

# Shape From Suggestive Contours Using 3D Priors

Stefanie Wuhrer  
Cluster of Excellence MMCI  
Saarland University  
Germany

Email: [swuhrer@mmci.uni-saarland.de](mailto:swuhrer@mmci.uni-saarland.de)

Chang Shu  
Institute for Information Technology  
National Research Council  
Canada

Email: [chang.shu@nrc-cnrc.gc.ca](mailto:chang.shu@nrc-cnrc.gc.ca)

**Abstract**—This paper introduces an approach to predict the three-dimensional shape of an object belonging to a specific class of shapes shown in an input image. We use suggestive contour, a shape-suggesting image feature developed in computer graphics in the context of non-photorealistic rendering, to reconstruct 3D shapes. We learn a functional mapping from the shape space of suggestive contours to the space of 3D shapes and use this mapping to predict 3D shapes based on a single input image. We demonstrate that the method can be used to predict the shape of deformable objects and to predict the shape of human faces using synthetic experiments and experiments based on artist drawn sketches and photographs.

**Keywords**—shape inference, statistical priors, suggestive contours

## I. INTRODUCTION

Artists use contour lines to convey 3D shapes on a 2D plane. Simple line drawings can be strikingly effective in rendering realistic objects and scenes. Contour lines can either be the boundary (silhouette) of an object or feature lines that depict the transition between light and shadow in the interior of an object. In computer vision, silhouette is the most often used contour, because it is a strong visual cue and it is relatively easy to extract from images. However, silhouette only encodes shape information on the boundary of an object, and therefore, multiple images are necessary to reconstruct shapes. Occluding contour, which has limited shape information in the interior of the object, is also used in multi-view reconstruction [7]. In the other extreme, every pixel can be used for reconstruction, as in the case of various stereo algorithms. However, for single-view reconstruction, the majority of the pixels depend on color and lighting conditions and do not contain shape information.

In computer graphics, a non-photorealistic rendering technique called *suggestive contour* has been developed to produce images in a style that imitates the line drawings in artworks [8]. Figure 1 illustrates suggestive contour with comparisons to silhouette and occluding contour. Note that suggestive contour reveals view-dependent shape information succinctly. This motivates us to use suggestive contours to reconstruct 3D shapes from a single image.

In this paper, we predict the three-dimensional shape of an object belonging to a specific class such as the



Figure 1: From left to right: silhouette, occluding contour, and suggestive contour of a face model.

human face. We use a database of 3D shapes to learn a functional mapping from the image space to the space of 3D shapes. From a 3D shape, we can generate the corresponding suggestive contour images from different viewpoints for training. In order to establish the functional mapping, both the 3D shape space and the image space are represented by statistical shape models. We use principal component analysis (PCA) on parameterized 3D shapes and images to model the data variations.

Once the learning is complete, the approach predicts the 3D shape of an object belonging to the class of objects used for training based on an input image as follows. First, the approach computes the valley image of the input image. This is the image of local pixel intensity minima and is expected to resemble the suggestive contour image of the 3D shape. In fact, if the image is obtained by rendering a smoothly shaded 3D object with a diffuse light source at the origin, it can be proven that the valley image and the suggestive contour image are equivalent [8]. Second, the approach searches the PCA space of the suggestive contours to find the contour that is most similar to the valley image. Finally, the point in PCA space corresponding to the contour that is most similar to the valley image along with the learned functional mapping is used to predict the 3D shape.

This way of predicting the shape of an object belonging to a specific class can be viewed as estimating a set of pose parameters of an a priori known parameterized 3D deformable model based on a single input image.

We demonstrate experimentally that the method can be used to predict the shape of deformable objects and to predict the shape of human faces based on a single artist drawn sketch or photograph.

## II. RELATED WORK

Many algorithms, such as [13], reconstruct the three-dimensional shape of an object from images or drawings using smoothness constraints or using template shapes as shape priors. Unlike the approach proposed in this paper, these approaches do not learn a statistical model of the shapes to be reconstructed. Since this work proposes a training-based approach, we limit the discussion of related literature to this scenario.

Various methods learn a multi-variate function that associates silhouettes and 3D shapes. This requires prior knowledge about the type of object being reconstructed and its shape variability. In general, these approaches treat the space of all shapes of interest as a shape space and aim to find a mapping from the space of silhouettes to this shape space. Agarwal and Triggs [1] use relevance vector machines (RMVs) to recover the pose of a human body from a single photograph. Gond et al. [10] use RMVs to predict the pose of a human body from a set of  $n$  silhouettes.

Chen and Cipolla [6] estimate the human body shape in a fixed pose based on a given silhouette. Ek et al. [9] estimate the pose of a human body based on a given silhouette. Sigal et al. [18] predict human pose and body shape from a single image. They encode the prior model using the SCAPE model [3]. Guan et al. [11] estimate both the shape and pose of a human body from a single photograph with a set of markers identified by the user.

