Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. 160

Automatic 3D Face Reconstruction from Single Images or Video

P. Breuer,[1] K. I. Kim,[2] W. Kienzle,[2] V. Blanz,[1]
B. Schölkopf[2]

February 2007

[1] Department for Empirical Inference, email: kimki;kienzle;bs@tuebingen.mpg.de
[2] Universität Siegen, email: pbreuer@informatik.uni-siegen.de; blanz@mpi-sb.mpg.de

This report is available in PDF–format via anonymous ftp at ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/tr160.pdf. The complete series of Technical Reports is documented at: http://www.kyb.tuebingen.mpg.de/techreports.html

Automatic 3D Face Reconstruction from Single Images or Video

P. Breuer, K. I. Kim, W. Kienzle, V. Blanz, B. Schölkopf

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. 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,
- involves zero user interaction,
- produces close-to-photorealistic 3D reconstructions.

To perform this task which, to the best of our knowledge, has so far not been accomplished within the given specifications, we are building a system which integrates two well-known techniques: Support Vector Machines (SVMs) and Morphable Models. The processing steps of our algorithm are

1. Face Detection using SVM
2. For video data: selection of a frontal view
3. Facial component detection using regression and classification
4. Selection of the most plausible combination of components based on Gaussian distributions
5. Selection of the most plausible nose position based on a Morphable Model
6. 3D reconstruction, initialized with the components.

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. 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.

For reconstruction from video, our system selects a single frontal 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 [1], structure-from-motion with subsequent refinement by a deformable face model [2], nonrigid

structure-from-motion with intrinsic model constraints [3] and feature tracking and factorization of the tracking matrix for non-rigid shape estimation [4]. Zhang et al. [5] 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 [6, 7, 8], the combined algorithm no longer requires manual rigid pre-alignment of the 3D model or manual labeling of features on the 2D image. Due to our automated face and feature detection components, these restrictions do not apply for our system. Xiao et al.[9] 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.

For related work in the feature detection literature, which involves SVM-based methods [10, 11], we refer the readers to the excellent survey of Yang et al. [12].

2 Detection of faces and facial components

2.1 Face detection

As a first step, a face detector is applied to the input image. For this purpose, we tried two publicly available face detection libraries for Matlab: an approach based on SVMs [13], and an implementation of the widely used Boosting based detector [14]. We found the detection results very similar and both implementations sufficiently fast for our purposes. Although the Boosting approach seemed to be more efficient, we chose the SVM implementation since it also returns a confidence value together with each detected face in the image. 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. Also, we believe that the better a detection matches the prototypical face that the detector responds to, the better subsequent processing stages (facial component detection, 3D model fitting, etc.) work. Therefore, we run the detector on an image and a video and pick the most confident detection among all frames. The input image containing the best detection is cropped to a square region around the face, and is then rescaled to 200×200 pixels. This is the reference coordinate system used in all subsequent processing steps.

2.2 Facial component detection based on regression

The second stage computes position estimates of eye and mouth corners (will be referred to as component of interest (COI), hereafter) 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 [15]. 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 kernel ridge regression (KRR) is adopted for this purpose (see Sec. 2.4 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 [15]. 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. 1, 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 makes the detector extremely redundant and therefore robust to occlusions and other local changes. This effect is shown in Fig. 1.

2.3 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 a pixel of the window (cf. below), 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 25×25 window, excluding the central 9×9. The reference point can be slightly offset to the side in the window (see Fig. 2 for details).

