Incremental Face Alignment in the Wild

Akshay Asthana[1] Stefanos Zafeiriou[1] Shiyang Cheng[1] Maja Pantic[1,2]
[1]Department of Computing, Imperial College London, United Kingdom
[2]EEMCS, University of Twente, Netherlands
{a.asthana, s.zafeiriou, shiyang.cheng11, m.pantic}@imperial.ac.uk

Abstract

The development of facial databases with an abundance of annotated facial data captured under unconstrained 'in-the-wild' conditions have made discriminative facial deformable models the de facto choice for generic facial landmark localization. Even though very good performance for the facial landmark localization has been shown by many recently proposed discriminative techniques, when it comes to the applications that require excellent accuracy, such as facial behaviour analysis and facial motion capture, the semi-automatic person-specific or even tedious manual tracking is still the preferred choice. One way to construct a person-specific model automatically is through incremental updating of the generic model. This paper deals with the problem of updating a discriminative facial deformable model, a problem that has not been thoroughly studied in the literature. In particular, we study for the first time, to the best of our knowledge, the strategies to update a discriminative model that is trained by a cascade of regressors. We propose very efficient strategies to update the model and we show that is possible to automatically construct robust discriminative person and imaging condition specific models 'in-the-wild' that outperform state-of-the-art generic face alignment strategies.

1. Introduction

The problem of construction and alignment[1] of generic deformable models capable of capturing the variability of a non-rigid object is among the most popular and well-studied problem in the field of computer vision. Arguably, the most studied non-rigid object is the human face. Based on the ways the various deformable models are built and their respective alignment procedure, the existing methodologies can be broadly classified into *Generative* and *Discriminative*. The *Generative* methods use an analysis-by-synthesis loop where the optimization strategy attempts to find the required parameters by maximizing the probability of the input image being constructed by the facial deformable model. Most notable example of this category is the Active Appearance Model (AAM) [10, 21].

The *Discriminative* methods rely on the use of discriminative information (i.e. a set of facial landmark classifiers [28] or discriminative functions [19, 15, 32] or both [2, 27, 29]). Many discriminative methods use part-based approaches, most notable example being the Constrained Local Model (CLM) [7, 28] paradigm, that represents the face via a set of local image patches cropped around the landmark points. Recently, a number of discriminative methodologies have shown excellent results for facial landmark localization [4, 2, 32]. The common characteristic of these methods is that they used a cascade of regression functions to map the textual features to shape directly [4, 32] or to shape parameters [2][2]. Furthermore, the authors of [32] went a step further arguing that the cascaded linear regression can be presented as a supervised gradient descent methodology.

Many of the above discriminative methodologies have shown to be successful for facial landmark localization under uncontrolled environments, recently referred to as *in-the-wild* settings [4, 2, 32], achieving even real time performance [2, 32, 5]. Without exception, these methods rely on a static generic model that is built completely on off-line training data. Nevertheless, when it comes to the applications that require perfect facial alignment and tracking accuracy, such as the analysis of human facial behavior (e.g., facial expression and action unit recognition [6]) and the facial motion capture, person-specific rather than generic models are mainly applied [1, 6, 11].

One way to automatically create a personalized facial deformable model from a generic one is through incremental learning. Very limited research has been conducted towards incremental deformable models, mostly restricted to AAM [30, 22] in which the incremental Principal Component Analysis (iPCA) [18] is applied to the fittings produced by a generic AAM or via update of the mean template of the AAM [23]. Apart from the problems associated with the

[1]Problem of deformable model alignment can be encountered under different names in literature, including fitting, landmark localization etc.

[2]Similar ideas have been explored for human pose estimation [9].

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.240

AAM framework in handling generic face alignment scenario and uncontrolled natural settings, the main drawback of these incremental approaches is that the erroneous fitting, which are very difficult to spot by simply thresholding the fitting score [30, 22], may arbitrarily bias iPCA and results in model drifting. Moreover, these incremental methodologies are applicable only to the generative AAMs.

In this paper, we study the problem of incremental training for the discriminative facial deformable models. The incremental training of discriminative models is not only important for building person-specific models but also to update a generic model in case a new annotated data arrives, since the training procedure is very expensive and time consuming. In particular, we study incremental training of discriminative models that use a cascade of linear regressors to learn the mapping from facial texture to the shape, a problem that, to the best of our knowledge, has not been studied in the literature before. For this, we exploit the fact that the cascade of regressors is trained using the Monte-Carlo sampling methodologies [2, 32] and present a very efficient methodology which can incrementally update all linear regressors in cascade in parallel. We demonstrate that the proposed incremental methods for deformable model alignment: (1) Are capable of adding new training samples and updating the model, without re-training from scratch, thereby, constantly increasing robustness of the generic model; (2) Can automatically tailor themselves to the subject being tracked and the imaging conditions using image sequences, and hence, become person-specific over time.

Note that it has been shown in [12, 33] that the main challenge for the deformable face models is the difficulty encountered in modeling the facial texture, whereas, the generative model of the sparse facial shape, trained even on the faces captured under constrained conditions, is capable of faithfully representing the facial shape of unseen faces captured under unconstrained conditions. Hence, we do not deal with the problem of updating the shape model and focus entirely on the problem of incrementally updating the function that maps facial texture to facial shape.

2. Problem and Motivation

In this section, we describe the general framework of cascade linear regression for discriminative face alignment [32]. Then, we show that the incremental update of the cascade of linear regression is a very challenging task, since the results from one level have to propagated to the next. Due to this sequential nature of the training procedure, we refer to this method [32] as *Sequential Cascade of Linear Regression* in the rest of the paper. And finally, since learning the cascade of regression is by nature a Monte-Carlo procedure [32], we argue that we can train every level independently using only the statistics of the previous level. To this end, we propose a *Parallel Cascade of Linear Regression* method which not only performs as accurately as (if not better than) the sequential method [32], but also allows for the incremental update of the cascades in a feasible and a computationally efficient manner.

