Facial Feature Detection using Generalized LVQ and Facial Shape Model

Yusuke Morishita
Information and Media Processing Lab.
NEC Corporation
y-morishita@bp.jp.nec.com

Hitoshi Imaoka
Information and Media Processing Lab.
NEC Corporation
h-imaoka@cb.jp.nec.com

Abstract

A method for detecting the facial feature points, such as the pupil, subnasal point, and corners of the mouth, is proposed. The proposed method is composed of two stages: candidate detection of facial feature points and optimization of these points by using a facial shape model. The candidates for each facial-feature-point are extracted from a face image by using generalized learning vector quantization classifiers, and the most suitable facial feature points are then selected from the facial feature point candidates obtained in the first stage. The facial shape model is utilized to constrain the alignment of facial feature points while abnormal candidates are estimated by the least-median-of-squares method. Experiments using a large still-face dataset with various illumination conditions demonstrate that the proposed method can extract facial features precisely under varying illumination and facial-expression conditions.

1 Introduction

Automatic facial feature detection, namely, detecting facial features such as the pupil, subnasal point, and corners of the mouth, is becoming a very important task in applications such as accurate face identification, face verification, and facial-expression recognition. Numerous approaches for facial feature detection have been proposed in the last decade; however, it remains a problem to determine the precise positions of facial features under significant variations of facial appearance such as shape, pose, illumination, expression, and occlusion.

The active shape model (ASM)[1] is one of the early approaches for facial feature detection. It models grey-level texture by using a local linear template and the configuration of feature points by using a statistical shape model. The active appearance model (AAM)[2], which combines shape and texture in one PCA space, is an extension of the ASM. Although these approaches perform well if the models are built with a limited number of known subjects, the alignment performance of ASM and AAM degrades quickly if the models are either trained on a large dataset or fitted to unseen subjects not in the training set[3].

To tackle this problem, the "shape optimized search AAM" (SOS-AAM) [4] has been proposed. The SOS-AAM uses a boosted classifier[5] as each facial-feature detector and a statistical shape model to give the spatial distribution of features over the face. The shape parameters corresponding to the facial feature points are optimized to maximize the sum of feature responses obeying a non-linear maximization scheme. The SOS-AAM approach outperforms the AAM approach; however, it might fail under severe conditions of facial appearance, such as uncontrolled lighting conditions, owing to the optimization scheme for facial feature points by not discarding the abnormal output of the feature detector.

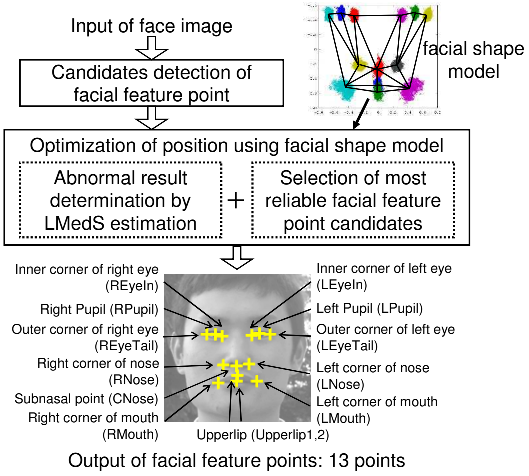

Input of face image
Candidates detection of facial feature point
facial shape model
Optimization of position using facial shape model
Abnormal result determination by LMedS estimation
+
Selection of most reliable facial feature point candidates
Inner corner of right eye (REyeIn)
Right Pupil (RPupil)
Outer corner of right eye (REyeTail)
Right corner of nose (RNose)
Subnasal point (CNose)
Right corner of mouth (RMouth)
Upperlip (Upperlip1,2)
Inner corner of left eye (LEyeIn)
Left Pupil (LPupil)
Outer corner of left eye (LEyeTail)
Left corner of nose (LNose)
Left corner of mouth (LMouth)
Output of facial feature points: 13 points