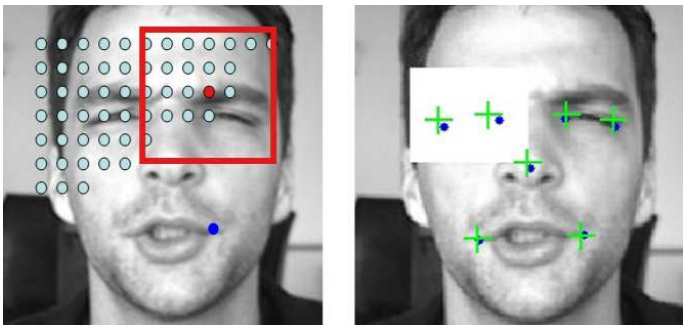

Figure 1: Regression-based detection of facial components. *Left*: illustration of the regressor array. For each component (e.g., a corner of the mouth, here marked dark blue) 144 regressors are learned. Each one operates on a different image region, centered at one of the light blue points. *Right*: prediction on a corrupted test image (the left eye region is covered with a white rectangle). Plus marks indicate desired component locations.

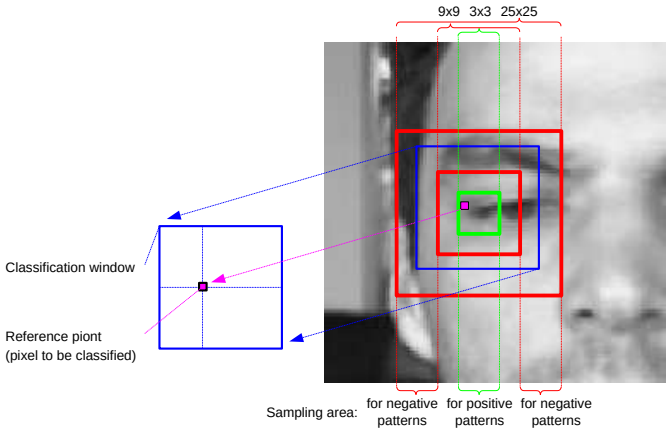

Figure 2: The configuration for sampling training data for outer corner of the left eye: the 25×25 window for sampling negative examples is motivated by the typical location error of the regression-based method which lies within the range of 0 to 9 pixel distances from the true eye corner locations (with 3-pixel margins for all directions from the 19×19 area); The 9×9 window is excluded in sampling the negative patterns such that the training is not affected by ambiguous patterns; The right offset of the classification window from a reference point is larger than the left offset to include more pixels from the actual eye area.

In the classification stage, instead of scanning the entire face image, the search space for a given COI is restricted as a 19×19 window surrounding the regression-based detection. Then, each pixel within the search window is classified into the COI if the corresponding SVM output is larger than a given threshold t which is initially set at 0.

While more accurate than the regression-based method in general, 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 for $t = 0$. When the size of detection blob is too small (or zero), we thus adapt t 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 12-dimensional vector $H = (h_1, \ldots, h_6) = ((h_1^x, h_1^y), \ldots, (h_6^x, h_6^y))$ ((x, y)-coordinates values for six components). Then, the best detection vector is obtained by firstly generating randomly 100,000 vectors, by sampling two dimensional vectors ((x, y)-coordinates)) from each component blob and concatenating them to constitute 12-dimensional vectors, and then choosing the maximizer of the following objective function.

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

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 motive this cost function as follows. Suppose we want to obtain the most probable configuration

$$\begin{aligned} H^* &= arg\max P(H = O|I) \\ &= arg\max P(I|H = O)P(H = O), \end{aligned} \tag{2}$$

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

$$\begin{aligned} H^* &\approx arg\max P(W_1, \ldots, W_6|H = O)P(H = O) \\ &\approx arg\max \left(\prod_{i=1,\ldots,6} P(W_i|h_i = o_i) \right) P(H = O) \\ &\approx arg\max \left(\sum_{i=1,\ldots,6} \log P(W_i|h_i = o_i) \right) - \frac{1}{\alpha} M(H), \end{aligned}$$

where the first line replaces the image by the small windows (W_i), the second line corresponds to an independence assumption of the component likelihoods, and the third line assumes a Gaussian distribution of the configurations H. The last step is to substitute $P(W_i|h_i = o_i)$ with the component detection posterior $P(h_i = o_i|W_i)$ [1] calculated by wrapping the SVM output with a sigmoidal function based on the idea of [16].[2] This technique of replacing the likelihood by the posterior estimated from a discriminative classifier is common in speech and character recognition applications and has shown to improve the discrimination capability of generative models [17, 18].