While these approaches were shown to yield satisfactory results in many cases, they do not consider features inside the 2D silhouette. Hence, these approaches are not suitable in certain scenarios. For instance, when we want to predict the shape of a human face from a single frontal image, the silhouette does not contain enough information for an accurate prediction. We use suggestive contours instead of silhouettes for training to obtain an approach that can predict a variety of shapes from a single view.

Since we consider the application of predicting the shape of a human face in our experiments, we briefly review literature related to predicting the three-dimensional shape of a human face from a single input image. The training-based methods by Blanz and Vetter [4], Romdhani and Vetter [16], Romdhani et al. [15], and Jiang et al. [12] all predict a 3D face shape from a single image based on the 3D morphable model [4]. Sucontphunt et al. [19] use the 3D morphable model to create a 3D face shape from a sketch. With the exception of the last approach, all of these approaches rely mostly on markers and pixel intensities in the input image for shape prediction. By using suggestive contour images instead of pixel intensities, we provide a unified framework for shape- and posture prediction.

Yoon et al. [23] have recently used suggestive contours for the application of shape retrieval of 3D models.

## III. SUGGESTIVE CONTOURS

Suggestive contours [8] are a powerful tool to convey shape information. Suggestive contours are view-dependent features that become contours in nearby views. Figure 1 shows that suggestive contours contain more shape information than silhouettes or occluding contours.

More formally, suggestive contours are the set of points where the radial curvature is zero and where the directional derivative of the radial curvature in the direction of the projection of the viewing direction into the tangent plane is positive. Alternatively, suggestive contours can be expressed as minima of the dot product of the normal direction and the viewing direction in the direction of the projection of the viewing direction into the tangent plane. These two definitions allow for two alternative algorithms to compute suggestive contours. The first algorithm is based on the first definition using radial curvature and operates on the 3D object directly. The second algorithm is based on the second definition using the dot product and operates on a rendered image of the 3D object. We take advantage of this definition to estimate a suggestive contour from an image directly when the 3D shape is unknown.

## IV. TRAINING

Our method proceeds by learning a correlation between a set of suggestive contour images and a set of shapes starting from a data base of  $n$  3D models represented as triangular manifold meshes. Let  $X_0, \dots, X_{n-1}$  denote the 3D models of a specific class of objects used for training. First, the models are parameterized and compressed. Second, suggestive contour images are derived from the shapes  $X_i$  [8]. Third, the suggestive contour images are parameterized and compressed. Finally, a functional mapping from the shape space of contour images to the shape space of 3D models is computed.

### A. Parameterizing 3D Models

We first parameterize the database of 3D models. In general, computing point-to-point correspondences between a set of surfaces is a difficult problem [21]. In this work, we only consider databases of shapes belonging to the same class of object. Furthermore, we assume that the database contains a set of marker positions. In this case, the known marker positions can be used to deform a template shape to each subject of the database [2], [22]. This allows us to express each 3D shape as an ordered vector of coordinate values  $[x_0 \ y_0 \ z_0 \ \dots \ x_{m-1} \ y_{m-1} \ z_{m-1}]^T$ , where  $m$  is the number of vertices on each shape, with the property that the vertices with the same index on different shapes of the database correspond to the same anatomical position on the shapes.

After parameterization, we perform PCA of the 3D models  $X_i$ . We denote the shape space of the 3D models by  $\mathcal{S}^{3D}$ . In PCA space, each shape  $X_i$  is represented by a

vector  $W_i^{(X)}$  of PCA weights. PCA yields a mean shape  $\mu^{(X)}$  and a matrix  $A^{(X)}$  that can be used to compute a new shape  $X_{new}$  based on a vector of PCA weights  $W_{new}^{(X)}$  as  $X_{new} = A^{(X)}W_{new}^{(X)} + \mu^{(X)}$ . This same matrix can also be used to compute the PCA weights of a new shape  $X_{new}$  as  $W_{new}^{(X)} = A^{(X)T}(X_{new} - \mu^{(X)})$ .

### B. Generating Suggestive Contours

For each shape  $X_i$ , we use  $k$  view points to render  $k$  suggestive contour images. Let  $I_0^0, \dots, I_{n-1}^0, I_0^1, \dots, I_{n-1}^1, \dots, I_{n-1}^{k-1}$  denote the suggestive contour images. Note that all of the images have the same size. We use the code by DeCarlo et al. [8] with the parameters set to the program’s default values to compute the suggestive contours.

