$$

$$\mathbf{U} = \left(\mathbb{I}_K + \mathbf{X}_\mathcal{S}^T\mathbf{V}_\mathcal{T}\mathbf{X}_\mathcal{S}\right)^{-1} \tag{8}$$

$$\mathbf{V}_{\mathcal{T}\cup\mathcal{S}} = \mathbf{V}_\mathcal{T} - \mathbf{V}_\mathcal{T}\mathbf{Q} \tag{9}$$

where $\mathbb{I}_K$ is the K-dimensional identity matrix.

The cost for these incremental updates is dominated by the multiplication $\mathbf{V}_\mathcal{T}\mathbf{Q}$, where both matrices have dimensionality $d \times d$, which has a computational complexity of $\mathcal{O}(d^3)$. Since d is high-dimensional (> 1000), the cost of updating the models becomes prohibitive for real-time performance. Once real time is abandoned, offline techniques that do not analyse every frame in a sequential manner can be used for fitting, e.g., [9]. We provide a full analysis of the computational complexity in Sect. 4.

3 Incremental Cascaded Continuous Regression (iCCR)

In this section we describe the proposed Incremental Cascaded Continuous Regression, which to the best of our knowledge is the first cascaded regression tracker with **real-time incremental learning** capabilities. To do so, we

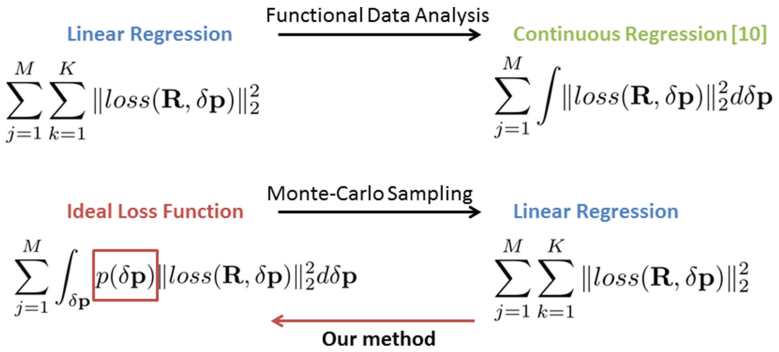

Fig. 2. Main difference between original Continuous Regression [10] and our method.

first extend the Continuous Regression framework into a fully fledged cascaded regression algorithm capable of performance on par with the SDM (see Sects. 3.1 and 3.2). Then, we derive the incremental learning update rules within our Cascaded Continuous Regression formulation (see Sect. 3.3). We will show in Sect. 4 that our newly-derived formulas have complexity of one order of magnitude less than previous incremental update formulations.

3.1 Continuous Regression Revisited

We first modify the original formulation of Continuous Regression. In particular, we add a "data term", which is tasked with encoding the probability of a certain perturbed shape, allowing for the modelling of correlations in the shape dimensions. Plainly speaking, the previous formulation assumed an i.i.d. uniform sampling distribution. We instead propose using a data-driven full covariance distribution, resulting in regressors that model the test-time scenario much better. In particular, we can see the loss function to be optimised as:

$$\arg\min_{\mathbf{R}} \sum_{j=1}^{M} \int_{\delta\mathbf{p}} p(\delta\mathbf{p}) \|\delta\mathbf{p} - \mathbf{R} f(\mathbf{I}_j, \mathbf{p}_j^* + \delta\mathbf{p})\|_2^2 d\delta\mathbf{p}. \tag{10}$$

It is interesting to note that this equation appears in [17], where the SDM equations are interpreted as a MCMC sampling-based approximation of this equation. Contrariwise, the Continuous Regression proposes to use a different approximation based on a first-order Taylor approximation of the *ideal loss function* defined in Eq. 10. However, the Continuous Regression proposed in [10] extends the Functional Data Analysis [32] framework to the imaging domain, without considering any possible data correlation. Instead, the "data term" in Eq. 10 (which defines how the data is sampled in the MCMC approach), will serve to correlate the different dimensions in the Continuous Regression. That is to say, the "data term" does not play the role of how samples are taken, but

rather helps to find an analytical solution in which dimensions can be correlated. These differences are illustrated in Fig. 2.

