

# The One Triangle Three Parallelograms Sampling Strategy and Its Application in Shape Regression

Mikael Nilsson  
Centre of Mathematical Sciences  
Lund University, Lund, Sweden  
mikael.nilsson@math.lth.se

## Abstract

The purpose of this paper is threefold. Firstly, the paper introduces the One Triangle Three Parallelograms (OTTP) sampling strategy, which can be viewed as a way to index pixels from a given shape and image. Secondly, a framework for cascaded shape regression, including the OTTP sampling, is presented. In short, this framework involves binary pixel tests for appearance features combined with shape features followed by a large linear system for each regression stage in the cascade. The proposed solution is found to produce state-of-the-art results on the task of facial landmark estimation. Thirdly, the dependence of accuracy of the landmark predictions and the placement of the mean shape within the detection box is discussed and a method to visualize it is presented.

## 1. Introduction

Registration of shape landmarks in an image is an essential step in many computer vision applications. Early work on shape landmark estimations commonly resorted to the Active Appearance Model (AAM) [8, 4]. More recently, Cascaded Pose Regression (CPR) [7] and Supervised Descent Methods (SDM) [21] have shown state-of-the-art results in shape estimation. Variants of CPR and SDM are currently the focus in state-of-the-art facial landmark estimation frameworks [16, 2, 1, 15, 19]. A key element in these shape regression approaches is that they predict shapes in a cascaded manner. That is, starting with an initial shape, this shape is progressively refined by estimating a shape increment stage-by-stage. At each stage a regressor is learned on some features, these features are often variants of shape-indexed features [10]. The way pixels are indexed can, for example, be by using local coordinates around each landmark [2] or indexing between two landmarks [1]. The learning method in the cascade can involve several methods, for example, gradient boosting, regression trees and linear

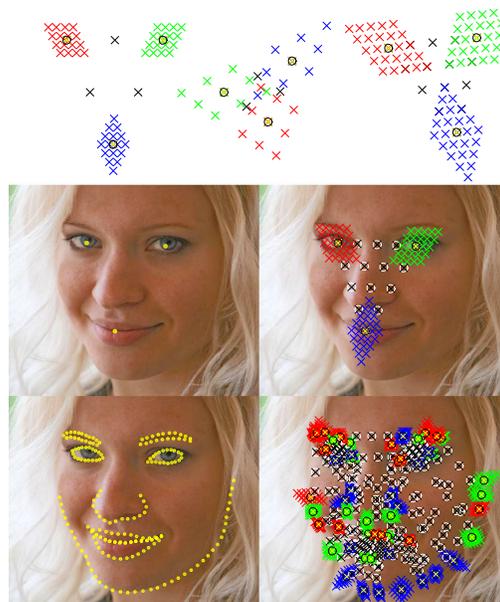


Figure 1. Top row are examples of OTTP shapes. Middle row left is a shape of three points on a face; right and left eye and the mouth, and right presents an OTTP placed on those three points. Bottom row left demonstrates a shape with many points and right are example of OTTPs created and placed on several chosen triplets of points.

regression [19, 15, 1, 21, 14].

In this work, a new way to index pixels in a shape is proposed. Furthermore, a cascaded shape regression framework is presented with novel parts. The general principle in the work builds on ideas found in recent research in shape alignment [7, 16, 2, 21, 12, 1, 15, 19].

The main contributions of this work are:

1. The One Triangle Three Parallelogram (OTTP) sample strategy. This is covered in section 2.
2. A novel cascaded shape regression framework involv-

ing the OTTP sampling, a way of learning spatial locations for binary pixel tests and the use of explicit landmark positions to capture landmark correlations. The framework is presented in section 3.

3. Performance results indicating state-of-the-art results on two facial landmark datasets, see Fig. 4 and Fig. 5, and a way to visualize the effect of shape initialization, see Fig. 9. These results are found in section 4.

## 2. The One Triangle Three Parallelograms Sampling Strategy

The One Triangle Three Parallelograms (OTTP) sampling strategy proposed here is intended to form a way from landmarks on a shape to shape-index features. A shape is here defined as set of landmark points,  $\mathcal{S}_p = [x_p, y_p]$ ,  $p \in 1, 2, \dots, P$ . Shape-indexed features, introduced by Fleuret *et al.* [10], depends on both the image and the current shape estimate. The motivation for the proposed sampling strategy stems from the observation that it is often desirable to extract and capture appearance information *between* as well as context *around* points in a shape. As an example, consider the right and left eye as well as the mouth as thee landmarks on a face, see middle row in Fig. 1. In addition, it is advantageous if this sampling could be performed in a simple and quick manner.

