Self-Adapting Feature Layers

Pia Breuer and Volker Blanz

Institute for Vision and Graphics – University of Siegen
{pbreuer,blanz}@informatik.uni-siegen.de

Abstract. This paper presents a new approach for fitting a 3D morphable model to images of faces, using self-adapting feature layers (SAFL). The algorithm integrates feature detection into an iterative analysis-by-synthesis framework, combining the robustness of feature search with the flexibility of model fitting. Templates for facial features are created and updated while the fitting algorithm converges, so the templates adapt to the pose, illumination, shape and texture of the individual face. Unlike most existing feature-based methods, the algorithm does not search for the image locations with maximum response, which may be prone to errors, but forms a tradeoff between feature likeness, global feature configuration and image reconstruction error.

The benefit of the proposed method is an increased robustness of model fitting with respect to errors in the initial feature point positions. Such residual errors are a problem when feature detection and model fitting are combined to form a fully automated face reconstruction or recognition system. We analyze the robustness in a face recognition scenario on images from two databases: FRGC and FERET.

1 Introduction

Fitting generative models such as 3D morphable models (3DMM) or active appearance models (AAM) to images of faces has turned out to be a promising approach to obtain a face-specific encoding of faces for recognition purposes. Due to the 3D representation, 3DMMs can help to recognize faces at arbitrary poses and illuminations [1]. A bottleneck in the development of automated fitting algorithms is the initialization of the optimization. While early work has started from a coarse alignment [2], later versions have relied on manually defined feature point positions [1]. Recently, a fully automated 3DMM fitting algorithm has been presented [3] which uses Support Vector Machines (SVM) for the detection of faces and facial features. However, the quality of the fit turned out to depend critically on the precision of the facial features. The goal of this paper is to integrate feature detection into the 3DMM fitting procedure.

In order to leverage the fact that 3DMM fitting can be applied to any pose and illumination, it is important to have feature detectors that are either invariant, or to rely on a set of different detectors, or - as we propose here - to have adaptive feature detectors. In our approach, the feature detector is updated by rendering an image of the current estimate of the face at the current estimate of the imaging parameters several times during the optimization, and forming

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 299–312, 2010.

templates from predefined face regions. Unlike more powerful feature detectors such as SVM or AdaBoost [4], template matching (TM) does not require multiple training samples.

The second contribution of this paper is a novel way to include facial features into model fitting. Most existing algorithms find the image position with maximum response of the feature detector and pull the corresponding point of the model towards this position. However, on difficult images, it may occur that the feature response at the correct position is not the global maximum. Therefore, we propose a strategy that forms a tradeoff between high feature detector response and a plausible overall configuration. This is achieved by including the value of the feature detector response as an additional term in the cost function of 3DMM fitting, rather than the 2D distance between the current feature position of the model and the position of the global maximum. Each feature detector response forms an additional 2D array, or layer, that is used along with the three color channel layers which form the image. On a more general level, the approach introduces a new, high-level criterion for image similarity to analysis-by-synthesis strategies. In fact, this can be implemented with any feature detector or any other local descriptor of image properties.

2 Related Work

Detection of facial features and integration into face recognition systems have been studied extensively in recent years. Still, robust feature detection in difficult imaging conditions continues to be a challenge.

AdaBoost [4] is a well-known approach for face and facial feature detection. [5] use it to first detect candidates for eyes, nose and lips separately. From the candidates, the combination with highest log-likelihood is chosen. Many approaches use coarse-to-fine strategies: [6] detect the head using AdaBoost and get a first guess of the iris position using linear regression. At the next step a weighted support vector machine (SVM), using only a small number of pixels of the whole search area, refines the iris position. [7] use a cascade of global deformation, texture fitting and feature refinement to refine eye, mouth, nose and eyebrow positions. [8] use a hierarchical face model composed of a coarse, global AAM and local, detailed AAMs for each feature for refinement. This restricts the influence of noise to the features directly nearby, and prevents it from affecting the rest of the face. [9] and [10] both use a prior distribution map analyzing AdaBoost face detection output as a starting condition, and refine the feature positions using color values and a decision tree classifier [9], or using a HarrisCornerDetector and a SVM to classify whether the detected corners belong to a feature or not [10]. [11] find facial features indirectly by using templates for parts of the face in connection with vectors that point from these regions to the positions of the features. The final feature positions are weighted combinations of the vectors.

