Face Alignment using Cascade Gaussian Process Regression Trees

Donghoon Lee Hyunsin Park Chang D. Yoo
Korea Advanced institute of Science and Technology
291 Daehak-ro, Yuseong-gu, Daejeon, Korea
{iamdh, hs.park, cd_yoo}@kaist.ac.kr

Abstract

In this paper, we propose a face alignment method that uses cascade Gaussian process regression trees (cGPRT) constructed by combining Gaussian process regression trees (GPRT) in a cascade stage-wise manner. Here, GPRT is a Gaussian process with a kernel defined by a set of trees. The kernel measures the similarity between two inputs as the number of trees where the two inputs fall in the same leaves. Without increasing prediction time, the prediction of cGPRT can be performed in the same framework as the cascade regression trees (CRT) but with better generalization. Features for GPRT are designed using shape-indexed difference of Gaussian (DoG) filter responses sampled from local retinal patterns to increase stability and to attain robustness against geometric variances. Compared with the previous CRT-based face alignment methods that have shown state-of-the-art performances, cGPRT using shape-indexed DoG features performed best on the HELEN and 300-W datasets which are the most challenging dataset today.

1. Introduction

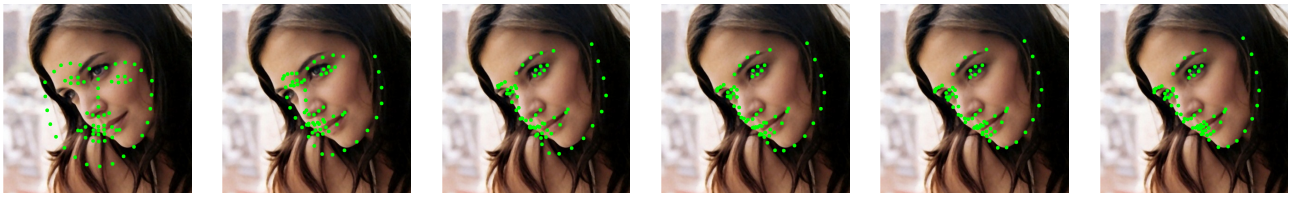
Face alignment is a task to locate fiducial facial landmark points, such as eye corners, nose tip, mouth corners, and chin, in a face image. Accurate and robust face alignment is conducive in achieving the goals of various applications involving a face, such as face recognition [3, 21], facial expression recognition [7], face synthesis [22], and age estimation [11].

Shape regression has become an accurate, robust, and fast framework for face alignment [4, 5, 9, 13, 17]. In shape regression, face shape $\mathbf{s} = (x_1, y_1, \cdots, x_p, y_p)^\top$, that is a concatenation of p facial landmark coordinates $\{(x_i, y_i)\}_{i=1}^{p}$, is initialized and iteratively updated through a cascade regression trees (CRT) as shown in Figure 1. Each tree estimates the shape increment from the current shape estimate, and the final shape estimate is given by a cumulated sum of the outputs of the trees to the initial estimate. The two key elements of shape regression that impact to the prediction performance are gradient boosting [10] for learning the CRT and the shape-indexed features [5] which the trees are based.

The CRT learned through gradient boosting generally exhibits overfitting [10, 13]. In gradient boosting, each stage iteratively fits training data in a greedy stage-wise manner by reducing the regression residuals that are defined as the differences between the ground truth shapes and shape estimates. Overfitting occurs when there is a discrepancy between the fitting rates during learning and prediction. Fitting the training data too quickly within a few stages, which often happened without regularization, can lead to poor generalization and inaccurate shape estimations during prediction.

Overfitting is even more critical when using the shape-indexed features [5, 13, 17] which are closely coupled with the shape estimate: the shape estimate is determined by the shape-indexed features, and the shape-indexed features are extracted from the pixel coordinates referenced by the shape estimate. A discrepancy between the fitting rates lead to irrelevant shape-indexed features to be extracted during prediction which in turn leads even more irrelevant features to be extracted.

Various regularization methods have been considered in shape regression to reduce overfitting and to attain better generalization. Cao *et al*. [5] augmented training data by generating multiple initial shape estimates for one face image, and this data augmentation method has been adopted in subsequent studies [13, 17]. Kazemi and Sullivan [13] considered shrinkage and averaging as regularization methods: in the gradient boosting learning procedure, a learning rate parameter $0 < \nu < 1$ is multiplied to each regression tree (shrinkage) or multiple trees are individually learned and averaged (averaging). Ren *et al*. [17] split up the learning procedure into two steps: (1) learning binary mapping function and (2) learning linear regression matrix. The binary mapping function consists of a set of local binary mapping functions that are induced from independently learned trees using a single facial landmark point. The linear regression



(a) (b) (c) (d) (e) (f)

Figure 1. A selected prediction result on the 300-W dataset using cGPRT. The shape estimate is initialized and iteratively updated through a cascade of regression trees: (a) initial shape estimate, (b)–(f) shape estimates at different stages of cGPRT.

matrix is then learned by minimizing the squared loss function with l_2 regularization, known as Ridge regression [12].