The way a single OTTP is formed is from three selected points in a shape. Lets denote these three points  $\mathcal{A} = \mathcal{S}_i$ ,  $\mathcal{B} = \mathcal{S}_j$  and  $\mathcal{C} = \mathcal{S}_k$  for some chosen  $i, j$  and  $k$ . Three parallelograms are created from the triangle and a given scale  $\alpha$ . The creation and placement for the parallelogram at point  $\mathcal{A}$  is performed by mirroring  $\mathcal{A}$ , creating  $\tilde{\mathcal{A}}$ , on the line between  $\mathcal{B}$  and  $\mathcal{C}$ , now  $\mathcal{A}, \mathcal{B}, \mathcal{C}$  and  $\tilde{\mathcal{A}}$  forms a parallelogram. Further, this parallelogram is centered with its midpoint at the origin, scaled with  $\alpha$  and translated back to have its midpoint at  $\mathcal{A}$ , see Fig. 2. In a similar manner, the parallelograms at  $\mathcal{B}$  and  $\mathcal{C}$  are created.

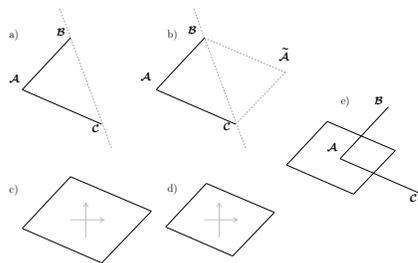


Figure 2. Steps to find the parallelogram at point  $\mathcal{A}$  for an OTTP.

Given the location of the triangle, as well as the three parallelograms, sampling points are calculated using Barycentric coordinates [17]. The sample size is determined by set-

ting samples along the sides of the triangle, denoted  $S_{tri}$  leading to  $N_{tri} = \frac{(S_{tri}+1)S_{tri}}{2}$  sampling points in total for the triangle. Given this  $S_{tri}$ , weights for  $\mathcal{A}, \mathcal{B}$  and  $\mathcal{C}$  can be found as described in Algo. 1. Using these weights, it is now possible to find the sampling points for the triangle as

$$\mathcal{A} \cdot w_{\mathcal{A},tri}(k) + \mathcal{B} \cdot w_{\mathcal{B},tri}(k) + \mathcal{C} \cdot w_{\mathcal{C},tri}(k), \forall k \in 1, \dots, N_{tri}. \quad (1)$$

---

### Algorithm 1 Triangle sample point weights

---

```

Input  $S_{tri}$ 
 $k = 0$ 
1: for  $i = 1, 2, \dots, S_{tri}$  do
2:    $u = \frac{i-1}{S_{tri}-1}$ 
3:   for  $j = 1, 2, \dots, S_{tri}$  do
4:      $v = \frac{j-1}{S_{tri}-1}$ 
5:     if  $u + v \leq 1$  then
6:        $k = k + 1$ 
7:        $w_{\mathcal{A},tri}(k) = u$ 
8:        $w_{\mathcal{B},tri}(k) = v$ 
9:        $w_{\mathcal{C},tri}(k) = 1 - (u + v)$ 
10:    end if
11:  end for
12: end for

```

---

The sampling points for the parallelogram around  $\mathcal{A}$  is determined by setting the number of samples along the sides of the parallelogram, denoted  $S_{par}$  with  $N_{par} = S_{par}^2$  being the number of sampling points, as well as the scale  $\alpha$ . Given this  $S_{par}$  and the scale  $\alpha$ , weights for  $\mathcal{A}, \mathcal{B}$  and  $\mathcal{C}$  can be found as described in Algo. 2. Using these weights, it is now possible to find the sampling points for the parallelogram at  $\mathcal{A}$  as

$$\mathcal{A} \cdot w_{\mathcal{A},par}(k) + \mathcal{B} \cdot w_{\mathcal{B},par}(k) + \mathcal{C} \cdot w_{\mathcal{C},par}(k), \forall k \in 1, \dots, N_{par}. \quad (2)$$

The weights for the parallelogram at  $\mathcal{A}$  can be reused for

---

### Algorithm 2 Parallelogram sample point weights

---

```

Input  $S_{par}, \alpha$ 
 $k = 0$ 
1: for  $i = 1, 2, \dots, S_{par}$  do
2:    $u = \frac{i-1}{S_{par}-1}$ 
3:   for  $j = 1, 2, \dots, S_{par}$  do
4:      $v = \frac{j-1}{S_{par}-1}$ 
5:      $k = k + 1$ 
6:      $w_{\mathcal{A},par}(k) = 1 + \alpha - \alpha u - \alpha v$ 
7:      $w_{\mathcal{B},par}(k) = -\frac{\alpha}{2} + \alpha u$ 
8:      $w_{\mathcal{C},par}(k) = -\frac{\alpha}{2} + \alpha v$ 
9:   end for
10: end for