Instead of refining the positions as in a coarse-to-fine approach, [12] combine conventional algorithms in a sequence to get better initial values for characteristic points: face detection by skin-color and luminance constraints, eye detection

by TM and symmetry enforcement, mouth and eyebrow detection both using luminance and geometry constraints. Other combined approaches have been proposed by [13] who classify SURF local descriptors with SVMs: one SVM to decide whether they belong to the face or not, followed by special SVMs for each feature. [14] combine four feature detectors (DCT, GaborWavelets, ICA, non-negative Matrix Factorization) on images at a reduced resolution. SVM is performed to get the most reliable positions (highest SVM scores) for each feature, and a graph based post-processing method is used to pick the best combination of feature positions. Refinement of the feature positions at the end is done using DCT again on full resolution.

A number of algorithms introduce local features to Active Shape Models (ASM) and Active Appearance Models (AAM): [15] extend the ASM by fitting more landmarks, using 2D-templates at some landmark positions and relaxing the shape model wherever it is advantageous. To improve the result further, they use the first alignment as start value for a second fitting with the new ASM. [16] combine ASM and Haar wavelets. [17] use a similar approach to ours, yet their 2D AAM model is designed for frontal or nearly frontal views of faces only. They form facial feature detectors from an AAM and update them in an iterative search algorithm. In each iteration, they find the feature positions with a plausible 2D configuration (high prior probability) and, at the same time, a high feature detector output. In contrast, we use a 3D model that contains additional parameters for pose and illumination, use different methods to create feature detectors and to fit the model, and we integrate the feature point criterion into a cost function that includes overall image difference for a global analysis-by-synthesis.

The first combination of feature detection with 3D morphable models (3DMM) was presented by [18] who created local feature detectors for face recognition from a 3DMM. Unlike our approach, they first reconstructed a face from a gallery image, created virtual images using the 3DMM and then relied on SVM-based local classifiers for recognition. [19] presented a patch based approach that is related to ours because it combines local feature detectors and a 3D shape model. In contrast to our algorithm, however, the feature detectors are trained prior to fitting, and the model fitting minimizes the 2D distances between image points with maximum response of the feature detectors and the corresponding model points. [20] first identify all potential feature points in the image by using SIFT as a criterion for saliency, then reject those that are similar to none of the points in the appearance model, and subsequently find the configuration of features in the image and the mapping to features of the 3DMM that has a maximum likelihood. The resulting feature locations can be used to initialize a 3DMM fitting procedure. [3] use SVM for detecting faces, estimating pose angles and finding facial features. From a number of nose point candidates, a model-based criterion selects the most plausible position. Then, these data are used to initialize a 3DMM fitting algorithm and compute 3D reconstructions. Our approach may be used in a similar general approach, but with an increased robustness to unprecise initial feature positions. In a Multi-Features Fitting Algorithm for the 3DMM, [21] use a cost function that adds color difference, edge information and

the presence of specular highlights in each pixel. The algorithm is related to ours because multiple features are used and a tradeoff is found for the match of each feature plus a prior probability term. However, we use features that are derived from facial appearance, and we update these features during the optimization.

3 Morphable Model of 3D Faces

For the reconstruction of a high-resolution 3D mesh, we use a Morphable Model of 3D faces (3DMM, [2]), which was built by establishing dense correspondence on scans of 200 individuals who are not in the test sets used below. Shape vectors are formed by the x, y, z-coordinates of all vertices $k \in \{1, \ldots, n\}$, $n = 75,972$ of a polygon mesh, and texture vectors are formed by red, green, and blue values:

$$\mathbf{S} = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T \tag{1}$$

$$\mathbf{T} = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \tag{2}$$

By Principal Component Analysis (PCA), we obtain a set of m orthogonal principal components $\mathbf{s}_j$, $\mathbf{t}_j$, and the standard deviations $\sigma_{S,j}$ and $\sigma_{T,j}$ around the averages $\overline{\mathbf{s}}$ and $\overline{\mathbf{t}}$. In this paper, only the first 99 principal components of shape and texture are used, because they cover most of the variance observed in the training set. A larger number would increase the computation time while not improving the results significantly.

In an analysis-by-synthesis loop, we find the face vector from the Morphable Model that fits the image best in terms of pixel-by-pixel color difference between the synthetic image I_{model} (rendered by standard computer graphics techniques), and the input image I:

$$E_I = \sum_{x,y} (I(x,y) - I_{model}(x,y))^2. \tag{3}$$

The squared differences in all three color channels are added in E_I. We suppress the indices for the separate color channels throughout this paper. The optimization is achieved by an algorithm that was presented in [1,2]. In each iteration, the algorithm evaluates E_I not on the entire image, but only on 40 random vertices. For the optimization to converge, the algorithm has to be initialized with the feature coordinates of at least 5 feature points.

