3D Reconstruction of Human Faces from Occluding Contours

Michael Keller, Reinhard Knothe, and Thomas Vetter

University of Basel, Computer Science Department, Basel, Switzerland
{michael.keller, reinhard.knothe, thomas.vetter}@unibas.ch

Abstract. In this paper we take a fresh look at the problem of extracting shape from contours of human faces. We focus on two key questions: how can we robustly fit a 3D face model to a given input contour; and, how much information about shape does a single contour image convey. Our system matches silhouettes and inner contours of a PCA based Morphable Model to an input contour image. We discuss different types of contours in terms of their effect on the continuity and differentiability of related error functions and justify our choices of error function (modified Euclidean Distance Transform) and optimization algorithm (Downhill Simplex).
In a synthetic test setting we explore the limits of accuracy when recovering shape and pose from a single correct input contour and find that pose is much better captured by contours than is shape. In a semi-synthetic test setting – the input images are edges extracted from photorealistic renderings of the PCA model – we investigate the robustness of our method and argue that not all discrepancies between edges and contours can be solved by the fitting process alone.

1 Introduction

Automatic face recognition from a given image is one of the most challenging research topics in computer vision and it has been demonstrated that variations in pose and light are the major problems [Zha00]. Since 2D view based methods are limited in their representation, most face recognition systems show good results only for faces under frontal or near frontal pose. Methods that are based on the fitting of deformable, parametrized 3D models of human heads have been proposed to overcome this issue [Bla03].

Our work investigates the problem of recovering the 3D shape of human faces from a single contour image. Previous work shows that shape recovery methods based on pixel intensities or color are inappropriate for matching edges accurately, while multi-feature approaches such as [Rom05], where contour features are used among other features, achieve higher accuracies at the edges. To further understand the benefits and limits of the contour feature, here, we take a step back to look at this feature in isolation. Our first interest lies in building a system to robustly find a solution with a small contour reconstruction error. Our second interest lies in determining the degree to which such a solution and therefore the 3D shape of the face is constrained by the input contour.

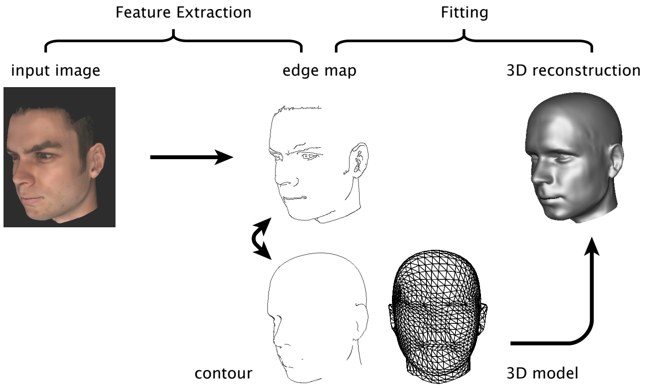


Fig. 1. Schematic view of the problem (see Section 1.1)

1.1 The Problem

Adapting a model to contours in an image can be decomposed into two subproblems: feature extraction and fitting (see Fig. 1). In a naive approach, feature extraction is performed by applying a general-purpose edge detection algorithm, such as [Can86]. Face model and camera parameters are then optimized by minimizing a distance between the model contours and the edge image.

There is an important discrepancy where the two processes interface: edge extraction computes edges and not contours. Even though most contours are found among the edges, they are incomplete and noisy. There are several ways to deal with this problem, **(a)** using a robust distance function which ignores both unmatched edges and unmatched contours, **(b)** using the model to identify which edges correspond to contours, **(c)** using a more specific form of contour detection in the first place.

In this paper we present and discuss a system based on the first approach: can we robustly fit model contours against edges by using an appropriate image distance measure. Additionally we investigate the ultimately more interesting question: assuming contour extraction was ideally solved, how well could we fit our model to a correct and complete input contour (synthesized from the model) and how good is the reconstruction of shape and pose.

1.2 Related Work

Moghaddam *et al.* [Mog03] built a system similar to our own and applied it both to video sequences and to photos acquired with a multi-camera rig – single images are not discussed. Only silhouettes are used, detected using background subtraction. Another difference lies in the error function used in their system.