Figure 1. Overview of proposed method

In light of the above-described circumstances, a novel method for detecting facial features is proposed in the following. This method is composed of two stages (as shown in Figure 1): first, detection of candidate facial feature points and, second, optimization of these points by using a facial shape model. In the first stage, candidates for each facial-feature-point are determined by generating confidence maps for each feature point by using a generalized learning vector quantization (GLVQ)[6] classifier. In the second stage, the most suitable facial feature points are selected from the candidates for facial feature points obtained in the first stage. The facial shape model , which is formed by concatenating the coordinate values of feature points, is utilized to constrain the alignment of facial feature points while abnormal candidates are determined by least-median-of-squares (LMedS) estimation. The proposed scheme enables highly accurate position detection even when the confidence of facial features cannot be calculated correctly owing to a change in illumination or facial expression.

2 Candidate Detection of Facial Feature Points

With the proposed method, the candidates for each facial-feature-point, namely, candidate detection of facial feature points, are determined first. In order to detect feature points accurately and stably, we choose thirteen feature points whose surrounding area contains rich edge and texture. GLVQ classifiers are utilized as the candidate detector. It was clarified that the accuracy of the face detection algorithm using GLVQ is equivalent to or better than that of the support vector machine (SVM) and that the face detection speed is much higher than that of SVM owing to a fewer number of reference vectors (corresponding to the support vectors of SVM)[7].

The GLVQ algorithm and how to find the candidates for each facial-feature-point by GLVQ are described in section 2.1 and section 2.2, respectively.

2.1 Generalized LVQ

GLVQ[6] is a method of learning templates, as used in nearest-neighbor classifiers, based on a minimum-classification-error (MCE) criterion. MCE minimizes the smoothed empirical risk defined by

$$R_e(\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\ell(\rho_k(\boldsymbol{x}_n;\theta))1(\boldsymbol{x}_n \in \omega_k), \qquad (1)$$

where $\boldsymbol{x}_n (n = 1, \cdots, N)$ and $\omega_k (k = 1, \cdots, K)$ denote training samples and classes, respectively, and $1(\cdot)$ is an indicator function such that $1(true) = 1$ and $1(false) = 0$. Function $\ell(\cdot)$ is a smoothed loss function defined by $\ell(\rho) = 1/(1 + \exp(\xi\rho))$, where $\xi (> 0)$ controls the slant of the sigmoid function, and when ξ goes to infinity, equation (1) becomes identical to the empirical loss in Bayes decision theory. $\rho_k(\boldsymbol{x}_n;\theta)$ is called a misclassification measure (as explained later). The classifier parameter θ can be updated for a given $\boldsymbol{x}_n$ to minimize the smoothed empirical risk as follows in an online learning form called probabilistic descent:

$$\theta \leftarrow \theta - \varepsilon\frac{\partial R_e(\theta)}{\partial\theta}, \qquad (2)$$

$$\frac{\partial R_e(\theta)}{\partial\theta} = \sum_{k=1}^{K}\frac{\partial\ell(\rho_k(\boldsymbol{x}_n;\theta))}{\partial\theta}1(\boldsymbol{x}_n \in \omega_k). \qquad (3)$$

For nearest-neighbor classifiers, the classifier parameter consists of reference vectors called templates; that is, $\theta = \{\boldsymbol{m}_{ki} | k = 1, \cdots, K; i = 1, \cdots, N_k\}$ where N_k is the number of reference vectors in class ω_k. In GLVQ, the misclassification measure is defined as follows to ensure convergence of reference vectors:

$$\rho_k(\boldsymbol{x}_n;\theta) = \frac{d_k(\boldsymbol{x}_n;\theta) - d_l(\boldsymbol{x}_n;\theta)}{d_k(\boldsymbol{x}_n;\theta) + d_l(\boldsymbol{x}_n;\theta)}, \qquad (4)$$

