Automatic 3D Face Reconstruction from Single Images or Video

Pia Breuer[†], Kwang-In Kim[‡], Wolf Kienzle[‡], Bernhard Schölkopf[‡], Volker Blanz[†]
[†]University of Siegen, [‡]Max Planck Institute for Biological Cybernetics
[†]{pbreuer, blanz}@informatik.uni-siegen.de
[‡]{kwangin.kim, kienzle, bernhard.schoelkopf}@tuebingen.mpg.de

Abstract

This paper presents a fully automated algorithm for reconstructing a textured 3D model of a face from a single photograph or a raw video stream. The algorithm is based on a combination of Support Vector Machines (SVMs) and a Morphable Model of 3D faces. After SVM face detection, individual facial features are detected using a novel regression- and classification-based approach, and probabilistically plausible configurations of features are selected to produce a list of candidates for several facial feature positions. In the next step, the configurations of feature points are evaluated using a novel criterion that is based on a Morphable Model and a combination of linear projections. To make the algorithm robust with respect to head orientation, this process is iterated while the estimate of pose is refined. Finally, the feature points initialize a model-fitting procedure of the Morphable Model. The result is a high-resolution 3D surface model.

1. Introduction

For reconstruction of 3D faces from image data, there are a variety of approaches that rely on different sources of depth information: some perform triangulation from multiple simultaneous views, *e.g.* stereo or multiview-video methods. Others use multiple consecutive monocular views in video streams for structure-from-motion or silhouette-based approaches. Finally, there are algorithms that rely on single still images only, for example by exploiting shading information (shape-from-shading) or by fitting face models to single images. In this paper, we propose an algorithm for 3D reconstruction that

- can be applied either to single still images or to raw monocular video streams,
- operates at a wide range of poses and illuminations,
- involves zero user interaction,
- produces close-to-photorealistic 3D reconstructions.

To perform this task, we are building a system which integrates two well-known techniques: Support Vector Machines (SVMs) and 3D Morphable Models (3DMM). The processing steps of our algorithm (Figure 1) are

1. Face Detection using SVM
2. For video data: selection of the best frame
3. Coarse estimate of pose based on regression
4. Facial component detection using regression and classification
5. Selection of the most plausible combination of components based on Gaussian distributions
6. Selection of the most plausible nose position based on a Morphable Model
7. Fast fit of the 3DMM for pose estimation
8. Iteration of step 4 to 7 until pose estimation is stable
9. Full 3D reconstruction.

The integration involves several extensions of the system parts: For facial component detection, we train an array of regressors for each component, as opposed to only one regressor used in existing algorithms. The detection results are then refined by a classification-based approach which combines SVM-based component detections and the prior distribution of their joint configurations. We train and test the classifiers based on the results of the regression-based method. This helps to filter out image regions far from the facial components and accordingly prevents the training of an SVM from being disturbed by irrelevant image information. The component detectors are view-specific, and they are trained on synthetic data produced by the 3DMM. Moreover, we propose a novel, model-based criterion for plausibility of component configurations. This involves a new method for estimating texture from images (Section 4) that is more efficient than the iterative model-fitting that we use for the final reconstruction.

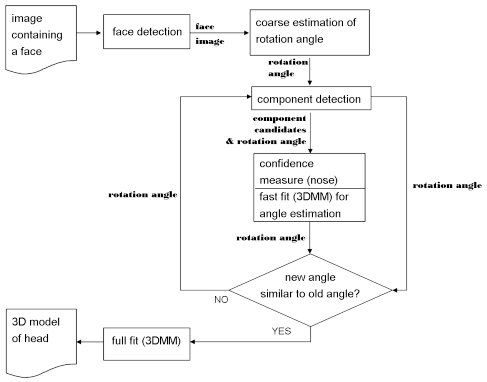

Figure 1. Flow diagram of the automatic face fitting system.

For reconstruction from video, our system selects a single, suitable frame from the video automatically, and performs model-based reconstruction from this frame. This is in contrast to previous work in model-based shape reconstruction from monocular video, which involved an analysis of multiple frames, such as model-driven bundle-adjustment [13], least-squares reconstruction from multiple successive frames [11], structure-from-motion with subsequent refinement by a deformable face model [8], nonrigid structure-from-motion with intrinsic model constraints [5] and feature tracking and factorization of the tracking matrix for non-rigid shape estimation [6]. Zhang et al. [28] presented an algorithm that involves tracking, model fitting and multiple-view bundle adjustment. Many of these algorithms require manual interaction such as a number of mouse clicks.

Unlike previous model-based algorithms for 3D face reconstruction from single images [3, 4, 17], the combined algorithm no longer requires manual rigid pre-alignment of the 3D model or manual labeling of features on the 2D image. Xiao et al.[25] presented a combination of Active Appearance Models and 3D Morphable Models that tracks features in realtime in videos, and reconstructs a face mesh for each frame. This system is very impressive, but so far the authors have only used a low-resolution face mesh that does not generate photo-realistic face reconstructions.