2.1. Sequential Cascade of Linear Regression

Let a set of M images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^{M}$ and the set of ground-truth shapes $\mathcal{S} = \{\mathbf{s}_i^*\}_{i=1}^{M}$ with $\mathbf{s}^o \in \mathbb{R}^{n\times 1}$. Also, let a feature function $\mathbf{f}(\mathbf{I}, \mathbf{s}) \in \mathbb{R}^{1\times f}$, where, f is the dimensionality of the feature. This function could return the vector of the concatenation of SIFT [20] or Histogram of Oriented Gradient (HoG) [8] features around each landmark of shape $\mathbf{s}$ [32, 2] from image $\mathbf{I}$. The training procedure of the discriminative methods in [15, 32, 2] can be summarized as follows: Find a function g that can map an initial shape $\mathbf{s}^a$ of image $\mathbf{I}_i$ to the ground-truth shape $\mathbf{s}_i^*$, as $g(\mathbf{s}^a, \mathbf{I}_t, \mathbf{f}) = \mathbf{s}_i^*$. The initial shape could be just the mean shape initialized in the bounding box returned by a face detector [32, 2].

In [32], function g is learned iteratively using a cascade of regression functions that maps the extracted feature vectors of images to shape [32] directly. In this paper, we use a parametric 3D shape model [2, 28] described as:

$$\mathbf{s}(\mathbf{p}) = s\mathbf{R}(\bar{\mathbf{s}} + \mathbf{\Phi}_\mathbf{s}\mathbf{g}) + \mathbf{t}, \tag{1}$$

where $\mathbf{R}$ (computed via pitch r_x, yaw r_y and roll r_z), s and $\mathbf{t} = [t_x; t_y; 0]$ control the rigid 3D rotation, scale and translations respectively, while $\mathbf{g}$ controls the non-rigid variations of the shape. Therefore, the parameters for the 3D shape model are $\mathbf{p} = [s; r_x; r_y; r_z; t_x; t_y; \mathbf{g}]^T$. Hence, instead of $\mathcal{S}$, we have a set of ground-truth of shape parameters $\mathcal{P}^* = \{\mathbf{p}_i^*\}_{i=1}^{M}$. Hence, the goal is to learn a function from an initial estimate of $\mathbf{p}$ that takes us to the ground-truth shape parameters $\mathbf{p}^*$, where, both $\mathbf{p}^*$ and $\mathbf{p} \in \mathbb{R}^{1\times l}$, and l is the total number of shape parameters.

The Monte-Carlo procedure to learn the sequential cascade of regression functions can be described as follows. For each of the training shapes in $\mathcal{S}$, the shape model parameter subspace is sampled within a pre-defined range around ground-truth shape parameters $\mathcal{P}^*$ and an initial set of L perturbed shapes is sampled which provides as set of L perturbed shape parameters $\{\mathbf{p}_j^{(1)}\}_{j=1}^{L}$. We want to learn a linear rule from the perturbed parameters $\mathbf{p}^{(1)}$ of image $\mathbf{I}$ such that

$$\begin{aligned}\mathbf{p}^* &= \mathbf{p}^{(1)} + \mathbf{f}(\mathbf{I}, \mathbf{s}(\mathbf{p}^{(1)}))\mathbf{W} + \mathbf{b} \\ &= \mathbf{p}^{(1)} + [\mathbf{f}(\mathbf{I}, \mathbf{s}(\mathbf{p}^{(1)}))\ 1]\tilde{\mathbf{W}} \\ &= \mathbf{p}^{(1)} + \tilde{\mathbf{f}}(\mathbf{I}, \mathbf{s}(\mathbf{p}^{(1)}))\tilde{\mathbf{W}}\end{aligned}$$

where, $\tilde{\mathbf{W}} = [\mathbf{W}; \mathbf{b}]$ and $\tilde{\mathbf{f}}(\mathbf{I}, \mathbf{s}(\mathbf{p}^{(1)})) = [\mathbf{f}(\mathbf{I}, \mathbf{s}(\mathbf{p}^{(1)}))\ 1]$. Since it is difficult to learn only one $\tilde{\mathbf{W}}$ that directly maps the perturbed $\mathbf{p}^{(1)}$ to the ground-truth $\mathbf{p}^*$, we can train a cascade of regression functions in a sequential manner as following. We learn the first $\tilde{\mathbf{W}}^{(1)}$ by solving the following least squares problem [32]:

$$\underset{\mathbf{W}^{(1)},\mathbf{b}^{(1)}}{\arg\min} \sum_{i=1}^{M} \sum_{j} ||\Delta \mathbf{p}_{ij}^{(1)} - \tilde{\mathbf{f}}(\mathbf{I}_i, \mathbf{p}_{ij}^{(1)}) \tilde{\mathbf{W}}^{(1)}||^2 \quad (2)$$

where $\Delta \mathbf{p}_{ij}^{(1)} = \mathbf{p}_i^* - \mathbf{p}_{ij}^{(1)}$ and j counts the perturbations. For notation simplicity, let $\tilde{\mathbf{f}}(\mathbf{I}_i, \mathbf{p}_{ij}) = \tilde{\mathbf{f}}_{ij}$, $\mathbf{X}^{(1)} = [\tilde{\mathbf{f}}_{ij}]$ and $\mathbf{Y}^{(1)} = [\Delta \mathbf{p}_{ij}^{(1)}]$, $\tilde{\mathbf{W}}^{(1)}$ can be estimated as:

$$\tilde{\mathbf{W}}^{(1)} = \left[(\mathbf{X}^{(1)})^T \mathbf{X}^{(1)} + \lambda \mathbf{E} \right]^{-1} (\mathbf{X}^{(1)})^T \mathbf{Y}^{(1)} \quad , \quad (3)$$

where, $\mathbf{E}$ is the identity matrix and the term $\lambda \mathbf{E}$ is included in case that $(\mathbf{X}^{(1)})^T \mathbf{X}^{(1)}$ is singular. This is also known as Ridge Regression[14].