A similar approach has been proposed in [19] where they restricted the search space of an SVM classifier based on joint configurations. However, unlike [19], we restrict the search space based on the regression based detection and use both the prior on joint configuration and the approximation of likelihood provided by the SVM to choose the best configuration.

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 10-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. It should be noted that even complete knowledge of the other components does not accurately determine the nose position, since the nose position determines the rotation angle of the head, and the other components alone do not. 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 (cf., Fig. 3).

2.4 Training

For training the regression-based component detector, randomly selected 552 faces of the BioID database [20] were used. A Gaussian kernel ($K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2)$) was utilized for the KRR. The parameters were found by cross-validation: length scale γ and ridge parameter for KRR were obtained as 0.05 and 1, respectively while 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.

[1] By applying the Bayes formula; Here $P(W_i)$ is a constant and accordingly can be discarded. Originally, the marginal $P(h_i)$ should be included in the substitution which can easily be calculated based on training configurations. However, it is assumed to be uniform in the current paper as within the search window given by the regression-based method, relatively small variations of single component coordinate might not be that informative.

[2] The original formulation of Platt's method requires tuning a parameter in the sigmoidal function for each component class which in this paper is replaced by a single parameter α in Eq. (1). Accordingly, the objective function (1) is not an exact implementation of (2).

For estimating the distribution of components in the classification-based component detector, again 552 faces of the BioID database were used. Then, for each component SVM, 8,832 training patterns (2,208 and 6,624 positive and negative patterns, respectively) are collected from these 552 faces. Exploiting the vertical symmetry of faces, SVMs are only trained on components lying in the left side of the faces (outer and inner corners of left eye and left corner of mouth). For all SVM models, Gaussian kernels were utilized with hyperparameters chosen by cross validation. The input window size for eye and mouth detection was determined as (31×31) (with the reference point horizontally placed 8 pixels from the side, and vertically in the center; cf. above) which for the eye case, roughly corresponds to the average length of the eye in (200×200)-size 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 (Fig. 3).

3 A Morphable Model of 3D Faces

For selecting the optimal nose position and for the 3D reconstruction of faces, we use a Morphable Model of 3D faces [21, 6], which is a vector space of 3D shapes and textures spanned by a set of examples. We use laser scans of 200 individuals who are not in the test sets used below. Shape and texture vectors $\mathbf{S}_i$, $\mathbf{T}_i$ are defined such that any linear combination of examples within a few standard deviations from the average face is a realistic face. 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:

$$\mathbf{S}_i = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T \quad (3)$$

$$\mathbf{T}_i = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T. \quad (4)$$

It is essential that these face vectors are in dense correspondence, so each vector component describes the same point, such as the tip of the nose, in all faces. Correspondence can be established using optical flow [6]. 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 this basis, faces can be written as

$$\mathbf{S} = \bar{\mathbf{s}} + \sum_{i=1}^{m'} \alpha_i \cdot \mathbf{s}_i, \qquad \mathbf{T} = \bar{\mathbf{t}} + \sum_{i=1}^{m'} \beta_i \cdot \mathbf{t}_i. \quad (5)$$

In the following, we use the $m' = 149$ most relevant principal components only.

4 Model-Based Confidence Measure for Feature Points

In this section, we use the 2D locations of facial components as feature points and compute a 3D-based confidence measure for the plausibility of a configuration, using a Morphable Model. It is more important to consider depth for the nose position than for the eyes and mouth positions, which are approximately coplanar. We consider the following feature points: the tip of the nose, the corners of the mouth, and the external corners of the eyes. The internal corners of the eyes turned out to be dispensable for model fitting.