Roy-Chowdhury *et al.* [Roy05] also model faces from video sequences, but as opposed to this work a locally deformable model is used. In the second part of their paper silhouettes and control points of the model are matched against edges extracted from the video frames. Model and optimization technique are very different from ours, except that an Euclidean Distance Transform is used for image comparison.

Ilic *et al.* [Ili05] adapt an implicit surface representation of a morphable PCA model to silhouette contours by solving a system of differential equations. Very interesting is their integrated approach of detecting the contour based on the location and direction of the model contour.

Both [Mog03] and [Roy05] use multiple images to achieve higher accuracy. While this is legitimate, our basic question of the constraint imposed by a single contour cannot be answered. None of the cited papers provide quantitative results describing reconstruction performance ([Roy05] do, but only for their SfM experiment, which is not based on contours).

In the remainder of the paper we first discuss types of contours and their properties, to develop exact terminology. We then explain in detail the design and implementation of our fitting system. Finally we motivate and describe our experiments, and discuss results.

2 Contours

Different authors often mean different things when using the term "contour". Sometimes, contours are "image contours" (f.i. in the context of Active Contour Models) a synonym of what we call "edges". In the context of shapes, contour is often used interchangeably with "silhouette". Other definitions of contours are implied by terms such as "inner contours", "texture contours", or "shadow contours".

2.1 Types of Contours

We define contours as the collection of those edges whose *image locations are invariant to lighting.* Contours fall then into two classes: those originating in the geometry of the solid ("occlusion contours"), and those originating in material properties ("texture contours"). Please note that a line drawing of a face would characteristically consist of just these edges.

Occluding Contours. Assuming a smooth solid shape [Koe90], occluding contours can be formally defined as the projection of those points P of the surface whose surface normal $\boldsymbol{n}$ is perpendicular to the direction $\boldsymbol{e}$ from the eye to P, i.e. $\langle \boldsymbol{n}, \boldsymbol{e} \rangle = 0$. This definition encompasses both "outer" contours (silhouettes) and "inner" contours (self-occluding contours), where the latter only appear on non-convex solids.

Texture Contours. Texture Contours are salient edges on texture maps. For faces, such contours are found at the lips, the eyes, the eyebrows, as well as at hair boundaries and within hair.

2.2 Continuity and Differentiability

We will now investigate how different types of contours behave differently under parameter changes in terms of continuity and differentiability of related error functions. These characteristics are very important when choosing an optimization algorithm.

For simplicity's sake we describe the problem in a 2D analogy (Fig. 2). We observe the projections of silhouette (S_1, S_2), inner contour (U) and texture contour (T) with respect to a rotation ρ around R.

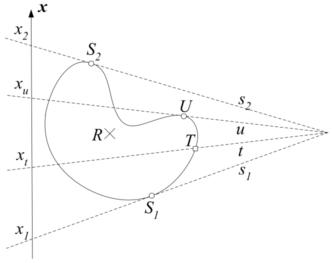


Fig. 2. 2D analogy of contour projection (see 2.2)

Only T is fixed with respect to the rotating body. Without knowledge of the shape of the body its projection x_t and its speed $\frac{\mathrm{d}x_t}{\mathrm{d}\rho}$ can be analytically determined, as long as T is not occluded.

Let now s_1, s_2 be defined as the two tangent rays with the maximum angle between them, and u the third tangent ray in the figure. When rotating the body clockwise, u will disappear. When rotating it counter-clockwise, u will coincide with s_2 to form a bitangent, then disappear.

Therefore the projection x_u is not defined for all ρ. Whether absence of the contour is penalized or not, x_u will contribute a discontinuity to a compound error measure. The projections x_1, x_2 are always present and unique, but where s_2 becomes a bitangent the derivative $\frac{\mathrm{d}x_2}{\mathrm{d}\rho}$ is undefined.

In 3D the situation is analogous: the movement of texture contour points with respect to parameters is analytically differentiable [Rom05], silhouette movement is piecewise differentiable (albeit generally not analytically), and inner contour movement is discontinuous (with 3D solids, contours appear, disappear, split and merge under parameter changes [Koe90]).

2.3 Contour Matching

Which contour features are accessible for matching depends of course on the application context. If indeed one was reconstructing faces from a shadow play, then only silhouettes would be available. Much more common is a scenario where the input is an ill-lit photograph and all types of contours are in principle accessible. While it is our goal to use all types of contours, we concentrate for now on silhouettes and inner contours, where the latter are arguably the hardest feature to work with.[1] Because the error function is neither differentiable nor even continuous, our chosen optimization algorithm is Downhill Simplex [Nel65], which uses only function evaluations and deals well with ill-behaved functions.