Let us apply the update rule $\mathbf{p}_{ij}^{(2)} = \mathbf{p}_{ij}^{(1)} + \tilde{\mathbf{f}}(\mathbf{I}_i, \mathbf{s}(\mathbf{p}_{ij}^{(1)})) \tilde{\mathbf{W}}^{(1)}$, and get a new set of estimates $\mathcal{P}^{(2)} = \{\mathbf{p}_{ij}^{(2)}\}$. Now, we want to find a new $\tilde{\mathbf{W}}^{(2)}$ that takes us closer to $\mathbf{p}_i^*$. We can now generalize to find $\tilde{\mathbf{W}}^{(k)}$ for the k-th step and the updated rule for the next set of shape parameters $\mathbf{p}^{(k+1)}$. At step k:

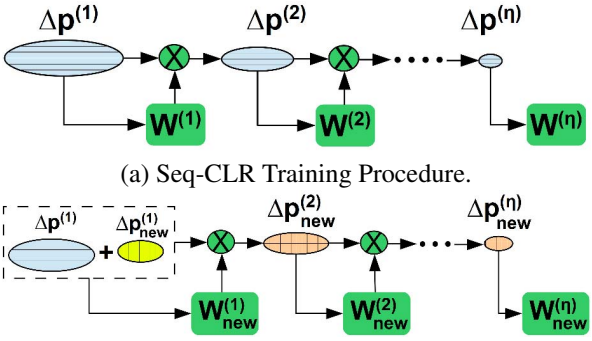
$$\begin{aligned} \tilde{\mathbf{W}}^{(k)} &= \left[(\mathbf{X}^{(k)})^T \mathbf{X}^{(k)} + \lambda \mathbf{E} \right]^{-1} (\mathbf{X}^{(k)})^T \mathbf{Y}^{(k)} \\ \mathbf{p}_{ij}^{(k+1)} &= \mathbf{p}_{ij}^{(k)} + \tilde{\mathbf{f}}(\mathbf{I}_i, \mathbf{s}(\mathbf{p}_{ij}^{(k)})) \tilde{\mathbf{W}}^{(k)} \end{aligned} \quad (4)$$

This procedure is sequentially repeated such that at each step, we get closer to the ground-truth parameters $\mathbf{p}^*$ i.e. the variance for the perturbations $\Delta \mathbf{p}^{(k)}$ decreases as the number of iterations k increases. We refer to this as the Sequential Cascade of Linear Regression (Seq-CLR) method.

2.2. Problem with Incremental Seq-CLR

While the above discussed Seq-CLR method (Section 2.1) has been shown to give state-of-the-art face alignment results, the sequential procedure involved in training the cascade of regression functions is not well suited for the task of incremental update. In that, if new data samples have to be added (for example, images captured under previously unseen imaging conditions), the entire cascade of regression functions have to be re-trained from scratch which is extremely expensive and time consuming. The problem is illustrated in Figure 1. Given the initial set of perturbations $\Delta \mathbf{p}^{(1)}$, we compute the initial regression function $\tilde{\mathbf{W}}^{(1)}$. We then propagate $\Delta \mathbf{p}^{(1)}$ through $\tilde{\mathbf{W}}^{(1)}$ to generate the subsequent set of perturbations $\Delta \mathbf{p}^{(2)}$ and compute the regression function $\tilde{\mathbf{W}}^{(2)}$. Similarly, to compute $\tilde{\mathbf{W}}^{(3)}$, we generate $\Delta \mathbf{p}^{(3)}$ by propagating $\Delta \mathbf{p}^{(2)}$ through $\tilde{\mathbf{W}}^{(2)}$. This procedure is repeated until the convergence criteria has been met. See Eqn. 4 for details.

Now, if a new sample or a set of new samples $\mathbf{p}_{\text{new}}$ have to be added, the initial regression function $\tilde{\mathbf{W}}^{(1)}$ can be incrementally updated to $\tilde{\mathbf{W}}_{\text{new}}^{(1)}$ (Section 3.1) which is computed simply by using the augmented set of samples $\mathbf{p}_{\text{new}}^{(1)} = \{\mathbf{p}^{(1)}, \mathbf{p}_{\text{new}}\}$. However, since the initial regression function $\tilde{\mathbf{W}}_{\text{new}}^{(1)}$ has changed, the subsequent set of samples



(a) Seq-CLR Training Procedure.

(b) Updating Seq-CLR after adding new samples.

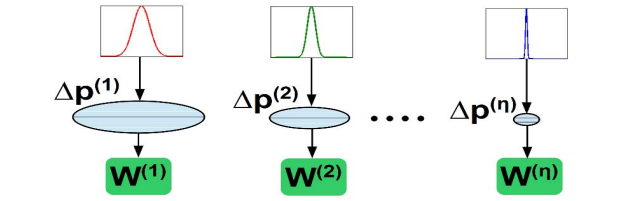
Figure 1: Problem with Incremental Seq-CLR

$\mathbf{p}_{\text{new}}^{(2)}$ will be re-computed by propagating the entire augmented set $\mathbf{p}_{\text{new}}^{(1)}$ through $\tilde{\mathbf{W}}_{\text{new}}^{(1)}$. As a result, the regression function $\tilde{\mathbf{W}}_{\text{new}}^{(2)}$ will also have to re-computed from scratch using Eqn. 4 which is computationally extremely intensive (requires huge matrix inversion) and time consuming. The same also applies to the subsequent iterations. As such, using the Seq-CLR training procedure (Section 2.1), *only the computation of the initial regression function* $\tilde{\mathbf{W}}^{(1)}$ *can be formulated in an incremental framework*, while all the other regression functions have to be computed from scratch (following the usual sequential training procedure and update rules given in Eqn. 4) as they rely on the perturbations generated from the previous iterations.