An automated algorithm for aligning a 3D face model to a single image has been proposed by Gu and Kanade [14]. Based on a Morphable Model of 3D shape and view-based patches located at 190 points, the model is iteratively aligned with the features in the image using an EM algorithm. The Morphable Model, the feature-based approach and the usage of a pose estimate are similar to our system. However, we use different feature detection algorithms and a different optimization strategy and for the final reconstruction, we fit all model vertices to the image in an analysis-by-synthesis loop that optimizes all facial details and compensates for lighting and other imaging parameters.

Xin et al. [26] recovered 3D face shape from a video containing a face rotating from frontal view to profile view. Based on image sequence segmentation, they estimate 2D features, head poses and underlying 3D face shape, using a morphable model resembling ours. In contrast, we use features for initialisation only, and then fit to greylevels in the image for more detailed reconstruction.

Heisele et al. used synthetic images to train SVM-based viewpoint-independent feature detectors [15]. For more related work in the feature detection literature, which involves SVM-based methods [18, 21], we refer the readers to the excellent survey of Yang et al. [27].

2. Detection of faces and facial components

2.1. Face detection

As a first step, a face detector is applied to the input image or video. For this purpose, two publicly available face detection libraries for Matlab were considered: an approach based on SVMs [16], and an implementation of the widely used Boosting based detector [24]. We chose the SVM implementation which also returns a confidence value together with each detected face. In our fully-automatic system, the confidence estimates are used to resolve ambiguities: if there are more than one detections in an image or a video, we discard all but the one for which the detector is most confident. The input image is cropped to a square region around the most confident position, and is then rescaled to 200×200 pixels. This is the reference coordinate system used in all subsequent processing steps.

2.2. Categorization of faces based on rotation angles

Faces show a significant variation in shapes in the viewing plane depending on the 3d rotations. To facilitate the training of subsequent *view-based* component detection (Secs. 2.3-2.4), the face images are categorized based on the rotation angles such that a component detector is trained for each category. This allows each component detector to focus on similar views. Training categories are found by uniformly quantizing the interval of tangent values of horizontal rotation $[-1, 1]$ (corresponding to $[-45°, 45°]$) into seven bins.

For testing, the rotation angle of the input face image is estimated by regression. The input face image is scaled down to 40×40 and the Kernel Ridge Regression (KRR) is applied to get the tangent of the angle. The estimated angle is then used to choose a proper component detector. Because the KRR provides only rough estimation of angles with an average error rate around $7°$, the estimated angles are refined later by iteration through alternating the component detection and face fitting (cf., Fig. 1 and Sec. 4).

2.3. Facial component detection based on regression

The third stage computes position estimates of eye and mouth corners, which we here refer to as the Components Of Interest (COI), in the 200×200 image. For this purpose, we developed a novel algorithm, which can be seen as a generalization of the regression method proposed by [12]. It predicts the position of a COI from pixel intensities within a $k \times k$ window. Invariance under intensity changes is achieved by subtracting the mean value from each window, and dividing it by its standard deviation. The KRR is adopted for this purpose (see Sec. 2.6 for details). The novelty of our approach is that for each facial component we train an array of $12 \times 12 = 144$ regressors, as opposed to only one [12]. All of them predict the same quantity, but they are trained on different $k \times k$ regions on the 200×200 image, evenly spaced on a 12×12 grid (see Fig. 2, left image). To predict the position of that component in a test image, all 144 estimates are computed, and then binned into 1-pixel-sized bins. The bin with the most votes is chosen as the predicted location. The rationale behind this is that faces cannot be arbitrarily deformed, and thus the appearance of facial regions away from the component in question can be informative about its position. The use of 144 regressors

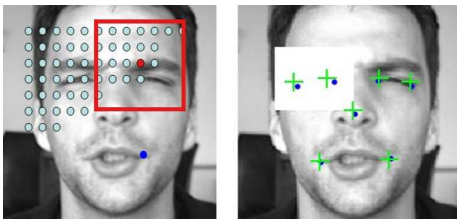

Figure 2. Regression-based detection of facial components. *Left*: illustration of the regressor array. For each component (e.g., a corner of the mouth, here marked dark blue) 144 regressors are learned. Each one operates on a different image region, centered at one of the light blue points. *Right*: prediction on a corrupted test image (the left eye region is covered with a white rectangle). Plus marks indicate desired component locations.

makes the detector extremely redundant and therefore robust to occlusions and other local changes. This effect is shown in Fig. 2.

2.4. Refinement of component detection based on classification

The regression-based approach is fast and robust. However, it turns out that its accuracy is not sufficient for subsequently fitting the 3D face model. In the present section, we present a classification-based method which is built on top of the regression-based component detection. The basic idea is to scan the input face image I with a small window and classify the central pixel of the window, using an SVM, as belonging to either the COI, or background.

We generated training examples for the classifier by sampling small windows from locations with pre-defined distances of the ground truth locations. Positive examples have their reference point in a 3×3 window around the ground truth location; negative examples, on the other hand, have it inside a 29×29 window, excluding the central 9×9 such that the training is not affected by ambiguous patterns. The 29×29 window for sampling negative examples is motivated by the typical location error of the regression-based method which lies within the range of 0 to 12 pixel distances from the true COI locations.

