Facial Landmark Detection via Progressive Initialization

Shengtao Xiao  Shuicheng Yan  Ashraf A. Kassim
Department of Electrical and Computer Engineering, National University of Singapore
Singapore 117576
xiao_shengtao@u.nus.edu, eleyans@nus.edu.sg, ashraf@nus.edu.sg

Abstract

In this paper, we present a multi-stage regression-based approach for the 300 Videos in-the-Wild (300-VW) Challenge, which progressively initializes the shape from obvious landmarks with strong semantic meanings, e.g., eyes and mouth corners, to landmarks on face contour, eyebrows and nose bridge which have more challenging features. Compared with initialization based on mean shape and multiple random shapes, our proposed progressive initialization can very robustly handle challenging poses. It also guarantees an accurate landmark localization result and shows smooth tracking performance in real-time.

1. Introduction

Face alignment plays a very important role in many computer vision research topics and applications, such as face recognition [20], face synthesis/morphing [11, 9], face detection [6], and 3D face modeling [4]. The accuracy of landmark position localization can directly influence the final performance of these applications. Though many efforts [5, 18, 8] have been devoted during the past decades, facial landmark localization is still very challenging in the cases of large pose variations, abrupt illumination changes, extreme facial expressions and heavy occlusion.

Regression-based facial points detection methods [3, 18, 5] have become very popular recently. Such models basically consist of two steps, feature extraction and regression. Features are extracted around the predicted shape at each cascade stage. The shape update for the current prediction can be easily generated by projecting the features to a learned regression matrix. After a few steps, the estimation error can converge to an arbitrarily small error. A major reason behind the popularity of the regression-based methods is their efficiency. Compared with other methods, e.g., Deformable Part Model [14, 22] and Active Appearance Models [17, 12], the regression-based models can be much faster [5, 18, 13].

Though regression-based models are fast and accurate in most cases, landmark detection is still challenging for faces with large poses and expressions. One major limitation of recent model-based regression methods [5, 18, 13] is that they may be easily trapped by a local optimum if the starting shape is far away from the ground-truth shape. Multiple random initializations [5] can improve the localization performance for simple cases. For faces with large pose, poor illumination or a large expression, multiple random initializations cannot guarantee good performance [21]. Burgos et al. proposed smart initialization in [3] where the set of initial shapes is stopped if its variance of regressed shapes is large and a new set of shapes is randomly selected for regression initialization. While this enables selection of good starting points, but too many uncertainties are introduced due to randomness. The performance enhancement due to smart restart is limited as reported in [3].

In this paper, we present progressive initialization for the facial landmark detection and tracking. The work is based on the observation that obvious points which have very strong discriminative features, e.g., eyes and mouth corners, are usually positioned first with strong confidence...
when we manually annotate these landmarks. Landmarks on the face contour, nose bridge and eyebrows, which are more challenging to be positioned, are localized at last usually with reference to the early defined points. This is actually a simple but very efficient strategy which guides the annotating process and ensures the great efficiency even if the to-be-processed face image is under a challenging condition. These points can be roughly inferred with reference to the early-positioned points. Zhang et al. [19] proposed a deep Convolutional Neural Network-based framework to learn a model which solves multiple objectives, i.e., regression of five facial landmarks, smile detection and sunglasses detection. They have validated that the five facial points detected by their approach can be used for more robust initialization and improve the landmark detection results for more points. Motivated by their findings, we propose to progressively initialize the face shape from a few easy landmarks to challenging landmarks. The major limitation of [19], which is its heavy computational resource requirement, is solved with the popular cascaded regression tree model. Fig. 1 shows some landmark detection results for a few selected face images under challenging conditions, e.g. expressions, poses and occlusion.

The main contributions of this paper are summarized as follows:

- An efficient face tracking approach is proposed.
- A simple but efficient facial landmark tracking approach which guarantees real-time performance in an unconstrained environment is presented.
- Progressive initialization is proposed to ensure a robust initialization for landmark detection of 68 points.