### C. Parameterizing Suggestive Contours

Once the suggestive contour images are generated, they need to be parameterized in order to be used effectively. We parameterize the suggestive contour images for each view point separately. Computing a parameterization of the suggestive contours of a general object class automatically is a challenging problem because the contours differ in both shape and topology. To assist establishing correspondences, we manually place markers on anatomically corresponding positions of the images when parameterizing  $I_0^j, \dots, I_{n-1}^j$ . Note that these markers can alternatively be obtained by placing markers on the 3D surfaces (only needs to be placed manually on one surface since the 3D correspondences are now known) and by projecting the markers to the suggestive contour images. We compute a reference frame by computing the mean position over  $I_0^j, \dots, I_{n-1}^j$  for each of the markers. We parameterize the database by aligning each image  $I_i^j$  to the reference frame using a pair of thin-plate splines computed using the marker positions [5]. This allows us to express each image as an ordered vector of intensities and offsets  $[i_0 \text{ } ox_0 \text{ } oy_0 \dots i_{m-1} \text{ } ox_{m-1} \text{ } oy_{m-1}]^T$ , where  $i_j$  is the intensity of the parameterized image at pixel  $j$ ,  $ox_j$  and  $oy_j$  are the horizontal and vertical offsets between the original image and the parameterized image at pixel  $j$ , and  $m$  is the number of pixels in each image. Note that the offsets  $ox_j$  and  $oy_j$  define a deformation field of the image with the property that after deforming the images according to their deformation fields, all of the images are aligned. By using thin-plate splines to compute the deformation fields, we minimize the amount of bending that the image undergoes. Figure 2 shows a suggestive contour image before and after alignment.

As before, we perform PCA of the suggestive contour images  $I_i^j$ . We denote the shape space of these images by  $\mathcal{S}^{2D}$ . In PCA space, each suggestive contour image  $I_i^j$  is represented by a vector  $W_i^{j(I)}$  of PCA weights. We denote the mean and matrix corresponding to this shape space by  $\mu^{(I)}$  and  $A^{(I)}$ .

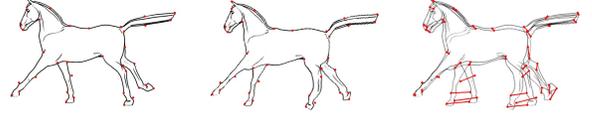


Figure 2: *From left to right: image with markers before alignment, image after alignment, and visualization of the deformation field at the marker positions.*

### D. Computing a Functional Mapping

As a last training step, we learn a functional mapping between  $\mathcal{S}^{2D}$  and  $\mathcal{S}^{3D}$  that is similar to the feature analysis approach by Allen et al. [2]. That is, we compute a linear mapping from the PCA space of the suggestive contour data to the PCA space of the 3D data. This approach yields a matrix  $B$  that can be used to compute a new vector of 3D PCA weights based on a new vector of PCA weights of a suggestive contour as  $W^{(X)} = BW^{(I)}$ .

## V. PREDICTION

This section outlines how to predict a 3D shape given a cropped image  $I_{new}$  of an object with solid background that belongs to the class of objects used for training. First, the algorithm processes the input image to find valleys. Second, we compute the PCA weight that corresponds to the image in PCA space that matches the valley image optimally. Finally, we find the 3D shape corresponding to that PCA weight.

### A. Processing Input Image

DeCarlo et al. [8] show that the suggestive contours of a three-dimensional model can be computed by rendering a smoothly shaded image with a light source placed at the camera origin and by computing the valleys in intensity in this image. Since we do not know the three-dimensional model, we compute the valleys of intensity of  $I_{new}$  to approximate the suggestive contour. Our algorithm to process  $I_{new}$  proceeds as follows. First, the algorithm resizes the image to have the same dimension as the training images. Second, the algorithm finds the valleys of the image as follows. Pixel  $p[i, j]$  is a valley if no more than  $s$  percent of the pixels in the disk of radius  $r$  centered at  $p[i, j]$  are darker than  $p[i, j]$  and if  $p_{max} - p[i, j]$  exceeds a threshold  $d$ , where  $p_{max}$  is the pixel with maximum intensity in the disk of radius  $r$  centered at  $p[i, j]$ . We set  $r = 4$ ,  $s = 20$ , and  $d = 0.06$ . We store the valleys in an image  $E$  with the same dimensions as the training images.

Note that using this algorithm to estimate suggestive contours is only provably correct if the image is a smoothly shaded image of the 3D object with a light source places at the camera origin. However, we found experimentally that good contours were found in case of cropped facial images.

### B. Computing the Optimal PCA Weight

Given the valley image  $E$ , we aim to find the point in  $\mathcal{S}^{2D}$  that corresponds to the suggestive contour that is most similar to  $E$ . Recall that given a weight  $W^{(I)}$  in  $\mathcal{S}^{2D}$ , we can compute a vector  $V = A^{(I)}W^{(I)} + \mu^{(I)}$  containing the image intensities as well as the horizontal and vertical offsets for each pixel. Let  $V[i, j, 1]$  contain the image intensity at the pixel in row  $i$  and column  $j$ , let  $V[i, j, 2]$  contain the horizontal offset at the pixel in row  $i$  and column  $j$ , and let  $V[i, j, 3]$  contain the vertical offset at the pixel in row  $i$  and column  $j$ . Then, the image corresponding to  $W^{(I)}$  contains the pixel intensities  $Img(W^{(I)})[i, j] = V[i + V[i, j, 3], j + V[i, j, 2], 1]$ . If  $i + V[i, j, 3]$  or  $j + V[i, j, 2]$  exceeds the valid image dimensions, we crop the values to stay inside the image.