In the classification stage, instead of scanning the entire face image, the search space for a given COI is restricted as a 25×25 window surrounding the regression-based detection. While it is more accurate than the regression-based method, the problem of the classification method is that it does not automatically single out a detection. Instead, it either produces a detection blob around the COI or sometimes produces no detection. Thus, if the detection blob for a given COI is too small (or zero), we adjust the decision threshold of the SVM such that the blob size is larger than or equal to a predetermined threshold r.

In the next step, the individual detection results need to be combined, taking into account a prior in the space of joint configurations. A configuration of eye and mouth corners constitutes a 13-dimensional vector $H = (h_1, \ldots, h_6, h_\phi) = ((h_1^x, h_1^y), \ldots, (h_6^x, h_6^y), h_\phi)$ ((x,y)-coordinates values for six components and the horizontal rotation angle ϕ). Then, the best detection vector is obtained by first generating 100,000 random vectors (by sampling two dimensional vectors ((x,y)-coordinates)) from each component blob and concatenating them to constitute 12(+1)-dimensional vectors, and then choosing the maximizer of the following objective function.

$$C(H) = \alpha \sum_{i=1,\ldots,6} \log\left(\frac{1}{1+\exp\left(-g^i(W_i)\right)}\right) - M(H), \tag{1}$$

where $g^i(W_i)$ is the real-valued output of the i-th SVM for the input image window W_i corresponding to the coordinate h_i, and $M(H)$ is the Mahalanobis distance of configuration H to the mean of a Gaussian distribution estimated based on training configurations. We motivate this cost function as follows. Suppose we want to obtain the most probable configuration

$$H^* \quad = \quad arg\max P(H = O|I),$$

where $O = (o_1, \ldots, o_6, o_\phi)$ is the unknown ground truth configuration. The cost function (1) is then obtained as a result of the following series of approximations

$$\begin{aligned} C(H) \quad &\approx \quad P(W_1, \ldots, W_6|H = O)P(H = O) \\ &\approx \quad \prod_{i=1,\ldots,6} P(W_i|h_i = o_i, h_\phi = o_\phi)P(H = O) \\ &\approx \quad \sum_{i=1,\ldots,6} \log P(W_i|h_i = o_i, h_\phi = o_\phi) - M(H), \end{aligned}$$

where the first line applies the Bayes formula and replaces the image by the small windows (W_i), the second line corresponds to an independence assumption of the component likelihoods plus pre-categorization of face images, and the third line assumes a Gaussian distribution of the configurations H. The last step is to substitute $P(W_i|h_i = o_i, h_\phi = o_\phi)$ with the component detection posterior $P(h_i = o_i, h_\phi = o_\phi|W_i)$ calculated based on Platt's method [20], assuming a uniform distribution over h_i, h_ϕ within a small region generated by the rough detector.

This technique of replacing the likelihood by the posterior estimated from a discriminative classifier has shown to improve the discrimination capability of generative models [22, 9].

In the computation of $P(H)$, the (x,y)-coordinate of the outer corner of the left eye was used as the origin $(0,0)$, and the width and height of the bounding box for the joint configuration were normalized by dividing by the width of the original box. Accordingly, H is a 11-dimensional vector.

In addition to the eye and mouth corners, we also use nose tip for fitting the 3D face model. However, it turns out that the nose is very hard to identify from local features alone which both the regression-based and classification based methods rely on. Accordingly, the nose detection is performed in a separate step which will be explained in

Sec. 4. To facilitate this, we generate several nose candidates based on detected eye and mouth corners. A conditional Gaussian model of nose tip location given the eye and mouth corners are estimated from which the nose candidates are obtained by thresholding based on Mahalanobis distance.

2.5. Reduced set method

In the preliminary experiments, the number of support vectors for SVM-based component detectors ranged from around 2,000 to 5,000. This resulted in processing time of more than 10 minutes per image. To reduce the time complexity, this paper adopts the *reduced set method* [7, 23] which finds a *sparse* solution

$$g(\cdot) = \sum_{i=1}^{N_{\mathbf{b}}} \alpha_i k(\cdot, \mathbf{b}_i)$$

as an approximation of the original SVM solution f. In the original reduced set method, the approximation error is measured by $\mathcal{F}(g) = \|g - f\|^2_{\mathcal{H}}$, where $\mathcal{H}$ is the *Reproducing Kernel Hilbert Space* corresponding to the kernel function of SVM. However, this does not consider the distribution of data $P(\mathbf{x})$. In this paper, we find the solution $g(\in \mathcal{H})$ as the minimizer of a new cost functional

$$\mathcal{G}(g) = \sum_{\mathbf{x}\in\mathbf{X}} (g(\mathbf{x}) - f(\mathbf{x}))^2,$$

where $\mathbf{X}$ is a finite set of data sampled according to $P(\mathbf{x})$. Informally, this cost functional allows us to find the sparse solution by putting more emphasis on high-density regions.