3 Implementation

Our face model is computed from a database of $m = 200$ laser scanned faces. Each face is defined by $n = 23888$ 3D vertices which are in correspondence across the database [Bla03]. Let $\boldsymbol{x}_i \in \mathbb{R}^{3n}, 1 \le i \le m$ be vector representation of these faces, then our (m-1)-dimensional face space is the affine subspace

[1] Please note that in most related works only the silhouette feature is used, which does not suffer from discontinuities.

$\{\sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i | \sum_{i=1}^{m} \alpha_i = 1\}$, or $\{\bar{\boldsymbol{x}} + \boldsymbol{v} | \boldsymbol{v} \in V\}$ with $\bar{\boldsymbol{x}}$ the mean face and V the vector subspace spanned by the mean-free vectors $(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}, ..., \boldsymbol{x}_m - \bar{\boldsymbol{x}})$.

We estimate the probability of faces by fitting a multi-variate Gaussian distribution to the mean-free data, yielding the pdf $p(\mathbf{p}) = c \exp\left(-\frac{1}{2}\mathbf{p}^{\mathbf{T}}\mathbf{\Sigma}^{-\mathbf{1}}\mathbf{p}\right)$ for coefficient vectors $\mathbf{p} \in \mathbb{R}^m$, where $\mathbf{\Sigma} = \frac{1}{m}\mathbf{X}\mathbf{X}^{\mathbf{T}}$ and c a normalization constant. This assumption allows us to judge plausibility of results and to randomly sample test cases.

In practise we work with an orthonormal basis $\mathbf{U} = (\boldsymbol{u}_1, ..., \boldsymbol{u}_m)$, where $\mathbf{U}$ is such that $\mathbf{U}^{\mathbf{T}}\mathbf{\Sigma}\mathbf{U} = \mathbf{\Lambda}$, with $\mathbf{\Lambda}$ the diagonal matrix of eigenvalues of $\mathbf{\Sigma}$ (in descending order). This transformation is called Principal Component Analysis, and allows us to make a feature reduction which incurs a minimal L^2 error. Specifically if $\hat{\boldsymbol{p}} \in \mathbb{R}^k$ is a projection of $\boldsymbol{p} \in \mathbb{R}^m$ onto the k-dimensional subspace $\text{span}(\boldsymbol{u}_1, ..., \boldsymbol{u}_k)$, then $E[||\mathbf{U}\mathbf{p} - \mathbf{U}\hat{\mathbf{p}}||^2] = \sum_{i=k+1}^{m} \lambda_i$. In our case the sum of the first 30 eigenvalues comprise 96% of the sum of all 200 eigenvalues, therefore fitting in a 30-dimensional subspace is mostly sufficient for good results.

Hair. While the hair line for an individual person may remain steady for years, hair length and hair style do not. Additionally hair often covers more innate and salient features such as ears and chin. In our opinion, hair is a destructive factor when creating a PCA-based model, leading to many false correspondences.

For this reason our head model is completely without hair. When registering heads we then need robust procedures which simply ignore portions of the input picture covered by hair. Future work may attempt to use such portions of input pictures constructively, but for the time being our model has no concept of hair.

Shape Error. Although we understand that the L^2 error is not optimal to describe shape similarity as perceived by humans, it is sufficient in our context. We quote the L^2 error as root-mean-squared error, with unit mm. This error is easy to visualize as average distance of the reconstructed surface from the true surface (although this is of course only correct if the error is relatively uniform). With the definitions for m, n, λ_i from above the L^2 distance between two faces described by shape vectors $\boldsymbol{p}, \boldsymbol{q}$ is calculated as:

$$L^2(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\frac{1}{n} \sum_{i=1}^{m} \lambda_i (p_i - q_i)^2}. \tag{1}$$

3.1 Error Function

Let $\boldsymbol{p'} \in \mathbb{R}^k, k = m + 7$ be a shape vector extended by the seven camera parameters of the pinhole camera model. Then the error function to be minimized must map such parameter vectors $\boldsymbol{p'}$ to scalar values. Since the binary contour image $I \in \{0,1\}^{w \cdot h}$ is in the 2D domain, it is reasonable to compose the error function E as $E(\boldsymbol{p}) = D(I, R(\boldsymbol{p}))$ with $R : \mathbb{R}^k \to \{0,1\}^{w \cdot h}$ a rendering function and $D : \{0,1\}^{w \cdot h} \times \{0,1\}^{w \cdot h} \to \mathbb{R}$ an image comparison function ("distance"). As opposed to previous work, we do not add a prior probability term to our error function, as overfitting has not been an issue in our experiments.

