Real-time Facial Feature Extraction by Cascaded Parameter Prediction and Image Optimization

Fei Zuo[1] and Peter H. N. de With[2]

[1] Eindhoven University of Technology, Faculty E-Eng., 5600MB Eindhoven, NL
[2] LogicaCMG/Eindhoven Univ. of Technol., P.O.Box 7089, 5605JB Eindhoven, NL
Email: F.Zuo@tue.nl

Abstract. We propose a new fast facial-feature extraction technique for embedded face-recognition applications. A deformable feature model is adopted, of which the parameters are optimized to match with an input face image in two steps. First, we use a cascade of parameter predictors to directly estimate the pose (translation, scale and rotation) parameters of the facial feature. Each predictor is trained using Support Vector Regression, giving more robustness than a linear approach as used by AAM. Second, we use the generic Simplex algorithm to refine the fitting results in a constrained parameter space, in which both the pose and the shape deformation parameters are optimized. Experiments show that both the convergence and the accuracy improve significantly (doubled convergence area compared with AAM). Moreover, the algorithm is computationally efficient.

1 Introduction

Accurate facial feature extraction is an important step in face recognition. Our aim is to build a feature-extraction system that can be used for face recognition in embedded and/or consumer applications. This application field imposes additional requirements in addition to feature extraction accuracy, such as real-time performance under varying lighting conditions, etc.

One promising technique for facial feature extraction is to use a deformable model [5], which can adapt itself to optimally fit to individual images while satisfying certain model constraints. The constraints can be derived from the prior knowledge about the object properties (e.g. shape and texture). The feature extraction process can then be seen as an optimization process, where the model parameters are adjusted to minimize a cost function for fitting.

In earlier research [1], a parameterized deformable template is used for facial feature extraction. However, it is computationally expensive and the convergence is not guaranteed. Recently, the Active Shape Model (ASM) and Active Appearance Model (AAM) [2] have been proposed as two promising techniques for feature extraction. The ASM fits a shape model to a real image by using a local deformation process, constrained by a global variance model. However, the ASM searches for the 'best-fit' for each landmark independently, which sometimes leads to unstable results. The global constraints can maintain a plausible

shape, but they cannot 'correct' the wrong local adjustments. The AAM incorporates global texture modelling giving more matching robustness, but we have found that the linear model parameter prediction used by AAM only works well for very limited ranges.

The deformable model fitting can also be solved by applying a general optimization algorithm, which gives accurate fitting results, provided that the cost function is appropriately defined and its global minimum is found. However, the crude use of such a technique either leads to erroneous local minima (when a local optimization algorithm such as the gradient-descent algorithm is applied) or takes too much computation cost (when a global stochastic optimization algorithm such as the genetic algorithm is applied).

In this paper, we propose a novel model-based facial feature extraction technique, employing both fast parameter prediction and direct optimization for each individual image. The used feature model is a variant of the statistical model in [2]. The fitting of the model to a real image is performed in two steps. First, a cascade of parameter predictors are used to estimate in a single step the 'correct' pose parameters (translation, scale and rotation). Second, a general optimization algorithm is used to further improve the extraction accuracy. In our case, a Simplex algorithm [8] is adopted to jointly optimize the pose and the shape deformation parameters. The aim is to obtain fast and accurate feature extraction results, which may enable re-usage in the face-recognition stage.

2 Statistical feature model

2.1 Feature model with extended shape and texture structure

Motivated by ASM and AAM, we build our statistical feature model by incorporating both shape and texture information. The geometrical shape of a facial

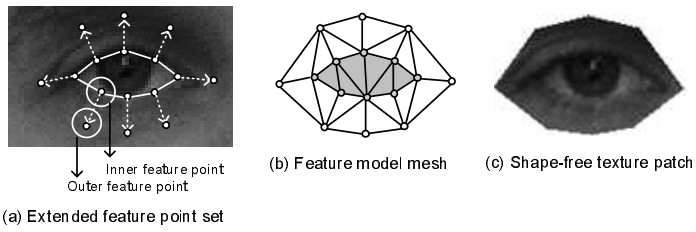

Fig. 1. The feature model.

feature (e.g. an eye) can be represented by a set of discrete feature points $FP = \{P_i = (x_i, y_i) | 1 \leq i \leq N\}$, where N is the number of the feature points. In contrast with ASM/AAM, where only corners and/or contour points are selected, we use an extended set of feature points covering a larger texture region. To this end, we introduce a set of auxiliary *outer* feature points (Fig. 1(a)), which can be derived from the original feature points FP (*inner* feature points) by extending each P_i to a neighboring point P_i' in the direction perpendicular to the

contour curvature. The extension range is proportional to the size of the feature. Although the outer feature points depend on the inner points and provide no new shape information, they encapsulate a larger texture region and incorporate more information (both inner texture and the surrounding texture).