Instead of using gradient boosting, we propose cascade Gaussian process regression trees (cGPRT) that can be incorporated as a learning method for a CRT prediction framework. Gaussian process regression (GPR) is known to give good generalization [16] but high computational complexity. By using a special kernel leading to low computational complexity in prediction, cGPRT provides good generalization compared with the CRT within the same prediction time. The proposed cGPRT is formed by a cascade of Gaussian process regression trees (GPRT), and each GPRT considers a kernel function that is defined by a set of trees. The kernel measures the similarity between two inputs based on the number of trees where the two inputs fall in the same leaves. The predictive mean of cGPRT can be computed as the summation of outputs of trees, and this provides the same computation time in prediction but with better generalization. Here, the predictive mean of cGPRT is designed to be proportional to the product of predictive variables from a set of GPRTs, and this explicitly leads to a greedy stage-wise learning method for cGPRT.

Input features to cGPRT are designed through shape-indexed difference of Gaussian (DoG) features computed on local retinal patterns [1] referenced by shape estimates. The shape-indexed DoG features are extracted in three steps: (1) smoothing face images with Gaussian filters at various scales to reduce noise sensitivity, (2) extracting pixel values from Gaussian-smoothed face images indexed by local retinal sampling patterns, shape estimates, and smoothing scales, and (3) computing the differences of extracted pixel values. Smoothing scale of each local retinal sampling point is determined to be proportional to the distance between the sampling point and the center point. Thus, distant sampling points cover larger regions than nearby sampling points, and this leads to increasing stability of the distant sampling points against to shape estimate errors, while the nearby sampling points are more discriminative with an accurate shape estimate. In a learning procedure of cGPRT, this trade-off allows for each stage to select reliable features based on the current shape estimate errors.

The remainder of the paper is organized as follows: Section 2 briefly reviews the CRT and describes the details of the proposed method. The experimental and comparative results are reported in Section 3. The conclusions are presented in Section 4.

2. Method

In Section 2.1, the CRT for shape regression is briefly reviewed to make the paper self-contained. Then, the details of the proposed cGPRT and the shape-indexed DoG features are described in Section 2.2 and 2.3, respectively.

2.1. Cascade regression trees

The CRT considers a set of T trees and formulates the shape regression as an additive cascade form of trees as follows:

$$\hat{\mathbf{s}}^T = \hat{\mathbf{s}}^0 + \sum_{t=1}^{T} f^t(\mathbf{x}^t; \boldsymbol{\theta}^t), \tag{1}$$

where t is an index that denotes the stage, $\hat{\mathbf{s}}^t$ is a shape estimate, $\mathbf{x}^t$ is a feature vector that is extracted from an input image I, and $f^t(\cdot;\cdot)$ is a tree that is parameterized by $\boldsymbol{\theta}^t$. Starting from the rough initial shape estimate $\hat{\mathbf{s}}^0$, each stage iteratively updates the shape estimate by $\hat{\mathbf{s}}^t = \hat{\mathbf{s}}^{t-1} + f^t(\mathbf{x}^t; \boldsymbol{\theta}^t)$.

Given training samples $\mathbf{S} = (\mathbf{s}_1, \cdots, \mathbf{s}_N)^\top$ and $\mathbf{X}^t = (\mathbf{x}_1^t, \cdots, \mathbf{x}_N^t)^\top$, the trees are learned in a greedy stage-wise manner to minimize the squared loss using regression residuals as follows:

$$\boldsymbol{\theta}^t = \underset{\boldsymbol{\theta}^*}{\operatorname{argmin}} \sum_{i=1}^{N} ||\mathbf{r}_i^t - f^t(\mathbf{x}^t; \boldsymbol{\theta}^*)||_2^2. \tag{2}$$

Here, the regression residual is given by $\mathbf{r}_i^t = \mathbf{s}_i - \hat{\mathbf{s}}_i^{t-1}$.

The tree parameter $\boldsymbol{\theta}^t$ consists of a split function $\tau^t(\mathbf{x}^t)$ and regression outputs $\{\bar{\mathbf{r}}^{t,b}\}_1^B$. The split function takes an input $\mathbf{x}^t$ and computes the leaf index $b \in \{1, \cdots, B\}$, and each regression output is associated with the corresponding leaf index b. The optimal regression outputs are obtained by averaging the regression residuals over all training data

points falling in the corresponding leaf:

$$\bar{\mathbf{r}}^{t,b} = \frac{1}{N^{t,b}} \sum_{i:\tau^t(\mathbf{x}_i^t)=b} \mathbf{r}_i^t, \tag{3}$$

where $N^{t,b}$ is the number of training data points that fall in leaf b. Now, Equation (1) can be re-written using the split function and regression outputs by $\hat{\mathbf{s}}^t = \hat{\mathbf{s}}^{t-1} + \bar{\mathbf{r}}^{t,\tau^t(\mathbf{x}^t)}$.