For each of the feature points $j = 1, ..., 5$, we have the image positions $(q_{x,j}, q_{y,j})$ 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 [22] that we summarize below. Unlike previous work, we are now using the Mahalanobis distance from the average face as a measure of 3D distortion.

To assess how well the reconstructed face fits to the pixel values in the image, we modify the above algorithm [22]: 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 [22]. In the following, let either $\mathbf{v} = \mathbf{S}$ or $\mathbf{v} = \mathbf{T}$, and

$$\mathbf{x} = \mathbf{v} - \overline{\mathbf{v}}, \qquad \overline{\mathbf{v}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{v}_i. \tag{6}$$

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

$$\mathbf{x} = \sum_{i=1}^{m'} c_i \sigma_i \mathbf{s}_i = (\sigma_1 \mathbf{s}_1, \sigma_2 \mathbf{s}_2, ...) \cdot \mathbf{c} \tag{7}$$

so the estimated normal distribution takes the simple form

$$p(\mathbf{c}) = \nu_c \cdot e^{-\frac{1}{2}\|\mathbf{c}\|^2}, \qquad \nu_c = (2\pi)^{-m'/2}. \tag{8}$$

Now let a reduced set of model data be a vector $\mathbf{r} \in I\!R^l$ of 3D coordinates of 5 feature points, or color values of the vertices in the nose region. These can be obtained by a projection operator from the full vectors $\mathbf{v}$. In addition, we may perform orthographic projection, rotation and scaling to geometry, or change contrast in the color channels. For the moment, assume that these operations are known, and they combine to a linear operator

$$\mathbf{r} = \mathbf{L}\mathbf{v} \qquad \mathbf{L} : I\!R^n \mapsto I\!R^l. \tag{9}$$

$$\mathbf{y} = \mathbf{r} - \mathbf{L}\overline{\mathbf{v}} = \mathbf{L}\mathbf{x} \tag{10}$$

The least-squares solution of this problem would be to minimize

$$E(\mathbf{x}) = \|\mathbf{L}\mathbf{x} - \mathbf{y}\|^2. \tag{11}$$

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

$$\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, ...) \in I\!R^{l \times m'}. \tag{12}$$

In terms of model coefficients c_i from (7), (11) is

$$E(\mathbf{c}) = \|\mathbf{L}\sum_i c_i \sigma_i \mathbf{s}_i - \mathbf{y}\|^2 = \|\mathbf{Q}\mathbf{c} - \mathbf{y}\|^2. \tag{13}$$

However, it has been shown that this simple approach would produce significant overfitting artifacts [22], so we use a Maximum Posterior Probability (MAP) approach [22]: Given the observed vector $\mathbf{y}$, 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 $\mathbf{c}$: In the noiseless case, $\mathbf{c}$ would define the vector

$$\mathbf{y}_{model} = \mathbf{L}\sum_i c_i \sigma_i \mathbf{s}_i = \sum_i c_i \mathbf{q}_i = \mathbf{Q}\mathbf{c} \tag{14}$$

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

$$P(\mathbf{y}|\mathbf{y}_{model}) = \prod_{j=1}^{l} P(y_j|y_{model,j}) \tag{15}$$

$$= \prod_{j=1}^{l} \nu_N \cdot e^{-\frac{1}{2\sigma_N^2}(y_{model,j} - y_j)^2} = \nu_N^l \cdot e^{-\frac{1}{2\sigma_N^2}\|\mathbf{y}_{model} - \mathbf{y}\|^2} \tag{16}$$

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

$$P(\mathbf{y}|\mathbf{c}) = \nu_N^l \cdot e^{-\frac{1}{2\sigma_N^2}\|\mathbf{Qc}-\mathbf{y}\|^2}. \tag{17}$$