The remainder of the paper is as follows. We provide a review of related work in Section 2 before introducing our progressive initialization algorithm in Section 3. The experimental results are presented and discussed in Section 4 before we conclude the paper in Section 5.

2. Related Work

Regression-based models have been very successful and popular in recent research. In this section, the cascaded regression tree and the local binary feature are briefly reviewed. More details can be found in [13 5 13 3].

2.1. Cascaded Regression Trees

Cascaded regression trees formulate the shape regression for $L$ landmarks into an additive cascade form as follows:

$$s^k = s^{k-1} + \mathcal{M}^k(\Phi^k(I, s^{k-1})), \quad (1)$$

where $s^k \in R^{2L \times 1}$ and $\Phi^k(\cdot)$ are the predicted shape and the feature extractor at the $k$-th cascade stage accordingly. $\mathcal{M}^k(\cdot)$ represents the mapping function which projects the feature extracted from the image $I$ at the predicted shape $\hat{s}^{k-1}$ to the target shape. For cascaded regression trees, this mapping function can be formulated as

$$\mathcal{M}^k(\Phi^k(I, \hat{s}^{k-1})) = \sum_{t=1}^{T} f^k_t(\phi^k_t(I, \hat{s}^{k-1})), \quad (2)$$

where $T$ is the number of trees in the current stage and $f^k_t(\cdot)$ generates shape increment with the given image $I$ and the estimated shape $\hat{s}^{k-1}$, $\phi^k_t(\cdot)$ is the feature extracted from the $t$-th regression tree in the $k$-th cascade stage. For a given input image $I$ and a current shape estimation $\hat{s}^{k-1}$, after passing a few node split tests from the $t$-th regression tree with $L_f$ leaf nodes, a leaf node with index $l_f \in \{1, 2, ..., L_f\}$ will be reached and the corresponding regression output $r^k_t, l_f = f^k_t(\phi^k_t(I, \hat{s}^{k-1}))$ is generated. The feature extracted is simply an $L_f$ bits binary value with the $l_f$-th bit being 1 and the rest bits being 0.

For clarification purposes, the cascade regression process of shape prediction can be further defined as

$$\hat{s}^K = \mathcal{M}^K(I, s^0) \quad (3)$$

where $\mathcal{M}^K$ represents a $K$-stage cascade regression which takes input image $I$ and initial shape $s^0$ and generates output $\hat{s}^K$.

2.2. Local Binary Feature [13]

To learn the structure of the regression trees, Ren et al. [13] divided the process into two steps: 1) learn tree structures locally and 2) learn tree output globally.

Consider a global feature $\Phi^k = [\phi^k_1, \phi^k_2, ..., \phi^k_L]$ where $\phi^k_i$ is the local feature for the $i$-th landmark. $\phi^k_i = [\phi^k_{1, x}, \phi^k_{2, x}, ..., \phi^k_{T_x, x}]$ where $T_x$ is the number of regression trees trained for the landmark $x$. For $T_x$ trees, the structure of the tree is learned locally via optimizing

$$\min_{w^k, \phi^k} \sum_{i=1}^{M} \|\pi_x \circ \hat{s}^k_i - w^k \phi^k\|_2^2, \quad (4)$$

where the global regression target $\hat{s}^k_i$ is defined as $\hat{s}^k_i = s^i - \hat{s}^k$, and $\pi_x$ is an operator which extracts the regression target of the $x$-th landmark. The node split is trained by maximizing the reduced variance of local regression targets, $\pi_x \circ \hat{s}^k$, from all training samples passed into the node.

When the structures of regression trees are fixed, the tree output is learned jointly by solving the regression problem similar as [13] but with a regularization term to prevent overfitting as stated below

$$\min_{W} \sum_{i=1}^{M} \|s^k_i - W \Phi^k\|_2^2 + \lambda_k \|W\|_2^2, \quad (5)$$

where $\lambda_k$ is a weight parameter. The training stage is the same as in [13].
where $M$ is the number of training samples, $\lambda_k$ is the regularization term at the $k$-th stage and $W^k \in \mathbb{R}^{2L \times TD}$ is the regression matrix with $T_D$ being the number of leaf nodes of all regression trees within $k$-th cascade stage. The leaf node output is simply given by the corresponding column vector from the regression matrix $W^k$.