To parameterize the image, we initialize  $W^{(I)}$  in a user-assisted way by asking the user to specify roughly the same markers on the image that were used to parameterize the images used for training. The algorithm uses this information to find an RBF mapping of the new image to the reference frame used for training. This results in a vector of image intensities and offsets that are projected to  $\mathcal{S}^{2D}$ .

Next, we aim to modify  $W^{(I)}$  to be as close to  $E$  as possible. To achieve this goal, we modify  $W^{(I)}$  such that the dissimilarity  $D = \sum_i \sum_j (Img(W^{(I)})[i, j] - E[i, j])^2$  is minimized, where  $i$  loops through the columns of the image and  $j$  loops through the rows of the image. We can compute the gradient with respect to the individual components of  $V$  as

$$\begin{aligned} \frac{\partial D}{\partial V[i, j, 1]} &= 2(Img(W^{(I)})[i, j] - E[i, j]), \\ \frac{\partial D}{\partial V[i, j, 2]} &= 2(Img(W^{(I)})[i, j] - E[i, j]) \frac{\partial Img(W^{(I)})[i, j]}{\partial V[i, j, 2]}, \\ \frac{\partial D}{\partial V[i, j, 3]} &= 2(Img(W^{(I)})[i, j] - E[i, j]) \frac{\partial Img(W^{(I)})[i, j]}{\partial V[i, j, 3]}, \end{aligned}$$

where  $\frac{\partial Img(W^{(I)})[i, j]}{\partial V[i, j, 2]}$  and  $\frac{\partial Img(W^{(I)})[i, j]}{\partial V[i, j, 3]}$  are evaluated numerically. This allows to compute the gradient  $\frac{\partial D}{\partial W^{(I)}} = A^{(I)T} \frac{\partial D}{\partial V}$ . We use the limited-memory Broyden-Fletcher-Goldfarb-Shanno scheme [14] for the optimization.

### C. Computing the 3D Shape

Once the PCA weight  $W^{(I)}$  corresponding to  $E$  is computed, we use the learned functional mapping to predict a shape  $X_{new}$ . The learned linear mapping is used to compute the weight vector  $W^{(X)}$  in  $\mathcal{S}^{3D}$  corresponding to  $W^{(I)}$ . Finally, the learned PCA space  $\mathcal{S}^{3D}$  is used to compute  $X_{new}$  as  $X_{new} = A^{(X)}W^{(X)} + \mu^{(X)}$ .

## VI. EXPERIMENTS

We conduct synthetic experiments and experiments with real artist drawn sketches and photographs. The first experiment demonstrates the ability of the proposed method to learn the posture space of a shape and to use images to predict this shape in different postures. The second experiment demonstrates the ability of the proposed method

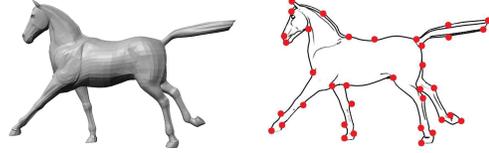


Figure 3: A 3D horse model and its corresponding suggestive contour with manually placed (36) markers.

to learn the shape space of a population of shapes in similar postures and to use images to predict new shapes of this class.

### A. Horse

This experiment aims to predict the three-dimensional posture and shape of a horse. We learn the shape space of the horse using different postures and we aim to predict the posture of the horse from an input image. Predicting the posture based on an input image is an important problem, particularly when the goal is to predict the posture of a human body [9], [18], [11].

*Training:* We use for training 10 horses with different postures from the dataset created and used by Sumner et al. [20]. Since the horses are already in correspondence, we do not need to compute the point-to-point correspondences of the 3D data. We set the dimension of  $\mathcal{S}^{3D}$  to 9.

In this experiment, we use the side view ( $k = 1$ ). We use a set of manually placed markers on the suggestive contour images for parameterization. For this approach to be feasible, we choose all of the horse models used in this experiment in a way that the order of the legs in the side view is always the same. Figure 3 shows a 3D model and its suggestive contour with the marker positions. We set the dimension of  $\mathcal{S}^{2D}$  to 9.

*Prediction:* We predict two 3D horse shapes based on synthetic data. We use two horse shapes of the dataset that were not used for training and compute their suggestive contours. These suggestive contours are then used directly to predict 3D shapes. Note that since we use a suggestive contour as input to the algorithm, we eliminate the possibility of errors from the computation of the valley image and from the search of the optimal PCA weight in  $\mathcal{S}^{2D}$ . Figure 4 shows the results. The color-coding shows the distance of vertices on the predicted shape to their corresponding vertices on the original 3D model. The distance is measured as percentage of the length of the diagonal of the bounding box of the model. Note that the overall posture of the horse is predicted correctly.