```

---

parallelogram  $\mathcal{B}$  and  $\mathcal{C}$ . The only important matter is to put

the weights for  $\mathcal{A}$  at the point you want to create the parallelogram. For example, sampling points for the parallelogram around  $\mathcal{B}$  can be found as

$$\mathcal{A} \cdot w_{\mathcal{B},par}(k) + \mathcal{B} \cdot w_{\mathcal{A},par}(k) + \mathcal{C} \cdot w_{\mathcal{C},par}(k), \forall k \in 1, \dots, N_{par}. \quad (3)$$

Similarly, the sampling points for the parallelogram around  $\mathcal{C}$  is found as

$$\mathcal{A} \cdot w_{\mathcal{B},par}(k) + \mathcal{B} \cdot w_{\mathcal{C},par}(k) + \mathcal{C} \cdot w_{\mathcal{A},par}(k), \forall k \in 1, \dots, N_{par}. \quad (4)$$

Given selected scale  $\alpha$ , sample size on the sides of the triangle  $S_{tri}$  and the sample size on the sides of the parallelogram  $S_{par}$ , the weights according to Algo. 1 and Algo. 2 can be pre-calculated. Finding sampling points is only a matter of weighting the triangle points as in Eq. (1), Eq. (2), Eq. (3) and Eq. (4) to get all the sampling points, hence it is a rapid operation. Furthermore, the notation  $OTTP_{\alpha, S_{tri}, S_{par}}$  will indicate the settings used in an OTTP. As an example, the top left OTTP in Fig. 1 is an  $OTTP_{0.25, 3, 4}$ .

### 3. Framework for Cascaded Shape Regression

The shape regression framework predicts a shape  $\mathcal{S}$  in a cascaded manner [7, 19, 21]. Beginning with an initial shape  $\mathcal{S}^{(0)}$ , which here is a mean shape placed in a detection box, a shape increment  $\Delta\mathcal{S}^{(1)}$  is sought to find an improved shape  $\mathcal{S}^{(1)}$ . This operation is then repeated  $T$  times in a cascade manner

$$\mathcal{S}^{(t)} = \mathcal{S}^{(t-1)} + \Delta\mathcal{S}^{(t)}. \quad (5)$$

At each step the goal is to find the ground truth shape  $\mathcal{G}$ , hence ideally  $\Delta\mathcal{S}^{(t)} = \mathcal{G} - \mathcal{S}^{(t-1)}$ .

The way  $\Delta\mathcal{S}^{(t)}$  is calculated here requires the image  $I$ , the previous shape estimate  $\mathcal{S}^{(t-1)}$  and a mean shape from training shapes. The mean shape from training shapes, denoted  $\mathcal{M}$ , is found by generalized Procrustes analysis [13] followed by translating the mean to the origin and uniform scaling.

In order to simplify the regression task, the regression targets are normalized [3]. This normalization implies that a scale  $s$ , rotation matrix  $\mathbf{R}$  and translation  $\mathbf{t}$  is found by the minimization

$$\min_{s, \mathbf{R}, \mathbf{t}} \sum_{p=1}^P \|\mathcal{M}_p - (s\mathbf{R}\mathcal{S}_p^{(t-1)} + \mathbf{t})\|^2 \quad (6)$$

where  $\mathcal{M}_p$  and  $\mathcal{S}_p^{(t-1)}$  are points in the mean shape and current shape, respectively. Only scale and rotation are used to normalize the target, thus regression targets are

$$s\mathbf{R}\Delta\mathcal{S}_p^{(t)} \forall p \in 1, \dots, P. \quad (7)$$

The following subsections will describe the features, learning and prediction method used to find the normalized regression targets and the final shape update  $\Delta\mathcal{S}^{(t)}$  utilizing the OTTP sampling strategy described in section 2.

### 3.1. Appearance Features as Binary Pixel Comparisons

Given  $K$  selected triangle indices for vertices  $V(k) = [v_{\mathcal{A}}(k), v_{\mathcal{B}}(k), v_{\mathcal{C}}(k)]$ ,  $\forall k \in 1, \dots, K$  the image is sampled with OTTPs. The triangle selection should be chosen to cover the appearance in a proper manner. It might be advantageous to have triangles of various sizes and have them cover each other in different parts due to the sparse sampling. Hence, Delaunay triangulation [6] might not be the best choice for this task. Note that there will be  $K$  OTTPs and each OTTP will have

$$N_{OTTP} = N_{tri} + 3N_{par} = \frac{(S_{tri} + 1)S_{tri}}{2} + 3S_{par}^2 \quad (8)$$

sample points, thus in total there are  $K \cdot N_{OTTP}$  sampling points.