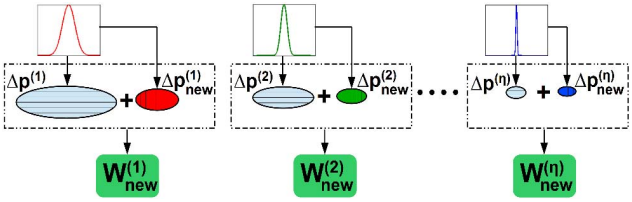
2.3. Parallel Cascade of Linear Regression

To address the problem of incremental formulation of Seq-CLR, discussed above in Section 2.2, we propose a Parallel Cascade of Linear Regression (Par-CLR) method that has the following properties: (1) The Par-CLR method shows the same level of alignment accuracy as the Seq-CLR method; and (2) In Par-CLR, the perturbations required for training or updating the cascade of regression functions do not rely on previous iterations; (3) The Par-CLR method is extremely well suited for the incremental formulation (Section 3) and is highly parallelizable making the incremental formulation real-time capable.

Note that for training the Seq-CLR method, the initial set of perturbations ($\Delta \mathbf{p}^{(1)}$) are obtained by Monte-Carlo sampling procedure [32], in that the perturbations are randomly drawn within a *fixed pre-defined range* around the ground-truth shape parameters ($\mathbf{p}^*$). For the experiments in this paper, this predefined range was set to ± 15 pixels for translation, $\pm 10°$ for rotation, ± 0.1 for scaling and 1.5 standard deviation (based on the available training set) for the non-rigid shape parameters ($\mathbf{g}$). In a Monte-Carlo setting, the aim of the cascade is to reduce the variance of the perturbations at each level. Motivated by this, we argue that the regression functions at all levels in a cascade can be trained (and updated) independently using only the statistics of the



(a) Par-CLR Training Procedure.



(b) Updating Par-CLR after adding new samples.

Figure 2: Incremental Formulation of Par-CLR.

previous level, thereby, eliminating the need for propagating the samples through all the previous iterations. We refer to this method as the Parallel Cascade of Linear Regression (Par-CLR) method.

For this, we collect the statistics for each shape parameter (in the form of its variance) at each iteration while training the cascade of regression functions using the Seq-CLR method (Section 2.1) on an offline database. Let this distribution be $\mathcal{D} = \{\sigma^{(1)}, \ldots, \sigma^{(\eta)}\}$, where, η is the maximum number of iterations. For the experiments presented in this paper, $\mathcal{D}$ was set so as to capture the spread of 98% of the samples at each iteration. In Par-CLR method, the perturbations for training the cascade of regression functions can be drawn directly from this distribution, without relying on the previous iteration. This modification in the training procedure not only makes it highly parallelizable in that all the regression functions can now be trained independently but, more importantly, this is achieved without any loss in the alignment accuracy as compared to the Seq-CLR method. See the next section for a motivating experiments.

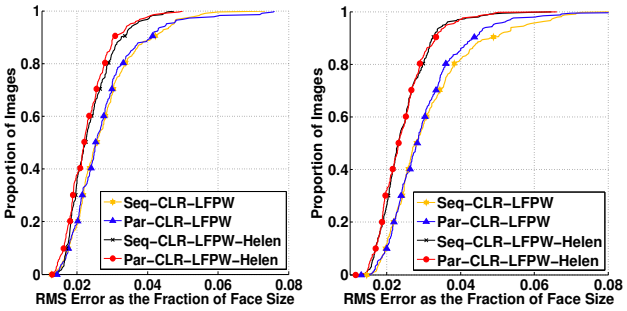
2.4. Experiment 1: Seq-CLR Vs. Par-CLR

The goal for this motivating experiment is to compare the performance of the Seq-CLR method (Section 2.1) against the Par-CLR method (Section 2.3). For this we use LFPW [3, 26, 25] and Helen [17, 26, 25] datasets as they contain images captured *in the wild*. The results are reported in Figure 3.

Firstly, we trained the cascade of regression functions using the Seq-CLR method using only the LFPW training set, referred to as Seq-CLR-LFPW. This model was used for aligning images in the LFPW testing set. Next, we compute the distribution $\mathcal{D}_{\text{LFPW}}$ (Section 2.3), signifying the spread of perturbations at each level of the cascade obtained during the training of Seq-CLR-LFPW. Using this distribution for drawing the perturbations, the training for each level of the cascade is also performed independently using the LFPW training set. We refer to this method as Par-CLR-LFPW. To validate the results, we also aligned images in the Helen testing set using the Seq-CLR-LFPW and Par-CLR-LFPW models. Overall, the results indicate that Seq-CLR and Par-CLR show identical performance (Figure 3).

Next, we augment the LFPW training set with the new Helen training set. To test the generalization capability of the Par-CLR method, we use the same distribution as above i.e. $\mathcal{D}_{\text{LFPW}}$, and train the cascade of regression functions independently using the Par-CLR method. We refer to this model as Par-CLR-LFPW-Helen. For comparison, we also train the model using the Seq-CLR method and refer to this as Seq-CLR-LFPW-Helen. Again, both methods show identical performance (Figure 3).

Overall, the results clearly indicate that by using simple statistic to model the spread of perturbations, the Monte-Carlo sampling based sequential training procedure can be compensated and each of the cascaded regression functions can be trained independently of the previous iteration *without loss in alignment accuracy*. In the next section, we will exploit this parallel training procedure and formulate a very efficient methodology to incrementally update the cascade of regression functions. The underlying assumption of the parallel assumption is that at each step the distribution of the perturbations is Gaussian. The assumption is valid in the first step by definition, since the perturbation have been drawn from a single multivariate Gaussian. To validate that this is true for the remaining steps of the cascade, we have employed the Kolmogorov-Smirnov (KS) statistical test [24], which validated our assumption.



(a) LFPW Test Set (b) Helen Test Set

Figure 3: Seq-CLR Vs. Par-CLR Results.