2.2. Cascade GPRT

The proposed cGPRT is formed by a cascade of GPRTs, and each GPRT considers a kernel function that is defined by a set of trees. In the following, the details of GPRT and cGPRT are described with a brief review on GPR. For the details of GPR, we refer to readers to [16].

Gaussian process regression trees In GPR, the relationship between inputs and outputs is modeled by a regression function $f(\mathbf{x})$ drawn from a Gaussian process with independent additive noise ε_i,

$$\begin{aligned} s_i &= f(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \cdots, N, && (4)\\ f(\mathbf{x}) &\sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')), && (5)\\ \varepsilon_i &\sim \mathcal{N}(0, \sigma_n^2). && (6) \end{aligned}$$

Given a test input $\mathbf{x}_*$, distribution over its predictive variable $\mathbf{f}_*$ is given as

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{S}) &= \mathcal{N}(\mathbf{f}_*|\bar{\mathbf{f}}_*, \sigma_*^2), && (7)\\ \bar{\mathbf{f}}_* &= \mathbf{k}_*^\top \mathbf{K}_s^{-1} \mathbf{S}, && (8)\\ \sigma_*^2 &= k_* - \mathbf{k}_*^\top \mathbf{K}_s^{-1} \mathbf{k}_*, && (9) \end{aligned}$$

where k_* and $\mathbf{k}_*$ are $k(\mathbf{x}_*, \mathbf{x}_*)$ and covariance vector between $\mathbf{x}_*$ and $\mathbf{X}$, respectively. Here, $\mathbf{K}_s$ is given by $\mathbf{K} + \sigma_n^2 \mathbf{I}_N$, and $\mathbf{K}$ is a covariance matrix of which $\mathbf{K}(i, j)$ is computed from the i-th and j-th row vector of $\mathbf{X}$. The predictive mean can also be written as a liner combination of N kernels as

$$\bar{\mathbf{f}}_* = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*), \tag{10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_N)^\top$ is given by $\mathbf{K}_s^{-1}\mathbf{S}$.

A kernel $k(\mathbf{x}, \mathbf{x}')$ in GPRT is defined by a set of M number of trees in a similar manner in [8]:

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \sigma_k^2 \sum_{m=1}^{M} \kappa^m(\mathbf{x}, \mathbf{x}'), && (11)\\ \kappa^m(\mathbf{x}, \mathbf{x}') &= \begin{cases} 1 & \text{if } \tau^m(\mathbf{x}) = \tau^m(\mathbf{x}') \\ 0 & \text{otherwise,} \end{cases} && (12) \end{aligned}$$

where σ_k^2 is the scaling parameter that represents the kernel power. This kernel computes the similarity of two inputs based on counting the number of trees in which the two inputs fall into the same leaf over trees.

Note that the method to learn split functions $\tau(\cdot)$ and the method to extract features $\mathbf{x}$ will be described in Section 2.3.

Optimization of GPRT Hyper-parameters of GPRT, σ_k^2 and σ_n^2, can be estimated by a gradient-based optimization method on log marginal likelihood:

$$\log p(\mathbf{S}|\mathbf{X}, \sigma_k^2, \sigma_n^2) = -\frac{1}{2}\mathbf{S}^\top \mathbf{K}_s^{-1}\mathbf{S} - \frac{1}{2}\log|\mathbf{K}_s| - \frac{n}{2}\log 2\pi. \tag{13}$$

Without loss of generality, the hyper-parameters σ_k^2 and σ_n^2 can be replaced by σ_k^2 and $\sigma_r^2 = \frac{\sigma_n^2}{\sigma_k^2}$. To set the σ_r^2 by maximizing the log marginal likelihood, we seek the partial derivatives with respect to σ_r:

$$\frac{\partial}{\partial \sigma_r} \log p(\mathbf{S}|\mathbf{X}, \sigma_k^2, \sigma_r^2) = \frac{1}{2}\mathrm{tr}\big((\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}_s^{-1})\frac{\partial \mathbf{K}_s}{\partial \sigma_r}\big). \tag{14}$$