### 3. Progressive Initialization

In this section, we introduce the components of our approach, including progressive initialization for both training and testing. Some techniques used for robust and efficient landmark tracking are also presented. The notations used in the rest of the paper are given in Table 1.

<table>
<thead>
<tr>
<th>Category</th>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalars</td>
<td>$K$</td>
<td>number of cascade stages of a model used</td>
</tr>
<tr>
<td></td>
<td>$M$</td>
<td>number of initializations used</td>
</tr>
<tr>
<td>Vectors</td>
<td>$I$</td>
<td>image window</td>
</tr>
<tr>
<td></td>
<td>$S_n$</td>
<td>$n$ points face shape</td>
</tr>
<tr>
<td></td>
<td>$N_S$</td>
<td>transformation matrix of $S$ to mean shape space</td>
</tr>
<tr>
<td></td>
<td>$D$</td>
<td>distances between prior shape and K-mean centers</td>
</tr>
<tr>
<td></td>
<td>$\hat{S}^0$</td>
<td>initial shapes $\hat{S}^0 = {\hat{S}^0_1, ..., \hat{S}^0_N}$</td>
</tr>
<tr>
<td>Functions</td>
<td>$\Phi_n(I, S_n)$</td>
<td>feature extraction for $S_n$ at $I$</td>
</tr>
<tr>
<td></td>
<td>$M_n(I, \Phi_n)$</td>
<td>mapping function for $n$ points</td>
</tr>
<tr>
<td></td>
<td>$M_n^K(I, S_n^0)$</td>
<td>shape prediction with model $M_n$ and starting shape $S_n^0$</td>
</tr>
<tr>
<td>Symbols</td>
<td>$\hat{x}$</td>
<td>value/vector estimated from ground-truth value/vector</td>
</tr>
</tbody>
</table>

#### 3.1. Normalized K-means Shape Centers for Initialization

All training shapes are first normalized by similarity transformation where a training shape is aligned to the mean shape to minimize their $L_2$ distance given by

$$N_S = \arg \min_{N} \| \hat{S} - N \circ S \|_2,$$  

(6)

where $\hat{S}$ is the mean shape and $N$ consists of rotation and scaling operations. The normalized shapes are then defined by $S_n = N_S \circ S$. K-means shape centers are formed with the normalized shapes. Fig. 3 and Fig. 4 show randomly selected K-means centers of normalized training shapes for 19 points and 68 points, respectively. From these two figures, we observe that all centers have a rotation angle of about zero. This is because all training samples undergo a similarity transformation, i.e., Eqn. 6. More details will be given on how to preserve the rotation information later.

![Figure 3](image-url) K-means centers, $S_{N_{19}}$, for normalized 19-points shapes in the mean shape space. There are 191 K-means centers in total and 49 K-means centers are randomly selected and shown here.

![Figure 4](image-url) K-means centers, $S_{N_{68}}$, for normalized 68-points shapes in the mean shape space. There are 681 K-means centers in total and 49 K-means centers are randomly selected and shown here.

#### 3.2. Guided Initialization with Prior Shape

A prior shape is a predicted face shape but with less points. For instance, $\hat{S}_5$ is the prior shape for 19P shape prediction and $\tilde{S}_{19}$ is the prior shape for 68P shape prediction. Prior shapes provide essential information for initial shape selection. For instance, to select initial shapes for 19P shape prediction, we first calculate the $L_2$ distance from the normalized prior shape $N_{S_5} \circ S_5$ to the 19P K-means centers, i.e.,

$$D_i = \| \mathcal{P}_{19 \rightarrow 5} \circ S_{N_{19}}^i - N_{S_5} \circ S_5 \|_2,$$  

(7)

where $S_{N_{19}} = \{S_{N_{19}}^1, S_{N_{19}}^2, ..., S_{N_{19}}^{N_{19}}\}$ denotes the 19P K-means centers with $N_{19}$ being the number of centers. $\mathcal{P}_{19 \rightarrow 5}$ extracts the 5P landmarks from 19P in a way that $\mathcal{P}_{19 \rightarrow 5} \circ S_{N_{19}}^i$ and the prior shape $S_5$ are within the same landmark space.