Based on the extended set of feature points, a Delaunay Triangularization is performed to construct a mesh over the feature region (see Fig. 1(b)). The triangular mesh is used for the texture warping to a standard shape (see Section 2.2).

2.2 Generic PCA-based feature model

To obtain a feature model that can adapt to individual shape variations, we adopt Principal Component Analysis (PCA) from ASM/AAM to model the shape variations. Suppose matrix $\mathbf{\Phi}$ contains L largest eigenvalues after the PCA decomposition, then any normalized (w.r.t position, scale and rotation) shape vector $\mathbf{x_n}$ can be approximated by

$$\mathbf{x_n} \approx \overline{\mathbf{x}} + \mathbf{\Phi} \cdot \mathbf{b}, \tag{1}$$

where $\overline{\mathbf{x}}$ is the mean normalized shape, and vector $\mathbf{b}$ defines a set of deformation parameters for the given class of features. If the geometric transformation (translation, scale and rotation) is incorporated, any feature shape vector $\mathbf{x}$ (not normalized) can be modelled using the normalized mean shape and a parameter set including both pose (x, y, s, θ) and deformation parameter $\mathbf{b}$ by

$$\mathbf{x} = T_{x,y,s,\theta}(\overline{\mathbf{x}} + \mathbf{\Phi} \cdot \mathbf{b}). \tag{2}$$

In Equation (2), $T_{x,y,s,\theta}$ denotes the geometric transformation by translation (x, y), scaling s and rotation θ. Based on the shape information, the texture overlayed by the shape can be sampled and warped to a mean shape by piecewise affine warping (Fig. 1(b) and (c)). The texture samples are then scanned on line basis and reordered into one vector $\mathbf{t}$, which is normalized by mean and standard deviation.

Given the feature model, the feature extraction in a new image can be formulated as a parameter estimation problem. The optimal shape parameters $(x, y, s, \theta, \mathbf{b})$ need to be located, so that the texture region covered by the estimated shape has the minimum matching error with a normalized template texture $\overline{\mathbf{x_t}}$.

3 Model fitting by prediction and optimization

3.1 Overview of model fitting

We search for an optimal set of model parameters for a new image by taking the following two steps: pose parameter prediction and direct local optimization.

Motivated by AAM, we utilize the prior knowledge of the properties of the feature and its neighboring areas. A set of learning-based predictors are trained,

which are able to directly predict the pose parameters given the incorrectly placed shape. Our prediction scheme has two distinct features.

1) We use *Support Vector Regression* (SVR) [3] to train the parameter predictors. Due to its nonlinearity, the SVR prediction is more reliable and robust than the limited linear prediction used by AAM. We have found in our experiments that the SVR is able to predict the model parameters correctly, even for very large pose deviations.

2) We use *a cascade of SVR predictors* to boost the prediction accuracy. We have found that the SVR predictors trained with varying pose variation ranges lead to different prediction errors. The cascading of these predictors can 'pull' the parameters to the correct position in a step-wise manner.

Parameter prediction quickly finds the approximately correct pose parameters. At the second stage, we use a direct image optimization of both the pose and deformation parameters within a small constrained area, based on the SVR prediction statistics.

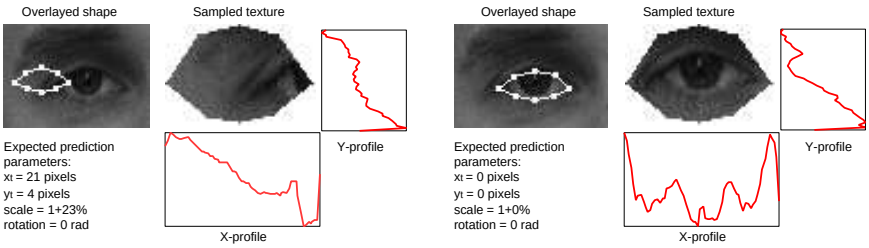

Fig. 2. Feature vector for SVR.