where $d_k(\boldsymbol{x}_n;\theta)$ is the squared Euclidean distance between $\boldsymbol{x}_n$ and the nearest reference vector $\boldsymbol{m}_{ki}$ of class ω_k to which $\boldsymbol{x}_n$ belongs; likewise, $d_l(\boldsymbol{x}_n;\theta)$ is the squared Euclidean distance between $\boldsymbol{x}_n$ and the nearest reference vector $\boldsymbol{m}_{lj}$ of the other classes. GLVQ learning rule for these two reference vectors are then obtained as follows:

$$\begin{aligned}\boldsymbol{m}_{ki} &\leftarrow \boldsymbol{m}_{ki} + \varepsilon w(\rho_k(\boldsymbol{x}_n;\theta)) \\ &\times \frac{d_k(\boldsymbol{x}_n;\theta)}{\{d_k(\boldsymbol{x}_n;\theta) + d_l(\boldsymbol{x}_n;\theta)\}^2}(\boldsymbol{x}_n - \boldsymbol{m}_{ki}), \qquad (5)\\ \boldsymbol{m}_{lj} &\leftarrow \boldsymbol{m}_{lj} - \varepsilon w(\rho_k(\boldsymbol{x}_n;\theta)) \\ &\times \frac{d_l(\boldsymbol{x}_n;\theta)}{\{d_k(\boldsymbol{x}_n;\theta) + d_l(\boldsymbol{x}_n;\theta)\}^2}(\boldsymbol{x}_n - \boldsymbol{m}_{lj}), \qquad (6)\end{aligned}$$

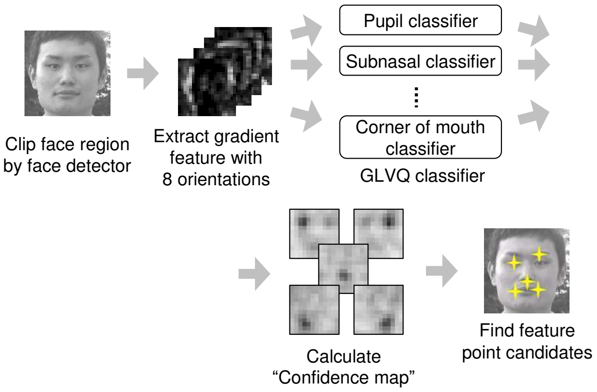

Clip face region by face detector
Extract gradient feature with 8 orientations
Pupil classifier
Subnasal classifier
Corner of mouth classifier
GLVQ classifier
Calculate "Confidence map"
Find feature point candidates

Figure 2. Flow of candidate detection of facial-feature points

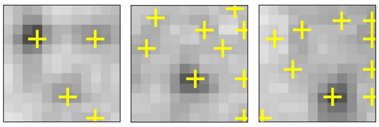


Figure 3. Example of confidence maps and each candidates: for right pupil (left), subnasal point (center), and left corner of mouth (right)

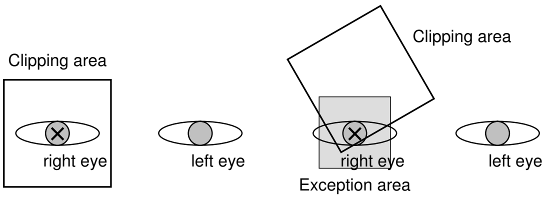
where $w(\rho_k(\boldsymbol{x};\theta)) = 4\ell(\rho_k(\boldsymbol{x};\theta))\{1 - \ell(\rho_k(\boldsymbol{x};\theta))\}$. In addition, to avoid getting trapped in local minima, GLVQ employs a simulated annealing technique, in which the slant parameter ξ is set to a small positive number at the beginning and is increased during learning.

2.2 Candidate Detector using GLVQ

How to find the feature point candidates for each facial-feature point by GLVQ is described in the following. The flow of the candidate detection stage is shown in Figure 2. First, to clip the face region, the face detector is applied to an input image, and the gradient features with eight orientations are extracted. Next, to construct a "Confidence map" for each feature point within the face region, two-class GLVQ classifiers are applied to calculate confidence values of each feature point. The candidates for a facial feature point are then extracted from the confidence maps by finding local maxima up to M for each facial-feature point. An example of the confidence maps of the right pupil, the subnasal point, and the left corner of the mouth obtained by GLVQ classifiers is shown in Figure 3.