The goal is to minimize the cost function

$$\mathbf{E} = \eta_I \cdot E_I + \eta_M \cdot E_M + \eta_P \cdot E_P \tag{4}$$

where E_M is the sum of the squared distances between the 2D positions of the marked feature points in the input image, and their current positions in the model. E_P is the Mahalanobis distance of the current solution from the average face, which is related to the log of the prior probability of the current solution. η_I, η_M and η_P are weights that are set heuristically: The optimization starts with a conservative fit (η_M and η_P are high), and in the final iterations $\eta_M = 0$ and η_P is small.

The algorithm optimizes the linear coefficients for shape and texture, but also 3D orientation and position, focal length of the camera, angle, color and intensity

of directed light, intensity and color of ambient light, color contrast as well as gains and offsets in each color channel.

4 Self-Adapting Features

Our proposed self-adapting feature approach is built on top of the 3DMM and introduces a novel criterion in the cost function. The goal is to reduce the influence of the (potentially unreliable) initial feature positions that are used in E_M: they are only used for the first coarse alignment of the head, and discarded later. After coarse alignment and a first estimation of the illumination, the term E_M in the cost function (Eqn. 4) is replaced by new E_{F_i}, that will be explained below, with $i = 1 \dots 7$, for the set of 7 feature positions to be refined, weighted with η_F:

$$\mathbf{E} = \eta_I \cdot E_I + \eta_F \cdot \sum_{i=1}^{7} E_{F_i}(x_{F_i}, y_{F_i}) + \eta_P \cdot E_P \tag{5}$$

The features are: the tip of the nose, the corners of the mouth, and the inner and outer corners of the eyes. For feature point i, we know which vertex k_i of the model it corresponds to, and using perspective projection we get the current position (x_{F_i}, y_{F_i}) in the image I_{model}.

Once every 1000 iterations, the entire current fitting result I_{model} is rendered, and templates are cut out around the current feature positions (x_{F_i}, y_{F_i}). Template sizes are pre-defined relative to the head size s_H (distance between a vertex on the top of the forehead and one on the bottom of the chin, in pixel units): eyes: $(\frac{1}{9}s_H) \times (\frac{2}{9}s_H)$, nose: $(\frac{1}{18}s_H) \times (\frac{1}{18}s_H)$ and mouth: $(\frac{2}{9}s_H) \times (\frac{1}{9}s_H)$. We chose these sizes to make sure that each template contained enough diagnostic features, such as part of the eyebrows in the eye template.

The new E_{F_i} in (5), based on TM, are

$$\mathbf{E_{F_i}}(x_{F_i}, y_{F_i}) = 1 - \mathbf{C_{F_i}}(x_{F_i}, y_{F_i}). \tag{6}$$

where C_{F_i} is the normalized cross correlation [22], which we found to be more reliable than alternative choices:

$$\mathbf{C_F}(x,y) = \frac{\sum_{(p,q)\in R} (I(x+p, y+q) \cdot R(p,q)) - N \cdot \bar{I}(x,y) \cdot \bar{R}}{\sqrt{\sum_{(p,q)\in R} (I(x+p, y+q))^2 - N \cdot (\bar{I}(x,y))^2 \cdot \sigma_R}} \tag{7}$$

where I is the original image and $\bar{I}(x,y)$ its local mean value around the current position (x,y) in a template-sized area, R is the current template (or reference image) and $\bar{R}$ its mean value (over all (p,q)), σ_R is the variance of the template values and N is the number of template values ($width \cdot height$). Only $\bar{I}$ has to be computed for every (x,y). The other three components ($\bar{R}$, σ_R, N) can be precomputed. Note that $\forall (x,y) \in I : \mathbf{C_F}(x,y) \in [-1,1]$, with 1 representing a maximum match and -1 a maximum mismatch. For color images, $\mathbf{C_F}(x,y) = \frac{1}{3}(\mathbf{C_{F,red}}(x,y) + \mathbf{C_{F,green}}(x,y) + \mathbf{C_{F,blue}}(x,y))$.

The weight η_F is constant and scales the sum of all E_{F_i} to the same range of values as the image difference E_I.

Templates R and cross correlations $\mathbf{C_{F_i}}$ are updated once every 1000 iterations of the fitting algorithm. To reduce computation time, $\mathbf{C_{F_i}}(x, y)$ for each feature i is calculated only in a region of interest (ROI: $\frac{1}{9}s_H \times \frac{1}{9}s_H$) around the current position (x_{F_i}, y_{F_i}). Even in the first iteration, these positions can be assumed to be approximately correct, and also the head size s_H that defines the (constant) template size will be in the right order of magnitude, due to the vague initial feature coordinates.