3.2 Cascaded prediction by Support Vector Regression

The cascaded prediction involves the following aspects.

Feature vector preparation. A reduction of the dimensionality of the texture vector decreases the training efforts and the computation complexity. Therefore, we extract the vertical and horizontal profiles from the normalized texture region and use the combined *profile vector* $\mathbf{v}$ for texture representation. In our experiments with eye extraction, the dimensionality of the feature space is reduced from 1700 to 100, giving more reliable training results and faster processing.

Parameter prediction using Support Vector Regression (SVR). Given an initial shape vector $\mathbf{x}$ and its associated profile vector $\mathbf{v}$, a geometric transformation correction $\delta\mathbf{p} = (\delta x, \delta y, \delta s, \delta\theta)^T$ can be applied to $\mathbf{x}$ for an optimal fit to the image. We try to build a prediction function f for the geometrical transformation δp to deform the shape towards the actual feature, thus, $f(\mathbf{v}) \simeq \delta\mathbf{p}$.

We obtain prediction function f by support vector regression. The SVR uses kernel functions to map data to a higher dimensional space and thus achieves nonlinear mapping. For each training image, we randomly displace each vector element of $\mathbf{p}$ from the manually annotated known optimal value $\mathbf{p}$ to $\mathbf{p_i}$ and obtain the displaced shape $\mathbf{x_i}$ and its corresponding profile vector $\mathbf{v_i}$. We then

use the training set $\{(\mathbf{v_i}, \delta\mathbf{p_i})|i = 1, 2, ...\}$ to train an ϵ-SVR function [4], where $\delta\mathbf{p_i} = \mathbf{p_i} - \mathbf{p}$.

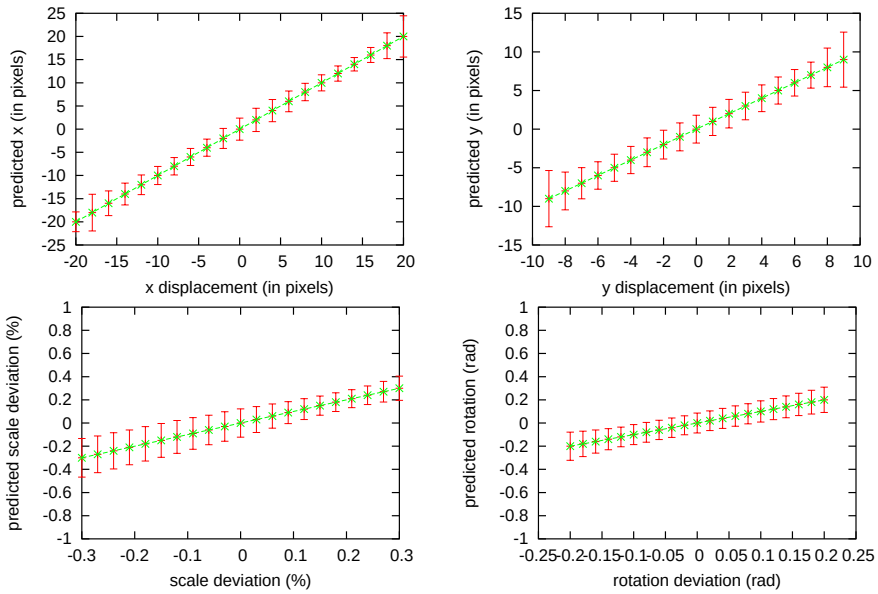

Fig. 3. SVR-based prediction vs. the actual parameter deviation. The total length of each vertical error bar corresponds with two standard deviations.

Experiments for the prediction. In our experiment, we used a face set composed of 37 labelled face images [7]. We randomly selected 27 images for training and the remaining 10 images were used for testing. For each training image, we randomly perturbed all the pose parameters and collected 100 data samples. The perturbation range is shown in Table 1.