To train the GLVQ classifiers for finding the candidates for facial feature points, about 15,000 positive and negative samples were used for each classifier. Positive samples are clipped (38 × 38 pixel) around each landmark point (manually annotated). Negative samples are clipped in the same way, but the center of the clipping area is located on the outside of the area around the landmark point, named exception area, which is the rectangle whose position, scale, and rotation vary randomly (see Figure 4). Some of the training samples are shown in Figure 5.

In the experiments described in section 4, 20 reference vectors of each GLVQ classifier were used, and up to $M = 10$ candidates for each facial-feature point were extracted.



(a) positive samples (b) negative samples

Figure 4. Clipping area of training samples (case of right pupil): for positive samples (a) and negative samples (b). The cross mark is the landmark point, the rectangle drawn with a solid line is the clipping area, and the shaded rectangle is the exception area.

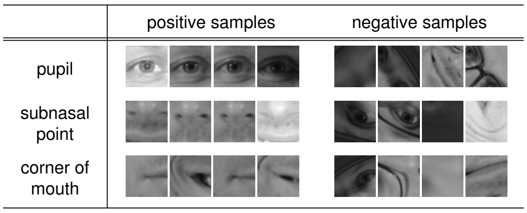


Figure 5. Some of the positive and negative training samples: for pupil (top), subnasal point (middle), and corner of mouth (bottom)

3 Optimization of Facial Feature Points using the Facial Shape Model

To determine the optimum facial feature points from the candidates, the facial shape model is applied to optimize these points after the candidate facial features are extracted.

Given a set of feature point candidates, the outlier candidates are first found by using equation (7). The most suitable combination of feature points is then extracted from a discrete search space with size up to M^K ($K = 13$ is the number of facial features) by using equation (8),

$$\tilde{\boldsymbol{p}} = \underset{\boldsymbol{p}}{\operatorname{argmin}} \operatorname{med} ||T_{\boldsymbol{p}}(\boldsymbol{x}_i^{(m)}) - \boldsymbol{z}_i||, \tag{7}$$

$$F = \sum_{i \in C} s_i(\boldsymbol{x}_i^{(m)}) + \lambda \sum_{i \in C} \sigma(-||T_{\boldsymbol{p}}(\boldsymbol{x}_i^{(m)}) - \boldsymbol{z}_i||). \tag{8}$$

In equation (7), $\boldsymbol{x}_i^{(m)}$ is the coordinate value of the m^{th} candidate in a set of the i^{th} extracted feature point candidates, $\boldsymbol{z}_i$ is the coordinate value of the i^{th} feature point in the facial shape model, $\boldsymbol{p}$ is the parameter of the Helmert transformation, and $T_{\boldsymbol{p}}(\cdot)$ is the Helmert transformation. In equation (8), C is the class of facial feature point candidates except for the outliers, $s_i(\boldsymbol{x})$ is the confidence value of the facial features at point $\boldsymbol{x}$, and $\sigma(\cdot)$ is the sigmoid function. The first term in equation (8) denotes the appearance likelihood of the facial features, and the second term denotes the geometric likelihood of alignment of the facial features. In addition, λ indicates the coefficient of adjustment between both terms. In this paper, the mean value of a number of facial feature positions manually annotated as facial shape model $\boldsymbol{z}$ is used (see Figure 1).

Figure 6. Example images from MBGC version 1.0 Still Face Database[8] (cropped face)

The proposed optimization procedure is described as follows. First, to eliminate the influence of outliers, the parameter of the Helmert transformation from the candidate facial feature points to the facial shape model are determined by LMedS estimation using equation (7). This process is described in detail below:

1. Select two facial features randomly from a set of K facial features, e.g., right pupil and left corner of mouth.

2. Determine parameter $\boldsymbol{p}$ of the Helmert transformation from facial feature point candidates $\boldsymbol{x}^{(m)}$ to facial shape model $\boldsymbol{z}$ by using the two facial features selected in step 1.