Rendering Function. The contour is found and rendered by determining front and back facing polygons on the mesh parametrized by $\boldsymbol{p}$. Edges between a back and a front facing polygon are contour candidates. Hidden contours are eliminated through z buffering. Finally the remaining edges are projected onto an image.

Distance Function. Choosing a good distance function is much more difficult. Edge comparison is a research topic in its own right with many interesting approaches, such as elastic matching, Fourier Descriptors, or Shock Graphs [Sid98], to name a few. All of them are computationally expensive and require relatively noise-free input. As long as we are dealing with unreliable input from the Canny edge detector, we choose the more primitive yet robust approach of distance transforms, which solves correspondence implicitly (albeit not necessarily well).

The Euclidean Distance Transform [Fel04] of an input contour image is a scalar field $d : \mathbb{R}^2 \to \mathbb{R}$ where $d(x, y)$ is the Euclidean Distance of the pixel at (x, y) to the nearest contour pixel. With $S = \{(x, y)|R(\boldsymbol{p})(x, y) = 1\}$ the set of "on" pixels in the synthetic contour image, we calculate the distance term as

$$D(I, R(\boldsymbol{p})) = \frac{1}{|S|} \sum_{(x,y)\in S} d(x, y). \tag{2}$$

Such a distance term is fairly robust against unmatched edges, although such edges can create an undesirable potential. On the other hand, contours not present in the input image will create large error terms and optimization will drive them toward completely unrelated edges. To prevent this, the gradient far away from input edges should be small or zero.

We experimented with three modifications of the EDT, of the form $g(x, y) = f(d(x, y))$: **(a)** "Linear with plateau" $f(d) = d$ for $d < c$, $f(d) = c$ for $d \geq c$, **(b)** "Quadratic with plateau" $f(d) = d^2$ for $d < c$, $f(d) = c^2$ for $d \geq c$, **(c)** "Exponential" $f(d) = \exp(-d/c) + 1$ for some $c > 0$. **(a)** and **(b)** result in distance transforms with a gradient of zero at pixels that are further than c pixels away from any input contour point, while **(c)** produces a monotonically decreasing gradient. Note that the latter is qualitatively most similar to the boundary-weighted XOR function of [Mog03].[2]

3.2 Minimizing Algorithm

For fitting against occluding contours, where the error function is not well behaved and analytical gradients are not available (see above), we chose an optimization algorithm based on function evaluations only, the Downhill Simplex algorithm due to Nelder and Mead [Nel65] in an implementation from [Pre99].

Much effort went into the setup of the initial simplex, and an appropriate restarting behavior. Our initial simplex is based on the *estimated standard deviation* of $\boldsymbol{p}_G - \boldsymbol{p}_T$ (the difference of initial guess to ground truth). Therefore,

[2] Moghaddam *et al.* integrate their error over the entire area between input and synthetic silhouette, with penalties of $1/d$ per pixel.

if $\boldsymbol{\sigma}$ is the vector of the estimated standard deviations of the k parameters to be adjusted, our initial simplex is made up of the $k+1$ points $\boldsymbol{p_i} = \boldsymbol{p_0} + \sigma_i$ for $1 \leq i \leq k$ and $\boldsymbol{p_0} = \boldsymbol{p_G} - \frac{1}{k}\boldsymbol{\sigma}$, the initial guess.[3]

After convergence of the algorithm, the algorithm was restarted with an initial simplex constructed in the same way as above, but with the point of convergence in place of the initial guess. The algorithm was started no more than ten times for a given test case, and aborted if three consecutive runs did not produce a new minimum. This resulted in an average of around 6000 error function evaluations per test case.