In intermediate iterations, $\mathbf{C_{F_i}}$ remain fixed, but the positions (x_{F_i}, y_{F_i}) for looking up $\mathbf{C_{F_i}}(x_{F_i}, y_{F_i})$ will change. This reflects the fact that the locations of feature points may change faster during the optimization than the appearances of features do. Fig. 1 gives an overview of the algorithm.

Fig. 2 shows how the templates (here: outer corner of the left eye) change over fitting iterations when fitting to different images (rows in Fig. 2). Over the first six template-adaptions, not much change is observed. At the seventh template in each row, the change is already visible at the eyebrow, after the eighth and ninth adaption the whole templates changed significantly. The major change can be observed at step eight and nine, because this is where fine adjustment starts: The head model is broken down in different regions, and these are optimized separately (see [2]).

Fig. 3 shows an example of how the cross correlation result, matching the left corner of the left eye, changes over the fitting iterations. The detail belongs to the result of the third line of Fig. 8. There, first the position of the corner of the eye has been displaced to the right to evaluate robustness. As the fitting proceeds, it moves to the left and upward until it reaches the correct position eventually. The position of the ROI also shows where the feature has been positioned when computing the cross correlation. The ROI position reveals the drift of the feature to the correct position.

It can be seen that the cross correlation turns into a single, wide optimum as the template adapts to the appearance in the image. Note that if the model adapts perfectly to the feature in the input image, $\mathbf{C_{F_i}}(x, y)$ will converge to the autocorrelation function, and the width of maxima and minima will be determined by the frequency spectrum of the template.

5 Results

We tested our algorithm on 300 randomly chosen images from the FRGC data base [23], using three images per person and a set of 50 women and 50 men. The only constraint in random selection was that the person did not show an extreme facial expression. Typical examples of the randomly chosen samples, some of them in difficult imaging conditions (focus, illumination, expressions) can be found in Fig. 4. The database contains front view images only. We show results on non-frontal views later in this section.

For ground truth in every image, five feature positions (outer corners of the eyes, nose and corners of the mouth) were labelled manually. To simulate scenarios with an unreliable initial feature detector, we perturbed the feature positions randomly:

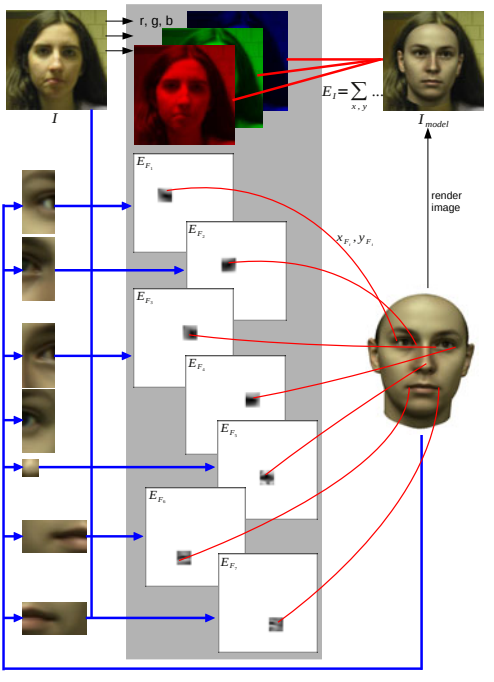

Fig. 1. Self-adapting feature layers: Blue arrows show actions performed only every 1000 iterations: Templates are cut out from the current fitting result. They are compared to the original image I using normalized cross correlation at a certain ROI and from these, the 'feature layers' are generated. Red lines show actions performed every iteration: I_{model} is compared to I, and for each feature i the error value E_{F_i} is taken from the corresponding 'feature layer' at the current feature position (x_{F_i}, y_{F_i}).

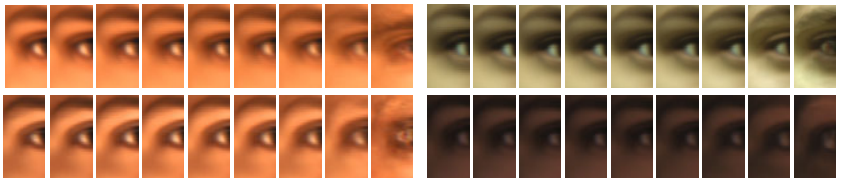

Fig. 2. Templates changing over fitting iterations. Each line is from fitting to one input image. In the first two examples (upper row), convergence was correct, in the last two (lower row), the corner of the eye moved to the eyebrow.