The first-order approximation of the feature vector is given by:

$$f(\mathbf{I}_j, \mathbf{p}_j^* + \delta\mathbf{p}) \approx f(\mathbf{I}_j, \mathbf{p}_j^*) + \mathbf{J}_j^* \delta\mathbf{p} \tag{11}$$

where $\mathbf{J}_j^*$ is the Jacobian of the feature representation of image $\mathbf{I}_j$ at $\mathbf{p}_j^*$. While [10] used a pixel-based representation, the Jacobian under an arbitrary representation can be computed empirically as:

$$\mathbf{J}_x = \frac{\partial f(\mathbf{I}, \mathbf{s})}{\partial x} \approx \frac{f(\mathbf{I}, [\mathbf{s}_x + \Delta x, \mathbf{s}_y]) - f(\mathbf{I}, [\mathbf{s}_x - \Delta x, \mathbf{s}_y])}{2\Delta x} \tag{12}$$

where $\mathbf{s}_x$ are the x coordinates of shape $\mathbf{s}$, and $\mathbf{s}_x + \Delta x$ indicates that Δx is added to each element of $\mathbf{s}_x$ (in practice, Δx is the smallest possible, 1 pixel). $\mathbf{J}_y$ can be computed similarly. Then $\mathbf{J}_j^* = [\mathbf{J}_x, \mathbf{J}_y] \frac{\partial \mathbf{s}}{\partial \mathbf{p}_j^*}$. Equation 10 has a closed form solution as[3]:

$$\begin{aligned} \mathbf{R}_{\mathcal{T}} = & \left(\sum_{j=1}^{M} \boldsymbol{\mu} \mathbf{x}_j^{*T} + (\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T) \mathbf{J}_j^{*T} \right) \cdot \\ & \left(\sum_{j=1}^{M} \mathbf{x}_j^* \mathbf{x}_j^{*T} + 2\mathbf{x}_j^* \boldsymbol{\mu}^T \mathbf{J}_j^{*T} + \mathbf{J}_j^* (\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T) \mathbf{J}_j^{*T} \right)^{-1} \end{aligned} \tag{13}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the data term, $p(\delta\mathbf{p})$.

Finally, we can see that Eq. 13 can be expressed in a more compact form. Let us first define the following shorthand notation: $\mathbf{A} = [\boldsymbol{\mu}, \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T]$, $\mathbf{B} = \begin{pmatrix} 1 & \boldsymbol{\mu}^T \\ \boldsymbol{\mu} & \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \end{pmatrix}$, $\mathbf{D}_j^* = [\mathbf{x}_j^*, \mathbf{J}_j^*]$ and $\bar{\mathbf{D}}_{\mathcal{T}}^* = [\mathbf{D}_1^*, \ldots, \mathbf{D}_M^*]$. Then:

$$\mathbf{R}_{\mathcal{T}} = \mathbf{A} \left(\sum_{j=1}^{M} \mathbf{D}_j^* \right)^T \left(\bar{\mathbf{D}}_{\mathcal{T}}^* \hat{\mathbf{B}} (\bar{\mathbf{D}}_{\mathcal{T}}^*)^T \right)^{-1} \tag{14}$$

where $\hat{\mathbf{B}} = \mathbf{B} \otimes \mathbb{I}_M$. Through this arrangement, the parallels with the sampling-based regression formula are clear (see Eq. 2).

It is interesting that, while the standard linear regression formulation needs to sample perturbed shapes from a distribution, the Continuous Regression training formulation only needs to extract the features and the Jacobians on the ground-truth locations. This means that once these features are obtained, re-training a new model under a different distribution takes seconds, as it only requires the computation of Eq. 14.

[3] A full mathematical derivation is included in the Supplementary Material.

3.2 Cascaded Continuous Regression (CCR)

Now that we have introduced a new formulation with the Continuous Regression capable of incorporating a data term, it is straightforward to extend the CR into the cascade regression formulation: we take the distribution in Eq. 5 as the *data term* in Eq. 10.

One might argue that due to the first-order Taylor approximation required to solve Eq. 10, CCR might not work as well as the SDM. One of the main experimental contributions of this paper is to show that in reality this is not the case: in fact CCR and SDM have equivalent performance (see Sect. 5). This is important because, contrary to previous works on Cascaded Regression, incremental learning within CCR allows for real time performance.