For ϵ-SVR, we used the Radial Basis Function kernel, and all the SVR parameters were selected by cross-validation. We tested the learned prediction function on the test set. The experimental results are shown in Fig. 3. It can be seen that the SVR-based prediction gives good results even for very large parameter deviations. The prediction error is distributed uniformly for various parameter displacements. For comparison, the prediction accuracy of the linear prediction scheme as used by AAM deteriorates sharply when the parameter displacement exceeds a small value, e.g. a horizontal (x) displacement of only $\pm 20\%$ of the shape width. Our system performs better because the SVM-based approaches are more flexible and can learn and adapt to the complexity of the problem.

Cascaded prediction scheme. Although the use of SVR yields robust prediction results for large parameter perturbations, the prediction accuracy with small parameter displacements is still not satisfactory. The use of an iterative approach [2] will not give much gain, since the prediction error over different parameter displacements mostly remains the same. However, if a second predic-

Table 1. Perturbation range for parameter prediction.

Parameter	Perturbation
x	±50% of the width of the ground-truth shape
y	±50% of the height of the ground-truth shape
scale	±30%
rotation	±0.2 rad

tion function is applied with smaller capture range but higher accuracy, then the error of the final prediction can be significantly reduced.

To this end, we propose a *cascaded prediction* approach in which a set of SVR functions are trained over varying data perturbation ranges. These cascaded functions form a prediction chain. The initial functions in the chain are trained with large parameter displacements but only have coarse prediction accuracy. On the other hand, the succeeding functions are trained with smaller parameter displacements but have approximately double accuracy. With this prediction chain, the incorrectly displaced model parameters can be gradually 'pulled' to the correct position. In practice, the prediction chain only contains a few SVR functions. In our case, three SVR functions are used (more does not improve), each of which is trained over a training set by halving the perturbation range of the previous one. Fig. 4 shows the prediction performance of the second and third functions for horizontal (x) prediction. In Section 4, we provide experimental results that demonstrate the effectiveness of the cascaded prediction scheme.

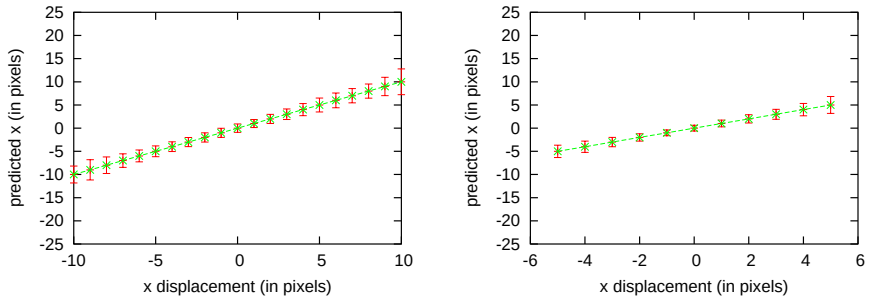

Fig. 4. The horizontal(x)-prediction performance trained over halved perturbation ranges (left: the 2nd function, right: the 3rd function).

3.3 Improving accuracy using direct local optimization

The prediction results achieved in the previous section largely depend on the feature appearance in the training set. The use of prior knowledge leads to a fast and robust 'jump' to the right position. However, it is not well adapted to individual features. Therefore, we apply a general optimization technique to refine the matching result. The procedure minimizes the fitting cost function w.r.t. both the pose and the deformation parameters. Based on the prediction statistics in the previous section, the optimization needs only to perform a con-

strained search over a small parameter subspace. We have used the following optimization techniques in our experiments.

1) **Gradient-descent**: Although gradient-based algorithms are fast, they fail to yield satisfactory results in our case. Since the target function can have many local minima, the gradient-descent-based search can easily converge to local minima. Moreover, the computation of function derivatives is time-consuming.

2) **Simulated annealing**: Simulated annealing is a 'global' optimization technique, which makes use of random sampling in the parameter space. However, the tuning of the annealing parameters is difficult (e.g. the cooling rate and the sampling step). Preliminary experiments showed that the use of simulated annealing is much more computationally expensive than the Simplex method (illustrated below) and yields no better results.