The computational burden in Equations (13) and (14) is to compute $\mathbf{K}_s^{-1}$ and $\log|\mathbf{K}_s|$ which is in $O(N^3)$. However, the inverse can be computed efficiently because the rank of $\mathbf{K}$ is in maximumly the number of leaves over trees BM. Let $\mathbf{q}_i = (\mathbf{q}_i^1, \cdots, \mathbf{q}_i^M)^\top$ and let $\mathbf{q}_i^m$ be the *one-of-B* coding vector that indexes the leaf node of the m-th tree that the i-th training data point falls in. Then $\mathbf{K} = \sigma_k^2 \mathbf{Q}\mathbf{Q}^\top$, where $\mathbf{Q} = (\mathbf{q}_i, \cdots, \mathbf{q}_N)^\top$. From this, we obtain

$$\mathbf{K}_s^{-1} = \sigma_k^{-2}\Big(\sigma_r^{-2}\mathbf{I}_N - \sigma_r^{-2}\mathbf{Q}\mathbf{K}_r^{-1}\mathbf{Q}^\top\Big), \tag{15}$$

$$\log|\mathbf{K}_s| = N\log\sigma_k^2 + (N - BM)\log\sigma_r^2 + \log|\mathbf{K}_r|, \tag{16}$$

in which the computation of inverse is in $O\big((BM)^3\big)$. Here, $\mathbf{K}_r = \mathbf{Q}^\top\mathbf{Q} + \sigma_r^2\mathbf{I}_{BM}$ is $BM \times BM$ matrix.

When σ_r^2 is estimated, σ_k^2 can be estimated in a closed form as follows:

$$\sigma_k^2 = \frac{\mathbf{S}^\top(\sigma_r^{-2}\mathbf{I}_N - \sigma_r^{-2}\mathbf{Q}\mathbf{K}_r^{-1}\mathbf{Q}^\top)\mathbf{S}}{N}. \tag{17}$$

Prediction of GPRT In GPRT, predictive variable $\mathbf{f}_*$ of the input $\mathbf{x}_*$ is a Gaussian random variable with the predictive mean and variance given in Equations (10) and (9), respectively. Computation of Equation (10) is in $O(N)$; however, this can be more efficient as follows:

$$\begin{aligned} \bar{\mathbf{f}}_* &= \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*) && (18)\\ &= \sum_{m=1}^{M}\sum_{i=1}^{N} \alpha_i \sigma_k^2 \kappa^m(\mathbf{x}_i, \mathbf{x}_*) && (19)\\ &= \sum_{m=1}^{M} \bar{\alpha}^{m,\tau^m(\mathbf{x}_*)}, && (20) \end{aligned}$$

Algorithm 1 Greedy stage-wise learning of cGPRT.

Input: training data $\{s_i, I_i\}_{i=1}^N$
Output: cGPRT parameters for prediction $\{\bar{\boldsymbol{\alpha}}^t, \tau^t\}_{t=1}^T$
Procedure:

1: Initialize $\hat{s}_1^0, \cdots, \hat{s}_N^0$
2: **for** $t = 1$ to T **do**
3: Set regression residuals for $i = 1, \cdots, N$
$$r_i^t \leftarrow s_i^t - \hat{s}_i^{t-1}$$
4: Extract features $\{\mathbf{x}_i^t\}_{i=1}^N$
5: Learning tree split functions $\{\tau^{t,m}\}_{m=1}^M$
6: Optimize t-th stage GPRT
(a) GPRT model:
$$r_i^t = f^t(\mathbf{x}_i^t) + r_i^{t+1},$$
$$f^t \sim \mathcal{GP}(0, k^t(\mathbf{x}, \mathbf{x}')),$$
$$r_i^{t+1} \sim \mathcal{N}(0, \sigma_n^2))$$
(b) Optimize σ_n^2, σ_k^2 using Equations (14), (17)
(c) Compute $\{\bar{\alpha}^{t,m}, \bar{\sigma}^{t,m}\}_{m=1}^M$
7: Re-weighting $\{\bar{\alpha}^{t,m}\}_{m=1}^M$ for $b = 1, \cdots, B$
$$\bar{\alpha}^{t,m,b} \leftarrow \frac{(\bar{\sigma}_k^{t,m,b})^{-2}}{(\bar{\sigma}_k^{t,m,b})^{-2} + \sigma_n^{-2}} \bar{\alpha}^{t,m,b}$$
8: Update estimates for $i = 1, \cdots, N$
$$\hat{s}_i^t \leftarrow \hat{s}_i^{t-1} + \sum_{m=1}^M \bar{\alpha}^{t,m,\tau^{t,m}(\mathbf{x}_i^t)}$$
9: **end for**

where $\bar{\alpha}^{m,b} = \sigma_k^2 \sum_{i:\tau^m(\mathbf{x}_i)=b} \alpha_i$ is a summation of all α_i that the corresponding $\mathbf{x}_i$ falls into the leaf b. More intuitively, $\bar{\alpha}^{m,b}$ can be interpreted as a predictive mean of the pseudo input that falls on leaf b of the m-th tree and does not fall on the other trees.