3.3 Incremental Learning Update Rules for CCR

Once frame j is tracked, the incremental learning step updates the existing training set $\mathcal{T}$ with $\mathbf{S} = \{\mathbf{I}_j, \hat{\mathbf{p}}_j\}$, where $\hat{\mathbf{p}}_j$ denotes the predicted shape parameters for frame j. Note that in this case $\mathbf{S}$ consists of only one example compared to K examples in the incremental SDM case.

The update process consists of computing matrix $\mathbf{D}_j$, which stores the feature vector and its Jacobian at $\hat{\mathbf{p}}_j$ and then, using the shorthand notation $\mathbf{V}_{\mathcal{T}} = \bar{\mathbf{D}}^*_{\mathcal{T}}\hat{\mathbf{B}}(\bar{\mathbf{D}}^*_{\mathcal{T}})^T$, updating continuous regressor as:

$$\mathbf{R}_{\mathcal{T}\cup\mathcal{S}} = \mathbf{A}\left(\sum_{j=1}^{M}\mathbf{D}^*_j + \mathbf{D}^*_{\mathcal{S}}\right)^T (\mathbf{V}_{\mathcal{T}\cup\mathcal{S}})^{-1} \tag{15}$$

In order to avoid the expensive re-computation of $\mathbf{V}_{\mathcal{T}}^{-1}$, it suffices to update its value using the Woodbury identity [32]:

$$\mathbf{V}_{\mathcal{T}\cup\mathcal{S}}{}^{-1} = \mathbf{V}_{\mathcal{T}}{}^{-1} - \mathbf{V}_{\mathcal{T}}{}^{-1}\mathbf{D}^*_{\mathcal{S}}\left(\mathbf{B}^{-1} + \mathbf{D}^{*T}_{\mathcal{S}}\mathbf{V}_{\mathcal{T}}{}^{-1}\mathbf{D}^*_{\mathcal{S}}\right)^{-1}\mathbf{D}^{*T}_{\mathcal{S}}\mathbf{V}_{\mathcal{T}}{}^{-1} \tag{16}$$

Note that $\mathbf{D}^*_{\mathcal{S}} \in \mathbb{R}^{d\times(m+1)}$, where m accounts for the number of shape parameters. We can see that computing Eq. 16 requires computing first $\mathbf{D}^{*T}_{\mathcal{S}}\mathbf{V}_{\mathcal{T}}{}^{-1}$, which is $\mathcal{O}(md^2)$. This is a central result of this paper, and reflects a property previously unknown. We will examine in Sect. 4 its practical implications in terms of real-time capabilities.

4 Computational Complexity

In this section we first detail the computational complexity of the proposed iCCR, and show that it is real-time capable. Then, we compare its cost with that of incremental SDM, showing that our update rules are an order of magnitude faster.

iCCR update complexity: Let us note the computational cost of the feature extraction as $\mathcal{O}(q)$. The update only requires the computation of the feature vector at the ground truth, and in two adjacent locations to compute the Jacobian, thus resulting in $\mathcal{O}(3q)$ complexity. Interestingly, this is independent from the number of cascade levels.

Then, the update equation (Eq. 16), has a complexity dominated by the operation ${\mathbf{D}_{\mathcal{S}}}^T {\mathbf{V}_{\mathcal{I}}^{\mathcal{C}}}^{-1}$, which has a cost of $\mathcal{O}(d^2 m)$. It is interesting to note that $\mathbf{B}^{-1} + {\mathbf{D}_{\mathcal{S}}}^T {\mathbf{V}_{\mathcal{I}}^{\mathcal{C}}}^{-1} \mathbf{D}_{\mathcal{S}}$ is a matrix of size $(m+1) \times (m+1)$ and thus its inversion is extremely efficient. The detailed cost of the incremental update is:

$$\mathcal{O}(3md^2) + \mathcal{O}(3m^2 d) + \mathcal{O}(m^3). \tag{17}$$

Incremental SDM update complexity: Incremental learning for SDM requires sampling at each level of the cascade. The cost per cascade level is $\mathcal{O}(qK)$, where K denotes the number of samples. Thus, for L cascade levels the total cost of sampling is $\mathcal{O}(LKq)$. The cost of the incremental update equations (Eqs. (6–9)), is in this case dominated by the multiplication $\mathbf{V}_{\mathcal{T}}\mathbf{Q}$, which is $\mathcal{O}(d^3)$. The detailed computational cost is:

$$\mathcal{O}(d^3) + \mathcal{O}((3m+k)d^2) + \mathcal{O}((2K^2 + mk)d) + \mathcal{O}(K^3). \tag{18}$$