3.3 Camera Model and Initial Pose Estimate

Our camera model is the pinhole camera model which has seven degrees of freedom: three rotations (azimuth α, declination δ, rotation ρ around viewing axis), and four translations (left x, up y, distance from object to focal point f, distance from focal point to photographic plate d). Only α, δ, f have influence on occluding contour formation, while the other parameters describe a 2D similarity transform. When randomizing test cases the contour-defining parameters were sampled uniformly: $\alpha \in [-90°, 90°], \delta \in [-30°, 30°], 1/f \in [0\text{m}^{-1}, 2\text{m}^{-1}]$.

From the rotations we derived our pose error measure, the *aspect error*. The aspect error is the angle of the compound rotation required to align the coordinate frame of the ground truth with the coordinate frame of the estimate. For small angles and frontal views this is approximately the sum $\alpha + \delta + \rho$.

Like in previous work we always assume that an initial guess for $\alpha, \delta, \rho, x, y$ and the scale f/d has been provided by either a manual procedure or another program. When randomizing test cases we simulate this condition by randomly scattering the parameters in question. Examples of the low accuracy of the initial guess can be seen in Fig. 4 and 5.

4 Experiments

4.1 Test Settings

We differentiate between three types of test settings: realistic, synthetic and semi-synthetic. The realistic test setting is an application of the system to real photos; while being the most difficult test setting, reconstruction accuracy cannot be quantified, unless the test subject is laser scanned.

In the synthetic test setting, geometrically correct contours are generated from random configurations of the model (Fig. 3 (a)). This simulates the condition where feature extraction has been optimally solved (Fig. 1) and an exact match is possible.

In the semi-synthetic test setting, random configurations of the model are rendered photorealistically and edges extracted [Can86]. This mimics the realistic case, except that reconstruction accuracy can be easily measured since the ground truth is known. Additionally, the difficulty of the problem can be controlled and specific aspects of contour fitting can be investigated by f.i. enabling or disabling hair texture and hard shadows (see Fig. 3 (b)-(d)).

[3] When using an uncentered simplex with $\boldsymbol{p_0} = \boldsymbol{p_G}$ we observed a bias of the error of the reconstructed parameters toward positive values.

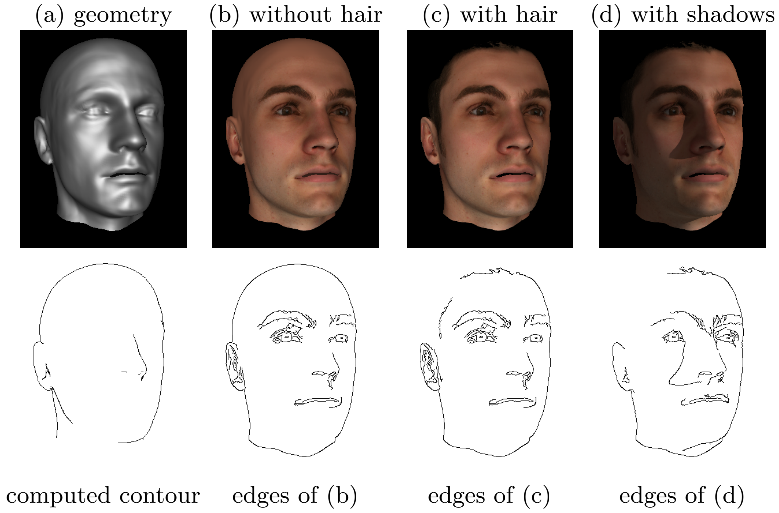


Fig. 3. Hierarchy of the used test settings: (a) synthetic: contour computed from geometry. (b)-(d) semi-synthetic: edges extracted from renderings with increasingly many features.