According to Bayes Rule, the posterior probability is

$$P(\mathbf{c}|\mathbf{y}) = \nu \cdot P(\mathbf{y}|\mathbf{c}) \cdot p(\mathbf{c}). \tag{18}$$

with a constant factor $\nu = (\int P(\mathbf{y}|\mathbf{c}') \cdot p(\mathbf{c}')d\mathbf{c}')^{-1}$.
Substituting (8) and (17) yields

$$P(\mathbf{c}|\mathbf{y}) = \nu \cdot \nu_N^l \cdot \nu_c \cdot e^{-\frac{1}{2\sigma_N^2}\|\mathbf{Qc}-\mathbf{y}\|^2} \cdot e^{-\frac{1}{2}\|\mathbf{c}\|^2}, \tag{19}$$

which is maximized by minimizing the cost function

$$E = -2 \cdot logP(\mathbf{c}|\mathbf{y}) = \frac{1}{\sigma_N^2}\|\mathbf{Qc}-\mathbf{y}\|^2 + \|\mathbf{c}\|^2. + const. \tag{20}$$

To simplify the calculation, we introduce a regularization factor $\eta = \sigma_N^2 \geq 0$ and minimize

$$E = \|\mathbf{Qc}-\mathbf{y}\|^2 + \eta \cdot \|\mathbf{c}\|^2. \tag{21}$$

Using a Singular Value Decomposition $\mathbf{Q} = \mathbf{UWV}^T$ with a diagonal matrix $\mathbf{W} = diag(w_i)$, it can be shown [22] that the optimal coefficients are

$$\mathbf{c} = \mathbf{V}diag(\frac{w_i}{w_i^2+\eta})\mathbf{U}^T\mathbf{y}. \tag{22}$$

Our confidence measure for feature points is
$\|\mathbf{c}_{shape}\| + \|\mathbf{c}_{texture}\|$.

In order to deal with unknown position, orientation and scale, we use the method of [22], 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 [22]. For the estimation of texture, we apply the same method to deal with gains and offsets in the color channels.

5 3D Face Reconstruction

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 distance. This optimization is achieved by an algorithm that was presented in [7]. 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.

In image synthesis, a given set of model parameters α and β (5) define a 3D face, and we can compute a color image $\mathbf{I}_{model}(x, y)$ by standard computer graphics procedures, including rigid transformation, perspective projection, computation of surface normals, Phong-Illumination, and rasterization. The image depends on a number of rendering parameters ρ. In our system, these are 22 variables for 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.

All parameters are estimated simultaneously in an analysis-by-synthesis loop. The main goal of the analysis is to find the parameters α, β, ρ that make the synthetic image $\mathbf{I}_{model}$ as similar as possible to the original image $\mathbf{I}_{input}$ by minimizing

$$E_I = \sum_x \sum_y \sum_{c\in\{r,g,b\}} (I_{c,input}(x, y) - I_{c,model}(x, y))^2. \tag{23}$$

All scene parameters are recovered automatically, starting from a frontal pose in the center of the image, at frontal illumination and with color contrast 0. For initialization, the 2D feature points $(q_{x,j}, q_{y,j})$ and the image positions (p_{x,k_j}, p_{y,k_j}) of the corresponding vertices k_j define a function

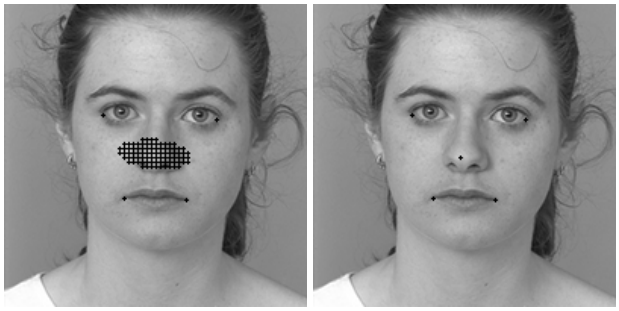

Figure 3: Left: nose candidates given by the component detection; Right: nose position chosen by the model-based measure.