The solution g is found by first initializing the basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_{N_{\mathbf{b}}}\}$ by k-means on a set of *unlabeled* data $\mathbf{X}_U$, and then performing the gradient descent on the regularized cost functional

$$\mathcal{F}(g) = \lambda\|g\|^2_{\mathcal{H}} + \sum_{\mathbf{x}\in\mathbf{X}_U} (g(\mathbf{x}) - f(\mathbf{x}))^2,$$

where $f(\mathbf{x})$ denotes the output of the SVM at $\mathbf{x}$. λ is obtained by cross-validation. For all component detectors, we set the number of basis vectors $N_{\mathbf{b}}$ to 200. To gain an insight into the performance of the proposed reduced method, the classification error rates for a test set which is disjoint from the training set is compared with two implementations of existing reduced set method for case of the outer corner of left eye component: fixed point iteration ([23]) and gradient descent ([7]). Fig. 3 summarizes the results.

2.6. Training

For training the rough angle estimator and the component detectors, we used 3,070 synthetic images obtained from 307 3D faces. 107 of these 3D faces were reconstructed from single images of the FERET database [19] using the algorithm described in [4], and 200 3D faces were from laser scans [4]. The synthetic images were rendered from the 3D faces by varying the azimuth angle randomly in $[-45°, 45°]$ with a uniform distribution, varying

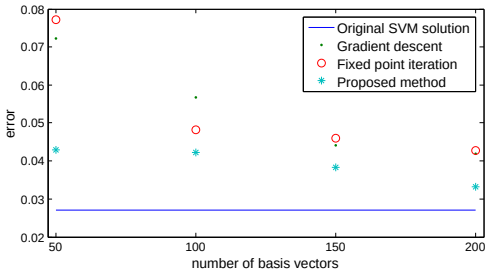

Figure 3. Comparison between different reduced set methods.

pitch in $[-20°, 20°]$, and the azimuth and height of the light in $[-45°, 45°]$. Also, the relative contribution of directed and ambient light was varied randomly. Gaussian kernels ($K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x}-\mathbf{y}\|^2)$) were utilized for the KRRs and SVMs whose parameters were found by cross-validation. For the KRR componenet detector, the size of the regression input k and a scaling factor s by which the image was downscaled before sampling the $k \times k$ window were set to 9 and 0.1, respectively. For each SVM detector, around 20,000 to 40,000 training patterns are collected from these 3,070 faces. The input window size for eye and mouth detection was determined as (31×31) which for the eye case, roughly corresponds to the average length of the eye in (200×200)-size frontal face images. The blob size threshold r and α in Eq. (1) were empirically set to 25 and 4, respectively. We did not find the parameters to affect the results significantly, but would expect that a future choice by cross validation could somewhat improve the performance.

The threshold for choosing nose candidates from eye and mouth corner detection is set to be 1.1. This value ensures that the resulting candidate set includes desired nose points for the entire training set. However, instead of investigating all the candidates, we use only a small subset sampled with a regular interval (3 pixels for each dimension) in an image domain so that on average, the number of actual candidates are around 100.

3. A Morphable Model of 3D Faces

For selecting the optimal nose position, estimating pose and for the subsequent reconstruction of a high-resolution 3D mesh, we use a Morphable Model of 3D faces [3], 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 $j \in \{1, \ldots, n\}$, $n = 75,972$ of a polygon mesh, and texture vectors are formed by red, green, and blue values:

$$\mathbf{S}_i = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T \quad (2)$$

$$\mathbf{T}_i = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \quad (3)$$

By Principal Component Analysis (PCA), we obtain a set of m' orthogonal principal components $\mathbf{s}_i$, $\mathbf{t}_i$, and the standard deviations $\sigma_{S,i}$ and $\sigma_{T,i}$ around the averages $\bar{\mathbf{s}}$ and $\bar{\mathbf{t}}$. In the following, we use the $m' = 50$ most relevant principal components only.

4. Model-Based Feature Confidence

In this section, we propose a novel, 3D-based confidence measure for the plausibility of a configuration of 2D features. We consider the following feature points: the tip of the nose, the corners of the mouth, and the external corners of the eyes.

For each of the feature points $j = 1, ..., 5$, we have the image positions (h_j^x, h_j^y) and we know which vertex k_j of the model it corresponds to. We can now find the linear combination of examples and the 3D rotation, scale and translation that reproduces these positions best. We do this with an efficient, quasi-linear approach [2] that we summarize below. This defines a new criterion for confidence in terms of 3D distortion, based on the Mahalanobis distance from the average face.

To assess how well the reconstructed face fits to the pixel values in the image, we modify the quasi-linear algorithm: After shape fitting, we can look up the desired color or grey values of the image for each vertex. Unlike the algorithm in Section 5, we assume simple ambient illumination here. For finding the optimal nose position, it has turned out to be best to use only vertices in the nose region. The color values $(R_{k_j}, G_{k_j}, B_{k_j})$ for vertices k_j are reconstructed by the textures of the Morphable Model using the algorithm described in this section. Again, Mahalanobis distance is used as a confidence measure. For grey-level images, we replace colors in the Morphable Model by grey-levels.