Detailed comparison and timing: One advantage of iCCR comes from the much lower number of feature computations, being as low as 3 vs. the LK computations required for incremental SDM. However, the main difference is the $\mathcal{O}(d^3)$ complexity of the regressor update equation for the incremental SDM compared to $\mathcal{O}(d^2 m)$ for the iCCR. In our case, $d = 2000$, while $m = 24$. The feature dimensionality results from performing PCA over the feature space, which is a standard procedure for SDM. Note that if we avoided the use of PCA, the complexity comparison would be even more in our favour. A detailed summary of the operations required by both algorithms, together with their computational complexity and the execution time on our computer are given in Algorithm 1. Note that $\mathcal{O}(D)$ is the cost of projecting the output vector into the PCA space. Note as well that for incremental SDM, the "Sampling and Feature extraction" step is repeated L times.

5 Experimental Results

This section describes the experimental results. First, we empirically demonstrate the performance of CCR is equivalent to SDM. In order to do so, we assess both methods under the same settings, avoiding artefacts to appear, such as face detection accuracy. We follow the VOT Challenge protocol [33]. Then, we develop a fully automated system, and we evaluate both the CCR and iCCR in the same settings as the 300VW, and show that our fully automated system achieves state of the art results, illustrating the benefit of incremental learning to achieve it.

Algorithm 1. Computational costs for iCCR and incremental SDM [8] updates

iCCR update (Total: 72 ms.):
precompute: Feature and Jacobian extraction : $\langle \mathcal{O}(3q) : 9 \text{ ms.} \rangle$

for $i \leftarrow 1$ **to** $L = 3$ *cascade levels* **do**
  PCA Projection : $\langle \mathcal{O}(Dm) : 6 \text{ ms.} \rangle$;
  Update $\mathbf{R}$ (Eq. 16) : $\langle \mathcal{O}(md^2) : 15 \text{ ms.} \rangle$;
end

---

iSDM [8] update (Total: 705 ms.):
for $i \leftarrow 1$ **to** $L = 3$ *cascade levels* **do**
  Sampling and Feature extraction : $\langle \mathcal{O}(Kq) : 30 \text{ ms.} \rangle$;
  PCA Projection : $\langle \mathcal{O}(DK) : 5 \text{ ms.} \rangle$;
  Update $\mathbf{R}$ (Eqs. 6–9) : $\langle \mathcal{O}(d^3) : 200 \text{ ms.} \rangle$;
end

5.1 Experimental Set-Up

Training Data: We use data from different datasets of static images to construct our training set. Specifically, we use Helen [34], LFPW [35], AFW [36], IBUG [37], and a subset of MultiPIE [38]. The training set comprises $\sim$ 7000 images. We have used the facial landmark annotations provided by the 300 faces in the wild challenge [37], as they offer consistency across datasets. The *statistics* are computed across the training sequences, by computing the differences of ground-truth shape parameters between consecutive frames. Given the easiness of the training set with respect to the test set, we also included differences of several frames ahead. This way, higher displacements are also captured.

Features: We use the SIFT [39] implementation provided by Xiong and De la Torre [17]. We apply PCA on the output, retaining 2000 dimensions. We apply the same PCA to all of the methods, computed during our SDM training.

Test Data: All the methods are evaluated on the test partition of the 300 Videos in the Wild challenge (300VW) [13]. The 300VW is the only publicly-available large-scale dataset for facial landmark tracking. Its test partition has been divided into categories 1, 2 and 3, intended to represent increasingly unconstrained scenarios. In particular, category 3 contains videos captured in totally unconstrained scenarios. The ground truth has been created in a semi-supervised manner using two different methods [29,40].