3. Incremental Face Alignment Framework

The above discussed Par-CLR method (Section 2.3) is the foundation for the proposed incremental face alignment framework. The Par-CLR method not only has an *exact* incremental solution per level, but it also allows for all the regression functions in a cascade to be updated independently of each other in parallel. This makes the proposed incremental framework highly parallelizable and real-time capable. In Section 3.1, we derive the solution for the Incremental Linear Least-Squares problem and list the update

rules. Next, in Section 3.2, we present the Incremental Parallel Cascade of Linear Regression (iPar-CLR) method.

3.1. Incremental Linear Least-Squares Problem

Given the feature matrix $\mathbf{X}(T)$ and perturbation ($\Delta\mathbf{p}$) matrix $\mathbf{Y}(T)$, where, T is the number of training samples, the regression function $\tilde{\mathbf{W}}(T)$ is computed as follows:

$$\begin{aligned}\tilde{\mathbf{W}}(T) =& \mathbf{V}(T)\mathbf{X}(T)^T\mathbf{Y}(T)\\ \mathbf{V}(T) =& [\mathbf{X}(T)^T\mathbf{X}(T) + \lambda\mathbf{E}]^{-1}\end{aligned} \tag{5}$$

See Eqn. 2 and Eqn. 3 for details. Now, let us assume that R new training samples are added i.e. $\mathbf{X}(R)$ and $\mathbf{Y}(R)$. The *Update Rules* are as follows (Derivation in Appendix A):

$$\mathbf{V}(T+R) = \mathbf{V}(T) - \mathbf{Q}\mathbf{V}(T) \tag{6}$$

$$\begin{aligned}\tilde{\mathbf{W}}(T+R) =& \tilde{\mathbf{W}}(T) - \mathbf{Q}\tilde{\mathbf{W}}(T)\\ &+ \mathbf{V}(T+R)\mathbf{X}(R)^T\mathbf{Y}(R)\end{aligned} \tag{7}$$

$$\text{where, } \mathbf{Q} = \mathbf{V}(T)\mathbf{X}(R)^T\mathbf{U}\mathbf{X}(R) \tag{8}$$

$$\text{and } \mathbf{U} = \left[\mathbf{E} + \mathbf{X}(R)\mathbf{V}(T)\mathbf{X}(R)^T\right]^{-1} \tag{9}$$

Properties

- Solution in Eqn. 7 is an exact mathematical equivalent of the closed-form solution of $\tilde{\mathbf{W}}(T+R)$.
- Computationally very efficient and update for adding R new samples is achieved in just one step.
- Does not requires data to be stored. Only $\tilde{\mathbf{W}}(T)$ and $\mathbf{V}(T)$ need to be saved.
- Matrix inversion is required just once for $\mathbf{U}$ and the size of this matrix is just $R \times R$. The closed-form solution requires the inversion of matrix of size $\tilde{f} \times \tilde{f}$, where $\tilde{f}$ is dimensionality of feature (usually large).

Special Case: If one sample at a time, say $\mathbf{x}$ and $\mathbf{y}$, is added (i.e. $R = 1$), the *Update Rules* is as follows:

$$\mathbf{V}(T+1) = \mathbf{V}(T) - \frac{\mathbf{V}(T)\mathbf{x}^T\mathbf{x}\mathbf{V}(T)}{1 + \mathbf{x}\mathbf{V}(T)\mathbf{x}^T} \tag{10}$$

$$\tilde{\mathbf{W}}(T+1) = \tilde{\mathbf{W}}(T) - \mathbf{V}(T+1)\mathbf{x}^T\left(\mathbf{y} - \mathbf{x}\tilde{\mathbf{W}}(T)\right) \tag{11}$$

This is the well-known recursive linear least-squares solution [13]. Similar to Eqn. 7, this is an exact mathematical equivalent of the closed-form solution of $\tilde{\mathbf{W}}(T+1)$. Moreover, this method is computationally extremely efficient as the update procedure requires only matrix/vector multiplications and no matrix inversion is required.

3.2. Incremental Par-CLR Formulation

In this section, we present an incremental formulation for the Parallel Cascade of Linear Regression method (Section 2.3). We state the update rules for incrementally adding new training samples and updating the cascade of regression functions in an efficient manner. We call this incremental Par-CLR (iPar-CLR) method and an overview of the method is shown in Figure 2.

Given the initial cascade of regression functions (Eqn. 5), represented by $\mathcal{V} = \{\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(\eta)}\}$ and $\mathcal{W} = \{\tilde{\mathbf{W}}^{(1)}, \ldots, \tilde{\mathbf{W}}^{(\eta)}\}$, and the distribution $\mathcal{D} = \{\sigma^{(1)}, \ldots, \sigma^{(\eta)}\}$ to be used for sampling the perturbations, the goal is to add a new training image $\{\mathbf{I}_{\text{new}}, \mathbf{S}_{\text{new}}\}$ and update the cascade of regression functions in $\mathcal{V}$ and $\mathcal{W}$. Let us sample R perturbation from the new training image using $\mathcal{D}$ for each iteration $i = \{1, \ldots, \eta\}$. The step-by-step procedure to update each of the cascaded regression functions is given in Algorithm 1.

As discussed in Section 3.1, the update procedure is very efficient in that the update for adding R samples is achieved in one step. Also, the update procedure for each cascaded regression function is independent of the previous iterations and hence, the entire update can be performed in parallel.