Also, to measure the uncertainty of predictions of each leaf of trees, we consider $(\bar{\sigma}^{m,b})^2$ that is a predictive variance of the pseudo input that falls on leaf b of the m-th tree and does not fall on the other trees.

Using Equations (20), the predictive mean can be computed in $O(M \log B)$, and the computation of the predictive mean to be performed in the same framework with prediction in the CRT.

Cascade GPRT The cGPRT consists of T number of GPRTs and combines GPRTs based on the following product-based rule [6]:

$$p(\mathbf{f}_*|\mathbf{x}_*, \mathcal{M}) \propto \prod_{t=1}^T p(\mathbf{f}_*|\mathbf{x}_*^t, \mathcal{M}^t), \tag{21}$$

where $\mathcal{M}$, and $\mathcal{M}^1, \cdots, \mathcal{M}^T$ are cGPRT model and the T number of GPRT models, respectively. Because each predictive variable from GPRTs are Gaussian random variables with means $\{\bar{\mathbf{f}}_*^t\}_{t=1}^T$ and variances $\{(\sigma_*^t)^2\}_{t=1}^T$, the predictive variable from cGPRT $\mathbf{f}_*$ is still a Gaussian random variable with predictive mean and variance defined by

$$\bar{\mathbf{f}}_* = \sigma_*^{-2} \sum_{t=1}^T (\sigma_*^t)^{-2} \bar{\mathbf{f}}_*^t, \tag{22}$$

$$\sigma_*^2 = \left(\sum_{t=1}^T (\sigma_*^t)^{-2} \right)^{-1}. \tag{23}$$

In Equation (22), $\bar{\mathbf{f}}_*$ is defined by a weighted summation of $\{\bar{\mathbf{f}}_*^t\}_{t=1}^T$ with the weights that represent the uncertainty of predictions of each GPRT.

The additive form of the predictive mean in Equation (22) explicitly induces a greedy stage-wise learning of cGPRT using regression residuals as described in Algorithm 1. Each GPRT is optimized through line 6–(a) to 6–(c), and re-weighted through line 7. The intuition behind this re-weighting process is to model the current residual r^t as a summation of the regression function f^t and the subsequence residual r_i^{t+1} that is assumed to be a Gaussian random variable with zero mean and variance σ_n^2. Then, Equation (22) explicitly induces the re-weighting process. Note that the computation of predictive mean of test input can be utilized in the CRT prediction framework as described in Algorithm 2.

Algorithm 2 Prediction of cGPRT.

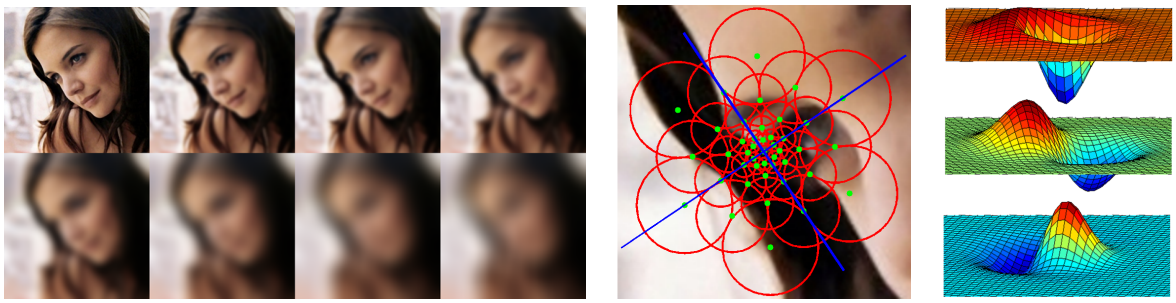
Input: test input I_*
Output: shape estimate $\hat{s}_*^T$
Procedure:

1: Initialize $\hat{s}_*^0$
2: **for** $t = 1$ to T **do**
3: Extract feature $\mathbf{x}_*^t$
4: Update estimates $\hat{s}_*^t \leftarrow \hat{s}_*^{t-1} + \sum_{m=1}^M \bar{\alpha}^{t,m,\tau^{t,m}(\mathbf{x}_*^t)}$
5: **end for**

2.3. Features & learning split functions

The shape-indexed difference of Gaussian (DoG) features are extracted as follows: (1) smoothing images with Gaussian filters at various scales as depicted in Figure 2–(a), (2) computing the similarity transform that maps a mean shape to the shape estimate, (3) applying similarity transform into local retinal sampling patterns [1] as depicted in Figure 2–(b), (4) computing global coordinates using transformed local retinal sampling patterns and the reference shape estimate, and (5) extracting Gaussian filter responses by taking pixel values at the global coordinates on Gaussian smoothed images corresponds to scale parameter of each sampling point.