3. Use parameter $\boldsymbol{p}$ to calculate the median of the squared residuals (i.e., med $||T_{\boldsymbol{p}}(\boldsymbol{x}_i^{(m)}) - \boldsymbol{z}_i||$).

4. Repeat steps 1 through 3 to find parameter $\tilde{\boldsymbol{p}}$ corresponding the least median value.

Next, F in equation (8) is calculated from parameter $\tilde{\boldsymbol{p}}$ obtained above, and the most suitable candidate that minimizes F is found by changing the candidate with the largest residual. For example, if $\boldsymbol{x}_j^{(m)}$ has the largest residual, F through $m = 1$ to M is calculated, and the candidate of the j^{th} feature point is replaced with the one that minimizes F.

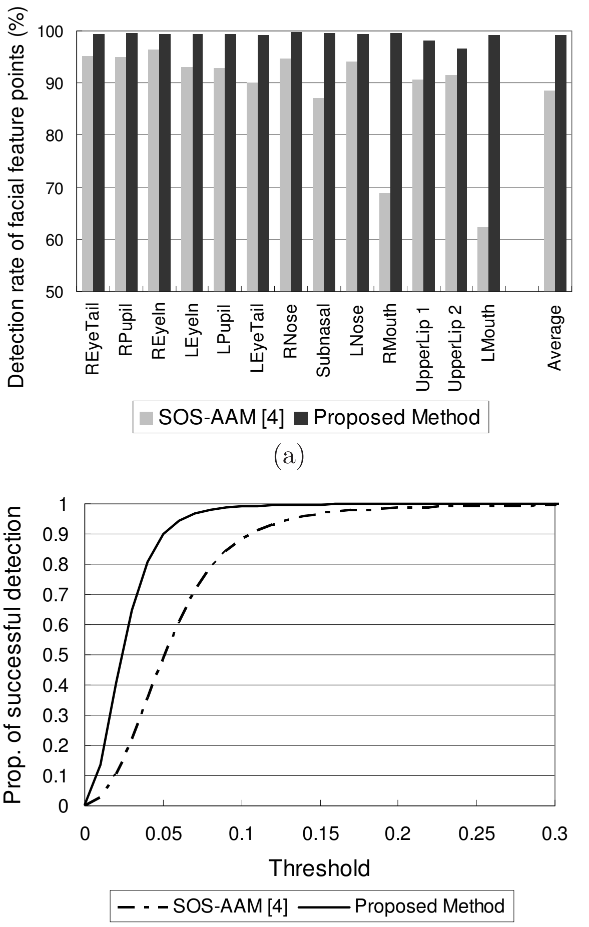
By repeating the above-mentioned process for different candidate facial feature points while all residuals are kept below a threshold value, it is possible to obtain the most suitable combination of facial feature points.

4 Experiments and Results

To evaluate the detection accuracy of the proposed facial feature point detection method, it was tested on facial images in the MBGC version 1.0 Still Face Database[8]. This database mainly consists of frontal faces taken in both indoor and outdoor environments with various illumination and facial-expression conditions. Example images from the MBGC Still Face Database are shown in Figure 6. In this experiment, 2,000 of uncontrolled still-face images in this database were used.

Each facial-feature detection process is considered successful if the distance from the detected facial feature position to the true location (annotated manually) is less than 10% of the true inter-ocular distance. The average error of all thirteen feature points was also calculated. To concentrate on the facial feature detection, those examples in which the face detection failed were discarded.