4.2 Synthetic Experiments

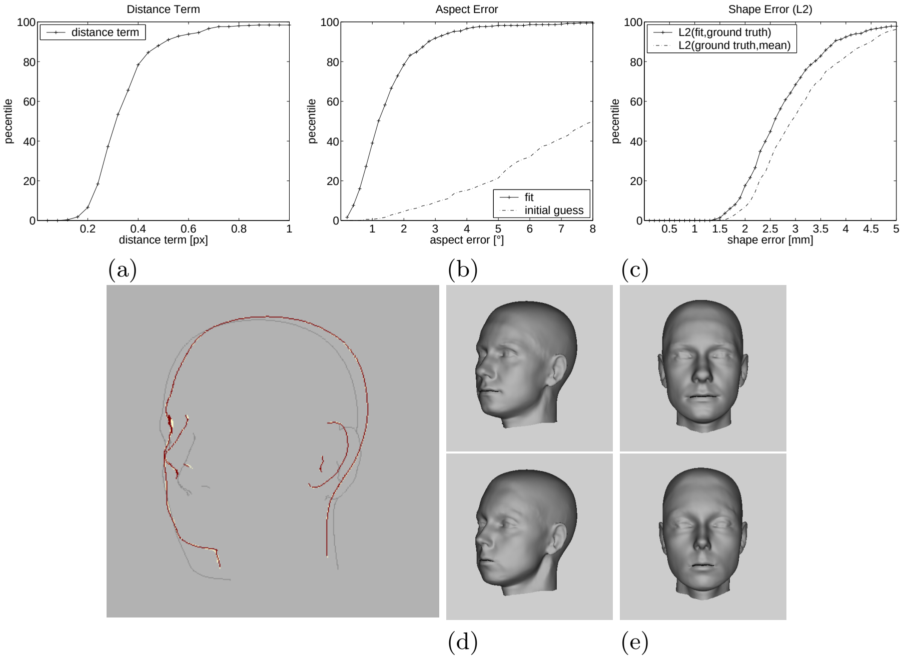
The purpose of this experiment is to answer our second key question, how much information about shape does a single contour image convey *in the best imaginable case.*

We achieved best results matching 30 principal components and an unmodified Euclidean Distance Transform (the modifications in 3.1 are designed for the semi-synthetic case). On a large test set with 500 test cases the distance term was under 0.5px for 90% of the results (Fig. 4).

The aspect error of the fits are on average very small: 92% with aspect error $e_a < 3°$. Please note, that in a first approximation $e_a \approx e_\alpha + e_\delta + e_\rho$. Therefore the pose is indeed very well estimated.

In terms of shape error the fits are not significantly nearer to the ground truth than the mean is. It is worth mentioning that this by no means signifies that the recovered head is "practically random". Let $\boldsymbol{p}, \boldsymbol{q}$ be independent random heads, where all components p_i, q_i are taken from a normal distribution with $\mu = 0$ and $\sigma = 1$. With $E[(p_i - q_i)^2] = 2$ and (1) it can easily be shown that $E[L^2(\boldsymbol{p}, \boldsymbol{q})^2] = 2E[L^2(\boldsymbol{p}, \boldsymbol{0})^2]$. Therefore L^2(fit, ground truth) would be on average $\sqrt{2}$ times larger than L^2(ground truth, mean) if the reconstructed heads were indeed random.

Nevertheless, the fact that the mean is almost as similar in L^2 terms as the reconstructed heads is very interesting. Obviously reconstructed heads can be very different from the ground truth, while displaying virtually the same contours (Fig. 4). This supports the view that contour does not constrain shape tightly, even with a holistic head model.



(a) (b) (c)

(d) (e)

Fig. 4. Synthetic Case: (a) Matched contour (black), input contour (white, mostly covered by matched contour), contour of initial guess (gray). (b),(c) ground truth (actual camera parameters, frontal view). (d),(e) reconstruction (estimated camera parameters, frontal view). Comparing 3D reconstruction to ground truth demonstrates that even near-perfect fitting of contour may result in very dissimilar 3D shape. Distance error and aspect error were near the median ($e_{\text{dist}} = 0.29\text{px}, e_{\text{asp}} = 1.47°, e_{L^2} = 3.61\text{mm}$).

More evidence is provided by a look at correlations: while aspect error correlates strongly with contour distance (correlation coefficient $c = 0.62$), L^2 error correlates less ($c = 0.44$).[4]

4.3 Semi-Synthetic Experiments

The purpose of this experiment is to see whether we can formulate the distance function in such a way, that comparing an edge map to model contours automatically leads to correct correspondence between contours and edges.

We used the same test set as in the previous paragraph, and generated edge maps for the four “unwanted feature” combinations, test persons with/without hair (predicate H), hard shadows enabled/disabled (predicate S). For brevity we notate *e.g.* the case “with hair, shadows disabled” as $H, \neg S$.

Image Errors. Since the input is now an edge map, which is generally quite different from the synthetic contour $R(\boldsymbol{p_T})$ (in the notation of 3.1 with $\boldsymbol{p_T}$ the parameters of the ground truth), we are now faced with a *residual error* $e_{\text{res}} =$

[4] To determine correlation, results with a distance term > 1px (1.6% of results) were excluded.