*Evaluation of Robustness:* We evaluate the robustness of the approach with respect to errors in the manually clicked marker positions used for image alignment and with respect to changes in the pose of the 3D model used to compute the suggestive contours. To evaluate the robustness with respect to errors in the manually clicked marker positions, we perturb

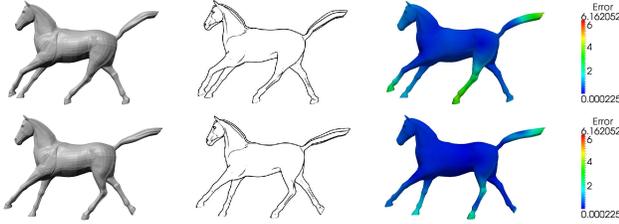


Figure 4: Prediction of horses in different postures. From left to right: the true 3D model, the suggestive contour image used for prediction, and the color-coded predicted shape  $X_{new}$ .

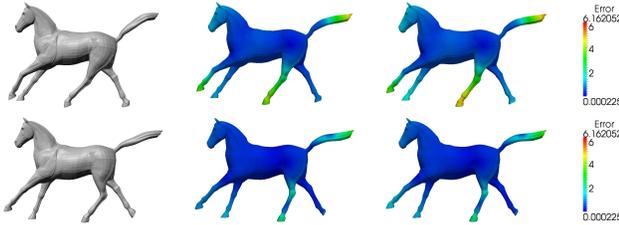


Figure 5: Robustness with respect to inaccurate marker positions. From left to right: the true 3D model, the color-coded predicted shape after perturbing the marker positions using a normal distribution with a mean of 3 and 5 pixels, respectively.

the marker positions randomly using a normal distribution with a mean of 3 and 5 pixels, respectively. Figure 5 shows the results using the same color-coding as above. Note that even inaccurate marker positions yield accurate predictions.

To evaluate the robustness with respect to changes in the pose of the 3D model, we rotate the 3D model before computing the suggestive contour. We rotate the model by  $-30^\circ$ ,  $-20^\circ$ ,  $-10^\circ$ ,  $10^\circ$ ,  $20^\circ$ , and  $30^\circ$  around the axis that is orthogonal to the floor on which the horse is running. Figure 6 shows the results using the same color-coding as above. Note that even inaccurate poses yield accurate predictions.

Table I summarizes the accuracy of the results. Note that

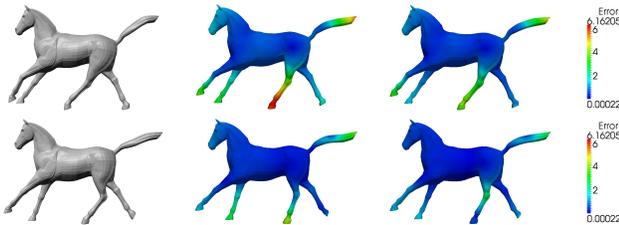


Figure 6: Robustness with respect to changes in pose. From left to right: the true 3D model, the color-coded shapes predicted from suggestive contours computed after rotating the model by  $-30^\circ$  and  $30^\circ$  degrees, respectively.

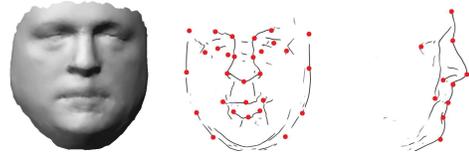


Figure 7: A 3D face model and its corresponding suggestive contours with manually placed markers (29 markers on front view and 10 markers on side view).

for all experiments, the maximum error is less than 7% of the length of the diagonal of the bounding box of the model.

### B. Face

This experiment aims to predict the three-dimensional shape of a face shown in an input image. Capturing the three-dimensional shape of a human face has various applications, such as face recognition [12]. When capturing 3D shapes for face recognition, typically only a single input image is available.

*Training:* We use for training 100 faces with neutral facial expression. Hence, we can only predict face shapes with neutral expression. We use Xi et al.’s approach [22] to parameterize the database based on known marker positions. We set the dimension of  $S^{3D}$  to 99.

For the first experiment, we use two sets of views ( $k = 2$ ), namely the front and side views of the models. For the remaining experiments, we retrain the PCA space to only use the front view ( $k = 1$ ) of the models. We use a set of manually placed markers on the suggestive contour images for parameterization. Figure 7 shows a 3D model and its two suggestive contours with the marker positions. We set the dimension of  $S^{2D}$  to 99.