$$E_F = \sum_j \| \begin{pmatrix} q_{x,j} \\ q_{x,j} \end{pmatrix} - \begin{pmatrix} p_{x,k_j} \\ p_{y,k_j} \end{pmatrix} \|^2. \tag{24}$$

that is added to the image difference E_I in the first iterations. Adding regularization terms to the cost function, we obtain

$$E = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_F^2} E_F + \sum_i \frac{\alpha_i^2}{\sigma_{S,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{T,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho}_i)^2}{\sigma_{R,i}^2} \tag{25}$$

using standard deviations from PCA, and ad-hoc estimates for $\sigma_{R,i}$. Similar to Section 4, the cost function (25) can be derived from a maximum-a-posteriori approach. The optimization is performed with a Stochastic Newton Algorithm [7]. 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 [6].

6 Results

We tested our algorithm on 50 still images and 6 videos.

As stills, we used the first 50 individuals from the FERET database [23] in frontal views (*ba*). The face detection algorithm succeeded in 48 out of 50 images, at a computation time of less than 50 ms per image on a standard PC. Detecting the facial components and classification-based refinement took around 4 minutes per face.

We evaluated the results by measuring the average Euclidean distance of the 6 components from manually labeled ground truth within the rescaled face images (200×200 pixels). On the BioDB [20] (276 test images from the DB used in training; disjoint from the training set however) we measured an error of 2.07 pixels for the regression based system. On the FERET data we measured an error of 4.67 pixels for the regression based system and an error of 3.13 pixels for the classification based refinement.[3]

Given four feature points for the corners of the eyes and mouth, along with around 100 candidates for the nose, the model-based confidence measure returns the most likely nose position, as illustrated in Fig. 3. The shape-based confidence measure is computed in 25ms on a standard PC, while the texture-based measure takes approximately 30s due to higher number l of samples. Fig. 4 shows the quality of the returned nose positions.

Reconstruction was 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. For evaluation, the results of all 48 fully automated reconstructed heads from the still images were shown to six students. We gave them 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. 1, and typical examples are shown in Fig. 5.

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

[3] Cf. the pure classification-based method (i.e., without the regression stage) produced an average error of 4.01 pixels. In this case, the negative training sample is collected from a rather large image area of size 60×60 around the ground truth.

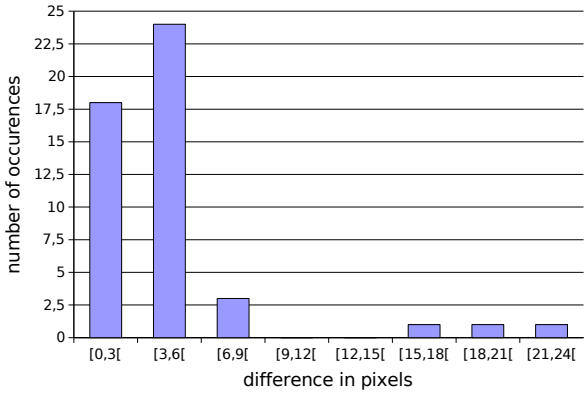

Figure 4: Difference between desired and measured nose positions in the 50 rescaled FERET images (2 examples failed already at face detection).

participant	1	2	3	4	5	6	mean
very good	6	8	3	9	11	8	7.5
good	21	20	16	14	12	18	16.8
acceptable	13	14	14	11	13	14	13.2
bad	8	6	15	14	12	8	10.5

Table 1: Rating of 48 examples by six participants (2 examples failed at face detection).

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 streams. The system has proved to be robust with respect to a variety of imaging conditions, such as those found in our example videos. Our algorithm scales well in terms of the resolution and quality of the 3D reconstructions, which is due to the model-based approach and the explicit representation of imaging parameters. The results and the rating scores by human participants demonstrate that the system produces a high percentage of photo-realistic reconstructions and it can be used for other practical applications e.g., as a preprocessing step for face recognition.