1. randomly select two (of the five) features to be perturbed
2. randomly select a displacement direction for each
3. displace feature positions by a fixed distance

In three different test conditions, we used distances of 5%, 12% and 25% of the vertical distance between eyes and nose. This corresponds to distances of 0.2cm, 0.48cm and 1.0cm in reality on an average sized head. The perturbation ranges

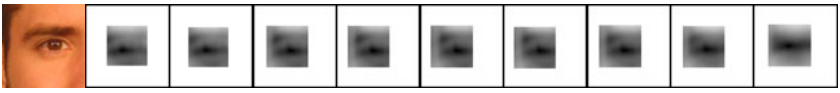

Fig. 3. Cross correlation results in the ROI, changing over fitting iterations. Dark pixels indicate good matches. These results correspond to the templates in the first line of Fig. 2).

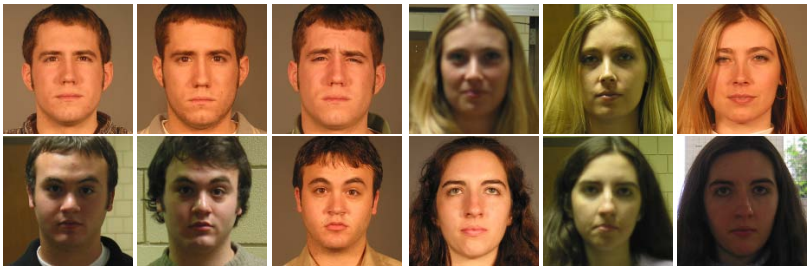

Fig. 4. Typical examples of images per person: Each line shows three pictures of the same individual [23]. Here not the whole pictures, but only the facial regions (scaled to the same size) are shown.

are visualized as the radii of the circles in the upper row of Fig. 5. The lower row shows a typical example for each test condition. By using displacement distances relative to the eye-nose distance in the image, rather than fixed pixel distances, we were able to use images at different resolutions.

To have an independent criterion for the quality of the reconstructions, we evaluate recognition rates from model coefficients in an identification task. Given the linear 3DMM coefficients for shape and texture of the entire face and the facial regions (eyes, nose, mouth and surrounding area), which are concatenated into coefficient vectors $\mathbf{c}$, the algorithm finds the individual from the gallery set with a minimum distance, measured in terms of a cosine criterion $d = \frac{\langle \mathbf{c}_1, \mathbf{c}_2 \rangle}{\|\mathbf{c}_1\| \cdot \|\mathbf{c}_2\|}$ (see [1,3]). For each probe image, a comparison with the other two images of that person and with all three images of all 99 other individuals is performed.

Recognition is tested with the standard 3DMM fitting algorithm ([1], see Section 3) and with our new SAFL approach for the manually marked feature positions and each perturbation range. The percentages of correct identification can be found on the left side of Fig. 7.

Due to the difficult imaging conditions, the overall recognition rate is below 50%. In the unperturbed case, both the standard algorithm and the new self-adapting feature layers (SAFL) deliver similar results, indicating that SAFL do not downgrade the system when correct feature positions are given. However, with perturbed features, the recognition rate for the standard algorithm rapidly decreases as the displacements get larger. In contrast, SAFL identification rates remain stable. This demonstrates that SAFL increases the robustness of the fitting for face recognition.

Fig. 5. Perturbation ranges and typical examples of perturbed positions: In the upper row circles mark the perturbation ranges of the three different test conditions, from left to right: 5%, 12% and 25%. In the lower row green crosses mark manually labelled feature positions and red crosses mark perturbed positions.

We have also evaluated the distances between the ground truth feature positions and the optimized positions after fitting. Fig. 6 shows the distribution of the average 2D distances of the five features in each test image: The vertical axis is the absolute number of test images (out of a total of 300) where the average feature distance is below the distance threshold indicated on the horizontal axis.

If we do not perturb the starting positions of features, most test images have an average error in final feature positions of 5% to 10% of the vertical distance between eyes and nose, which corresponds to approximately $2mm$ to $4mm$. The standard algorithm performs slightly better than SAFL because E_M keeps the features fixed to the ground truth positions during part of the optimization. It should be noted that the ground truth positions may have some residual uncertainty, because it is difficult to identify corresponding feature positions (pixel in the image - vertex on the model) exactly by hand. This may explain why the benefit of SAFL in this evaluation criterion becomes visible only on a larger scale of feature distances, i.e. when larger perturbations are applied (Fig. 6, second diagram). These results are consistent with the face identification rates on the left of Fig. 7, where we found similar performance for unperturbed initial features, but a significant improvement for perturbed features.