The way the features are created involves pixel comparisons in each triangle and parallelogram individually. For example, consider  $N_{tri}$  sample points in a triangle. In order to find feature for pixel  $i$  it is compared to each of the other  $j \neq i$  pixels and a binary feature vector of size  $N_{tri} - 1$  is the result. Similarly, a parallelogram results in a binary feature vector of size  $N_{par} - 1$  for one pixel. Let the matrix  $\mathbf{x}_{bin}$  be such a binary feature matrix for one pixel in either a triangle or parallelogram for every training sample, hence  $\mathbf{x}_{bin}$  will be of size  $N \times (N_{tri} - 1)$  or  $N \times (N_{par} - 1)$  where  $N$  is number of training samples. The aim with these binary features is to find a *spatial configuration* of  $B$ , out of  $N_{tri}$  or  $N_{par}$ , binary tests from each sample point in either a triangle or a parallelogram. Let  $\mathbf{y}_{bin}$  be a vector, size  $N \times 1$ , of regression values from the *mean of the normalized targets*, see Eq. (7), for each training sample. Note that, in general, one could have used different features for each regression target, but this would imply  $2P$  feature extractions in general. Therefore, the mean of all the regression targets are chosen resulting in only one feature extraction.

Given  $\mathbf{x}_{bin}$  and  $\mathbf{y}_{bin}$ , a weight vector  $\mathbf{w}_{bin}$  and a bias  $b_{bin}$  are found by minimizing

$$\min_{\mathbf{w}_{bin}, b_{bin}} \|\mathbf{y}_{bin} - (b_{bin} + \mathbf{x}_{bin}\mathbf{w}_{bin})\|_2^2. \quad (9)$$

Then the absolute value of each of value in  $\mathbf{w}_{bin}$  is taken and these absolute values are sorted from high to low and the first  $B$  indices are kept for future feature calculations. The regression result from this operation is discarded and only the indices indicating the spatial location of the points to be used for comparisons are kept. Furthermore, note that this operation is *repeated for all  $K \cdot N_{OTTP}$  sampling points* resulting in indices of points to compare with for each sampling point used.

With  $B$  found indices for each sampling point, the features form an index in range  $[0, Q - 1]$  where  $Q = 2^B$ . The

appearance features extracted as described are denoted

$$\Phi_a \left( I, \mathcal{S}^{(t-1)}, V \right) \quad (10)$$

and will have a dimension of

$$D_a = K \cdot N_{OTTP} \cdot Q. \quad (11)$$

These feature are sparse since only one value in  $Q$  is active for each of the  $K \cdot N_{OTTP}$  sample points.

This selection of binary tests located within each triangle and parallelogram enforces a form of *locality principle*. Other locality principles has been argued for in previous works. For example, the two step, locality before global, regression proposed by Ren *et al.* [19] or the use of an exponential prior as proposed by Kazemi and Sullivan [15].

The binary features produced here have similarities to randomized ferns [22] and Local Binary Patterns (LBPs) [18]. However, the binary features proposed here differ in that they find the spatial configuration by learning over a larger set of points rather than randomized location of pixel pairs [22] or the use of a predefined spatial configuration [18].

### 3.2. Shape Features to Capture Point Correlations

The appearance features described make use of points on the shape *implicitly*. With occlusions in mind, it may be desirable to make use of the points on the shape *explicitly* as well. The benefit to incorporate a way to capture correlations between points was discusses, and argued for, as a possible future improvement in conclusions of the work of Kazemi and Sullivan [15].

The shape to update,  $\mathcal{S}_p^{(t-1)} = [x_p, y_p] \forall p \in 1, \dots, P$ , is scaled, rotated and translated by the normalization to the mean shape, see Eq. (6), then a normalized shape can be found as

$$\overline{\mathcal{S}}_p^{(t-1)} = s \mathbf{R} \mathcal{S}_p^{(t-1)} + \mathbf{t}, \forall p \in 1, \dots, P. \quad (12)$$

Now the normalized shape  $\overline{\mathcal{S}}_p^{(t-1)} = [\overline{x}_p, \overline{y}_p], \forall p \in 1, \dots, P$  will be used to create  $2P$  shape features as

$$\Phi_s \left( \mathcal{S}^{(t-1)}, \mathcal{M} \right) = [\overline{x}_1, \overline{y}_1, \overline{x}_2, \overline{y}_2, \dots, \overline{x}_P, \overline{y}_P]. \quad (13)$$