Error Measure: To compute the error for a specific frame, we use the error measure defined in the 300VW challenge [13]. The error is computed by dividing the average point-to-point Euclidean error by the inter-ocular distance, understood as the distance between the two outer eye corners.

5.2 CCR vs. SDM

In order to demonstrate the performance capability of our CCR method against SDM, we followed the protocol established by the Visual Object Tracking (VOT) Challenge organisers for evaluating the submitted tracking methods [33]. Specifically, if the tracker error exceeds a certain threshold (0.1 in our case, which is a common definition of alignment failure), we proceed by re-initialising the tracker. In this case, the starting point will be the ground truth of the previous frame. This protocol is adopted to avoid the pernicious influence on our comparison of some early large failure from which the tracker is not able to recover, which would mean that successive frames would yield a very large error. Results are shown in Fig. 3 (**Left**). We show that the CCR and the SDM provide similar performance, thus ensuring that the CCR is a good starting point for developing an incremental learning algorithm. It is possible to see from the results shown in Fig. 3 that the CCR compares better and even sometimes surpasses the SDM on the lower levels of the error, while the SDM systematically provides a gain for larger errors with respect to the CCR. This is likely due to the use of first-order Taylor approximation, which means that larger displacements are less accurately approximated. Instead, the use of *infinite* shape perturbations rather than a handful of sampled perturbations compensates this problem for smaller errors, and even sometimes provides some performance improvement.

5.3 CCR vs. iCCR

We now show the benefit of incremental learning with respect to generic models. The incremental learning needs to filter frames to decide whether a fitting is suitable or harmful to update the models. That is, in practice, it is beneficial to filter out badly-tracked frames by avoiding performing incremental updates in these cases. We follow [8] and use a linear SVM trained to decide whether a particular fitting is "correct", understood as being under a threshold error. Despite its simplicity, this tactic provides a solid performance increase. Results on the test set are shown in Fig. 3 (**Right**).

5.4 Comparison with State of the Art

We developed a fully automated system to compare against state of the art methods. Our fully automated system is initialised with a standard SDM [41], and an SVM is used to detect whether the tracker gets lost. We assessed both our CCR and iCCR in the most challenging category of the 300VW, consisting of 14 videos recorded in unconstrained settings. For a fair comparison, we have reproduced the challenge settings (a brief description of the challenge and submitted methods can be found in [13]). We compare our method against the top two participants [42,43]. Results are shown in Fig. 4. The influence of the incremental learning to achieve state of the art results is clear. Importantly, as shown in the paper, our iCCR allows for real-time implementation. That is to say, our iCCR reports state of the art results whilst working in near real-time,

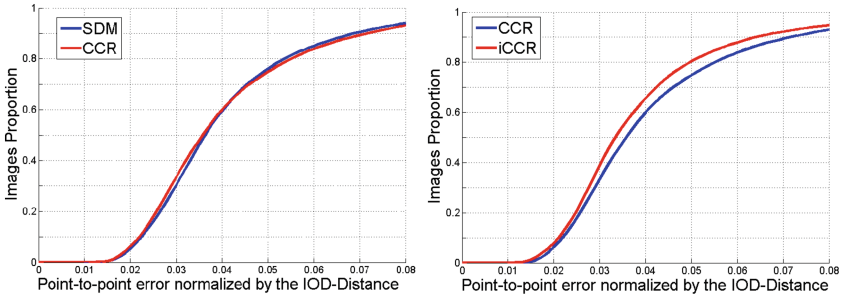

Fig. 3. Left: Accumulated graph across all three categories for SDM and CCR methods. In both cases, the Area Under the Curve (AUC) is 0.49, meaning that CCR shows better capabilities for lower errors, whereas SDM fits better in higher errors. **Right**: Accumulated graph across all three categories for CCR and iCCR methods. The contribution of incremental learning is clear.