To demonstrates that SAFL is not restricted to frontal views we did some additional tests on the FERET database. The setting was chosen like for the FRGC data. In a rank 1 identification experiment (1 out of 194) we used *ba* images as gallery and *bb* (rotated views with a mean rotation angle ϕ of 38.9°, cf. [1]) as query images also considering perturbation ranges from 0% to 25%. The percentages of correct identification can be found on the right of Fig. 7. Compared to [1] the recognition rates of both (standard and SAFL) are lower. This is due to the fact that in [1] more than the five feature positions are used, e.g. at the ear and at the contour, which are useful for non-frontal views. But our goal here is to demonstrate the usefulness of the new algorithm compared with the standard one.

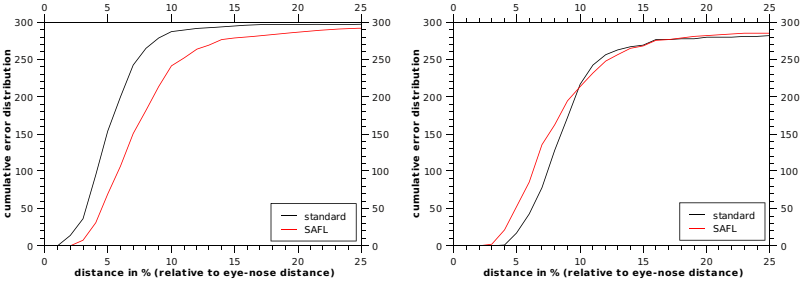

Fig. 6. Movement of feature positions: Left diagram: The manually labelled feature positions (without perturbation) were used for initialization of the reconstruction. Right diagram: 25% perturbation. x-axis: Average distance between the manually labelled positions and the resulting positions after reconstruction for a given test image. y-axis: Cumulative error distribution (absolute number of images with distance below threshold, total 300.) Black line: standard 3DMM fitting algorithm, red line: proposed algorithm SAFL.

To test the new algorithm in a *real world scenario* we chose the feature detector of [3] to automatically detect the feature positions on the 300 faces taken from the FRGC database. Performing a rank 1 identification experiment again the standard algorithm delivers a recognition rate of $29.\bar{6}\%$ and the new algorithm yields a recognition rate of 39.0%. This results are comparable to the recognition rates of the former experiment with random perturbation of 12%.

To confirm our choice of making the features self-adapting and of using the image layer approach rather than considering only the position of maximum feature response, we evaluated some alternative versions of the algorithm:

a. Standard algorithm, but the initial (perturbed) feature positions (which contribute to E_M) are replaced after iteration 1000 by the position of the maximum output of template matching (TM). The idea is that a single TM early in the process would be enough to refine the perturbed feature positions. The template is created after a coarse estimation of pose, lighting and appearance.
b. Use self-adapting templates that are adapted every 1000 iterations, but consider only the maximum TM output rather than layers E_{F_i}, and use it in E_M instead of the initial features (standard algorithm). This condition tests whether the layer approach E_{F_i} is superior to E_M which just pulls features to the positions of maximum TM output.
c. Compute the cross correlation results only once for each feature, and use this to get $E_{F_i}(x_{F_i}, y_{F_i})$ for the rest of the reconstruction algorithm. This condition verifies the benefit of adaptiveness in the SAFL algorithm.

Table 1 shows the recognition rates of the standard algorithm, the proposed SAFL and the additional tested versions a,b and c listed above. The results

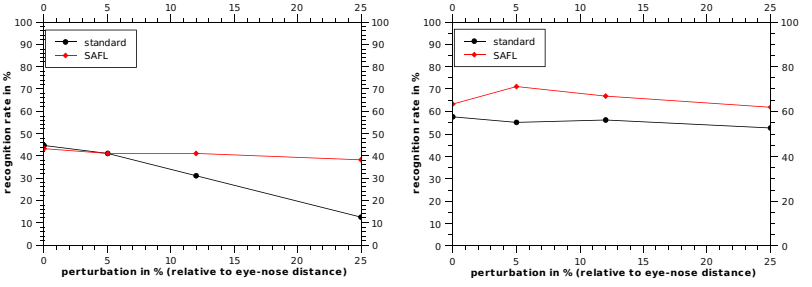

Fig. 7. Comparison of recognition rates: measured in terms of correct identifications (one out of n individuals). The left diagram is for frontal faces of the FRGC database ($n = 100$) and the right diagram is for non-frontal faces using images from the FERET database *ba* as gallery and *bb* as query images ($n = 194$). The accuracy in the right plot is better than in the left plot because FERET is easier to classify (the images are from a single session).