*Prediction:* First, we predict 3D face shapes based on synthetic image data. We generate a set of face shapes that were not used for training from the learned PCA space  $S^{3D}$  and compute their suggestive contours. These suggestive contours are then used directly to predict 3D face shapes. Note that since we use a suggestive contour as input to the algorithm, we eliminate the possibility of errors from the computation of the valley image and from the search of the optimal PCA weight in  $S^{2D}$ . Figure 8 shows the results for four models. We used the front view to predict two of the models and the side view to predict the remaining two models. Note that the true shape and the predicted shape are visually similar and that in the interior of the face, the distance between the predicted shape and the true 3D model is at most  $5mm$  (for comparison, note that movements of the mouth region due to a slight expression change may result in distances of this magnitude).

Second, we predict 3D face shapes based on artist drawn sketches of faces. Note that by using sketches as input, we eliminate the possibility of errors from the computation of the valley image. However, unlike in the previous

	Original	Perturb Markers by 3 Pixels	Perturb Markers by 5 Pixels	Rotate by $-10^\circ$	Rotate by $-20^\circ$	Rotate by $-30^\circ$	Rotate by $10^\circ$	Rotate by $20^\circ$	Rotate by $30^\circ$
Model in top row of Figures 4, 5, 6									
Max. error	4.30	5.35	5.69	4.56	5.20	6.16	4.69	4.35	4.83
Average error	1.05	1.21	1.42	1.11	1.35	1.63	1.03	0.93	1.18
Model in bottom row of Figures 4, 5, 6									
Max. error	2.62	3.40	3.79	3.48	4.08	4.25	2.91	3.22	2.93
Average error	0.62	0.73	0.78	0.75	0.85	0.97	0.64	0.72	0.67

Table I: We compute the error as distances of vertices on the predicted shape to their corresponding vertices on the original 3D model. The table gives for each experiment the maximum and average error over all vertices of the model. The distances are measured as percentage of the length of the diagonal of the bounding box of the model. The first column corresponds to the experiment shown in Figure 4, the next two columns correspond to the experiments shown in Figure 5, and the remaining columns correspond to the experiments shown in Figure 6.

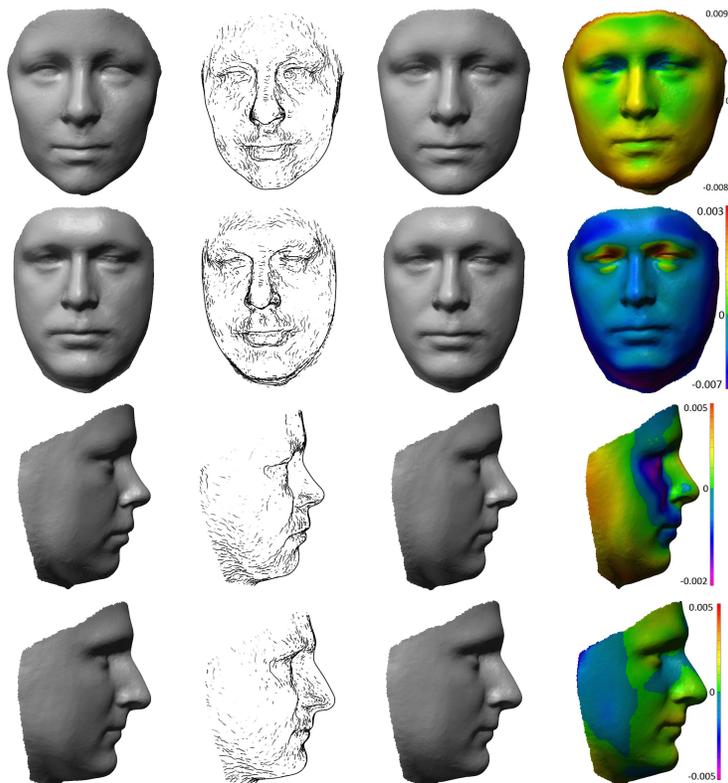


Figure 8: Prediction of faces. From left to right: the true 3D model, the suggestive contour image used for prediction, the predicted shape  $X_{new}$ , and the color-coded signed distance between the predicted shape and the true 3D model (in meters).

experiment, we search for the optimal PCA weight in  $S^{2D}$ . Figure 9 shows two sketches and the corresponding predicted face shapes. Note that the overall shape of the faces is captured well in the predicted results.

Third, we predict 3D face shapes based on images of faces. Figure 10 shows the results for three subjects from the Yale face database<sup>1</sup>. Note that the overall shape of the faces is captured well in the predicted results. Note that the algorithm produces visually pleasing results although

the valley images differ in appearance from the suggestive contour images extracted from 3D faces. The reason is that the valley image is not used as input for prediction directly, but instead is used to find the closest suggestive contour image in the learned shape space.