Algorithm 1: iPar-CLR Update Procedure

Require: $\mathcal{V}$, $\mathcal{W}$, $\mathcal{D}$, $\mathbf{I}_{\text{new}}$, $\mathbf{S}_{\text{new}}$, R

parfor $i = 1 \rightarrow \eta$ **do**
- Get R samples $\{\Delta\mathbf{p}_j^{(i)}\}_{j=1}^{R}$ using distribution $\sigma^{(i)}$
- Compute $\{\tilde{\mathbf{f}}_j^{(i)}\}_{j=1}^{R}$ using the perturbed shapes generated from $\{\Delta\mathbf{p}_j^{(i)}\}_{j=1}^{R}$.
- Generate $\mathbf{X}(R)^{(i)} \in \mathbb{R}^{R \times \tilde{f}}$ and $\mathbf{Y}(R)^{(i)} \in \mathbb{R}^{R \times l}$
- Compute $\mathbf{V}_{\text{new}}^{(i)}$ using Eqn. 6, where, $\mathbf{V}(T+R) = \mathbf{V}_{\text{new}}^{(i)}$ and $\mathbf{V}(T) = \mathbf{V}^{(i)}$
- Compute $\tilde{\mathbf{W}}_{\text{new}}^{(i)}$ using Eqn. 7, where, $\tilde{\mathbf{W}}(T+R) = \tilde{\mathbf{W}}_{\text{new}}^{(i)}$, $\tilde{\mathbf{W}}(T) = \tilde{\mathbf{W}}^{(i)}$, $\mathbf{V}(T+R) = \mathbf{V}_{\text{new}}^{(i)}$ and $\mathbf{V}(T) = \mathbf{V}^{(i)}$

Output : Updated Cascade $\mathcal{V}_{\text{new}}$ and $\mathcal{W}_{\text{new}}$.

4. Experiments

In this section, we present detailed experiments for face alignment both in static images and videos. The first experiment investigates the ability of the incremental iPar-CLR method to continuously update the generic model, as the new annotated data arrives, thereby increasing its accuracy and robustness as more and more new training images are added. The second experiment investigates the incremental iPar-CLR method in a face tracking scenario with the particular focus on automatically updating the generic model on-the-fly and accessing its ability to adapt to the subject's face being tracked and the imaging conditions. Finally, in our experiments we have also considered a simple alternative of the Seq-CLR in which only the new arriving sample was propagated to the next levels. This alternative is much faster than the orginal Seq-CLR procedure since does not need to propagate the whole training set but, since we found out that it performs significantly worst than Seq-CLR and iPar-CLR, we opted not to include so that we do not clutter our graphs. Furthermore, according to our experiments this alternative of Seq-CLR was more susceptible to

outliers propagation and model-drifting. While, we verified that parallel approaches are more resilient to outliers, since its step is updated independently.

4.1. Face Alignment in Static Images

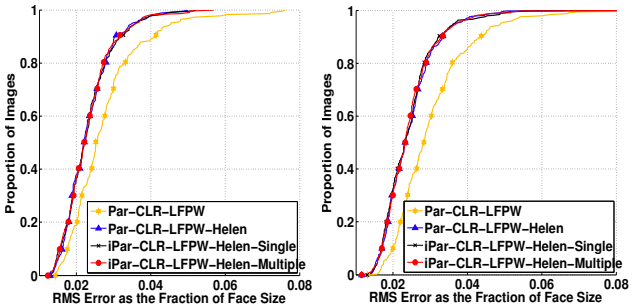
The goal of this experiment is to investigate the utility of the incremental iPar-CLR method (Section 3.2) in case a new annotated data arrives. More specifically, we investigate the scenario in which a new batch of training samples is added (for example, images captured under previously unseen imaging conditions). Obviously, the Seq-CLR (Section 2.1) and the Par-CLR (Section 2.3) methods have a static model in that the entire cascade of regression functions will have to be re-trained from scratch in order to incorporate the new samples. The proposed iPar-CLR framework, however, can update all the regression functions in a cascade on the fly. Similar to the experiment in Section 2.4, we use both the LFPW [3, 26, 25] and Helen [17, 26, 25] datasets.

We use the previously trained Par-CLR-LFPW model (Section 2.4) as the baseline and use the same distribution $\mathcal{D}_{\text{LFPW}}$ for drawing the perturbation from. Also, we initialize the cascades for iPar-CLR method (Section 3.2) with the cascade of Par-CLR-LFPW model. Note that the update procedure for the iPar-CLR method can be performed in two different ways: (1) By adding one sample at a time (See Eqn. 11 for update rules). We refer to this method as iPar-CLR-LFPW-Helen-Single; and (2) By adding multiple samples at a time (See Eqn. 7 for update rules). We refer to this method as iPar-CLR-LFPW-Helen-Multiple.

Now, one by one, we begin to add images from the Helen training set and update the cascade of regression functions using the iPar-CLR method (Algorithm 1). To validate the performance of the update procedure, we use these models for aligning images from both the LFPW and Helen testing sets. We observe a consistent increase in the alignment accuracy as the iPar-CLR model is being incrementally updated with new training images. See supplement material for detailed experimental results obtained after adding 500, 1000, 1500 and 2000 images from Helen training set. But perhaps, the most important result is obtained after all the training images from the Helen training set have been incrementally added. From Figure 4, we can see that the performance of iPar-CLR-LFPW-Helen models is slightly better than the performance of the Par-CLR-LFPW-Helen models. This is significant because it shows that not only does iPar-CLR method present a very useful and efficient incremental training procedure but it does this without any loss in the face alignment accuracy.

4.2. Face Tracking in Videos

The goal of this experiment is to investigate the utility of the incremental iPar-CLR method (Section 3.2) to automatically tailor itself to the subject being tracked and become person-specific over time. For this experiment, we use the extremely challenging Youtube Celebrities database [16] which contains videos of celebrities captured *in the wild*. Since this database do not provide facial landmark annotations, we manually annotated nearly 1000 frames containing 6 sequences (Sequence IDs 0292, 0293, 0294, 0502, 0504, 1198) for this experiment.



(a) LFPW Test Set (b) Helen Test Set

Figure 4: Par-CLR vs. iPar-CLR Results.

We use the Par-CLR-LFPW-Helen models (See Section 2.4) as the baseline for this experiment. Moreover, we also use this model to initialize the cascade for iPar-CLR method (Section 3.2) and the distribution $\mathcal{D}_{\text{LFPW}}$, for drawing the perturbation from the incoming new images. All the tracking experiments are conducted under fully-automatic settings, in that the initialization for the first frame is provided by face detector while the subsequent frames are initialized using the fitting from the previous frame.