Here, the difference of extracted two Gaussian filter responses is a shape-indexed DoG feature which eventually computes the response of predefined DoG filter as depicted in Figure 2–(c). Note that by applying similarity transform into the local retinal sampling patterns, computation



(a) (b) (c)

Figure 2. A extraction procedure of the shape-indexed DoG features: (a) Smoothed images using Gaussian filters at various scales, (b) Local retinal sampling pattern, where green dots and red circles represent sampling points and standard deviations for corresponding Gaussian filters, respectively (each sampling point is assigned to particular smoothing scale which is determined to be proportional to the distance between the local sampling point and the center point), and (c) DoG filters that are computed in practice during the feature extraction procedure.

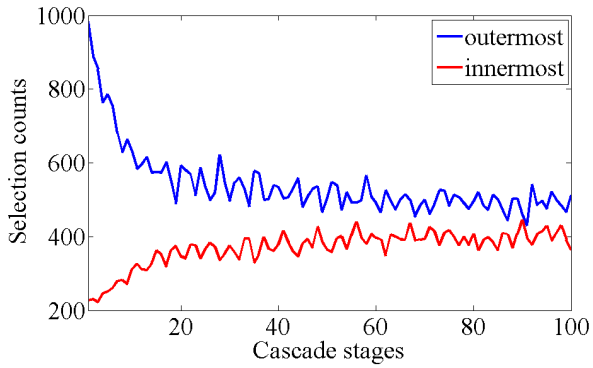


Figure 3. Counts of smoothing scale selections for split functions at different stages. Only two smoothing scales, the outermost and innermost except the center, are illustrated for better visualization.

of shape-indexed DoG features does not involve to transform whole image but transforms only sparse coordinates [5]. Also, computational complexity to obtain Gaussian-smooth images is not too high because smoothing process is performed only once: prior to the learning procedure.

The tree of cGPRT is learned with a single facial landmark [17]: the split functions of the tree are learned by randomly sampling thresholds and the DoG features referenced by l-th facial landmark. In order to obtain more discriminative split function, several split functions are tested and the best performing split function is selected. The performances of the split functions are measured in terms of squared loss on l-th facial landmark. Note that this procedure only learns split functions of the trees, and regression outputs are learned using cGPRT.

The learned trees at earlier stages tend to use the shape-indexed DoG features computed from distant sampling points while the trees at later stages tend to use the features computed from nearby sampling points as depicted in Figure 3. This is due to that the distant sampling points are more stable against shape estimate errors than the nearby sampling points because it cover larger regions. The nearby points are less stable than the distant sampling points, but much more discriminative when the shape estimate is accurate. Thus, in the learning procedure, the shape-indexed DoG features allows for each tree to adoptively select more reliable features respect to the current shape estimate error.

3. Experiments

The objectives of our experiments are two-folds: (1) to compare cGPRT using shape-indexed features with state-of-the-art methods, and (2) to verity two key elements of the proposed method: cGPRT and the shape-indexed DoG features.

3.1. Experimental settings

Implementation details To obtain the training data, face images are firstly cropped using the bounding boxes from Viola & Jones face detector [19] as [13]. Then, shape estimates are initialized into randomly sampled ground truth shapes from the other training data points. This initialization process is repeated twenty times for each face image in the training procedure. In prediction, we used the mean shape obtained from the training data points for the initialization.

We consider two configurations: (1) "cGPRT" configuration which is configured to give lower mean error but slower prediction and (2) "cGPRTfast" configuration which is configured to give faster prediction time but higher mean error. In cGPRT configuration, the number of trees for each GPRT and the number of GPRTs are set to $M = 10$ and $T = 500$, respectively. The cGPRT is formed by a two-level cascading of GPRTs likes [4, 5, 13, 17], and the num-

Dataset	ESR [5]	RCPR [4]	SDM [20]	EST [13]	LBF [17]	cGPRT
LFPW (29 landmarks)	3.47	3.50	3.49	3.80	**3.35**	3.51
HELEN (194 landmarks)	5.70	6.50	5.85	4.90	5.41	**4.63**
300-W (68 landmarks)	7.58	-	7.52	6.40	6.32	**5.71**

Table 1. Comparison of accuracy between the cGPRT and state-of-the-art methods on LFPW, HELEN and 300-W datasets.