*Evaluation of Robustness:* We evaluate the robustness of the approach with respect to changes in lighting conditions. While the theoretical foundation of the approach requires the light to be positioned at the camera center, we show that in practice, the approach can handle slight changes in lighting conditions. We evaluate the approach by predicting the 3D

<sup>1</sup><http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

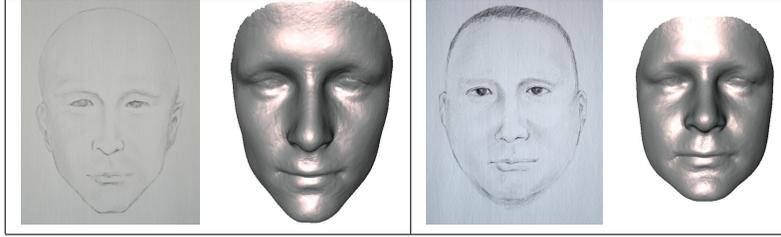


Figure 9: Prediction of faces from artist drawn sketches. The figure shows the input sketches and the predicted shapes  $X_{new}$ .

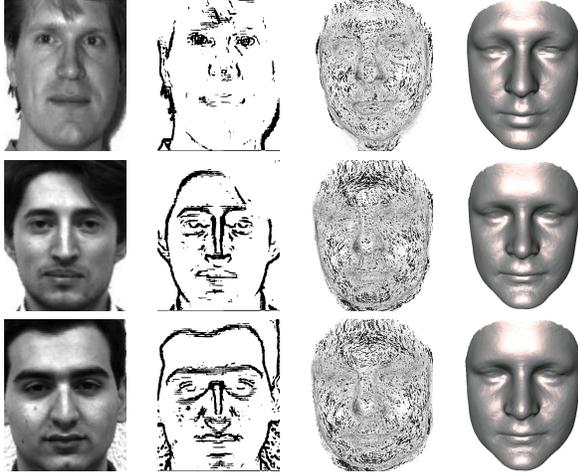


Figure 10: Prediction of faces from images. From left to right: the input image  $I_{new}$ , the valley image  $E$ , the suggestive contour in  $S^{2D}$  that is most similar to  $E$ , and the predicted shape  $X_{new}$ .

face shape of a subject from two images of the same subject and by comparing the geometry of the two predictions. The first image  $I_{new}^{center}$  is taken with the light approximately at the camera position, while the second image  $I_{new}^{left}$  is taken with the light on the left side of the face. We use the images to predict the face shapes  $X_{new}^{center}$  and  $X_{new}^{left}$ , and we compare the two shapes by computing the signed distance between the shapes. Figure 11 shows the results. We can see that the difference between the predicted shapes is always less than  $5mm$  although the facial expressions of the subjects are slightly different in the two images.

### C. Discussion

To summarize, we showed that suggestive contours contain rich shape information that can be used to accurately reconstruct the shape of an object of a known class of shapes. The reconstruction is robust with respect to inaccurately placed marker positions and changes in the pose of the 3D model. When using as input suggestive contours generated from known 3D models, the reconstructions are accurate (i.e. an error of less than  $5mm$  for face examples). Furthermore, we demonstrated that visually pleasing 3D reconstructions

can be computed from a single artist drawn sketch of a face or from a single input image of a face.

Certain aspects of the method remain to be further investigated. We finish with a list of limitations and topics for future work.

- Database for training. By using suggestive contours, we assume that the training data is not noisy. This is not always the case. New techniques need to be developed to cope with noisy training data. Furthermore, we can only predict shape variations, such as facial expressions, that are present in the training database.
- Coverage of viewpoint space. In this work, we use  $k$  views of suggestive contour images. This assumes that these views cover the viewpoint space well. That is, images used for prediction are taken from similar viewpoints as the images used for training.
- Controlled background. We can only predict shapes from an image with controlled background. This limitation may be overcome by first performing a grab cut of the image to find the background [17].
- Cropped images. We can only predict shapes from an image that only contains the object of interest.
- Parameterization. We use manually placed markers to aid in the parameterization of the suggestive contour images. One challenging avenue for future work is to automatically parameterize the suggestive contours.
- Thorough experimental validation. One avenue for future work is to further investigate the influence of noise and partial occlusions on the results.

## VII. CONCLUSIONS

Suggestive contours contain rich shape information. Although developed in a completely different context, they can be a useful shape-suggesting image feature. Since it is a rendering technique, it provides a natural training tool for associating the 3D shape with the image. In this paper, we proposed an approach to predict the three-dimensional shape of an object belonging to a specific class of shapes shown in an input image. The approach is a general training-based approach that can be applied to any class of object. We demonstrated experimentally that the method can be used to predict the shape of deformable objects such as human faces based on a single sketch or photograph.