Table 1. Recognition rates: percentage of correct identifications for all algorithms tested on all perturbation ranges

perturbation in %	0	5	12	25
standard	44.6	41.0	31.0	12.3
a	25.0	25.3	25.0	19.6
b	14.0	14.3	11.3	11.3
c	42.0	40.6	39.0	33.0
SAFL	43.3	41.0	41.0	38.3

Table 2. Computation times: in seconds, measured on an Intel^R Core^{TM}2 Duo CPU E8300 @ 2.83GHz (single threaded)

facial region	std.	a	b	c	SAFL
541^2px	64	92	103	146	160
1054^2px	67	120	232	331	456

of the versions (a) and (b) are much lower than all others, indicating that the cost function E_{F_i} performs better than searching for the maximum output of TM only. The recognition rates of setting (c) come close to the ones of the SAFL approach, but are still inferior, showing that self adaptation is useful. We conclude that both main ideas proposed in this paper make a significant contribution to the stability of 3DMM fitting in a recognition scenario with partially unreliable initial features.

The computation times for the different approaches according to different facial region sizes can be found in Tab. 2. When TM is used, computation times depend critically on the size of the facial region. It would have been worth considering to perform template matching at a lower image resolution.

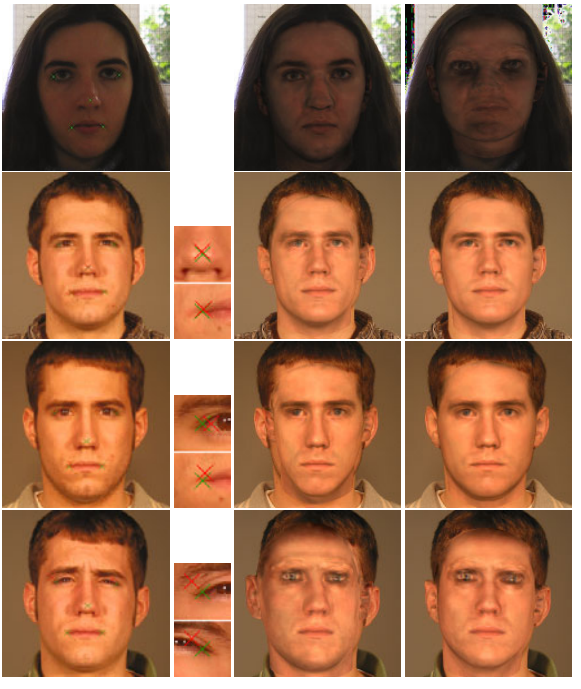

Fig. 8. Reconstruction examples: Each row shows 3DMM fitting for one test image. In the examples in row 1, 2, 3, 4, we used 0%, 5%, 12% and 25% perturbation, respectively, relative to eye-nose distance. From left to right: positions marked on the input image, close-ups of the perturbed feature positions (green: manually labelled, red: perturbed position used), reconstruction with the standard algorithm, reconstruction with the new SAFL approach.

Fig. 8 shows 4 reconstruction results of frontal views. For lack of space, we show only one perturbation level per example in this figure in order to give an idea of what the reconstructions look like but more results can be found in the supplementary material. In the left column, the feature positions are marked on the input images with colored crosses: green for manually labelled positions and red for perturbed positions. The second column shows close-ups of the features randomly chosen for perturbation. In the first row, the manually labelled feature positions were used. Here the SAFL approach got into a local minimum, moving the eyes to the eyebrow positions. In the second row, two randomly chosen feature positions were perturbed 5%. Here the perturbation is quite small, but SAFL outperforms the standard algorithm in reconstruction. In the third row, two randomly chosen feature positions were perturbed 12%. Here it can be seen how much the perturbed feature positions influence the standard algorithm. The reconstruction using SAFL is plausible. In the fourth row, two randomly chosen feature positions were perturbed 25%. We would like to add that both

Fig. 9. Reconstruction examples of rotated views: Examples were chosen randomly out of the reconstruction results with a perturbation range of 12%. Each pair of images shows reconstructions using the standard algorithm (left) and SAFL (right).

algorithms may produce suboptimal results occasionally, and we selected four typical examples here.