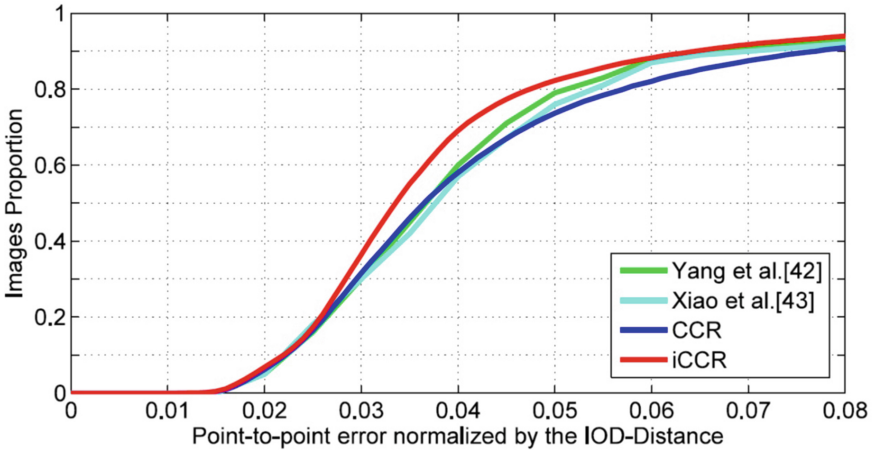

Fig. 4. Results given by our fully automated system in the most challenging category of the 300VW benchmark. Results are shown for the 49 inner points. The contribution of Incremental Learning in challenging sequences, and in a fully automated system, is even higher.

something that could not be achieved by previous works on Cascaded Regression. Code for our fully automated system is available for download at www.cs.nott.ac.uk/~psxes1.

6 Conclusion

In this article we have proposed a novel facial landmark tracking algorithm that is capable of performing on-line updates of the models through incremental learning. Compared to previous incremental learning methodologies, it can

produce much faster incremental updates without compromising on accuracy. This was achieved by firstly extending the Continuous Regression framework [10], and then incorporating it into the cascaded regression framework to lead to the CCR method, which we showed provides equivalent performance to the SDM. We then derived the incremental learning update formulas for the CCR, resulting in the iCCR algorithm. We further show the computational complexity of the incremental SDM, demonstrating that iCCR is an order of magnitude simpler computationally. This removes the bottleneck impeding real-time incremental cascaded regression methods, and thus results in the state of the art for real-time face tracking.

Acknowledgments. The work of Sánchez-Lozano, Martinez and Valstar was supported by the European Union Horizon 2020 research and innovation programme under grant agreement No 645378, ARIA-VALUSPA. The work of Sánchez-Lozano was also supported by the Vice-Chancellor's Scholarship for Research Excellence provided by the University of Nottingham. The work of Tzimiropoulos was supported in part by the EPSRC project EP/M02153X/1 Facial Deformable Models of Animals. We are also grateful for the given access to the University of Nottingham High Performance Computing Facility, and we would like to thank Jie Shen and Grigoris Chrysos for their insightful help in our tracking evaluation.

References

1. Dhall, A., Goecke, R., Joshi, J., Sikka, K., Gedeon, T.: Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In: International Conference on Multimodal Interaction, pp. 461–466 (2014)
2. Zhou, S., Krueger, V., Chellappa, R.: Probabilistic recognition of human faces from video. Comput. Vis. Image Underst. **91**(12), 214–245 (2003)
3. Gross, R., Matthews, I., Baker, S.: Generic vs. person specific active appearance models. Image Vis. Comput. **23**(11), 1080–1093 (2005)
4. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S., Torr, P.: Struck: Structured output tracking with kernels. Trans. Pattern Anal. Mach. Intell. (2016). doi:10.1109/TPAMI.2015.2509974
5. Wang, X., Valstar, M., Martinez, B., Khan, M.H., Pridmore, T.: Tric-track: tracking by regression with incrementally learned cascades. In: International Conference on Computer Vision (2015)
6. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1–3), 125–141 (2008)
7. Xiong, X., la Torre, F.D.: Supervised descent method for solving nonlinear least squares problems in computer vision. arXiv abs/1405.0601 (2014)
8. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
9. Sagonas, C., Panagakis, Y., Zafeiriou, S., Pantic, M.: RAPS: Robust and efficient automatic construction of person-specific deformable models. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
10. Sánchez-Lozano, E., De la Torre, F., González-Jiménez, D.: Continuous regression for non-rigid image alignment. In: European Conference on Computer Vision, pp. 250–263 (2012)