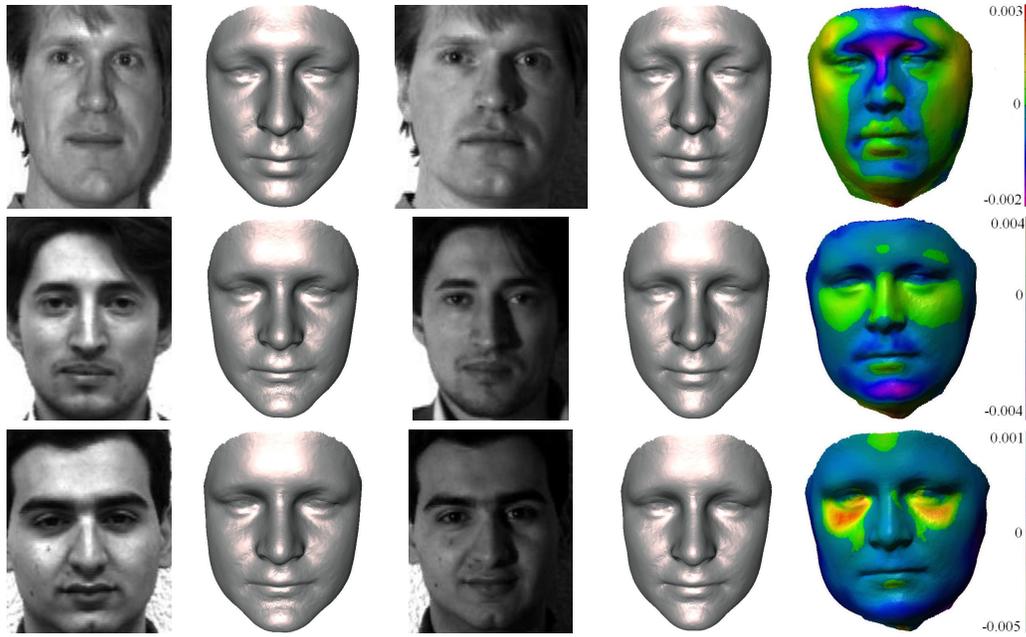


Figure 11: *Robustness with respect to changes in lighting conditions.* From left to right: the input image  $I_{new}^{center}$ , the 3D prediction  $X_{new}^{center}$ , the input image  $I_{new}^{left}$ , the predicted shape  $X_{new}^{left}$ , and the color-coded signed distance from  $X_{new}^{left}$  to  $X_{new}^{center}$  (in meters).

#### ACKNOWLEDGMENTS

We thank Pengcheng Xi for providing us with the face data. This work has partially been funded by the Cluster of Excellence *Multimodal Computing and Interaction* within the Excellence Initiative of the German Federal Government.

#### REFERENCES

- [1] A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *PAMI*, 28(1):44–58, 2006.
- [2] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. *TOG*, 22(3):587–594, 2003.
- [3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. *TOG*, 24(3):408–416, 2005.
- [4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.
- [5] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI*, 11:567–585, 1989.
- [6] Y. Chen and R. Cipolla. Learning shape priors for single view reconstruction. In *3DIM*, pages 1425–1432.
- [7] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge University Press, 2000.
- [8] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *TOG*, 22(3):848–855, 2003.
- [9] C. Ek, P. Torr, and N. Lawrence. Gaussian process latent variable models for human pose estimation. In *Machine learning for multimodal interaction*, pages 132–143, 2007.
- [10] L. Gond, P. Sayd, T. Chateau, and M. Dhome. A regression-based approach to recover human pose from voxel data. In *THEMIS*, pages 1012–1019, 2009.
- [11] P. Guan, A. Weiss, A. Balan, and M. Black. Estimating human shape and pose from a single image. In *ICCV*, 2009.
- [12] D. Jiang, Y. Hu, S. Yan, L. Zhang, H. Zhang, and W. Gao. Efficient 3D reconstruction for face recognition. *Pattern Recognition*, 38(6):787–798, 2005.
- [13] V. Kraevoy, A. Sheffer, and M. van de Panne. Modeling from contour drawings. In *Sketch-Based Interfaces and Modeling*, pages 37–44, 2009.
- [14] D. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.
- [15] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3D morphable model using linear shape and texture error functions. In *ECCV*, pages 3–19, 2002.
- [16] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *ICCV*, page 59, 2003.
- [17] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG*, 23(3):309–314, 2004.
- [18] L. Sigal, A. Balan, and M. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. *NIPS*, 2007.
- [19] T. Sucontphunt, B. Tunwattanapong, Z. Deng, and U. Neumann. Crafting 3D faces using free form portrait sketching and plausible texture inference. In *GI*, pages 209–216, 2010.
- [20] R. Sumner and J. Popović. Deformation transfer for triangle meshes. *TOG*, 23(3):399–405, 2004.
- [21] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Eurographics*, 2010.
- [22] P. Xi, W.-S. Lee, and C. Shu. Analysis of segmented human body scans. In *GI*, 2007.
- [23] S. M. Yoon, M. Scherer, T. Schreck, and A. Kuijper. Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours. In *Conference on Multimedia*, pages 193–200, 2010.