The corresponding initial shape for the $i$-th K-means center is defined as

$$S_{i}^{0,j} = N_{S_5}^{-1} \circ S_{N_{19}}^{i}.$$

(8)

Since the distance from the prior shape to K-mean centers is known, i.e., $D_i, i = \{1, 2, ..., N_{19}\}$, initial shapes...
Figure 2. The framework of our progressive initialization consists of three stages. Each stage consists of a shape regressor. Three regressors are trained progressively which predict shapes with 5, 19 and 68 points. The 5 points with strong features, i.e., eyes, mouth corners and nose tip, are predicted first. This predicted 5P shape is then used to guide the initialization process of the 19P face shape predictor. Similarly, the 68P shape predictor uses the predicted 19P shape as reference to select initial shapes.

which are close to the prior shapes can be easily selected by sorting $D = \{D_1, D_2, ..., D_{N_{19}}\}$. Compared to [21], the computational resources required are much lower for searching good initial shapes, as only the distances between the prior shape and a limited number of K-means centers are required. Rotation information can still be preserved by directly applying inverse similarity transformation, i.e., $N_{S_5}^{-1}$, to the selected K-means centers.

The general initial shape selection process can be summarized in Algorithm 1.

3.3. Perturbed Training for Robustness

Robustness of landmark tracking can be achieved in the training process. The key factors such as bounding box variation and bad prior shape initialization are handled explicitly in the training process.

3.3.1 Bounding Box Robustness

To ensure bounding box robustness, bounding box augmentation is performed when training the 5P shape predictor. The center of the detected bounding box is perturbed with

Algorithm 1 Initial Shape Selection from K-means Centers

1: **Input**: Prior shape detected: $\hat{S}_p$, K-means face centers: $S_{N_c} = \{S_{N_c}^1, S_{N_c}^2, ..., S_{N_c}^{N_{19}}\}$. Number of K-means centers: $N_c$. Number of shapes to be selected: $x$. Number of landmarks used in $S_{N_c}$: $c$. Number of landmarks used in $\hat{S}_p$: $p$.
2: **Output**: Initial shapes for next stage shape prediction: $\hat{S}_c^0 = \{\hat{S}_c^0, \hat{S}_c^0, 2, ..., \hat{S}_c^0, x\}$
3: Get $N_{\hat{S}_p}$ with Eqn. (6)
4: for i=1: $S_{N_c}.\text{length}$() 
5:     $D[i] = \|P_{c->p} \circ S_{N_c}^{i} - N_{\hat{S}_p} \circ \hat{S}_p\|_2$
6: end
7: Sort D to get a vector of distance order in ascending form: $I_{dx}$
8: for i=1:x 
9:     $\hat{S}_c^{0,i} = N_{\hat{S}_p}^{-1} \circ S_{N_c}^{I_{dx}[i]}$ # reverse transformation
10: end
a uniformly distributed random translational offset within 10% of the interocular distance calculated from the ground-truth shape. A scaling disturbance is also applied to the bounding box.

### 3.3.2 Initialization Robustness

Each prior shape provides initialization guidance for the face shape predictor of the next stage. In the training process, initial shapes are selected in a way that distant K-mean centers can also be selected. This mimics the situation when prior face landmark detection fails and a bad K-mean center is selected as the starting shape. Algorithm 2 is used to select centers to train a 19P predictor from the 5P shape prior.