Both the coarse shape and texture reconstruction is achieved by a Maximum a Posteriori estimate. In the following, let either $\mathbf{v} = \mathbf{S}$ or $\mathbf{v} = \mathbf{T}$, and

$$\mathbf{x} = \mathbf{v} - \overline{\mathbf{v}}, \qquad \overline{\mathbf{v}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{v}_i. \tag{4}$$

In this unified notation, let $\mathbf{s}_i$ be the eigenvectors from PCA, and σ_i the standard deviations which we include as explicit factors in the expansion

$$\mathbf{x} = \sum_{i=1}^{m'} c_i \sigma_i \mathbf{s}_i = (\sigma_1 \mathbf{s}_1, \sigma_2 \mathbf{s}_2, ...) \cdot \mathbf{c} \tag{5}$$

so the estimated normal distribution takes the simple form

$$p(\mathbf{c}) = \nu_c \cdot e^{-\frac{1}{2}\|\mathbf{c}\|^2}, \qquad \nu_c = (2\pi)^{-m'/2}. \tag{6}$$

Now let a reduced set of model data be a vector $\mathbf{r} \in I\!R^l$: By a projection operator, we select coordinates of 5 feature points k_j from the full vectors $\mathbf{v}$, perform orthographic projection, rotation and scaling to obtain 5 2D image positions $(l = 2 \cdot 5.)$ In a similar way, we select texture values of vertices and may change contrast in the color channels. For the moment, assume that these operations are known and combined to a linear operator, respectively:

$$\mathbf{r} = \mathbf{L}\mathbf{v} \qquad \mathbf{L} : I\!R^n \mapsto I\!R^l. \tag{7}$$

$$\mathbf{y}_{model} = \mathbf{r} - \mathbf{L}\overline{\mathbf{v}} = \mathbf{L}\mathbf{x} \tag{8}$$

$$\mathbf{y}_{model} = \mathbf{L}\sum_i c_i \sigma_i \mathbf{s}_i = \sum_i c_i \mathbf{q}_i = \mathbf{Q}\mathbf{c} \tag{9}$$

where $\mathbf{q}_i = \mathbf{L}(\sigma_i \mathbf{s}_i) \in I\!R^l$ are the reduced versions of the scaled eigenvectors, and

$$\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, ...) \in I\!R^{l \times m'}. \tag{10}$$

Given the observed vector $\mathbf{y} \in R^l$, we are looking for the coefficients $\mathbf{c}$ with maximum posterior probability $P(\mathbf{c}|\mathbf{y})$. As an intermediate step, consider the likelihood of measuring $\mathbf{y}$, given $\mathbf{c}$: We assume that each dimension j of the measured vector $\mathbf{y}$ is subject to uncorrelated Gaussian noise with a variance σ_N^2. Then, the likelihood of measuring $\mathbf{y} \in I\!R^l$ is given by

$$P(\mathbf{y}|\mathbf{y}_{model}) = \prod_{j=1}^{l} P(y_j|y_{model,j}) \tag{11}$$

$$= \prod_{j=1}^{l} \nu_N \cdot e^{-\frac{1}{2\sigma_N^2}(y_{model,j} - y_j)^2} = \nu_N^l \cdot e^{-\frac{1}{2\sigma_N^2}\|\mathbf{y}_{model} - \mathbf{y}\|^2} \tag{12}$$

with a normalization factor ν_N. In terms of the model parameters $\mathbf{c}$, the likelihood is

$$P(\mathbf{y}|\mathbf{c}) = \nu_N^l \cdot e^{-\frac{1}{2\sigma_N^2}\|\mathbf{Q}\mathbf{c} - \mathbf{y}\|^2}. \tag{13}$$

According to Bayes Rule, the posterior probability is

$$P(\mathbf{c}|\mathbf{y}) = \nu \cdot P(\mathbf{y}|\mathbf{c}) \cdot p(\mathbf{c}). \tag{14}$$

with a constant factor $\nu = (\int P(\mathbf{y}|\mathbf{c}') \cdot p(\mathbf{c}') d\mathbf{c}')^{-1}$. Substituting (6) and (13) yields

$$P(\mathbf{c}|\mathbf{y}) = \nu \cdot \nu_N^l \cdot \nu_c \cdot e^{-\frac{1}{2\sigma_N^2}\|\mathbf{Q}\mathbf{c} - \mathbf{y}\|^2} \cdot e^{-\frac{1}{2}\|\mathbf{c}\|^2}, \tag{15}$$

which is maximized by minimizing the cost function

$$E = -2 \cdot log P(\mathbf{c}|\mathbf{y}) = \frac{1}{\sigma_N^2}\|\mathbf{Q}\mathbf{c} - \mathbf{y}\|^2 + \|\mathbf{c}\|^2. + const. \tag{16}$$

Using a Singular Value Decomposition $\mathbf{Q} = \mathbf{U}\mathbf{W}\mathbf{V}^T$ with a diagonal matrix $\mathbf{W} = diag(w_i)$, it can be shown [2] that the optimal coefficients are

$$\mathbf{c} = \mathbf{V} diag(\frac{w_i}{w_i^2 + \sigma_N^2})\mathbf{U}^T\mathbf{y}. \tag{17}$$

As a confidence measure for feature points, we propose $\|\mathbf{c}_{shape}\| + \|\mathbf{c}_{texture}\|$.