Thus, the dimension of shape features will be

$$D_s = 2P. \quad (14)$$

### 3.3. Learning Linear Regression on Appearance and Shape Features

Given appearance features, see Eq. (10), and shape features, see Eq. (13), concatenated features

$$\Phi \left( I, \mathcal{S}^{(t-1)}, \mathcal{M}, V \right) = [\Phi_a \left( I, \mathcal{S}^{(t-1)}, V \right), \Phi_s \left( \mathcal{S}^{(t-1)}, \mathcal{M} \right)] \quad (15)$$

are calculated at each stage in the cascaded regression. In each iteration  $t$  in the cascade, consider  $\mathbf{x}$  to be a matrix of size  $N \times D_x$ , where  $N$  is number of training samples and each training sample is found according to Eq. (15) resulting in a feature dimension of size  $D_x = D_a + D_s$ . Let  $\mathbf{y}$  be a matrix of size  $N \times D_y$ , where each training sample, and its ground truth shape, produces a row in  $\mathbf{y}$  according to Eq. (7) resulting in  $D_y = 2P$  regression values to find. Let  $\mathbf{y}_{\bullet,j}$  be the  $j$ :th column in  $\mathbf{y}$ , then there are  $j = 1, \dots, D_y$  linear regressions to be found by minimizing

$$\min_{\mathbf{w}_{\bullet,j}, b_j} \frac{1}{N} \|\mathbf{y}_{\bullet,j} - (b_j + \mathbf{x} \mathbf{w}_{\bullet,j})\|_2^2 + \lambda_1 \|\mathbf{w}_{\bullet,j}\|_1 + \lambda_2 \|\mathbf{w}_{\bullet,j}\|_2^2, \forall j = 1, \dots, D_y \quad (16)$$

where  $\mathbf{w}_{\bullet,j}$  is the  $j$ :th column in the weight matrix  $\mathbf{w}$  of size  $D_x \times D_y$  and  $\mathbf{b}$  is vector with biases of size  $1 \times D_y$ . Note that an *elastic net* regularization is applied. Regularization is essential since the feature dimension is typically very high and overfitting is an issue without it. Note that the data involves sparse appearance features as well as dense shape features and elastic net regularization, for this reason a special coordinate solver for this problem was implemented. In general, this implementation follows principles found in works of Fan *et al.* [9] and Friedman *et al.* [11].

Given the  $T$  regressors,  $\mathbf{w}^{(t)}$  and  $\mathbf{b}^{(t)}$ , the values for the shape update  $\Delta \mathcal{S}^{(t)}$  is found at each iteration, recall Eq. (5) and Eq. (7), as

$$s^{-1} \mathbf{R}^{-1} \left( \Phi \left( I, \mathcal{S}^{(t-1)}, \mathcal{M}, V \right) \mathbf{w}^{(t)} + \mathbf{b}^{(t)} \right) \quad (17)$$

where  $s$  and  $\mathbf{R}$  are found by the minimization in Eq. (6) at each stage.

## 4. Experiments

Experimental results are reported on two highly challenging facial landmark annotated datasets. The first is HELEN [16] which consists of 2330 annotated images, 2000 training images and 330 testing images. This dataset is annotated with 194 landmarks and images are of high resolution. The second is Caltech Occluded Faces in the Wild (COFW) [1] which consists of 1852 annotated images, 1345 training images and 507 testing images. This dataset is annotated with 29 landmarks and was made to contain heavy occlusions and large shape variations.

The error reported is the average normalized distance of each landmark to its ground truth position normalized by the Inter Ocular Distance (IOD) from the ground truth shape. This error is reported in percent. An alignment resulting in error above 10% is considered as a failure, this rule has been adopted in other works as well [1, 5].

The system presented is fast and runs in a few milliseconds, depending on the number of landmarks and settings.

Parameter	Value
$T$	10
$B$	5
$S_{tri}$	6
$S_{par}$	6
$\alpha_t$	$1.1 - 0.1t$

Table 1. Parameters used in all experiments.

Parameter	HELEN	COFW
$P$	194	29
$K$	35	15
$D_a$	144480	61920
$D_s$	388	58
$D_x$	144868	61978
$D_y$	388	58

Table 2. Parameters and dimensions for the datasets HELEN and COFW.

Similarly to the work of Ren *et al.* [19], regressing on the sparse binary features that make up the appearance features can be implemented by an efficient look up table and vector addition. Calculating the transform in Eq. (6) at the test time is another factor affecting the processing time in the same manner as in other works [2, 15]. An additional factor in this work is the extra processing time related to the shape features as in Eq. (13). While inclusion of the shape features will require additional processing time, the result from Eq. (6) is reused and the overhead is small in comparison to the processing times for the appearance regression part and finding the transformation.