Method	Error	std	fps
ESR [5]	7.58	-	120
SDM [20]	7.52	-	70
EST [13]	6.40	-	1000
LBF [17]	6.32	-	320
LBFfast [17]	7.37	-	3100
cGPRT	5.71	0.06	93
cGPRTfast	6.32	0.07	871

Table 2. Detailed comparison of prediction time and accuracy between cGPRT and state-of-the-art methods on 300-W dataset.

ber of first level cascade stages and the number of second level cascade stages are set to 100 and 5, respectively. Note that the total number of trees is same with the numbers used in [4, 5, 13]. In cGPRTfast configuration, the number of trees for each GPRT and the number of GPRTs are set to $M = 10$ and $T = 100$, respectively. And the number of first level cascade stages and the number of second level cascade stages are set to 10 and 10, respectively. For both configurations, the depth of trees is set to 5 that is also same value used in [4, 5, 13]. Each split function is learned through 200 trials, and the number of smoothing scales is set to 8. The number of retinal sampling points per smoothing scale is set to 6, and the resulting number of sampling points is $6 \times 7 + 1 = 43$ for each facial landmark. All experiments are performed on single core on i5-3570 3.40GHz CPU.

Datasets Most of the experimental results are reported on the 300-W [18] dataset that is considered as the most challenging dataset. We also provide the comparison results with state-of-the-art methods on the LFPW [2] and HELEN [14].

- **LFPW** (29 landmarks): The LFPW [2] dataset consists of $1,132$ images for training and 300 images for testing. The LFPW dataset provides the URLs that link to images, and the some URLs are broken. We are only possible to collect 778 training images and 216 test images which make the direct comparison with the previously proposed methods are not possible.

- **HELEN** (194 landmarks): The HELEN [14] dataset consists of $2,330$ high-resolution images with dense 194 facial landmark annotations. The HELEN dataset provides a data division: $2,000$ for training and 330 for testing.

- **300-W** (68 landmarks): The 300-W [18] is extremely challenging due to the large variations in pose, expression, illumination, background, occlusion, and image quality. It is created from existing popular datasets, including the LFPW [2], AFW [23], HELEN [14], XM2VTS [15], and the new dataset IBUG [18]. In our experiments, the whole dataset is split into training and test images, following the previous work [17]. The training images consist of the AFW dataset and the training sets of the LFPW and HELEN datasets. The test images consist of the IBUG dataset and the test images of the LFPW and HELEN datasets. The number of images in the training and testing sets are 3,148 and 689, respectively.

Evaluation metric We measured the shape estimation error as a fraction of inter-ocular distance defined as the distance between ground truth shape and shape estimate normalized by the distance between two pupils. For all experiments, we reported averaged performances over 10 trials to reduce the effect of the randomness.

3.2. Comparison with state-of-the-art methods

We compared cGPRT using shape-indexed DoG features with the following state-of-the-art methods: explicit shape regression (ESR) [5], robust cascade pose regression [4], supervised descent method (SDM) [20], ensemble of regression trees (ESR) [13], and regression local binary features (LBF) [17].

The comparison results are summarized in Table 1 and 2. The experimental results on HELEN and 300-W datasets showed that cGPRT outperformed all other methods including EST and LBF which are the two leading methods on face alignment. The performance improvement was much larger on the 300-W dataset which is the most challenging dataset, and this demonstrated the better generalization of cGPRT than the others. The example results are depicted in Figure 6. The cGRPTfast, configured to give faster prediction time but higher mean error, provided faster prediction and same mean error compared with LBF [17].

The cGPRT performed comparatively compared to other state-of-the-art methods on the LFPW dataset. However, the LFPW dataset only provides links to the faces images, and the number of broken links vary year to year. It was not possible to make direct comparison to the previously proposed methods.

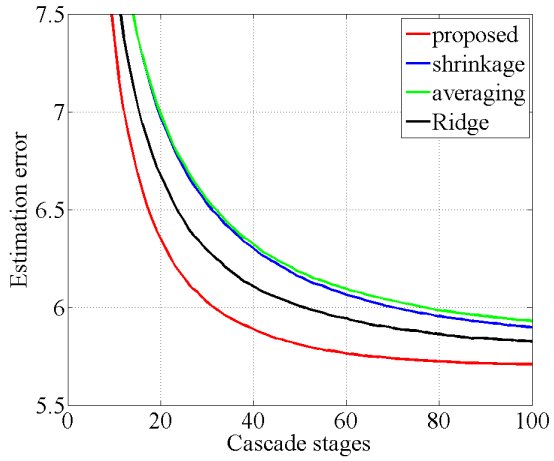


Figure 4. Comparison results on the 300-W dataset between the various regularization methods: the proposed cGPRT, shrinkage, averaging, and Ridge regression based method.