In order to deal with unknown position, orientation and scale, we use the method of [2], which is to treat not only translation, but also rotation and scaling as additive terms, and add a set of vectors $\mathbf{s}_i$ and coefficients c_i to the system. For rotation, this is a first-order approximation only. From c_γ, c_θ, c_ϕ, we recover the angles γ, θ, ϕ, then update $\mathbf{L}$ and iterate the process, which gives a stable solution after the second pass. For the estimation of texture, we apply the same method to deal with gains and offsets in the color channels.

5. 3D Face Reconstruction by Model Fitting

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. This optimization is achieved by an algorithm that was presented in [4]. For

the optimization to converge, the algorithm has to be initialized with the feature coordinates of the 5 feature points provided by the previous processing steps.

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.

In the algorithm, the fitting algorithm is used twice (Figure 1): In a fast fit with a reduced number of iterations, the system estimates the azimuth angle ϕ. Given the optimal set of feature points, the final reconstruction is then computed by a full fit. Since the linear combination of textures $\mathbf{T}_i$ cannot reproduce all local characteristics of the novel face, such as moles or scars, we extract the person's true texture from the image and correct for illumination [3].

6. Results

We tested our algorithm on 6 videos and 870 images corresponding to the 87 individuals from the FERET database [19] who were not in the training set. The face detection algorithm succeeded in 705 out of 870 images, at a computation time of less than 50 ms per image on a standard PC. A single step of detecting the facial components and classification-based refinement took around 7 seconds per face. The component detection results were evaluated by measuring the average Euclidean distance of the 7 components from manually labeled ground truth within the rescaled face images (200×200 pixels). For comparison, four variants of existing classification-based methods were implemented. The independent SVM-based method scans the input image within a candidate region for each COI (based on the estimateed Gaussian distribution of the COI location in the training set, hereafter referred to as Gaussian prior) and produces the detection as the location with the highest SVM score. The connected components method, inspired by [1], first generates a binary image for each COI with the pixel values corresponding to the SVM classification labels (COI or background). The location of the COI is then obtained as the mean of the connected component of the COI labels yielding the highest evidence, assuming the above-mentioned Gaussian prior. The Bayesian component detector, inspired by [12], models pixel values within a window around each COI and background as Gaussian distributions. The location of a COI is then obtained as the position of the window that yields the greatest log-likelihood ratio among a set of candidate locations within the smallest rectangle ecompassing all the training data locations. The pairwise reinforcement of feature responses (PRFR) is based on the idea of Cristinacce et al. [10], i.e., the individual COI detections obtained from each SVM are refined based on their distributions conditioned on the other components. Details of the refinement procedure can be found in [10].[1] Table 1 summarizes the results.

[1] In the original work of Cristinacce et al. [10], a boosting classifier is utilized while we used the SVM to facilitate comparison between different methods.

component detection method	average error (pixels)
independent SVM	8.30
connected components	7.05
PRFR	8.17
Bayesian	6.50
proposed (rough detection)	6.51
proposed (single iteration)	4.56
proposed	4.52

Table 1. Performance of different component detection methods: 'rough detection' refers to the KRR-based detection (cf., Sec. 2.3); 'single iteration' stands for a single iteration of the component detection and fitting (with the rotation angle obtained from the rough estimator).

	left eye o	right eye o	nose	left mouth	right mouth	left eye i	right eye i
Ø error	4.78	4.97	6.09	4.39	4.53	3.71	4.20

Table 2. Average errors (pixels) per feature over all views (o: outer corner; i: inner corner)

	ba $1.1°$	bb $38.9°$	bc $27.4°$	bd $18.9°$	be $11.2°$
Ø error	3.92	6.03	4.89	5.14	4.19
	bf $-7.1°$	**bg $-16.3°$**	**bh $-26.5°$**	**bi $-37.9°$**	**bk $0.1°$**
Ø error	3.35	3.66	4.82	6.29	4.40

Table 3. Average errors (pixels) over all features per view

Tables 2 and 3 list the average errors for different feature types and for different viewing angles of the original images. The angles in Table 3 are estimates from [4].

Reconstruction was performed based on four points given by the facial component detection (external corners of the eyes, and corners of the mouth) and the nose position returned by the model-based confidence measure. The computation time is approximately 3 minutes. Our system does not produce veridical, but plausible and photorealistic 3D models consistent with the input images. Hence we evaluated it using two perceptual experiments that reflect the demands of many potential real-world applications.

From all 705 successful reconstructions, we randomly selected 200, and collected ratings by 49 participants. Each participant saw the original image and two views of the reconstructed face. Some participants rated only a subset of faces, so we obtained an average of 24 ratings per face (4,815 ratings in total) according to the following instructions: "The following 3D reconstructions are supposed to be used as personalized avatars in a game. Divide the results into four groups: very good, good, acceptable and bad." Their ratings are shown in Tab. 4, and typical examples are shown in Fig. 4. Given the instructions, the criteria and standards were deliberately left open to the judgements of the participants. Still, we found that the ratings were overall positive and promising. To demonstrate the quality

	very good	good	acceptable	bad
%	12.86	24.09	28.93	34.12

Table 4. Average rating of 200 examples by 49 participants.

participant	better	same quality	worse
1	10%	26%	64%
2	22%	24%	54%
3	16.5%	29%	54.5%

Table 5. Percentage of faces where participants found the automated reconstruction better, equal or worse than the reconstruction from manual labeling in a side by side comparison.