The results of the proposed and the previous method (SOS-AAM[4] using a GLVQ classifier as a feature detector instead of Viola and Jones' AdaBoost cascade classifier[5]) are presented in Figure 7. Figure 7(a) shows that the detection rates for all facial feature



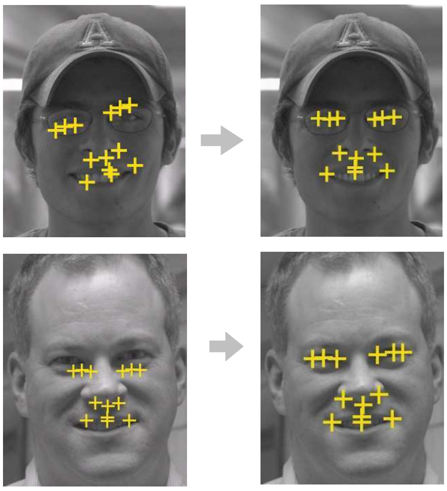
(a)

(b)

Figure 7. Detection rates of facial feature points in MBGC Still Face Database: (a) detection rates of each facial-feature point and (b) threshold versus successful detection rate

points of proposed method are more accurate than the corresponding rates for the previous method. Figure 7(b) is plotted as threshold values (i.e., radius of acceptable area) versus successful detection rate. In this case, a detection rate corresponding to threshold 0.1 represents the rate of detection of faces with average distance error of less than 10% of the true inter-ocular distance. Figure 7(b) also shows that the average detection error of the proposed method is less than that of the previous method over the whole range of threshold values. For example, with threshold < 0.1, the success rate of the proposed method is 99.1% of all faces, while that of the previous method is only 88.6% of faces.

The facial feature points detected by the proposed method and the previous method were compared (see Figure 8). As shown in Figure 8(a), facial feature detection by the previous method may fail under varying illumination and facial expressions because the confidence maps were not calculated correctly. The previous method optimizes the facial feature points while satisfying the constraints of the facial shape model under the influence of outliers, so the facial feature points are displaced as a whole. In contrast, the proposed method can detect all facial feature points precisely even in this condition, because of the influential outlier is eliminated by the proposed optimization process.



(a) (b)

Figure 8. Comparison of detected positions: previous method (a) and proposed method (b)

5 Conclusion

A novel method for detecting facial features was proposed. It is composed of two stages: candidate detection of facial feature point and optimization of these points by using a facial shape model. The candidates for each facial-feature-point are extracted from face image by using GLVQ classifiers, and the most suitable facial feature points are then found from candidates. The facial shape model is utilized to constrain the alignment of facial feature points while abnormal candidates are estimated by LMedS. Experiments using a large uncontrolled still-face dataset show that the proposed method is precise with respect to variations in illumination and facial expressions.

References

[1] T. F. Cootes, et al.: “Active shape models - their training and application,” In *Computer Vision and Image Understanding*, Vol.61(1), pp.38-59, 1995.

[2] T. F. Cootes, et al.: “Active appearance models,” In *IEEE Tran. on Pattern Analysis and Machine Intelligence (PAMI)*, Vol.23(6), pp.681-685, 2001.

[3] R. Gross, et al.: “Generic vs. person specific active appearance models,” In *Image and Vision Computing*, Vol. 23, pp. 1080-1093, 2005.

[4] D. Cristinacce, et al.: “A Comparison of Shape Constrained Facial Feature Detectors,” In 6^{th} *Intr. Conf. on Automatic Face and Gesture Recognition*, 2004.

[5] P. Viola, et al.: “Rapid Object Detection using a Boosted Cascade of Simple Features,” In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[6] A. Sato, et al.: “Generalized Learning Vector Quantization,” In *Advances in Neural Information Processing Systems, MIT Press*, 1996.

[7] A. Sato, et al.: “Advances in face detection and recognition technologies,” In *NEC Journal of Advanced Technology*, Vol. 2, pp. 28-34, 2005.

[8] P. Jonathon Phillips, et al.: “Overview of the Multiple Biometrics Grand Challenge,” In *Proc. of the* 3^{rd} *IAPR/IEEE Intr. Conf. on Biometrics*, 2009.