Another crucial component in an incremental tracking scenario is the tracking failure checker. Since the cascade of regression functions for iPar-CLR-LFPW-Helen model are updated automatically on-the-fly, the aim of this failure checker is to ensure that the update occurs only if the fitting score (that describes the goodness of fit) is higher than the set threshold. For this purpose, we use two separate failure checkers, one at global and another at local level, and the fitting is considered good enough to update the cascades only if the thresholds at both the levels are met. For the global failure checker, we trained an SVM classifier to differentiate between the aligned and misaligned images. For this, we warp the texture from all the LFPW and Helen training images to the canonical mean face using piecewise affine warping [10] to generate the positive samples (i.e. aligned images) and then randomly samples the region around the ground truth to generate the negative samples (i.e. misaligned images). The score from this SVM is used as the criteria to judge the goodness of fit at the global level. For the local failure checker, the trained patch-experts for each facial landmark point, as described in [2, 28], using the LFPW and Helen training images and use the score obtained from each of the patch-experts to judge the goodness

of fit at the local level. Notice in Figure 5(e) for the sequence 0502-0504, the failure checker did not allow for the model to be updated until roughly the first 40 frames as the fittings' scores were below the set threshold.

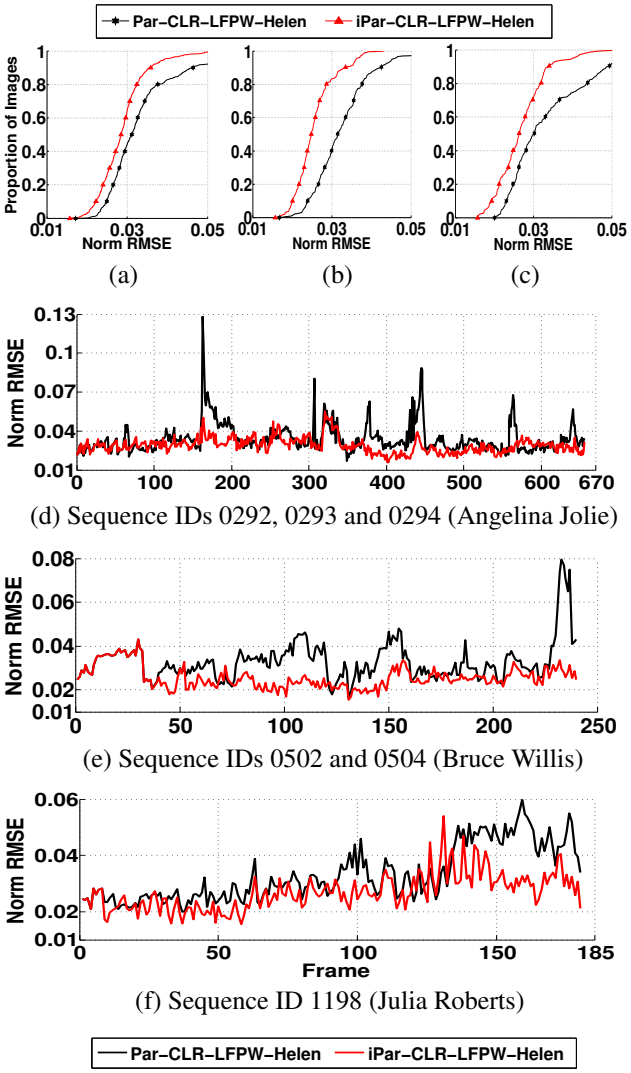


Figure 5: Empirical Results for Face Tracking. (a)-(c) Overall Tracking Results; (a) Sequence IDs 0292, 0293 and 0294 (Angelina Jolie); (b) Sequence IDs 0502 and 0504 (Bruce Willis); (c) Sequence ID 1198 (Julia Roberts). (d)-(e) Frame-by-Frame Comparison of Tracking Results.

From the empirical results in Figure 5 and the qualitative results in Figure 6, including the complete tracking videos[3], we can clearly infer that, in comparison to the generic Par-CLR method, the incremental iPar-CLR method adapts well to the face being tracked over time and shows robustness against occlusion (sequence 0033), fast head movement (sequences 0292-0294), hard shadows (sequences 0502-0504) and head pose variation (sequence 1198). For example, in a challenging sequence 0033 (Adam Sandler)[3], the tracking using the model Par-CLR-LFPW-Helen initially fails for first 88 frames and then it diverges again from frame 144 onwards. However, the iPar-CLR-LFPW-Helen shows more robustness by virtue of its update procedure, as it is able to utilize the frames 89–144 to tailor itself to the subject and the imaging conditions (i.e. occlusion in this case), and do not diverge in the later half of this video. See last two rows in Figure 6 for this sequence. Also, notice stability of iPar-CLR method in the stationary frames of sequence 1198 (Julia Roberts)[3] signifying the robustness of the proposed methodology against over-fitting.

[3] See Supplement Videos for complete tracking results.

Figure 6: Qualitative Face Tracking Results. For each sequence, the top row contains Par-CLR-LFPW-Helen results, and the bottom row contains the corresponding iPar-CLR-LFPW-Helen results.

5. Conclusion

We have proposed an incremental formulation for the discriminative deformable face alignment framework [32] and presented multiple ways for incrementally updating a cascade of regression functions in an efficient manner. Using our current MATALB implementation, the entire procedure (face alignment and model update) takes less than 4 seconds per image, without any parallel processing, on an Intel Xeon 3.80 GHz processor. In the future, we will implement the incremental method in C/CUDA to make it real-time. Also, we will investigate other discriminative methods that will allow the use of incremental updates at local level, say via use of patch-experts [2].

Acknowledgement: The work of Akshay Asthana is funded by Marie Curie Fellowship under FP7-PEOPLE-2011-IIF Grant agreement no. 302836 (FER in the Wild). The work of Shiyang

Cheng and Stefanos Zafeiriou is partially funded by the EPSRC project EP/J017787/1 (4D-FAB).