In experiments, a face detector [20] is applied and the bounding box for the face is used to place the mean shape in accordance to this box. In the cases where the face detector fails, a manual bounding box is inserted. Generic settings used for both datasets can be found in Table 1. Note that the OTTP scale shrinks as the stages in the cascade progresses. For the HELEN dataset, 35 triangles are selected and for COFW, 15 triangles are chosen. Given the generic settings and the number of selected triangles, the parameters and dimensions for the respective datasets can be found in Table 2. For each training image, 40 initializations are used in order to train the model. These initializations are created by various scaled, rotated and translated mean shapes as well as randomly selected shapes from all the training shapes. Given the trained models at every stage in the cascade, a final shape can be predicted on a test image, see Fig. 3.

Results and comparisons to results stated in other works on the HELEN dataset can be found in Fig. 4. Note that the result on HELEN is on par with the state-of-the-art. While HELEN is challenging due to the large amount of landmarks, COFW is challenging due to the larger amount of

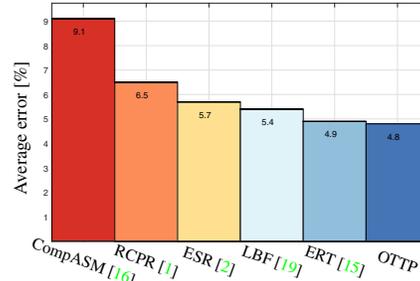


Figure 4. Average error results on the HELEN database.

occlusions. Comparisons and results on the COFW dataset can be found in Fig. 5. Additionally, similar to the work of Burgos-Artizzu *et al.* [1], the failures on the test set is reported in Fig. 6. As results on COFW indicate, the pro-

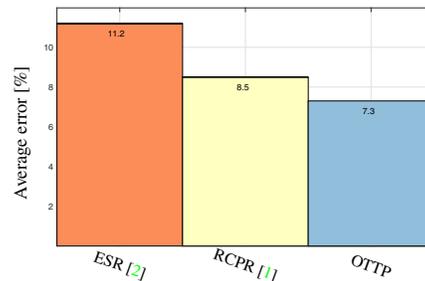


Figure 5. Average error results on the COFW database.

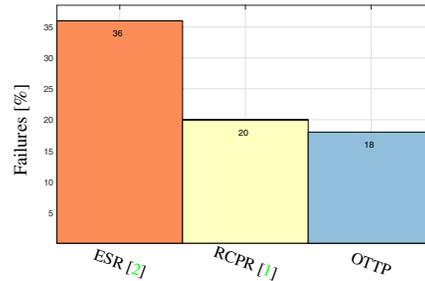


Figure 6. Failures on the COFW database.

posed method copes with occlusions fairly well. The reason for this could be various factors, such as the OTTP sampling, or the design of features, or the use of explicit shape features. Here the effect of the use of explicit shape features is investigated. Furthermore, as mentioned earlier, the triangles used are here manually selected and a comparison to Delaunay triangulation would be in place. The Delaunay triangulation on the mean shape results in 49 triangles, compared to the 15 selected ones, on the 29 landmarks in COFW. Results on COFW using Delaunay triangulation, omitting shape features, only triangles, and only parallelograms can be found in Fig. 7. This result opens the question on how to select the triangles, since the same results

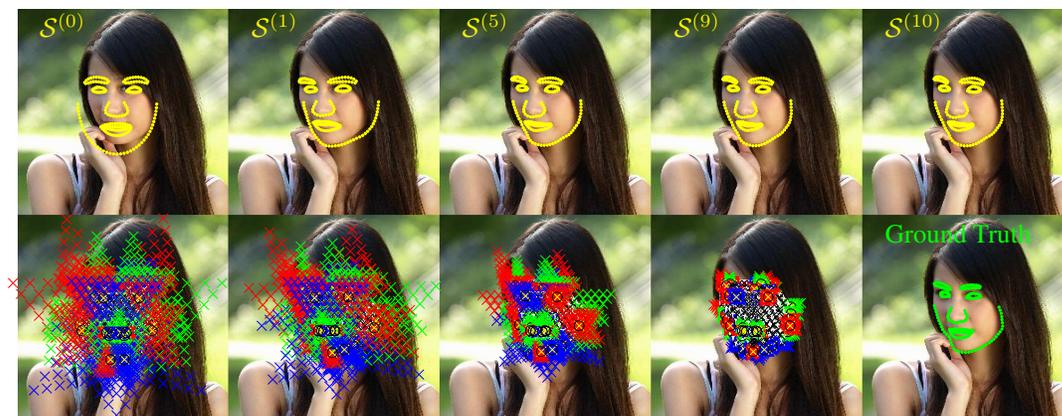


Figure 3. Shape estimates at different stages in the cascade. Top left is the initialization with mean shape and top right is the final estimate. Bottom shows the OTP sampling used for the shape above, note how the parallelograms shrink as stages progresses, and bottom right is the ground truth shape.