Results of non-frontal views are shown in Fig. 9. We show only this randomly chosen examples with a perturbation range of 12% in this figure in order to give an idea of what the reconstructions look like but more results can also be found in the supplementary material. At the upper line the SAFL approach improved the reconstruction. On the left side it is obvious but on the right side it can only be seen at the forehead and at the chin. At the lower line the SAFL approach does not really improve the reconstruction. On the left side the model fits better to the image (it is rotated) but it is still too small and on the right side it fits better at the forehead and at the ear but chin and nose are deformed.

6 Conclusion

We have presented a new approach for using feature detectors in 3DMM fitting. The algorithm involves adaptive features, which is crucial to leverage the advantages of 3DMMs, and it is based on a new type of cost function that forms a tradeoff between feature similarity and some more global criteria such as geometric configuration, correct reproduction of color values and high prior probability.

The evaluation is based on a scenario where the 3DMM fitting is initialized by a set of potentially unreliable feature detectors, and the algorithm iteratively refines the feature positions. The results indicate that the proposed algorithm improves recognition rates significantly. The second part of our evaluation is focused on the contributions of different design options in our algorithm, and it demonstrates that both the adaptiveness and the new type of cost function increase the robustness.

References

1. Blanz, V., Vetter, T.: Face recognition based on fitting a 3d morphable model. IEEE Trans. on Pattern Analysis and Machine Intelligence 25, 1063–1074 (2003)
2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Computer Graphics Proc. SIGGRAPH'99, pp. 187–194 (1999)
3. Breuer, P., Kim, K.I., Kienzle, W., Schölkopf, B., Blanz, V.: Automatic 3d face reconstruction from single images or video. In: FG'08 (2008)
4. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR'01 (2001)
5. Erukhimov, V., Lee, K.C.: A bottom-up framework for robust facial feature detection. In: FG'08 (2008)
6. Nguyen, M.H., Perez, J., la Torre Frade, F.D.: Facial feature detection with optimal pixel reduction svms. In: FG'08 (2008)
7. Zuo, F., de With, P.H.: Facial feature extraction using a cascade of model-based algorithms. In: AVSBS'05 (2005)
8. Tang, F., Wang, J., Tao, H., Peng, Q.: Probabilistic hierarchical face model for feature localization. In: WACV'07 (2007)
9. Wimmer, M., Mayer, C., Radig, B.: Robustly classifying facial components using a set of adjusted pixel features. In: FG'08 (2008)
10. Ardizzone, E., Cascia, M.L., Morana, M.: Probabilistic corner detection for facial feature extraction. In: Foggia, P., Sansone, C., Vento, M. (eds.) Image Analysis and Processing – ICIAP 2009. LNCS, vol. 5716, pp. 461–470. Springer, Heidelberg (2009)
11. Kozakaya, T., Shibata, T., Yuasa, M., Yamaguchi, O.: Facial feature localization using weighted vector concentration approach. In: FG'08 (2008)
12. Oh, J.S., Kim, D.W., Kim, J.T., Yoon, Y.I., Choi, J.S.: Facial component detection for efficient facial characteristic point extraction. In: Kamel, M.S., Campilho, A.C. (eds.) ICIAR 2005. LNCS, vol. 3656, pp. 1125–1132. Springer, Heidelberg (2005)
13. Kim, D.H., Dahyot, R.: Face components detection using surf descriptors and svms. In: IMVIP'08 (2008)
14. Celiktutan, O., Akakin, H.C., Sankur, B.: Multi-attribute robust facial feature localization. In: FG'08 (2008)
15. Milborrow, S., Nicolls, F.: Locating facial features with an extended active shape model. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 504–513. Springer, Heidelberg (2008)
16. Zuo, F., de With, P.: Fast facial feature extraction using a deformable shape model with haar-wavelet based local texture attributes. In: ICIP'04 (2004)
17. Cristinacce, D., Cootes, T.: Feature detection and tracking with constrained local models. In: BMVC'06 (2006)
18. Huang, J., Heisele, B., Blanz, V.: Component-based face recognition with 3d morphable models. In: Proc. of the 4th Int. Conf. on Audio- and Video-Based Biometric Person Authenticitation, Surrey, UK (2003)
19. Gu, L., Kanade, T.: 3d alignment of face in a single image. In: CVPR'06 (2006)
20. Romdhani, S., Vetter, T.: 3d probabilistic feature point model for object detection and recognition. In: CVPR'07 (2007)
21. Romdhani, S., Vetter, T.: Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: CVPR'05 (2005)
22. Burger, W., Burge, M.J.: Digital Image Processing. Springer, New York (2008), `http://www.imagingbook.com`
23. Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: CVPR'05 (2005)