#### Algorithm 2 K-means Centers Selection for Shape Initialization in Training Process

1. **Input:** Prior shape detected: \( \hat{S}_5 \). 19P K-means centers: \( S_{N_{19}} = \{ S_{N_{19}}^1, S_{N_{19}}^2, \ldots, S_{N_{19}}^N \} \). Number of K-means centers in \( S_{N_{19}} \): \( N_{19} \). Number of initializations for training: \( x \). K-means center selection sampling rate: \( r \).
2. **Output:** K-means centers selected: \( S_K = \{ S_K^1, S_K^2, \ldots, S_K^N \} \).
3. Get \( N_{S_5} \) with Eqn. (6).
4. For \( i = 1: N_{19} \):
5. \[ D[i] = \| P_{19} \rightarrow 5 \circ S_{N_{19}}^0 - N_{S_5} \circ \hat{S}_5 \|_2 \]
6. End
7. Sort \( D \) to get a vector of distance order in ascending form: \( I_d \).
8. For \( p = \text{rand} > 0.1 \) # random number
9. If \( p \):
10. \( S_K^i = S_{I_d[p]} \)
11. Else
12. \( S_K^i = S_{I_d[N_{19} - (i-1)r]} \)
13. End
14. \# i-th initial shape: \( S_{19}^{0,i} = N_{S_5}^{-1} \circ S_K^i \)

#### 3.4. Shape Prediction via Progressive Initialization

Our approach consists of three stages which predict shapes with 5, 19 and 68 points (the points selected are manually defined at each stage) in a cascaded way. Fig. 2 shows the framework of the proposed approach. The 5P predictor first locates the 5 key points, including eyes, mouth corners and nose tip. These 5 key points are chosen as they have obvious features and used by most face detectors to identify a face. They are relatively more robust to face detectors as compared to other landmarks, e.g. eyebrows and chin. The predicted 5P face shape acts as the prior shape for the 19P landmark detector to help choose initial shapes from the K-means centers. The shape estimated from the 19P predictor then guides the initial shapes selection for the 68P shape regressor. These steps can be summarized as Algorithm 3 in the testing process.

#### Algorithm 3 Shape Prediction via Progressive Initialization

1. **Input:** Image: \( I \); Bounding box: \( B \); models: \( M_{S_{19}}^{K_5}, M_{S_{19}}^{K_19}, M_{S_{68}}^{K_{68}} \); K-means shape centers: \( S_{S_{68}} \), \( S_{S_{68}} \); Previous Estimation: \( \hat{S}_{68} \). Number of initial shapes for 5P, 19P and 68P: \( x_5, x_{19}, x_{68} \).
2. If \( B \) is from face detector
3. \( S_{68}^0 = \hat{S}_{68} \)
4. \( S_{S_{19}}^{K_5} = M_{S_{19}}^{K_5}(I, \hat{S}_{68}^0) \)
5. Else # estimated bounding box from last frame
6. Select initial shapes \( S_{S_{19}}^0 \) from \( S_{S_{68}} \) based on \( \hat{S}_{68} \).
7. \( S_{S_{19}}^{K_5} = \frac{1}{x_{19}} \sum_{i=1}^{x_{19}} M_{S_{19}}^{K_5}(I, \hat{S}_{68}^0) \)
8. Given \( S_{S_{19}}^{K_5} \), select initial shapes \( S_{S_{19}}^0 \) with Algorithm 1.
9. 19P prediction: \( \hat{S}_{19}^{K_19} = \frac{1}{x_{19}} \sum_{i=1}^{x_{19}} M_{S_{19}}^{K_19}(I, \hat{S}_{S_{19}}^{0,i}) \)
10. Given \( \hat{S}_{S_{19}}^{K_19} \), select initial shapes \( S_{S_{68}}^0 \) with Algorithm 1.
11. 68P prediction: \( \hat{S}_{68}^{K_{68}} = \frac{1}{x_{68}} \sum_{i=1}^{x_{68}} M_{S_{68}}^{K_{68}}(I, \hat{S}_{S_{68}}^0) \)
12. End

#### 3.5. Face Detection and Tracking

Although the processing speed of the recent face landmark detection algorithms can be less than 1ms [13], the speed for processing face alignment in videos can hardly reach 1ms in practice. There are a few key factors which limit the face landmark tracking in video sequences, including the video decoding speed and the face detection speed. Loading all frames into memory and processing individual frames are hardly practical due to the huge amount of memory consumption for decoding and storing the decoded video frames. Popular existing OpenCV face detectors, e.g., Viola-Jones-based, can achieve almost real-time face detection on a normal PC. However, the detection performance deteriorates significantly for faces with large poses, slight occlusion and bad light illumination. Yu et al. [11] developed a face detection library which guarantees a satisfactory detection speed and accuracy for challenging cases. How-
ever, the speed decreases dramatically for faces under challenging conditions. To ensure efficiency, face tracking is essential as the computational resources required for face detection make real-time facial landmark tracking almost impossible with most existing algorithms.