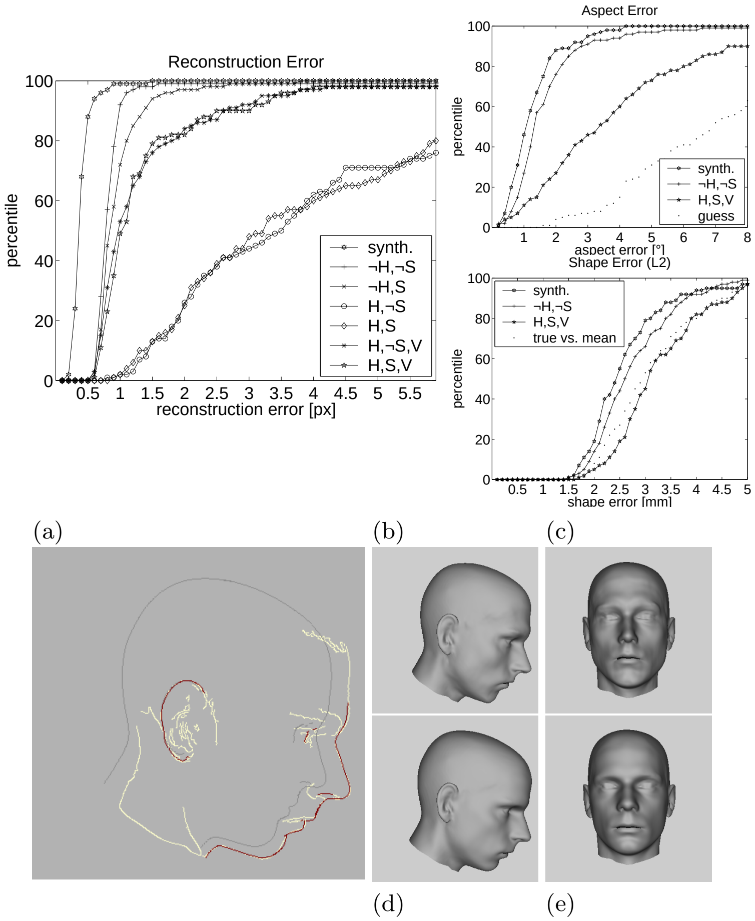


Fig. 5. Semi-Synthetic Case (H, S): Characteristic test result with vertex selection; reconstruction, aspect and L^2 error near the median ($e_{\text{rec}} = 0.97\text{px}, e_{\text{asp}} = 3.99°, e_{L^2} = 3.18\text{mm}$). (a) Matched contour (black), input contour (white), contour of initial guess (gray). (b),(c) ground truth (actual camera parameters, frontal view). (d),(e) reconstruction (estimated camera parameters, frontal view).

$D(I, R(\boldsymbol{p_T})), e_{\text{res}} \geq 0$. This residual error is highly dependent on the nature of each test case as well as edge extraction parameters. Therefore it would be a mistake to quote the distance error $e_d = D(I, R(\boldsymbol{p_R}))$ of a result vector $\boldsymbol{p_R}$, as its value does not describe reconstruction quality.

Instead we quote the *reconstruction error* $e_{\text{rec}} = D(R(\boldsymbol{p_T}), R(\boldsymbol{p_R}))$, the difference between the resulting contour and the synthetic input contour. This error measures the performance of the algorithm to match the model to the unknown true contour.

However, we must be aware that the algorithm can only minimize e_d and never e_{rec} (for which the ground truth must be known). If there is a solution with $e_d < e_{\text{res}}$ the system will prefer it, even if it means e_{rec} becomes larger.

Reconstruction. Our foremost goal is to reduce the above *reconstruction error*, e_{rec}. In a comparison of the three variants of the distance transform described above, "linear with plateau" was best in every respect. Especially the non-zero gradient of the exponential variant proved troublesome with incomplete input. The plateau constant c was chosen depending on the amount of noise in the input ($c = 25$ for $\neg H, \neg S$, until $c = 5$ for H, S).