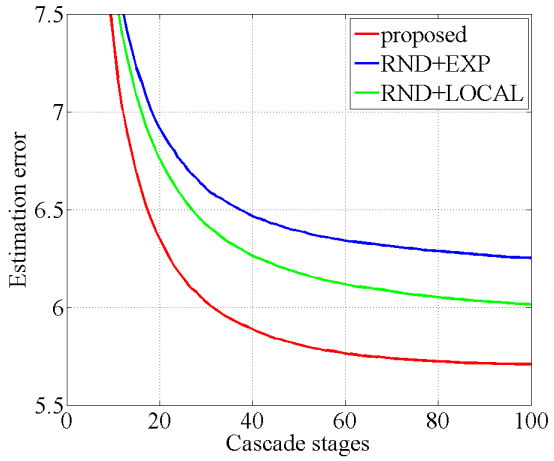


Figure 5. Comparison results on the 300-W dataset between the various feature extraction methods: the proposed shape-indexed DoG features (proposed), the randomly sampled shape-indexed pixel difference features with the exponential prior based feature selection method (RND+EXP) [13] and the local tree learning method (RND+LOCAL) [17].

3.3. Comparison with regularization methods

To verify the effectiveness of cGPRT, we compared cGPRT with the three base-line regularization methods used in ERT [13] and LBF [17]. The first and second methods are shrinkage and averaging, respectively, used in ERT. Third method is Ridge regression based regularization method used in LBF. We fixed the features to be the shape-indexed DoG features, and the parameters for each method are set to the same values with the original paper except the number of trees and the depth of trees. These parameters are not changed for the fair comparison, and the cGPRT configuration is used for the experiment.

The comparison results are depicted in Figure 4. The proposed cGPRT outperformed all the base-line regularization methods. We obtained similar performance for shrinkage and averaging methods as reported in [13]. Note that with the same feature extraction method, all methods have same computational complexity for prediction.

3.4. Comparison with features

To verify the effectiveness of the proposed shape-indexed DoG features, we compared with the two base-line shape-indexed features used in ERT and LBF. The first method samples the local pixel coordinates in randomly (RND) and selects relevant features using the exponential priors [13] (RND+EXP). The second method also samples the local pixel locations in randomly and learns the split functions to fit the single facial landmark errors using the local regions around the landmark (RND+LOCAL). We fixed the regression method and the number of sampling points to cGPRT and $68 \times 43 = 2924$, respectively, and the cGPRT configuration is used for the experiment.

The comparison results are depicted in Figure 5. The proposed shape-indexed DoG features performed best with large amount of error reduction. All feature extraction methods consider the locality to obtain discriminative trees, however, the difference is come from the correlation among the trees. The shape-indexed DoG features and RND+LOCAL method learn trees using a single facial landmark, and this reduces the correlation among the trees that can lead to performance improvement.

4. Conclusion

For the face alignment, cGPRT using shape-indexed DoG features has been proposed. The cGPRT is constructed by combining a set of GPRTs and learned in a greedy stage-wise manner. We have described the predictive mean of cGPRT can be computed in the CRT framework with better generalization. Further more, we have described the shape-indexed DoG features that are designed through difference of Gaussian filter responses computed on local retinal patterns referenced by shape estimates. The cGPRT using the shape-indexed DoG features has shown the best performances on the HELEN and 300-W datasets.

Acknowledgements

This work was partly supported by the ICT R&D program of MSIP/IITP [B0101-15-0307, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)] and the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIP) (No.NRF-2010-0028680).

Figure 6. Shape estimation results using cGPRT with the shape-indexed DoG features on three datasets: (a) LFPW (29 landmarks), (b) HELEN (194 landmarks), and (c) 300-W datasets (68 landmarks).

References

[1] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 510–517. IEEE, 2012.

[2] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 545–552. IEEE, 2011.

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[4] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *International Conference on Computer Vision (ICCV)*, pages 1513–1520. IEEE, 2013.

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[6] T. Chen and J. Ren. Bagging for gaussian process regression. *Neurocomputing*, 72(7):1605–1610, 2009.

[7] S. W. Chew, S. Lucey, P. Lucey, S. Sridharan, and J. F. Conn. Improved facial expression recognition via uni-hyperplane classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2554–2561. IEEE, 2012.

[8] A. Davies and Z. Ghahramani. The random forest kernel and other kernels for big data from random partitions. *arXiv*

preprint arXiv:1402.4293, 2014.

[9] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1078–1085. IEEE, 2010.

[10] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[11] X. Geng, C. Yin, and Z.-H. Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.

[12] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[13] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874. IEEE, 2014.

[14] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision (ECCV)*, pages 679–692. Springer, 2012.

[15] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, volume 964, pages 965–966. Citeseer, 1999.

[16] C. E. Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

[17] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692. IEEE, 2014.

[18] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 397–403. IEEE, 2013.

[19] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[20] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539. IEEE, 2013.

[21] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 497–504. IEEE, 2011.

[22] Q. Zhang, Z. Liu, G. Quo, D. Terzopoulos, and H. Y. Shum. Geometry-driven photorealistic facial expression synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):48–60, 2006.

[23] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886. IEEE, 2012.