11. Dollár, P., Welinder, P., Perona, P.: Cascaded pose regression. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1078–1085 (2010)
12. Yan, J., Lei, Z., Yang, Y., Li, S.: Stacked deformable part model with shape regression for object part localization. In: European Conference on Computer Vision, pp. 568–583 (2014)
13. Shen, J., Zafeiriou, S., Chrysos, G.S., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The first facial landmark tracking in-the-wild challenge: benchmark and results. In: International Conference on Computer Vision - Workshop (2015)
14. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001)
15. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Comput. Vis. **60**(2), 135–164 (2004)
16. Saragih, J.M., Lucey, S., Cohn, J.F.: Deformable model fitting by regularized landmark mean-shift. Int. J. Comput. Vis. **91**(2), 200–215 (2011)
17. Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: IEEE Conference on Computer Vision and Pattern Recognition (2013)
18. Tresadern, P., Ionita, M., Cootes, T.: Real-time facial feature tracking on a mobile device. Int. J. Comput. Vis. **96**(3), 280–289 (2012)
19. Tzimiropoulos, G., Pantic, M.: Optimization problems for fast AAM fitting in-the-wild. In: International Conference on Computer Vision (2013)
20. Tzimiropoulos, G., Pantic, M.: Gauss-newton deformable part models for face alignment in-the-wild. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1851–1858 (2014)
21. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Comput. Vis. Image Underst. **61**(1), 38–59 (1995)
22. Cristinacce, D., Cootes, T.: Feature detection and tracking with constrained local models. In: British Machine Vision Conference, pp. 929–938 (2006)
23. Cootes, T.F., Ionita, M.C., Lindner, C., Sauer, P.: Robust and accurate shape model fitting using random forest regression voting. In: European Conference on Computer Vision, pp. 278–291 (2012)
24. Valstar, M.F., Martinez, B., Binefa, X., Pantic, M.: Facial point detection using boosted regression and graph models. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2729–2736 (2010)
25. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. Int. J. Comput. Vis. **107**(2), 177–190 (2014)
26. Ren, S., Cao, X., Wei, Y., Sun, J.: Face alignment at 3000 FPS via regressing local binary features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1685–1692 (2014)
27. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
28. Yan, J., Lei, Z., Yi, D., Li, S.: Learn to combine multiple hypotheses for accurate face alignment. In: Internation Conference on Computer Vision - Workshop, pp. 392–396 (2013)
29. Tzimiropoulos, G.: Project-out cascaded regression with an application to face alignment. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3659–3667 (2015)
30. Cootes, T.F., Taylor, C.J.: Statistical models of appearance for computer vision (2004)

31. Cao, X., Wei, Y., Wen, F., Sun, J.: Face alignment by explicit shape regression. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2887–2894 (2012)
32. Brookes, M.: The matrix reference manual (2011)
33. Kristan, M., Matas, J., Leonardis, A., Vojir, T., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. arXiv (2015)
34. Le, V., Brandt, J., Lin, Z., Bourdev, L.D., Huang, T.S.: Interactive facial feature localization. In: European Conference on Computer Vision, pp. 679–692 (2012)
35. Belhumeur, P.N., Jacobs, D.W., Kriegman, D.J., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 545–552 (2011)
36. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2879–2886 (2012)
37. Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: A semi-automatic methodology for facial landmark annotation. In: IEEE Conference on Computer Vision and Pattern Recognition - Workshops (2013)
38. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie. Image Vis. Comput. **28**(5), 807–813 (2010)
39. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
40. Chrysos, G.S., Antonakos, E., Zafeiriou, S., Snape, P.: Offline deformable face tracking in arbitrary videos. In: International Conference on Computer Vision - Workshop (2015)
41. Sánchez-Lozano, E., Martinez, B., Valstar, M.: Cascaded regression with sparsified feature covariance matrix for facial landmark detection. Pattern Recogn. Lett. **73**, 19–25 (2016)
42. Yang, J., Deng, J., Zhang, K., Liu, Q.: Facial shape tracking via spatio-temporal cascade shape regression. In: Internationl Conference on Computer Vision - Workshop (2015)
43. Xiao, S., Yan, S., Kassim, A.: Facial landmark detection via progressive initialization. In: International Conference on Computer Vision - Workshop (2015)