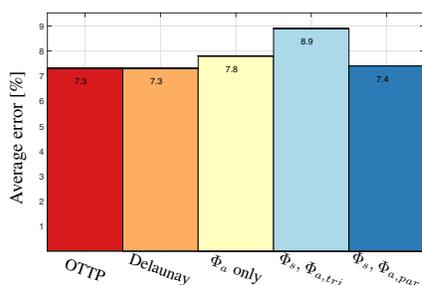


Figure 7. Comparison on the COFW dataset of using OTTP, with manually chosen triangles versus using Delaunay triangulation, omitting shape features, only triangles, and only parallelograms in the cascaded regression.

with only 15 manually selected triangles can be achieved compared to the 49 produced by Delaunay triangulation. A more systematic way to select them could possibly be found. Furthermore, the explicit shape features seem to complement the shape indexed appearance features and give an improvement. Using only parallelograms, which capture features around the landmarks, shows better results than using only triangles, which captures features between landmarks. However, using both gives slightly better performance.

#### 4.1. Face Detection Bounding Box and the Mean Shape Placement

In some images, the face detector fails and a manual box is put in. In general, the detection fails to large rotations of the head or severe occlusions. Hence, omitting the missed detections would simplify the landmark regression task. Using the COFW dataset, this statement can be justified. The proposed method results in 7.3% error with all test images and it would improve to 6.8% if the missed face detections was not accounted for.

While the face detection box is one factor, another is the placement of the initial shape. In some works, different schemes of multiple initializations and restarts have been explored [2, 1]. Initialization of the landmarks here uses one single mean shape placed in the face detection box, this approach has been used in other works as well [19, 15]. It is a known fact that pose regression frameworks typically require a good shape initialization provided by a face detector to accurately locate landmarks [12]. The way to place the mean shape in the face detection bounding box is here performed by a similarity map of the mean shape eye positions to a fixed position of the left and right eye inside the detection box. A way to control the insertion of the mean shape, with the eyes aligned, is to use two ratios,  $f_x$  and  $f_y$ , related to the bounding box width  $W$  and height  $H$  as

$$\begin{aligned} f_x &= \frac{\Delta x}{W} \\ f_y &= \frac{\Delta y}{H} \end{aligned} \quad (18)$$

where  $\Delta x$  and  $\Delta y$  are distances, see Fig. 8. All the results

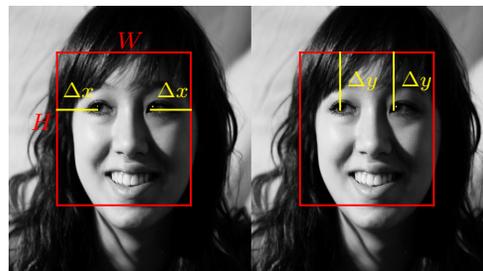


Figure 8. Positioning of the mean shape eyes in the face bounding box.

reported in this paper used  $f_x = 0.3$  and  $f_y = 0.375$ . Analyzing the effect of the initialization could be performed by elaborating on choices of  $f_x$  and  $f_y$ , for test results on

COFW see Fig. 9. This way, an objective visualization

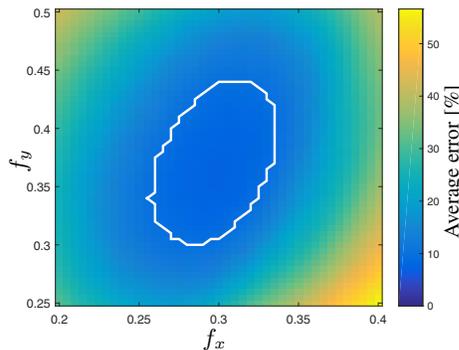


Figure 9. Average errors for different choices of  $f_x$  and  $f_y$  on the test set of the COFW database. White contour indicates the area in which the average error is below 10%.

of the performance related to the initialization can be presented. Research on methods that leads to an expansion of the area indicated in Fig. 9 should be encouraged.

## 5. Conclusion

The proposed One Triangle Three Parallelogram (OTTP) sampling strategy was presented. This sampling strategy was further included in a framework for cascaded shape regression. This framework included binary features created from the OTTPs, inclusion of explicit shape features and a large linear regression with elastic net regularization to perform the regression at each stage in the cascade. The proposed framework resulted in state-of-the-art results on the two tested facial landmarks datasets, HELEN and COFW. Additionally, it was shown that the explicit shape features complement the appearance shape index features and that the way to chose triangles is important and should get more attention. Finally, a way to visualize the effect of shape initialization was presented.