A. Incremental Linear Least-Squares Problem

Following from Section 3.1, the goal is to find $\tilde{\mathbf{W}}(T+R)$ as a function of **strictly** $\tilde{\mathbf{W}}(T)$, $\mathbf{V}(T)$, $\mathbf{X}(R)$ and $\mathbf{Y}(R)$.

Let, $\mathbf{X}(T+R) = \begin{bmatrix} \mathbf{X}(T) \\ \mathbf{X}(R) \end{bmatrix}$ and $\mathbf{Y}(T+R) = \begin{bmatrix} \mathbf{Y}(T) \\ \mathbf{Y}(R) \end{bmatrix}$.

From Eqn. 5,

$$\begin{aligned} \mathbf{V}(T+R) =& [\mathbf{X}(T+R)^T\mathbf{X}(T+R) + \lambda\mathbf{E}]^{-1} \\ =& \left[\mathbf{X}(T)^T\mathbf{X}(T) + \mathbf{X}(R)^T\mathbf{X}(R) + \lambda\mathbf{E}\right]^{-1} \end{aligned} \quad (12)$$

Using the Woodbury formula [31]:

$$(\mathbf{A}+\mathbf{BDC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{D}^{-1}+\mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1}$$

where, $\mathbf{A} = \left[\mathbf{X}(T)^T\mathbf{X}(T) + \lambda\mathbf{E}\right]$, $\mathbf{B} = \mathbf{X}(R)^T$, $\mathbf{C} = \mathbf{X}(R)$ and $\mathbf{D} = \mathbf{E}$, the term $\mathbf{V}(T+R)$ (Eqn.12) can be re-written as in Eqn 6.

Also, from Eqn. 5, 6, 8, 9:

$$\begin{aligned} \tilde{\mathbf{W}}(T+R) =& \mathbf{V}(T+R)\mathbf{X}(T+R)^T\mathbf{Y}(T+R) \\ =& \mathbf{V}(T+R)\left[\mathbf{X}(T)^T\mathbf{Y}(T) + \mathbf{X}(R)^T\mathbf{Y}(R)\right] \\ =& \mathbf{V}(T)\mathbf{X}(T)^T\mathbf{Y}(T) \\ & - \mathbf{V}(T)\mathbf{X}(R)^T\mathbf{U}\mathbf{X}(R)\mathbf{V}(T)\mathbf{X}(T)^T\mathbf{Y}(T) \\ & + \mathbf{V}(T+R)\mathbf{X}(R)^T\mathbf{Y}(R) \\ \tilde{\mathbf{W}}(T+R) =& \tilde{\mathbf{W}}(T) - \mathbf{Q}\tilde{\mathbf{W}}(T) \\ & + \mathbf{V}(T+R)\mathbf{X}(R)^T\mathbf{Y}(R) \end{aligned} \quad (13)$$

References

[1] A. Asthana, M. de la Hunty, A. Dhall, and R. Goecke. Facial performance transfer via deformable models and parametric correspondence. *IEEE TVCG*, 18(9):1511–1519, 2012. 1

[2] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, 2013. 1, 2, 6, 7

[3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011. 4, 6

[4] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012. 1

[5] S. Cheng, A. Asthana, S. Zafeiriou, J. Shen, and M. Pantic. Real-time generic face tracking in the wild with cuda. In *ACM MMSys*, 2014. 1

[6] S. Chew, P. Lucey, S. Lucey, J. Saragih, J. Cohn, I. Matthews, and S. Sridharan. In the pursuit of effective affective computing: The relationship between features and registration. *IEEE TSMCB*, 42(4):1006–1016, 2012. 1

[7] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006. 1

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[9] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, 2010. 1

[10] G. Edwards, C. Taylor, and T. Cootes. Interpreting Face Images Using Active Appearance Models. In *FG*, 1998. 1, 6

[11] F. D. F. Zhou and J. F. Cohn. Unsupervised discovery of facial events. In *CVPR*, 2010. 1

[12] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific Active Appearance Models. *IVC*, 23(11):1080–1093, 2005. 2

[13] M. Hayes. *Statistical digital signal processing and modeling*. 1996. 5

[14] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970. 3

[15] V. Kazemi and J. Sullivan. Face alignment with part-based modeling. In *BMVC*, 2011. 1, 2

[16] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 2008. 6

[17] V. Le, J. Brandt, Z. Lin, L. D. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012. 4, 6

[18] A. Levey and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE TIP*, 9(8):1371–1374, 2000. 1

[19] X. Liu. Discriminative face alignment. *IEEE PAMI*, 31(11):1941–1954, Nov. 2009. 1

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

[21] I. Matthews and S. Baker. Active Appearance Models Revisited. *IJCV*, 60(2):135–164, Nov. 2004. 1

[22] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE TPAMI*, 26(6):810–815, June 2004. 1, 2

[23] G. Papandreou and P. Maragos. Adaptive and constrained algorithms for inverse compositional active appearance model fitting. In *CVPR*, 2008. 1

[24] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002. 4

[25] 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. 4, 6

[26] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR-W*, 2013. 4, 6

[27] J. Saragih and R. Goecke. Learning AAM fitting through simulation. *PR*, 42(11):2628–2636, 2009. 1

[28] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, Jan. 2011. 1, 2, 6

[29] P. Sauer, T. Cootes, and C. Taylor. Accurate regression procedures for active appearance models. In *BMVC*, 2011. 1

[30] J. Sung and D. Kim. Adaptive active appearance model with incremental learning. *PRL*, 30(4):359 – 367, 2009. 1, 2

[31] M. A. Woodbury. *Inverting Modified Matrices*. Number 42. 1950. 8

[32] Xuehan-Xiong and F. De la Torre. Supervised descent method and its application to face alignment. In *CVPR*, 2013. 1, 2, 3, 7

[33] C. Zhao, W.-K. Cham, and X. Wang. Joint face alignment with a generic deformable face model. In *CVPR*, 2011. 2