Results were best if only 20 principal components were matched. Be aware though that if $\boldsymbol{p}$ is a random shape vector, $E[L^2(\boldsymbol{p}, \mathbf{0})] = 3.26\text{mm}$, while for $\boldsymbol{q}, q_i = p_i (1 \leq i \leq 20), q_i = 0$ (else) $E[L^2(\boldsymbol{q}, \mathbf{0})] = 3.14\text{mm}$. So on average the first 20 components carry 96% of the L^2-"content".

An interesting result in itself is that the presence of hard shadows made virtually no difference on reconstruction accuracy (Fig. 5) – when looking again at Fig. 3 (c) and (d), this makes a strong case for contour matching, considering how different the lighting is between these two pictures.

On the other hand, hair had a very destructive influence on reconstruction quality, mainly because our model is an entire head and in many test cases the silhouette of the head was matched against the hair line of the input image. For this reason we include results where the model was artificially restricted to the face area ("vertex selection", predicate V).

Results for both aspect and shape error (Fig. 5) are in line with expectations from reconstruction errors. The results are by no means bad in absolute terms (Fig. 5), but compared to the synthetic case much accuracy is lost.

5 Conclusions

We have shown that it is possible to compute a shape and a pose of a human face explaining a given input contour with consistently very high accuracy. While the original pose is recovered very well, the recovered shape often differs greatly from the ground truth. This predestines contour matching as part of a system combining several features, as it can ensure contour consistency while not imposing tight constraints on the shape.

More difficult is the case where the input contour is replaced by an edge map from a picture, because of the different nature of edges and model contours. While it is possible to design a distance function to solve correspondence implicitly, there are limits to robustness where an unwanted edge is too close to a missed contour. We showed that restricting fitting to areas where such misunderstandings are least likely to occur does alleviate the problem, but would like to point out that this solution suffers from the problem that the scope of the restriction is arbitrary and has to be appropriate for all test cases simultaneously.

In our problem description we have listed three basic options how to overcome the edge-contour correspondence problem: **(a)** using a robust distance function, **(b)** using the model to establish correspondence, **(c)** using a more elaborate feature detection process. In this paper we thoroughly explored the first option, showing on one hand that despite the simplicity of the method, good results can be achieved most of the time. On the other hand certain kinds of false correspondences are difficult to avoid without exploring the other options, which will be our focus in future work.

Furthermore we have laid out that different types of contours require different optimization techniques; we believe that combining subsystems for different types of contours may prove to be a powerful concept.

Acknowledgements

This work was supported by a grant from the Swiss National Science Foundation 200021-103814.

References

[Nel65] J. A. Nelder, R. A. Mead, "A Simplex Method for Function Minimization," *Comput. J.* 7, 308-313, 1965.

[Can86] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:679-698, November 1986.

[Mog03] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju, "Model-based 3-D face capture with shape-from-silhouettes," *IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Nice, France*, pages 2027, October 2003.

[Bla99] V. Blanz, Th. Vetter, "A morphable model for the synthesis of 3d faces," *SIGGRAPH '99*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[Bla03] V. Blanz, Th. Vetter, "Face Recognition Based on Fitting a 3D Morphable Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence 2003*, 25(9).

[Fel04] P. Felzenszwalb, D. Huttenlocher, "Distance transforms of sampled functions," *Cornell Computing and Information Science Technical Report* TR2004-1963, September 2004.

[Koe90] J. J. Koenderink, "Solid Shape," *The MIT Press, 1990.*

[Pre99] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, "Numerical Recipes in C / the Art of Scientific Computing," *Cambridge University Press, 1999.*

[Rom05] S, Romdhani, Th. Vetter, "Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior, " *CVPR'05, San Diego*, 2:986–993, 2005.

[Roy05] A. Roy-Chowdhury, R. Chellappa, H. Gupta, "3D Face Modeling From Monocular Video Sequences," *Face Processing: Advanced Modeling and Methods (Eds. R.Chellappa and W.Zhao)*, Academic Press, 2005.

[Ili05] S. Ilic, M. Salzmann, P. Fua, "Implicit surfaces make for better silhouettes," *CVPR'05, San Diego*, 1:1135–1141, 2005.

[Zha00] W. Zhao, R. Chellappa, A. Rosenfeld, P. Phillips, "Face recognition: A literature survey," *UMD CfAR Technical Report CAR-TR-948*, 2000.

[Sid98] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, S. W. Zucker, "Shock Graphs and Shape Matching," *Computer Vision*, pages 222–229, 1998.