## References

- [1] X. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, 2013. 1, 4, 5, 6
- [2] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, pages 2887–2894. IEEE, 2012. 1, 5, 6
- [3] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 2013. 3
- [4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *PAMI*, 23(6):681–685, June 2001. 1
- [5] M. Dantone, J. Gall, G. Fanelli, and L. J. V. Gool. Real-time facial feature detection using conditional regression forests. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2578–2585. IEEE, 2012. 4
- [6] B. Delaunay. Sur la sphère vide. a la mémoire de georges voronoi. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, pages 793–800, 1934. 3
- [7] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, pages 1078–1085. IEEE, 2010. 1, 3
- [8] G. Edwards, T. Cootes, and C. Taylor. Advances in active appearance models. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 137–142 vol.1, 1999. 1
- [9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 4
- [10] F. Fleuret and D. Geman. Stationary features and cat detection. *Journal of Machine Learning Research (JMLR)*, 9:2549–2578, 2008. 1, 2
- [11] J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2 2010. 4
- [12] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1899–1906, 2014. 1, 6
- [13] J. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 3
- [14] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009. 1
- [15] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014. 1, 4, 5, 6
- [16] V. Le, J. Brandt, Z. Lin, L. D. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV (3)*, volume 7574 of *Lecture Notes in Computer Science*, pages 679–692. Springer, 2012. 1, 4, 5
- [17] A. F. Möbius. Der barycentrische calcul. In *Gesammelte Werke*. Johann Ambrosius Barth, Leipzig, 1827. 2
- [18] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, January 1996. 4
- [19] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1, 3, 4, 5, 6
- [20] P. A. Viola and M. J. Jones. Robust real-time face detection. In *ICCV*, page 747, 2001. 5
- [21] X. Xiong and F. De la Torre Frade. Supervised descent method and its applications to face alignment. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, May 2013. 1, 3
- [22] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(3):448–461, March 2010. 4



Figure 10. Examples of OTTP shape regression on the HELEN dataset.

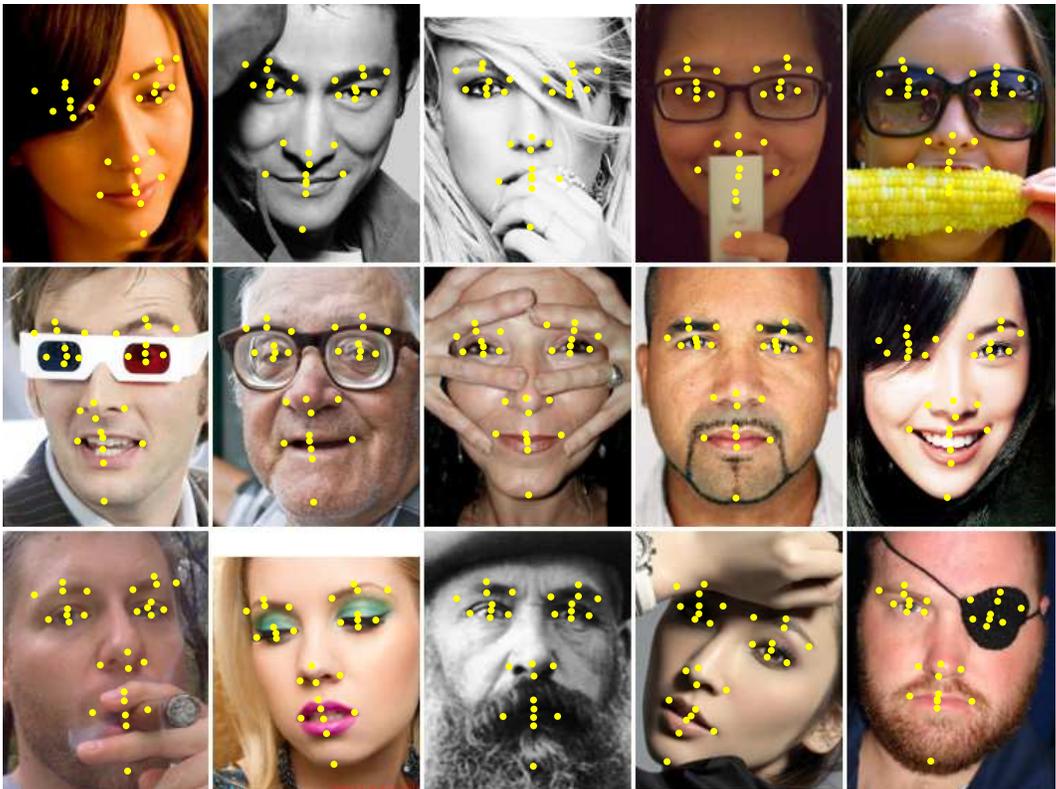


Figure 11. Examples of OTTP shape regression on the COFW dataset.