3) **Simplex algorithm**: Although still a local optimization technique, the Simplex method allows occasional 'jumps' out of local minima. In our experiments, it gives the best trade-off between fitting accuracy and computation cost.

4 Experimental results

In this section, we give the experimental results for eye extraction, using the same data set as given in Section 3.2.

Pose parameter prediction. To measure the robustness and accuracy of the parameter prediction, we randomly perturb the pose parameters in the test set within the range specified in Table 1. The predicted parameters are compared with the ground-truth parameters, and the results are given in Table 2. It can be seen that the cascaded SVR prediction generally yields higher prediction accuracy, especially in x/y prediction.

Table 2. The pixel accuracy of the pose parameter prediction.

Prediction scheme	Pose parameter	Mean error	Std. Deviation
One-stage/Cascaded SVR	x	2.20/0.94 (pixels)	2.21/1.60 (pixels)
	y	1.48/0.94	1.23/0.90
	scale	0.09/0.09	0.08/0.08
	rotation	0.08/0.06	0.06/0.05

Feature extraction accuracy. To measure the feature extraction accuracy, we randomly position a mean shape near the ground-truth position in the test image and perform the model fitting. The average point-to-point error between the fitted shape and the manually labelled shape is measured (see Table 3). It can be seen that the use of the Simplex optimization effectively improves the extraction accuracy. Fig. 5 gives two examples of the eye extraction. A typical execution takes approximately 40-60 ms on a Pentium-IV PC (3 GHz), in which the SVR prediction takes one-third and the Simplex optimization takes two-third of the total execution time. This is much more efficient than using a direct optimization alone, which takes 300-400 ms under the same conditions.

Table 3. Feature extraction accuracy for cascaded SVR with/without Simplex.

Applied technique	Mean pt-pt error (pixels)	One std. dev. (pixels)
Cascaded SVR prediction	2.56	1.67
Casc. SVR + Simplex	2.14	1.43

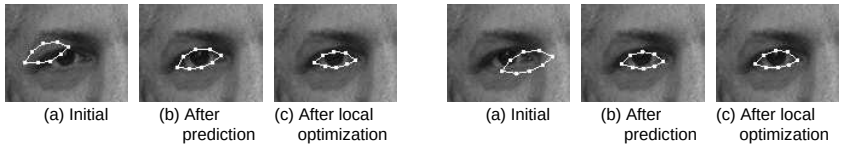

Fig. 5. Stages of the eye extraction for the complete algorithm.

5 Conclusions

In this paper, we have proposed a fast facial feature extraction technique for face recognition applications. The proposal contains three major contributions. First, we use support vector regression to train a parameter predictor for the feature model (Section 2), which is used to estimate the correct parameter displacements in a single step. Second, we use a cascade of SVR-based predictors with increasing convergence accuracy. The predictors are trained over data sampled with varying perturbation ranges, to give a performance that exchanges capture range with prediction accuracy. The cascading of these predictors thus combines a large capture range with a high prediction accuracy. Finally, a direct individual image optimization by the Simplex algorithm gives improved model parameters. The experimental results show an at least doubled convergence area compared to AAM with a higher accuracy. We are now applying the technique to a larger-scale database and insert it into an embedded/consumer face-recognition application.

References

1. Yuille, A., Cohen, D., and Hallinan, P.: Feature extraction from faces using deformable templates. Proc. CVPR. (1989) pp. 104–109
2. Cootes, T., Taylor, C.: Statistical models of appearance for computer vision. Tech. Rep. ISBE, Univ. Manchester. (2001)
3. Smola, A., Schlkopf, B.: A tutorial on support vector regression. Tech. Rep. NC-TR-98-030, Univ. London. (1998)
4. Chang, C. C., Lin C. J.: LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm. (2001)
5. Cheung, K-W., Yeung, D-Y, and Chin, R.: On deformable models for visual pattern recognition. Pattern Recog. 35 (2002) pp. 1507–1526
6. Duda, R., Hart, P., and Stork, D.: Pattern classification. (2001)
7. Stegmaan, M.: Analysis and segmentation of face images using point annotation and linear subspace techniques. Tech. Rep. DTU. (2002)
8. Nelder, J.A., and Mead, R.: A Simplex Method for Function Minimization. Computer Journal, vol. 7. (1965) pp. 308–313