Since our framework can easily regress to a very stable face shape, we can use this face shape to estimate a bounding box. The estimated bounding box can then be used as the face bounding box for the next frame under the assumption that the face moves within an arbitrary range. However, there are still cases where landmark tracking fails due to heavy occlusion and scenery boundary. Inspired by smart restart [3], a similar scheme is applied to trigger the face detector when the variance of regression results exceeds a pre-defined threshold. This successfully prevents error from accumulating due to bad bounding box estimation. Meanwhile, the computational resources required for face detection can be significantly reduced.

4. Experiment and Result

Our models are trained with the 300W dataset (LFPW, HELEN, and AFW) and the randomly selected 10% frames from the training videos provided by the event organizer. Standard evaluation measures are adopted in this event. The error of each frame can be calculated as follows:

$$ e_i = \frac{\|S^* - \hat{S}\|}{D_i}, $$

where $D_i$ is the inter-ocular distance determined by outer corners of left and right eyes, i.e. the 37-th and the 46-th point.

4.1. Results on 300-VW Testing Set

Our approach has been evaluated independently by the 300-VW Challenge [15] organizers using their own testing videos which are not disclosed to the participants. The details of annotation process for the training and testing set are presented in [7, 16]. There are 150 testing videos which are divided into three scenarios. Scenario 1 consists of videos taken under well-lit conditions; Scenario 2 contains videos taken in unconstrained conditions without heavy occlusion; Scenario 3 consists of videos recorded under completely unconstrained conditions. The 49 points (excluding points from the face contour) errors are returned together with the baseline performance [2]. For 68 points errors, only the performance of our approach is provided.

Fig. 5 shows that our approach yields much better performance in facial landmark tracking for all categories when compared with the baseline [2]. More than 90% of the testing frames are within 8% point-to-point error for all categories. Our approach has outperformed the baseline for more than 20% in each category.

4.2. Results on Real-time Video

Further evaluation is done to verify our framework on the challenging Youtube Celebrities Database [10] which contains videos of celebrities captured in the wild. Some frames of challenging poses and expressions are shown in Fig. 7. We notice that even in the cases of extreme poses,
our approach can still robustly track the landmarks without using the face detector which is normally computationally expensive. The face detection process will be triggered (e.g., Fig. 7 Row-3 to Row-4) only when the variance (Sec. 3.5) is too large. This mechanism can efficiently help us to prevent error accumulation.

Our approach is computationally efficient. Without much code optimization, our implementation can reach 30+ FPS for landmark tracking on a single core E5-1603 CPU. The speed may be further improved via code optimization and parallel computation. In fact, the current version can be used in most real-time face-related applications.

5. Conclusion

In this paper, we introduced a facial landmark tracking framework which progressively initializes and predicts the face shapes for the 300-VW competition. The proposed method locates simple and easy-to-detect facial landmarks first which are then used to guide the initial shapes selection process for the regressor in later stages. To ensure overall landmark tracking efficiency, our efficient face tracking approach uses the bounding box estimated by landmark prediction from the previous frame. Our method showed significant improvement over the baseline in all three testing scenarios and its real-time performance also makes it possible to be used in many face-related applications.

Currently, the shape at each stage is fixed and manually determined. In the future, a framework which automatically and gradually infers the locations of landmarks from the most obvious landmarks to the most difficult landmarks is to be developed.

References


Figure 7. Qualitative facial landmark tracking results of selected frames from Youtube Celebrities Database.