References

[1] P. Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. In *Proc. of the Int. Conf. on Computer Vision (ICCV)*, Corfu, Greece, September 1999.

[2] Amit K. Roy Chowdhury and Rama Chellappa. Face reconstruction from monocular video using uncertainty analysis and a generic model. *Comput. Vis. Image Underst.*, 91(1-2):188–213, 2003.

[3] M. Brand. Morphable 3d models from video. In *Conf. on Comp. Vis. and Pattern Recog.*, pages 456–463, 2001.

[4] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, pages 2690–2696, 2000.

[5] Zhengyou Zhang, Zicheng Liu, Dennis Adler, Michael F. Cohen, Erik Hanson, and Ying Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *Int. J. Comput. Vision*, 58(2):93–119, 2004.

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

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

[8] Jinho Lee, Hanspeter Pfister, Baback Moghaddam, and Raghu Machiraju. Estimation of 3d faces and illumination from single photographs using a bilinear illumination model. In *Rendering Techniques*, pages 73–82, 2005.

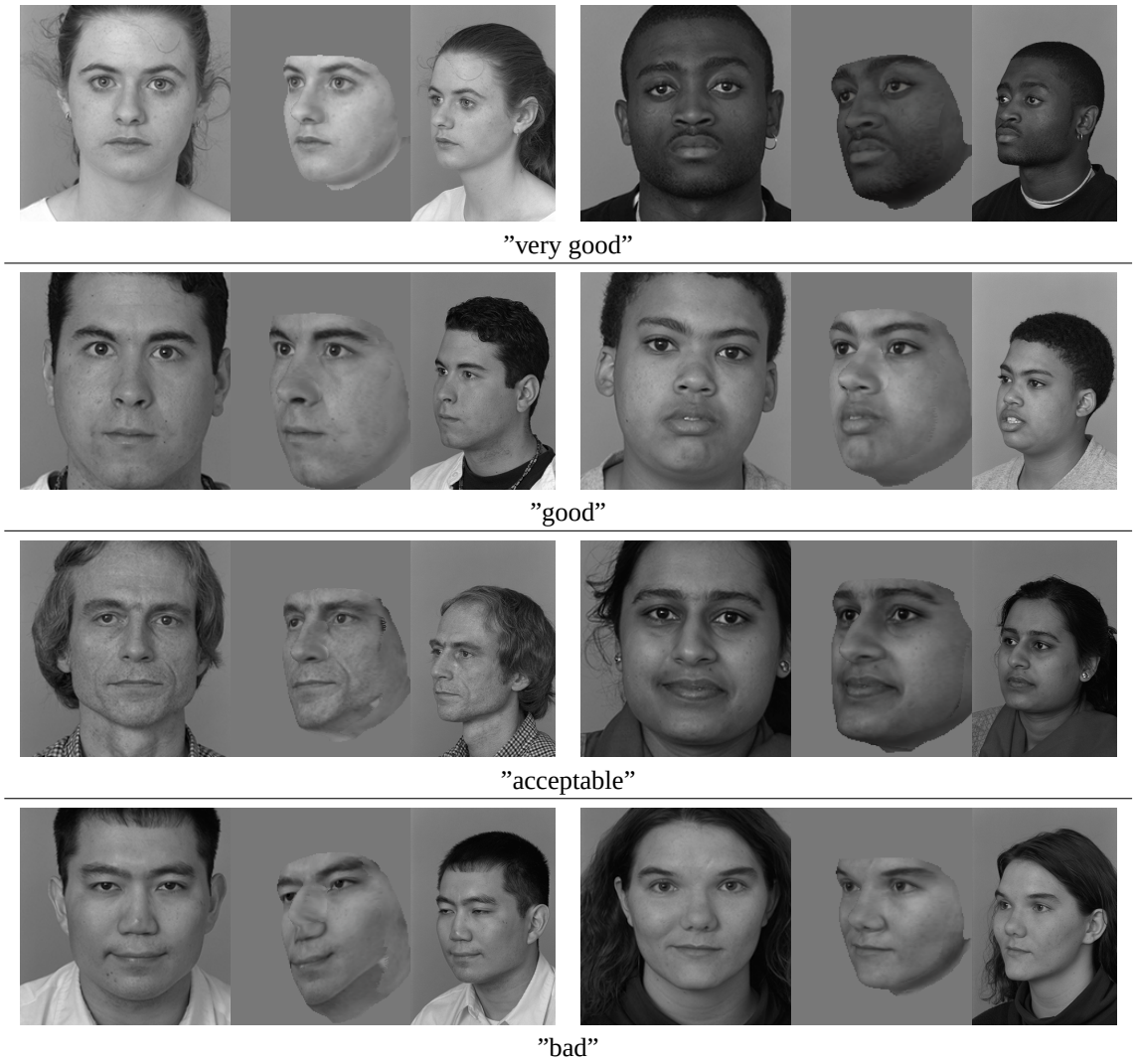

Figure 5: Reconstruction from still images: Left: face region that was automatically cropped from larger images after face detection. Center: side view of the reconstructed 3D model. Right: side view of the person for comparison. We show two typical examples from each rating score level.

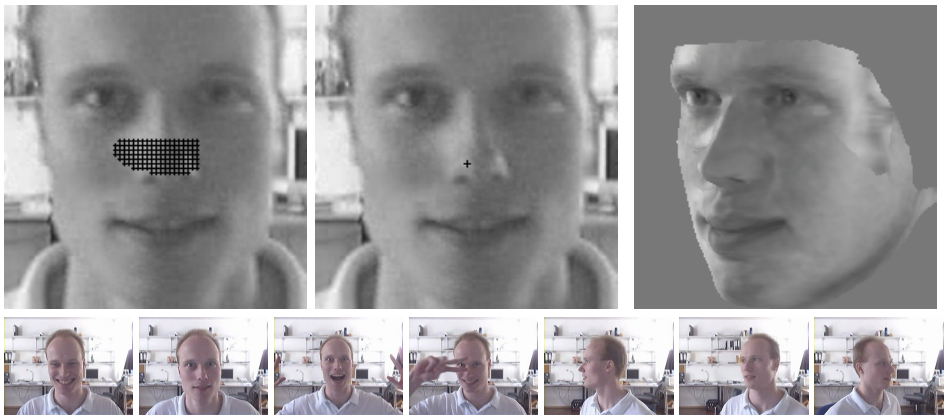

Figure 6: Fully automated reconstruction from an automatically detected single frame of a webcam video. Left: nose candidates given by the component detection, Center: nose position returned by the model-based measure, Right: automatically reconstructed head. The bottom row shows 7 sample frames.

[9] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In *Proc. IEEE CVPR*, June 2004.

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

[11] 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.

[12] M.-H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[13] W. Kienzle, G. Bakir, M. Franz, and B. Schölkopf. Face detection - efficient and rank deficient. In Y. Weiss, editor, *Advances in Neural Information Processing Systems*, pages 673–680, MA, 2005. MIT Press.

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

[15] 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.

[16] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large-Margin Classiers*, pages 61–74. MIT Press, MA, 1999.

[17] 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.

[18] 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.

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

[20] O. Jesorsky, K. Kirchberg, and R. Frischholz. Robust face detection using the hausdorff distance. In J. Bigun and F. Smeraldi, editors, *Audio and Video based Person Authentication*, pages 90–95. Springer, 2001.

[21] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

[22] Volker Blanz, Albert Mehl, Thomas Vetter, and Hans-Peter Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In Yiannis Aloimonos and Gabriel Taubin, editors, *2nd International Symposium on 3D Data Processing, Visualization, and Transmission, 3DPVT 2004*, pages 293–300, Thessaloniki, Greece, 2004. IEEE.

[23] 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.