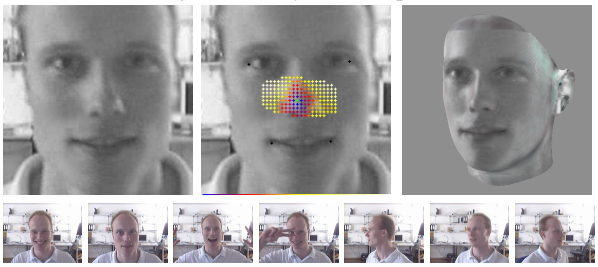

Figure 6. Fully automated reconstruction from an automatically detected single frame of a webcam video. From left to right: original frame, color coded results of the model-based measure (the darker the better, best position marked green) and the automatically reconstructed head. The bottom row shows 7 sample frames.

and variability of the results, images of the reconstructed faces without textures are shown in Fig. 5.

In the second experiment, we showed the automated reconstruction and the reconstruction from manually labeled features side by side in random order for each face, along with the original image. Participants were instructed to select the reconstruction that looks better. The results in Table 5 indicate that the automated algorithm is competitive in many cases, even though it does not fully match the quality of the manual initialization.

The 6 videos were recorded with a webcam (Logitech QuickCam pro 4000). Each video shows a moving person, e.g. turning their heads, taking glasses off and on, moving forward and backward, etc. The recording speed was 30 frames/sec. and the resolution of each frame was 320×240. Our face detection algorithm attempts to detect faces in every frame, and returns the single frame with maximum detection score. Component detection, confidence measure for the feature points and finally the reconstruction proceed the same way as for the still images. For all video examples we got similar results, one of which is shown in Fig. 6.

7. Conclusion

By combining Support Vector Machines and 3D Morphable Models, we have addressed the problem of fully automated 3D shape reconstruction from raw video frames or other images. The system has proved to be robust with respect to a variety of imaging conditions, such as pose and lighting. The results and the rating scores by human participants demonstrate that the system produces a high percentage of photo-realistic reconstructions which makes it useful for a wide range of applications.

References

[1] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *CVPR*, pages 860–867, 2005. 6

[2] V. Blanz, A. Mehl, T. Vetter, and H.-P. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In Y. Aloimonos and G. Taubin, editors, *2nd International Symposium on 3DPVT 2004*, pages 293–300, Thessaloniki, Greece, 2004. IEEE. 5

[3] V. Blanz and T. Vetter. A morphable model for the synth. of 3D faces. In *SIGGRAPH'99*, pages 187–194, 1999. 2, 4, 6

[4] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 25(9):1063– 1074, 2003. 2, 4, 5, 6

[5] M. Brand. Morphable 3d models from video. In *CVPR*, pages 456–463, 2001. 2

[6] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, pages 2690–2696, 2000. 2

[7] C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In *Advances in Neural Information Processing Systems*, volume 9, pages 375–381, 1997. 4

[8] A. K. R. Chowdhury and R. Chellappa. Face reconstruction from monocular video using uncertainty analysis and a generic model. *CVIU*, 91(1-2):188–213, 2003. 2

[9] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash. Hybrid neural network / hidden markov model continuous speech recognition. In *Proc. of Int. Conf. on Spoken Language Processing*, pages 915–918, 1992. 3

[10] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In *British Machine Vision Conference, London, England*, pages 277–286, 2004. 6

[11] M. Dimitrijevic, S. Ilic, and P. Fua. Accurate face models from uncalibrated and ill-lit video sequences. In *CVPR '04, Washington, DC*, June 2004. 2

[12] M. R. Everingham and A. Zisserman. Regression and classification approaches to eye localization in face images. In *Proc. of the 7th Int. Conf. on Automatic Face and Gesture Recognition (FG2006)*, pages 441–446, 2006. 2, 6

[13] P. Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. In *ICCV*, Corfu, Greece, September 1999. 2

[14] L. Gu and T. Kanade. 3d alignment of face in a single image. In *CVPR '06*, pages 1305–1312, 2006. 2

[15] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *NIPS*, 2002. 2

[16] W. Kienzle, G. Bakir, M. Franz, and B. Schölkopf. Face detection - efficient and rank deficient. In *Advances in Neural Information Processing Systems*, pages 673–680, 2005. 2

[17] J. Lee, H. Pfister, B. Moghaddam, and R. Machiraju. Estimation of 3d faces and illumination from single photographs using a bilinear illumination model. In *Rendering Techniques*, pages 73–82, 2005. 2

[18] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *CVPR*, pages 130–136, 1997. 2

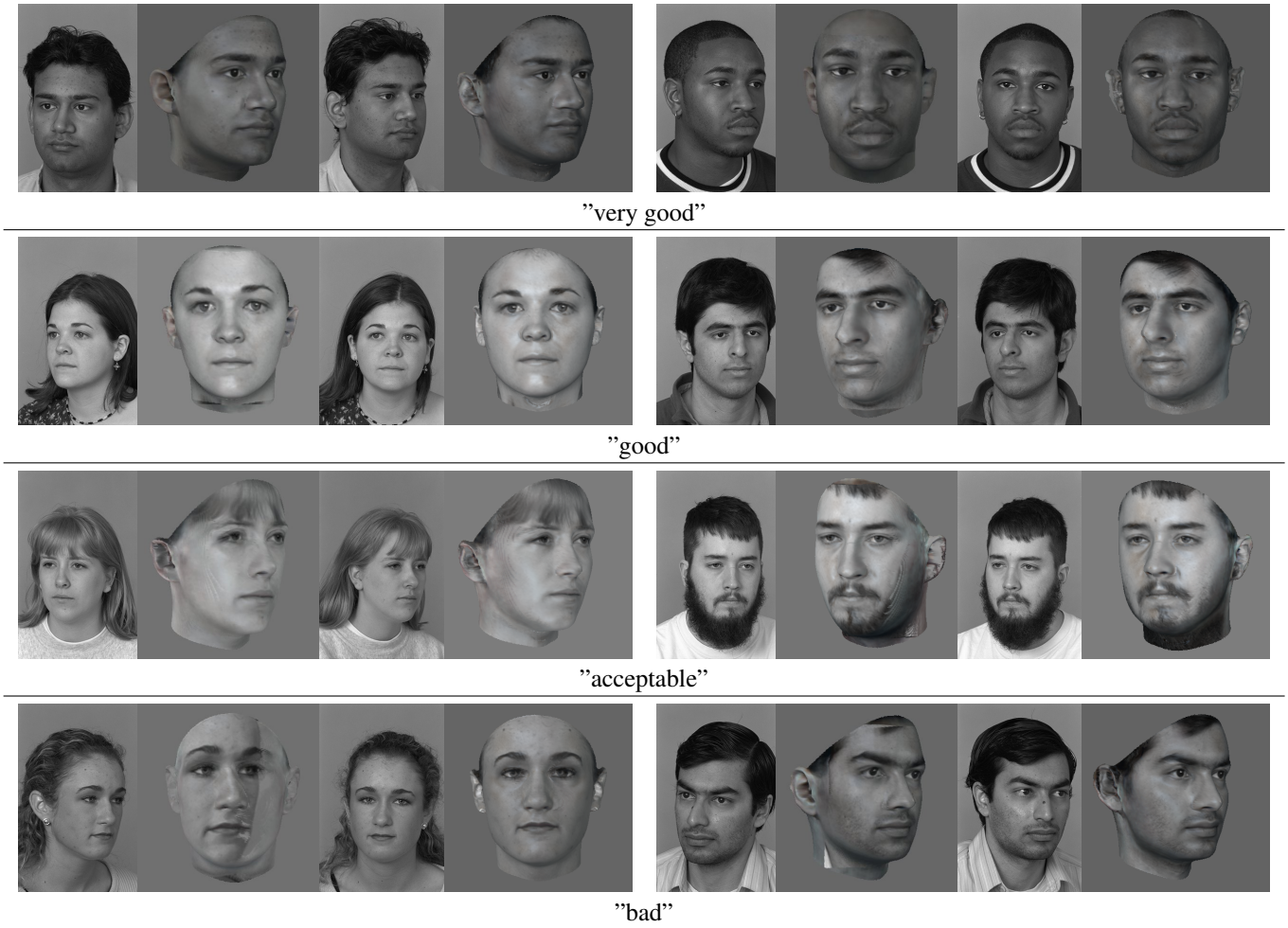

Figure 4. Two typical examples from each rating score level. From left to right: original image, novel view of the automated reconstruction, original second view of the person for comparison, novel view of the reconstructed 3D model if features are labeled manually.

Figure 5. Images of the automated reconstructions without texture. Same faces as in Fig. 4 line by line.

[19] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face recognition algorithms. *Img. and Vis. Comp. J*, 16(5):295–306, 1998. 4, 6

[20] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large-Margin Classiers*, pages 61–74. 1999. 3

[21] S. Romdhani, P. Torr, B. Schoelkopf, and A. Blake. Efficient face detection by a cascaded support-vector machine expansion. *Proc. - Royal Society. Mathematical, physical and engineering sciences*, 460(2051):3283–3297, 2004. 2

[22] M. Schenkel, I. Guyon, and D. Henderson. On-line cursive script recognition using time-delay neural networks and hidden markov models. *Machine Vision and Applications*, 8(4):215–223, 1995. 3

[23] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999. 4

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

[25] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *CVPR*, June 2004. 2

[26] L. Xin, Q. Wang, J. Tao, X. Tang, T. Tan, and H. Shum. Automatic 3d face modeling from video. In *ICCV*, 2005. 2

[27] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *PAMI*, 24(1):34–58, 2002. 2

[28] Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *IJCV*, 58(2):93–119, 2004. 2